INSTITUTO POLITÉCNICO NACIONAL

CENTRO DE INVESTIGACIÓN EN COMPUTACIÓN

CLASIFICACIÓN DE SERIES DE TIEMPO POR MINERÍA DE DATOS

TESIS QUE PARA OBTENER EL TÍTULO DE:
MAESTRO EN CIENCIAS DE LA COMPUTACIÓN

PRESENTA:
JOSÉ MARCELO RODRÍGUEZ ELIZALDE

ASESOR:
DR. JESÚS FIGUEROA NAZUNO

MEXICO, DF. 2006

SIP-14

INSTITUTO POLITECNICO NACIONAL

SECRETARIA DE INVESTIGACIÓN Y POSGRADO

ACTA DE REVISIÓN DE TESIS

En la Ciudad de México, D.F. siendo las 12:00 horas del día 27 del mes de Marzo de 2006 se reunieron los miembros de la Comisión Revisora de Tesis designada por el Colegio de Profesores de Estudios de Posgrado e Investigación del:

Centro de Investigación en Computación

para examinar la tesis de grado titulada:

"CLASIFICACIÓN DE SERIES DE TIEMPO POR MINERÍA DE DATOS"

RODRÍGUEZ	**ELIZALDE**	**JOSÉ MARCELO**
Apellido paterno	materno	nombre(s)

Con registro:

B	0	1	1	3	9	2

aspirante al grado de: ***MAESTRO EN CIENCIAS DE LA COMPUTACIÓN***

Después de intercambiar opiniones los miembros de la Comisión manifestaron ***SU APROBACIÓN DE LA TESIS***, en virtud de que satisface los requisitos señalados por las disposiciones reglamentarias vigentes.

LA COMISIÓN REVISORA

Presidente

Dr. Adolfo Guzmán Arenas

Secretario

Dr. Sergio Suárez Guerra

Primer vocal
(Director de Tesis)

Dr. Jesús Guillermo Figueroa Nazuno

Segundo vocal

Dr. Luis Pastor Sánchez Fernández

Tercer vocal

Dr. José de Jesús Medel Juárez

Suplente

Dr. Carlos Fernando Aguilar Ibañez

EL PRESIDENTE DEL COLEGIO

INSTITUTO POLITECNICO NACIONAL
CENTRO DE INVESTIGACION
EN COMPUTACION

Dr. Hugo César Coyote Estrada

Dedicatoria.

A mi esposa, por ser lo más importante.

Agradecimientos.

Al Centro de Investigación en Computación, por la oportunidad y facilidades para realizar y conocer la “investigación”.

Al Instituto Politécnico Nacional por haberme enseñado y mostrado su grandeza, como casa de estudios.

A CONACYT, por el apoyo económico otorgado para la realización de la maestría en ciencias de la computación.

A todos los sinodales: Dr. Guzmán-Arenas, Dr. Suárez-Guerra, Dr. Figueroa-Nazuno, Dr. Sánchez-Fernández, Dr. Medel-Juárez, Dr. Aguilar-Ibáñez, por sus comentarios para mejorar el presente trabajo.

Al Dr. Nazuno, por TODA su paciencia, amistad y consejos, así como a todos los amigos y compañeros del laboratorio de cómputo paralelo.

A Angélica por toda su comprensión, tiempo y apoyo.

A mi familia, amigos y a toda aquella persona, que en algún momento, me alentaron para finalizar.

A dios, por todo lo que he sido, soy y será.

¡MUCHAS GRACIAS!

CONTENIDO

Lista de figuras.

Resumen.

La minería de datos de series de tiempo (MDST) ha evolucionado considerablemente en la última década, proporcionando un marco de trabajo con diversos algoritmos. Estos algoritmos adaptan e innova las técnicas de minería de datos para su aplicación en análisis de series de tiempo.

En este trabajo se utiliza este marco y los conceptos de MDST, así como su aplicación para el análisis de series de tiempo, con la finalidad de proporcionar un método que integre técnicas de minería de datos de series de tiempo y mapas recurrentes en una herramienta computacional.

Los mapas recurrentes tienen un rol muy importante en este trabajo ya que nos permitirán tener indicadores paramétricos de similitud entre las series de tiempo, para su caracterización, clasificación y aplicación de análisis de las mismas.

Los resultados experimentales muestran, que los patrones temporales significativos generan clasificaciones, caracterizaciones y pueden ser identificados estadísticamente. Además de mostrar por métodos experimentales que dichas técnicas funcionan correctamente.

También que, la herramienta, permite el almacenamiento y recuperación de las series de tiempo.

Abstract.

The Times Series Data Mining (TSDM) has evolved considerably the last decade. These algorithms innovate and improve the data mining techniques to perform time series analysis.

In this dissertation, we shall provide an integration method with different techniques of TSDM and recurrence plots in a computing software tool.

The recurrence plot has a very important meaning, because, from there are taken some parametric attributes, which ones are used to perform classification and characterization of the time series.

The experimental result shows that the temporal patterns are using to create characteristics and classifications of time series. Based on it we will modeling those future events and could recognize statistically.

Also we shall show, by experimental methods, which theses techniques work properly. Therefore the tool is capable to storage and grabs time series for an easiest way to handled for its analysis.

Capítulo 1
Introducción

El presente capítulo es punto de partida para la realización y principal motivación para el desarrollo de esta tesis, se muestra la introducción, planteamiento del problema, objetivo, delimitación, contribución original y la organización del resto del trabajo de tesis.

1.1 Introducción

Las series de tiempo contienen mucha información relevante a un fenómeno y comportamiento observado por un sistema, como son la bolsa de valores, un sistema de control de producción, un ECG, el clima, los sismos, etc.

La **extracción, comparación, almacenamiento y recuperación** de dicha información es fundamental para realizar futuros análisis de extracción de modelos (reglas).

El análisis de series de tiempo requiere de la búsqueda e identificación de elementos o componentes que conforman a la serie de tiempo, estos proporcionan información sobre la dinámica del sistema que representa la serie de tiempo.

El presente trabajo está fundamentado en el desarrollo e integración de algoritmos de minería de datos para el análisis de series de tiempo. Esto debido al enorme crecimiento y desarrollo de herramientas automatizadas para gestión de colecciones de datos, sistemas manejadores de bases de datos y tecnologías de bodegas de datos. La comunidad académica ha mostrado un gran interés por descubrir información o *estructuras ocultas* en bases y bodegas de datos; que es conocido como ***Minería de datos,*** y más aún, en el conocimiento de dichas estructuras en las series de tiempo.

1.2 Planteamiento del problema

El planteamiento del problema, está fundamentado en lo siguiente:

"Será posible generar, diseñar y realizar un método que integre técnicas de minería de datos de series de tiempo y alguna técnica clásica de análisis de series de tiempo en una herramienta computacional, para proporcionar una caracterización y clasificación de series de tiempo"

1.3 Objetivo de la tesis

El objetivo principal de la tesis es el siguiente:

1. La generación, integración e implementación, en una herramienta computacional, de las técnicas de minería de datos de series de tiempo y mapas recurrentes.

Los objetivos particulares a continuación se enlistan:

- Proporcionar un método que integre y utilice diferentes técnicas de minería de datos para análisis de series de tiempo, es decir, utilizar la minería de datos y el análisis clásico de series de tiempo para proporcionar alguna caracterización para utilizar dichos resultados para su análisis.

- Mostrar y proporcionar en una herramienta computacional, que dicho método funciona en forma experimental y corroborando por una técnica clásica de análisis de series de tiempo. En bi-proceso de almacenamiento y recuperación.

- Proporcionar un lineamiento para futuras investigaciones en la minería de datos de series de tiempo.

1.4 Delimitación del tema

El presente trabajo se limita a proporcionar, al menos tres métodos de minería de datos en forma integral y funcional. Enfatizando su representación para almacenamiento y recuperación de las series de tiempo a analizar.

Este trabajo hace énfasis en el análisis de series de tiempo que representan sismos, pero no se limita para el análisis de otro tipo de eventos representados como series de tiempo, como pueden ser, datos científicos, financieros, médicos, ambientales o experimentales. Aunque no podemos asegurar que el método sea generalizable para cualquier tipo de series de tiempo, ya que existen teoremas limite que demuestran que no es posible y conocido como NFL. [31]

Este trabajo sólo utilizará como técnica de análisis clásico de series de tiempo el método llamado mapas recurrentes. Así como el análisis experimental de algoritmos de minería de datos como son técnicas indexadas, clasificación y agrupación de series de tiempo.

1.5 Contribución de la tesis.

- Generar una clasificación y caracterización de series de tiempo, utilizando similitud y los estimadores de los mapas recurrentes.

- Proporcionar una herramienta computacional que utiliza técnicas de minería de datos para la generación de clasificación y caracterización de las series de tiempo

- Mostrar que la similitud puede generar clasificaciones de las series de tiempo

Cabe mencionar que los mapas recurrentes son de gran importancia para corroborar y/o validar la clasificación de las series de tiempo utilizando las técnicas de minería de datos.

Además de que los mapas recurrentes nos proporcionan estimadores intrínsecos para analizar patrones ocultos, cambios de estructura en los datos, así como determinar similitud entre las series de tiempo a analizar. Esto con el fin de complementar las clasificaciones existentes [13].

1.6 Organización de la tesis.

Este trabajo consta de los siguientes capítulos que se presentan de la siguiente forma:

- Capítulo 1

  Aquí se presenta, y muestra una breve introducción, planteamiento del problema, objetivo de la tesis, delimitación del tema así como la contribución principal del trabajo.

- Capítulo2

  Este capítulo proporciona el marco de trabajo para la tesis, estado del arte, así como los conceptos fundamentales en los que se sustenta este trabajo, como son: series de tiempo, clasificación, similitud, descubrimiento de conocimiento en bases de datos y minería de datos.

- Capítulo 3

  En este capítulo se proporciona una introducción a lo que es la minería de datos de series de tiempo incluyendo sus fundamentos teóricos.

  Además de mostrar y proporcionar la problemática de cómo representar a las series de tiempo con diferentes técnicas para generar su análisis.

- Capítulo 4

  Este capítulo muestra el desarrollo de los algoritmos a utilizar para el método experimental, como son, la indexación de series de tiempo por métodos espaciales de acceso, la clasificación simbólica de series de tiempo y la generación de grupos de series de tiempo utilizando una representación de series de tiempo basado en ondeletas, así como mostrar dicho método en

términos experimentales e integrando dichos algoritmos para su aplicación a la clasificación de series de tiempo.

- Capítulo 5

Se proporcionan algunos de los resultados obtenidos por la aplicación del método así como los resultados obtenidos, además de mostrar las conclusiones y los posibles trabajos futuros que se pueden realizar en base a este trabajo o para dar continuidad al desarrollo de nuevos métodos de análisis de series de tiempo utilizando la minería de datos de series de tiempo.

- Capítulo 6

En este capítulo se muestran los anexos del trabajo de tesis, principalmente se muestra el diseño de una aplicación prototipo para el análisis de series de tiempo utilizando técnicas de minería de datos y mapas recurrentes.

- Capítulo 7

Aquí se proporcionan los requerimientos, instalación, descripción y uso de la herramienta computacional desarrollada en este trabajo.

- Capítulo 8

En este capítulo se presentan las principales referencias utilizadas en este trabajo, así como las referencias básicas en la minería de datos de series de tiempo.

La siguiente figura proporciona una representación visual del trabajo de tesis, así como la relación entre cada uno de los capítulos de la misma.

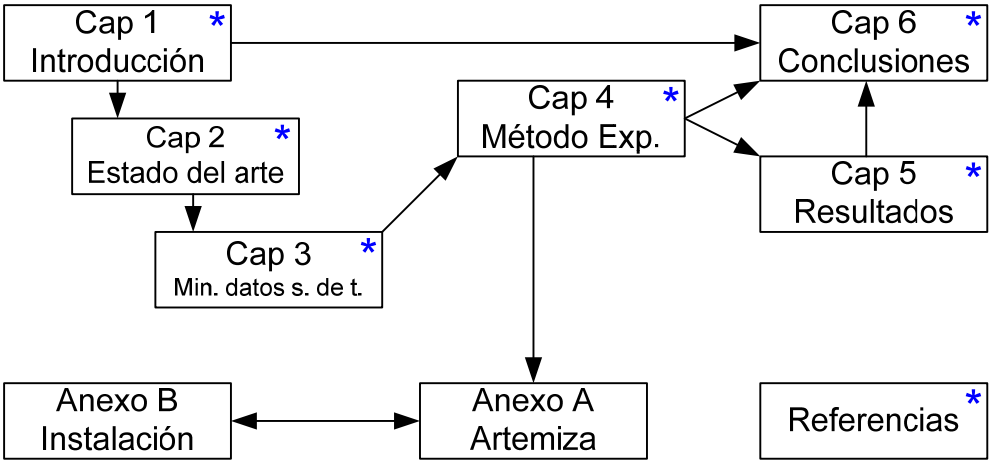


Fig. 1. Organización gráfica de la tesis.

Capítulo 2
Estado del arte.

El presente capítulo proporciona el estado del arte, introducción, antecedentes, conceptos y un contexto de trabajo referentes a series de tiempo, clasificación, similitud, mapas recurrentes, descubrimiento de conocimiento en datos y minería de datos, el cual será usado para el desarrollo de este trabajo.

2.1 Antecedentes

El problema de clasificar y encontrar similitud en bases de datos de series de tiempo ha tenido mucha atención en el campo de minería de datos, tanto que ahora se puede decir que es una gran herramienta para el análisis de series de tiempo y así extraer algún conocimiento de los fenómenos.

La minería de datos de series de tiempo permite la generación de alguna clasificación de éstas y así generar relaciones de semejanza a partir de un conjunto de características de los fenómenos que son observados para su análisis.

Además de que la minería de datos está relacionada con muchas áreas como son, machine learning, estadística y diseño de bases de datos, etc. Ésta usa técnicas como clasificación, reglas de asociación, visualización, árboles de decisión, regresiones no lineales, modelos probabilísticos para identificar y descubrir estructuras ocultas de forma novedosa en grandes bases de datos.

Algunos de los investigadores que aplican la minería de datos para encontrar patrones en las series de tiempo son Berndt y Clifford [17], Keogh y Lin [1, 2, 3, 8, 9], Rosenstein y Cohen [26], Agrawal, R y Faloutsos, C [4,16, 19].
Berndt y Clifford usan una técnica llamada “dynamic time warping” tomada del reconocimiento de patrones en voz. Su enfoque usa programación dinámica para alinear las series de tiempo y así predefinir un conjunto de plantillas de series de tiempo.

Rosestein y Cohen [26] también usan un conjunto de *templates* o platillas predefinidass y aquí ellos aplican un proceso embebido en el tiempo de retardo para encontrar los conjuntos predefinidos.

Keogh y Lin [1, 2, 3, 8, 9] representan a las series de tiempo usando un método llamado “Modelo de Porciones Lineales y Porciones Constantes”, aquí se utiliza un enfoque de estadística descriptiva, para encontrar los patrones en las series de tiempo. Además de introducir una representación simbólica de las series fundamentado en un modelo estadístico y así utilizar métodos de bioinformática, para la búsqueda de patrones, o mejor dicho, búsqueda de “motivos” en las series de tiempo.

Agrawal, R y Faloutsos [4, 19] proponen un método indexado de series de tiempo

fundamentado en la representación de que una serie de tiempo puede ser vista como un punto N-dimensional y en virtud de ello utilizar métodos espaciales de acceso, como lo son los R-Tree [6, 7].

En ambos casos de Keogh y Agrawal, proporcionan una representación, de las series de tiempo, diferente de la original, teniendo que generar una métrica que pueda ser utilizada en dicha representación. Estas representaciones pueden influir en el análisis de las series de tiempo. Además de que con ellas es necesario definir una métrica para esta representación, es muy importante ésta, ya que prescindir de ella, no seria posible analizar las series de tiempo.

2.2 Series de tiempo.

En el mundo existen diferentes tipos de fenómenos, estos pueden ser observados con diferentes dispositivos, pero cada uno de ellos es esencialmente observado en algún periodo de tiempo dando como resultado una serie de tiempo de este fenómeno.

Una serie de tiempo es un conjunto de valores en un periodo de tiempo, en la literatura existen diferentes definiciones. Algunos investigadores ven a las series de tiempo sólo como valores numéricos, otros son especificados en cada intervalo de tiempo, además de que las series de tiempo pueden ser continuas o discretas. En este texto se tomará una forma general, basada en todas las cuestiones anteriores. Como vemos en la definición 2.1, una ***serie de tiempo***, es un conjunto de valores sobre algún periodo de tiempo.

Def. 2.1 Una **serie de tiempo** es un conjunto de ***n*** valores $\{\langle t_1, a_1\rangle, \langle t_2, a_2\rangle, \ldots, \langle t_n, a_n\rangle\}$. Los valores son identificados por puntos específicos bien definidos en el tiempo, en tal caso los valores pueden ser vistos como un vector $\langle a_1, a_2, \ldots, a_n\rangle$.

Dada la anterior definición, podemos decir que es una **subserie o subsecuencia** de alguna serie de tiempo.

Def. 2.2 Una serie de tiempo $Y' = \langle y_{i_1}, \ldots, y_{i_m}\rangle$ es una **subserie** de otra serie de tiempo $Y = \langle y_1, \ldots, y_m\rangle$ *si* $\forall 1 \le j \le m-1, i_j < i_{j+1}$ $y \forall 1 \le \mathrm{j} \le \mathrm{m}, \exists 1 \le \mathrm{k} \le \mathrm{n}$ *tal* que $\mathrm{y}_{\mathrm{i}_\mathrm{j}} = y_k$

Para la realización de algún tipo de aplicación de minería de datos de series de tiempo se debe de considerar la ***similitud*** entre dos series de tiempo de tal forma que dados algunos valores conocidos se puedan llegar a determinar algunos valores. Alternativamente dada una serie de tiempo quisiéramos encontrar a cual grupo pertenece mejor en base a la similitud (clasificación). La similitud se presenta mas adelante en la sección 2.2.

Un tipo especial de análisis de similitud es aquel que trata de identificar ***patrones*** en las series de tiempo. En la Fig.1 se muestra una gráfica de una serie de tiempo.

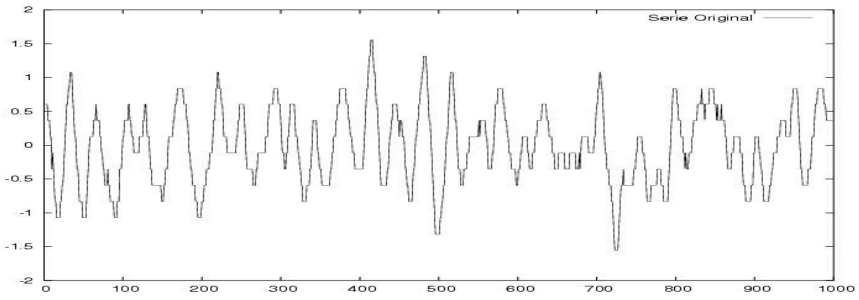


Fig. 2. Gráfica de una serie de tiempo.

2.1.1 Series de tiempo caóticas

En esta sección se presenta una definición de lo que son las series de tiempo caóticas ya que son un tópico en particular en la investigación del análisis de las series de tiempo y por lo tanto es importante mencionarlas.

El caos clasifica las señales en un rango de valores entre las señales sinusoidales, impredecibles, cuasi-periódicas y las de comportamiento completamente estocástico [13]. Se puede decir que una serie de tiempo caótica es aquella generada por un proceso no lineal determinístico con muchas condiciones iniciales que tienen un espectro muy grande de frecuencia [13].

El lenguaje para definir una serie de tiempo caótica se toma de la teoría de sistemas dinámicos, la cual estudia la trayectoria de fluidos (ecuaciones diferenciales) y dinámicas no lineales. El concepto clave para describir una serie de tiempo caótica es un atractor caótico.

Def. 2.3 En la teoría de sistemas dinámicos, un sistema con *dinámica* $f(t,\bullet)$, el atractor $\wedge$ es un subconjunto del espacio de fase tal que, existe un vecindario de $\wedge$, el atractor base, tal que converge a algún punto contenido en $\wedge$ y $f(t,\wedge) \supset \wedge$ para un valor “largo” de t.

En las siguientes figuras se muestra la serie de Lorenz, así como su correspondiente figura de su atractor.

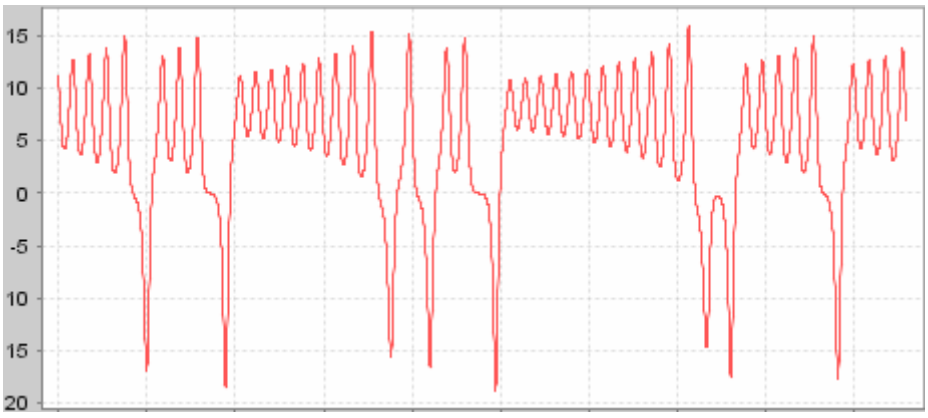


Fig. 3. Serie de tiempo Lorenz.

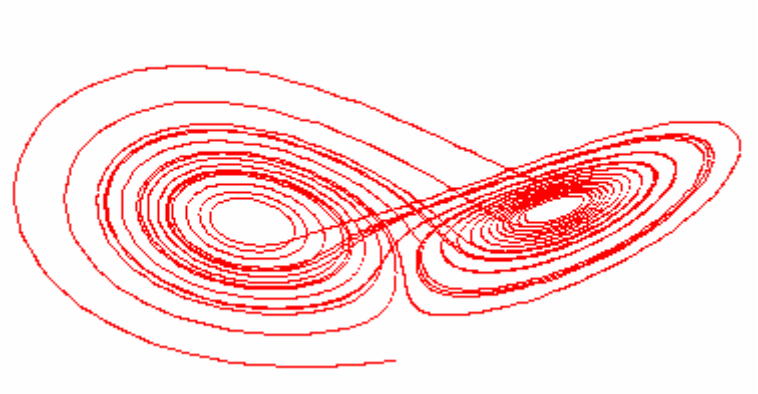


Fig. 4. Atractor de la serie de Lorenz.

2.1.2 Análisis de series de tiempo.

El análisis de series de tiempo puede ser visto como la tarea de encontrar patrones en los datos y predictibilidad de valores. La detección de patrones puede incluir:

- **Tendencias (Trend):** El análisis de tendencias puede ser visto como cambios sistemáticos no repetitivos (lineales o no lineales) de algún valor sobre el tiempo**.** Un ejemplo podría ser el valor de una acción cuando continuamente sube de precio.
- **Cíclicos:** Aquí el comportamiento observado es cíclico.
- **Periódicos:** En este los patrones detectados se repiten en base a un periodo de tiempo, ya sea por año, mensual o día. Un ejemplo de ello es cuando los volúmenes de venta aumenta en la temporada navideña.
- **Detección de anomalías (outliers):** Para ayudar a encontrar patrones, la técnica de detección de anomalías, elimina mucho de los llamados *falsos positivos*.

2.2 Clasificación

La clasificación es quizás la técnica más popular y familiar de la minería de datos. La clasificación mapea datos entre grupos predefinidos o *clases.*
La mayoría de los algoritmos de clasificación asumen algún conocimiento de los datos y/o se realizan fases de entrenamiento para estas clasificaciones. El problema de la clasificación se define de la siguiente forma:

Def. 2.4 Dada una *Base de Datos* $D = \{t_1, t_2, \ldots, t_n\}$ con series de tiempo, y un conjunto de clases $C = \{C_1, \ldots, C_m\}$, el **problema de clasificación** es el definir una función $f : D \to C$ donde cada t_i es asignado a una clase. Y una **clase**, C_j contiene, precisamente a las series mapeadas en ella, esto es: $C_j = \{t_i \mid f(t_i) = C_j, 1 \le i \le n, \text{ y } t_i \in D\}$

En la literatura existen tres métodos diferentes para solventar este problema:

- **Especificación de límites.** Aquí la clasificación se realiza dividiendo el espacio de entrada en regiones, donde cada región es asociada a una clase.

- **Aplicando distribuciones probabilísticas.** Para cualquier clase dada C_j, su función de distribución probabilística será: $P(t_i \mid C_j)$, si la probabilidad de ocurrencia para cada clase, $P(C_j)$, es conocida (quizás determinada por un dominio experto), entonces $P(C_j)\, P(t_i \mid C_j)$ es usada para estimar la probabilidad de que t_i está en la clase C_j.
- **Usando probabilidad a posteriori.** Dado un valor t_j, nos gustaría determinar la probabilidad de que t_j pertenece a la clase C_j. Esto es denotado como $P(C_j \mid t_i)$ y es llamada *probabilidad posteriori.* Un enfoque de la probabilidad posteriori puede ser determinarla para cada clase y después asignar el valor t_j a la clase con mayor probabilidad.

2.3 Problemática de la clasificación.

En el estudio de clasificación existen algunos problemas relacionados que a continuación se enlistan para cuando se quiera abordar la **Def. 2.4** y se deban de tomar en cuenta.
Datos faltantes (*Missing Data*): Los datos faltantes causan problemas en la fase de entrenamiento para la generación de las clases y en el proceso de clasificación. Para manejar este problema existen algunos enfoques:

- Ignorar los datos faltantes.
- Asumir el valor de los datos faltantes. Estos pueden ser determinados por métodos de modelado de datos.

- Asumir un valor especial para los datos faltantes. Esto significa que los datos faltantes, pueden asumir el valor especificó para cada uno de ellos.

Rendimiento de las métricas: El rendimiento de las métricas de los algoritmos de clasificación son evaluadas en base a la efectividad de la clasificación, es decir, dada una clase especifica C_j y algún valor t_j, este valor puede o no puede estar asignado a la clase, aunque dicho valor ya éste o no asignado a la clase.

Entonces en consecuencia se generan los siguientes valores que a continuación se describen:

- **Verdadero positivo:** t_j es clasificado en C_j y ya estaba en ella.
- **Falso positivo:** t_j es clasificado en C_j pero no estaba en ella.
- **Verdadero negativo:** t_j no está clasificado en C_j y no esta en ella.
- **Falso negativo:** t_j no está clasificado en C_j pero si esta en ella.

2.4 Algoritmos de clasificación.

A continuación se proporciona una breve explicación de los algoritmos de clasificación que por cuestiones históricas y para nuestro marco de estudio son importantes mencionar.

2.4.1 Algoritmos estadísticos.

Este tipo de algoritmos se fundamentan en la utilización de estadística no descriptiva.

2.4.1.1 Regresión

La regresión trata de estimar los valores de salida basado en los valores de entrada, y cuando es usada para clasificación los valores pueden ser tomados de la base de datos ***D*** y los valores de salida pueden ser representados por las clases. También puede ser usada para solventar el problema de la predictibilidad.

La regresión puede ser usada para realizar dos tipos de clasificación esencialmente:

1. **División**: Los datos son divididos en regiones utilizando las clases.
2. **Predicción**: Pueden generarse fórmulas que predigan el valor de la clase.

2.4.1.1 Clasificación bayesiana

Con la estadística inferencial y el Teorema de Bayes, proporciona una técnica que estima el vecindario de una propiedad, dado el conjunto de datos como entrada.

Supongamos que cualquiera de las *hipótesis* h_1 o la *hipótesis* h_2 pueden ocurrir, pero no las dos, además de que x_i es un evento *observable*.

Def. 2.5 Teorema de Bayes.

$$P(h_1 \mid x_i) = \frac{P(x_i \mid h_1)P(h_1)}{P(x_i \mid h_1)P(h_1) + P(x_i \mid h_2)P(h_2)}$$

Aquí $P(h_1 \mid x_i)$ es llamada ***probabilidad posterior,*** donde $P(h_1)$ es la probabilidad ***a priori*** asociada con la hipótesis h_1. $P(x_i)$, es la probabilidad de ocurrencia del valor x_1 y $P(x_i \mid h_1)$ es la probabilidad condicional que, dada la hipótesis, la satisface.

Entonces con esto se puede realizar una simple clasificación llamada ***Naive Bayes***. La contribución dada es que para cada atributo "independiente", se puede determinar su probabilidad condicional.

La clasificación se realiza por el impacto que tienen diferentes atributos sobre los valores predecidos. Esta es llamada "naive", porque, ésta asume la independencia entre varios valores de atributos.

Entonces dado un valor x_i, la probabilidad de que una serie de tiempo, t_i, esté en la clase C_j está dada por $P(C_j \mid x_i)$. Los datos de entrenamiento pueden ser usados para determinar $P(x_i)$, $P(x_i \mid C_j)$ y $P(C_j)$. Con estos valores, el teorema de Bayes, permite estimar la probabilidad posteriori $P(C_j \mid x_i)$ y entonces finalmente $P(C_j \mid t_i)$.

2.4.2 Algoritmos basados en distancia.

La idea de los algoritmos basados en distancia es usar métricas de similitud, dichas métricas pueden hacerse más abstractas y aplicarse a problemas generales de clasificación.

2.4.2.1 Enfoque simple

Estos algoritmos tratan de solventar la siguiente problemática:

Def. 2.6. Dada una base de datos $D = \{t_1, t_2, \ldots, t_n\}$ de series de tiempo, donde cada serie de tiempo $t_i = \{t_{i1}, t_{i2}, \ldots, t_{in}\}$ contiene valores numéricos y un conjunto de clases

$C = \{C_1, \ldots, Cm\}$ donde cada una de las clases $C_j = \{C_{j1}, C_{j2}, \ldots C_{jm}\}$ tiene valores numéricos, entonces el problema de clasificación es el asignar cada t_i a la clase C_j tal que la $sim(t_i, C_j) \geq sim(t_i, C_l) \forall C_l \in C$ donde $C_l \neq C_j$.

El algoritmo 2.1 ilustra una simple forma de clasificación para las series de tiempo, previamente calculando el centro de cada clase C_i, tal que cada serie de tiempo, debe ser comparada con cada uno de los centros de la clase, que generalmente son un grupo fijo de clases y usualmente pequeño.
La complejidad para generar la clasificación de la serie de tiempo, en el peor de los casos, es del orden ***O(n)***.

```
ALGORITMO 2.1
Entrada:
C1,…,Cm              //Centros para cada Clase Ci
t                    // Serie de tiempo a clasificar
Salida:
c                    //Clase donde t fue asignada
//Algoritmo simple utilizando la distancia
dist := ∞;
for i := 1 to m do{
  if sim(ci, t) < dist  then{
     c = i;
     dist = dis(ci,t);
  }
}
```

2.4.2.2 K Vecindarios cercanos (K Nearest Neighbors-- KNN)

Esta técnica, algoritmo 2.2, asume que los datos de entrenamiento son en sí el modelo de los datos, entonces cuando se tiene que hacer una clasificación para una nueva serie de datos a incluir en el modelo, se tiene que calcular su distancia con cada uno de los elementos en el modelo. La serie de tiempo es puesta en la clase que contiene los K elementos mas cercanos a ella. Entonces dado que cada serie de tiempo debe ser clasificada, ésta debe ser comparada para cada elemento en el modelo, si hay *q* elementos en el modelo esto significa que tiene complejidad de orden *O(q)*. Teniendo n elementos a clasificar, esto llega a ser del orden *O(nq),* pero como el tamaño del modelo es constante, se puede ver en forma generalizada como una complejidad de *O(n).*

```
ALGORITMO 2.2
Entrada:
T               //Modelo de datos
K               //Número de vecindarios
t               // Serie de tiempo a clasificar
Salida:
c               //Clase donde t fue asignada
//Descripción del algoritmo KNN
KNN:
N = ∅;
For each d ∈ T do
    if |N| ≤ K then
      N = N ∪ {d};
    else
      if ∃ u ∈ N tal que sim(t,u) ≤ sim(t,d)
then{
        N = N - {u};
        N = N ∪ {d};
      }
c = la clase donde u ∈ N es clasificada.
```

2.5 Clustering (Agrupamientos)

El clustering es similar a la clasificación, ya que trata de agrupar los datos. Sin embargo, aquí los grupos no están definidos. Algunos autores dicen que el clustering es un tipo especializado de "Clasificación", pero en nuestro caso lo mencionaremos como otro enfoque, es decir, temas relacionados pero con sus propias características y diferencias.

En la siguiente definición se menciona la problemática del clustering:

Def. 2.7. Dada una base de datos $D = \{t_1, t_2, \ldots, t_n\}$ de series de tiempo y un valor entero ***k,*** entonces el **clustering** se define como una función $f : D \to \{1, \ldots, k\}$ donde cada t_i es asignado a un cluster $K_j, 1 \le j \le k$. Un **cluster**, K_j, contienen precisamente los elementos mapeados en él, esto es, $K_j = \{t_i \mid f(t_i) = K\ , 1 \le i \le n, \text{y } t_i \in D\}$.

Comúnmente algunos algoritmos de clustering se aplican a datos numéricos y asumiendo algunas métricas para ellos, estos atributos ***Métricos*** satisfacen la desigualdad del triángulo.

Entonces un cluster puede ser descrito con varios valores característicos, es decir:

Def. 2.8 Dado un cluster, K_m de N puntos $\{t_{m1}, t_{m2}, \ldots, t_{mN}\}$ se consideran las siguientes definiciones:

centroide = $C_m = \dfrac{\sum_{i=1}^{N}(t_{mi})}{N}$

radio = $R_m = \sqrt{\dfrac{\sum_{i=1}^{N}(t_{mi} - C_m)^2}{N}}$

diámetro = $D_m = \sqrt{\dfrac{\sum_{i=1}^{N}\sum_{j=1}^{N}(t_{mi} - t_{mj})^2}{(N)(N-1)}}$

2.5.1 K-means

El algoritmo K-means es un algoritmo iterativo de clustering, en el cual los elementos se "mueven" u organizan, en el conjunto de clusters hasta que el conjunto de elementos es alcanzado. Se tiene un alto grado de similitud entre los elementos en cada cluster y un alto grado de disimilitud entre cada cluster.

Def. 2.9 La media del cluster $K_i = \{t_{i1}, t_{i2}, \ldots, t_{im}\}$ se define a continuación:

$$m_i = \frac{1}{m}\sum_{j=1}^{m} t_{ij}$$

El algoritmo 2.4 especifica el algoritmo K-means. Nótese que los valores iniciales de las medias son tomados arbitrariamente, esto puede asignarse aleatoriamente o quizás pueden usarse los primeros k valores.

```
Algoritmo 2.4
Entrada:
D = {t1, t2, ..., tn}   //Conjunto Series de Tiempo
k                       //Número de clusters deseado
Salida:
K                       //Conjunto de clusters
K-means:
asignar valores medias iniciales m1, m2, ..., mk;
 repetir
   asignar ti al cluster con media más cercana;
   calcular la nueva media para cada cluster;
 hasta conocer el criterio de convergencia;
```

La figura siguiente representa gráficamente el algoritmo k-means, con una k = 2 y éste para cuando el criterio de terminación es alcanzado.

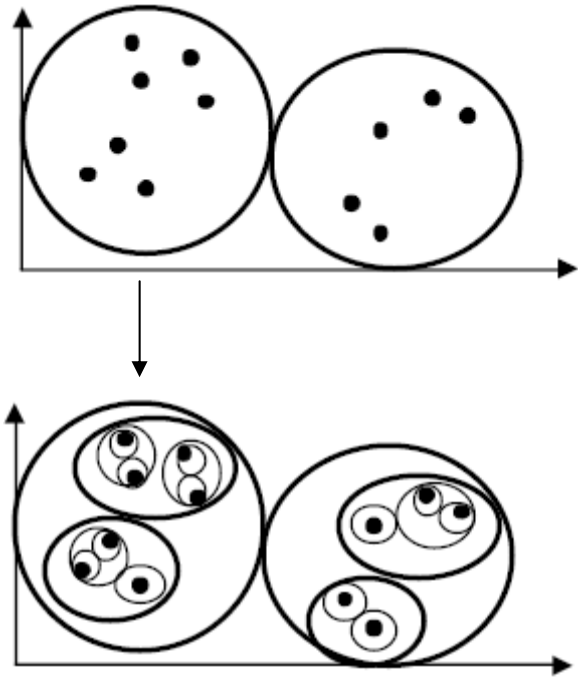


Fig. 5. Representación gráfica del algoritmo K-means.

2.6 Similitud.

El problema de similitud, es considerado desde la antigüedad, como algo que se puede resolver en base a la identificación o comparación de dos cosas, es decir se necesita la utilización de la vista o percepción, pero esta claro que en los problemas de cómputo eso podría ser inaceptable, motivo por el cual se usan ***métricas de similitud****, que se muestran en la siguiente sección.* Además de que la similitud es un problema abierto en el campo de las matemáticas y en cómputo, también ésta se fundamenta en lo que es la topología y los espacios métricos.
Topológicamente hablando, la similitud es una función que proporciona un valor positivo, para dos puntos cualesquiera que *son cercanos en un espacio métrico*.

La definición de similitud puede variar en la literatura y por autor, dependiendo de las propiedades deseadas, pero las propiedades básicas de la función de similitud son las siguientes:

- Es definida positivamente; es decir, $\forall(a,b), S(a,b) \geq 0$ talque a, b, c $\in T$ y *T* es una topología métrica.
- Proporcionan auto similitud; es decir, $S(a,b) \leq S(a,a)$ y $\forall(a,b), S(a,b) = S(a,a) \Leftrightarrow a = b$.
- Reflexiva; es decir, $\forall(a,b)\ S(a,b) = S(b,a)$.
- Finita; es decir, $\forall(a,b)\ S(a,b) < \infty$.
- Desigualdad del triangulo, es decir, $\forall(a,b,c)\, S(a,b) \leq S(a,c) + S(b,c)$

Fig. 6. Problema de similitud.

2.6.1 Métricas de similitud.

La mayoría de las métricas de similitud asumen valores numéricos y éstas pueden dificultar un uso generalizado para diferentes tipos de datos no numéricos. A continuación se define la similitud entre dos series de tiempo.

Def. 2.10 La similitud entre dos series de tiempo $X_i, \mathrm{y}\, \mathrm{X_j}, \mathrm{S}(X_i, \mathrm{X_j})$ en alguna base de datos de series de tiempo *D,* es una función de D x D a el rango [0, 1]. Entonces $\mathrm{S}(X_i, \mathrm{X_j}) \in [0,1]$.

Entonces las siguientes características son deseables para alguna métrica de similitud:

$\forall\, \mathrm{X_i} \in D, S(X_i, X_i) = 1$

$\forall\, \mathrm{X_i}, X_j \in D, S(X_i, X_j) = 0$

$\forall\, \mathrm{X_i}, X_j, X_t \in D, S(X_i, X_j) < S(X_j, X_t)$

Y a continuación se mencionan las métricas de similitud utilizadas en la literatura.

Definiciones 2.11 Métricas de similitud:

Dice: $S(X_i, X_j) = \dfrac{2\sum_{h=1}^{k} x_{ih} x_{jh}}{\sum_{h=1}^{k} x_{ih}^2 + \sum_{h=1}^{k} x_{jh}^2}$

Jaccard: $S(X_i, X_j) = \dfrac{\sum_{h=1}^{k} x_{ih} x_{jh}}{\sum_{h=1}^{k} x_{ih}^2 + \sum_{h=1}^{k} x_{jh}^2 - \sum_{h=1}^{k} x_i x_j}$

Coseno: $S(X_i, X_j) = \dfrac{\sum_{h=1}^{k} x_{ih} x_{jh}}{\sqrt{\sum_{h=1}^{k} x_{ih}^2 \sum_{h=1}^{k} x_{jh}^2}}$

Overlap: $S(X_i, X_j) = \dfrac{\sum_{h=1}^{k} x_{ih} x_{jh}}{\min(\sum_{h=1}^{k} x_{ih}^2, \sum_{h=1}^{k} x_{jh}^2)}$

Euclidiana: $S(X_i, X_j) = \sqrt{\sum_{h=1}^{k} (x_{ih} - x_{jh})^2}$

Manhatan: $S(X_i, X_j) = \sum_{h=1}^{k} | x_{ih} - x_{jh} |$

Algo que se debe mencionar es que para compensar las escalas entre los valores de cada variable, cada valor puede ser normalizado dentro de algún rango, por ejemplo el rango [0,1].

2.6.2 Problemática de las métricas de similitud.

Además de las métricas mostradas, existen otras como la correlación lineal, la transformada discreta de Fourier, la transformada discreta de ondeletas. Pero con ellas existen los siguientes problemas comunes:

- **Longitud:** *X* y *Y* pueden tener diferentes longitudes, pero pueden ser, muy similares.
- **Escala:** Aunque la forma general de *X* y *Y* pueden ser idénticas, la escala puede ser diferente.
- **Cortaduras (gaps):** Alguna de las series pueden tener valores faltantes que pueden existir en la otra serie.
- **Anomalías:** Es similar al problema de cortaduras, excepto que los valores pueden ser generados por lecturas erróneas.
- **Línea base o de fondo (Baseline):** El valor actual de la línea base puede diferir, esto significa, que el tiempo sucesivo entre dos series de tiempo *X* y *Y* puede ser diferente.

2.7 Mapas recurrentes.

Los mapas recurrentes son una técnica de análisis muy valiosa que proporciona una detección e identificación gráfica de patrones y estructuras ocultas en las series de tiempo. Además de proporcionar una técnica visual de similitud entre las mismas.

La idea principal de los mapas recurrentes, está fundamentado en que las series de tiempo observadas, están realizando algún proceso dinámico en el tiempo y que interactúan con algunas variables relevantes.

Def 2.12 Un Mapa Recurrente se define como:

$$R(i,j) = \Theta(\varepsilon - ||\vec{x}(i) - \vec{x}(j)||), \quad \vec{x}(i) \in \mathbf{R}^m, \quad i,j = 1,\ldots,N,$$

Donde *N* es el número de estados considerados para $\vec{x}(i)$, ε un factor de distancia, $||\cdot||$ una norma (i.e. Norma Euclidiana) y $\Theta(\cdot)$ una función de cambio de unidad.

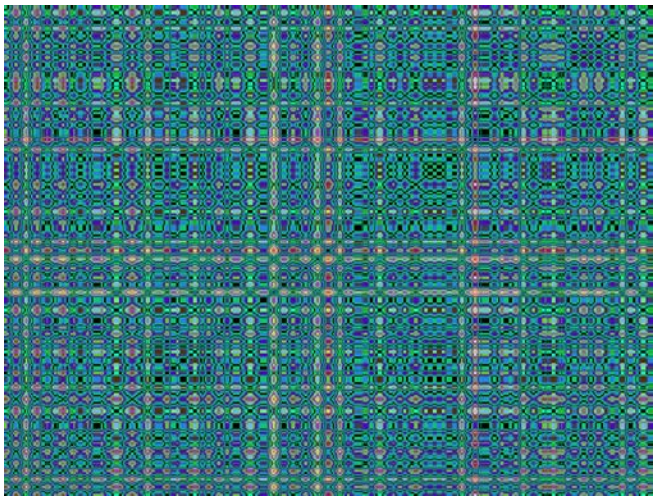


Fig. 7. Mapa recurrente de una serie de tiempo.

En particular estamos interesados en los siguientes estimadores de los mapas recurrentes para el análisis que proporcionan información correspondiente a la dinámica de la serie de tiempo en cuestión y que servirán para la verificación de nuestras técnicas de minería de datos:

Definiciones 2.13 Estimadores de los mapas recurrentes utilizados para este trabajo:

- **Entropía Shannon:** $H(x) = -\sum_{i=1}^{n} p(i) \log_2 p(i)$

- **Determinismo:** $DET = \dfrac{\sum_{l=l_{\min}}^{N} lP(l)}{\sum_{i,j=1}^{N} R(i,j)}$

  Donde *P(l)* es la distribución de frecuencia.

- **Recurrencia:** $RR = \dfrac{1}{N^2} \sum_{i,j=1}^{N} R(i,j)$, correlación.

- **Entropía Espacio Temporal:**

$$EST_{i,j,t}^{w} = -\sum_{Y,U,V \in \Omega} p_{i,j,t}^{w}(Y,U,V) \log p_{i,j,t}^{w}(Y,U,V)$$

Donde:

w: Es el tamaño de la ventana, es decir, área a analizar.

Y,U,V: Son el espacio de colores a utilizar.

Ω : Es la cuantificación del espacio de colores.

$p_{i,j,t}^{w}(Y,U,V)$: Es la función de distribución de frecuencia.

i,j: Denotan la posición del píxel.

t:: Es la duración de la observación.

2.8 Descubrimiento de conocimiento en bases de datos (KDD).

El descubrimiento del conocimiento de la base de datos se define como sigue:

Def. 2.13 KDD: Es un proceso no trivial de identificación válida, novedoso, potencialmente útil, y muy entendible para la búsqueda de patrones en conjuntos de datos. [32]

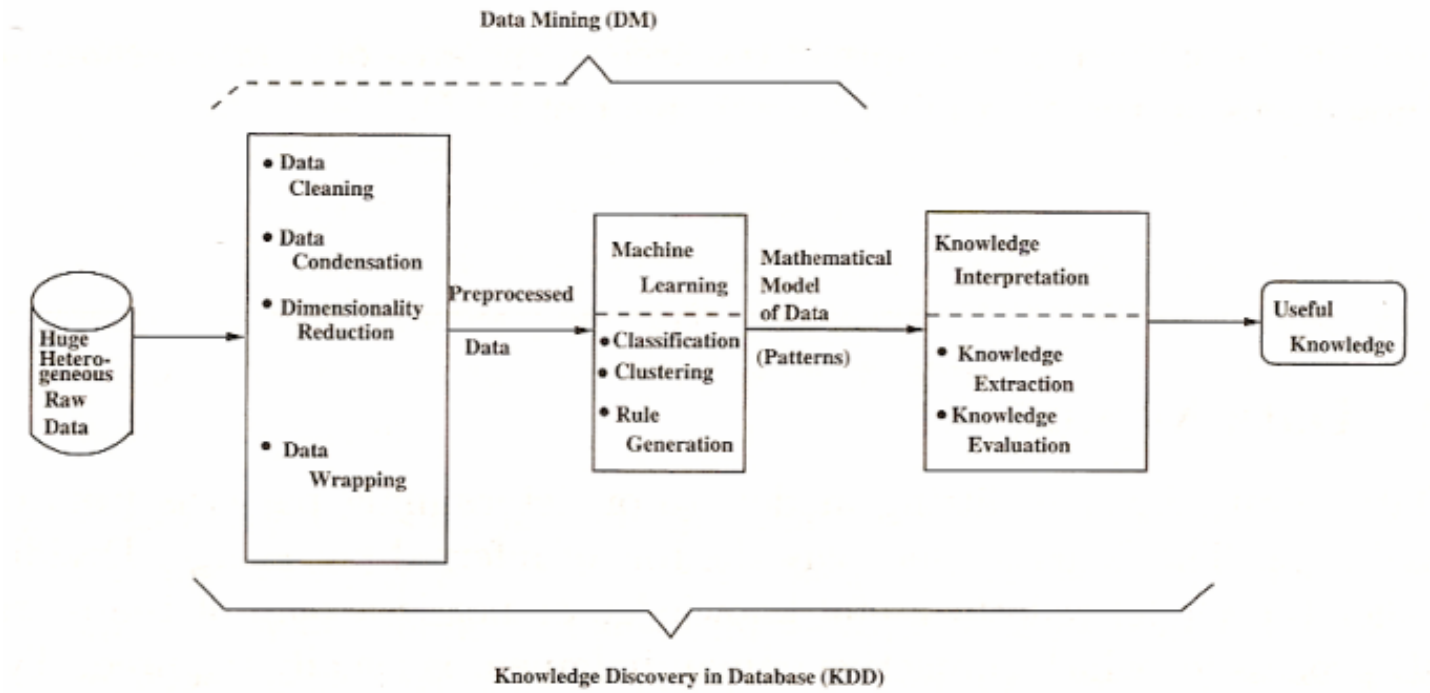


Fig. 8. Representación del descubrimiento de conocimiento en bases de datos. [32]

Este proceso es interactivo e involucra los siguientes pasos:

1. **Limpieza y preprocesamiento de datos:** Incluye operaciones básicas, tal como remover ruido y el manejo de datos perdidos.

2. **Proyección y condensación de datos:** Incluye encontrar características útiles y ejemplos para presentar los datos usando reducción de dimensiones o métodos de transformación.

3. **Descripción e integración de datos:** Integrar múltiples y heterogéneas fuentes de datos para proporcionar sus descripciones para un fácil manejo.

4. **Determinar funciones, minería de datos y los algoritmos:** Se define el propósito del modelo así como los métodos a usar para las búsquedas de patrones en los datos.

5. **Minería de datos:** Incluye la búsqueda de patrones, en una forma de representación o por un conjunto de representaciones.

6. **Interpretación y visualización:** Incluye la interpretación de los patrones descubiertos, así como la posible visualización de estos patrones.

7. **Usar el conocimiento encontrado:** Incluye la incorporación del conocimiento encontrado para tomar acciones con este conocimiento.

2.9 Minería de datos.

La minería de datos corresponde al ajuste de modelos o la determinación de patrones de un conjunto de datos observados. Típicamente, un algoritmo de minería de datos se constituye de alguno de los siguientes tres componentes.

- **El modelo:** Contiene los parámetros que serán determinados de los datos, en éste incluye su función y una forma de representación.
- **El criterio de preferencia:** Básicamente la preferencia de un modelo o el conjunto de parámetros de otro, dependen del conjunto de datos de entrada, entonces el criterio es básicamente ajustar la función del modelo hacia los datos o generar un modelo con grados de libertad que se ajuste con los datos de entrada.
- **El algoritmo de búsqueda:** Esto se refiere a la especificación del algoritmo para buscar en modelos particulares y parámetros, dados los datos, modelos y el criterio de preferencia.

Un algoritmo de minería de datos en particular es una instancia concreta del modelo, preferencias y búsquedas.

2.9.1 Aplicación de la minería de datos.

Comúnmente las tareas o funciones de un modelo de minería de datos incluyen:

1. **Descubrimiento de reglas de asociación:** Describe la relación asociada entre diferentes atributos. Su origen esta fundamentado en el análisis de mercados.
2. **Cluster:** Los datos son mapeados solamente a un cluster de muchos de ellos, donde los cluster son grupos naturales de datos que se generaron utilizando métricas de similitud o modelos probabilísticos.
3. **Clasificación:** Se clasifican los datos entre una de las muchas clases predefinidas.
4. **Análisis secuencial.** El objetivo es el modelado de procesos generando secuencias o tendencias sobre el tiempo, usualmente usado para el modelado de series de tiempo.
5. **Regresión:** Ésta es usada en diferentes tipos de predicción y en el modelado de aplicaciones.

6. **Condensación:** Proporciona una descripción compacta de un subconjunto de datos. Estas funciones se usan para el análisis interactivo, para la generación de reportes y minería de texto.
7. **Modelado de dependencias:** Describe dependencias significativas entre las variables.

Estas tareas pueden se utilizadas en diferentes campos siempre y cuando se cuente con un gran volumen de información ya sea centralizada o descentralizada y algunos de esos campos pueden ser los siguientes:

- **Investigación financiera.**
- **Cuidado de la salud.**
- **Manufactura y producción.**
- **Telecomunicaciones.**
- **Científica.**
- **WEB.**
- **Medio ambiente**

Capítulo 3
Minería de datos de series de tiempo.

En este capítulo se proporciona una breve y concreta introducción de la minería de datos de series de tiempo, su fundamento teórico, así como indicar las principales áreas de investigación. Además de mostrar los diferentes enfoques, en el ámbito computacional, de cómo ***representar*** una serie de tiempo para su análisis.

3.1 Introducción

La minería de datos de series de tiempos es una contribución importante en los campos del análisis de series de tiempos y la minería de datos. Los métodos utilizados en la minería de datos de series de tiempo son capaces de caracterizar satisfactoriamente series de tiempo periódicas, no periódicas, complejas y caóticas. Estos métodos cubren las limitaciones de las técnicas tradicionales de la series de tiempo ya que adapta los conceptos de la minería de datos para analizar las series de tiempo.

El campo de estudio de la minería de datos de series de tiempo, utiliza lo mejor de las siguientes áreas de estudio: minería de datos, análisis de series de tiempo, procesamiento de señales, ondeletas, algoritmos genéticos, redes neuronales, caos y sistemas dinámicos.

De la minería de datos utiliza el descubrimiento de patrones ocultos. Del análisis de las series de tiempo toma la teoría para analizar series de tiempo lineares y estacionarias. El procesamiento de señales toma la idea del mejoramiento de filtros para obtener una mejor señal, esto está muy relacionado a lo que son ondeletas. De los algoritmos genéticos viene la robustez de métodos de optimización y de los sistemas dinámicos se obtiene la justificación del método, especialmente el teorema de Takens [20] y extensión de *Saber* [28].

3.1.1 Fundamento teórico de la minería de datos de series de tiempo.

Como hemos mencionado la minería de datos de series de tiempo utiliza lo mejor de varias técnicas, para analizar a las series, pero el elemento principal o marco teórico de este análisis está fundamentado en el siguiente teorema:

Def. 3.1 Teorema de Takens [20], establece que, dada una serie de tiempo determinista $\{x_t\}_{t=1}^{N}$, existe una función $F:\Re^m \to \Re$ tal que $x_t = F(x_{t-\tau}, x_{t-2\tau}, ..., x_{t-m\tau})$ donde τ es el factor de retardo y *m* la dimensión embebida.

Por tanto, para un ***m*** suficientemente grande, el teorema garantiza la posibilidad de generar o modelar, la dinámica de una serie de tiempo determinista a partir de sus valores previos.

3.2 Líneas de Investigación de la minería de datos de series de tiempo.

La minería de datos de series de tiempo es del interés de muchos investigadores en la última década, a continuación se enlistan las principales áreas de investigación en la minería de datos de series de tiempo:

1. **Indexación**: Dada una serie de tiempo *Q*, y alguna métrica de similitud *D(Q,C)* encontrar la más similar en la bases de datos de series de tiempos.

2. **Agrupamiento**: Encontrar grupos naturales de series de tiempo en la base de datos utilizando alguna métrica de similitud.

3. **Clasificación**: Dada una serie de tiempo sin etiqueta, asignarla a una o más de las clases predefinidas para su etiquetación.

4. **Condensación:** Dada una serie de tiempo que contiene **n,** puntos de datos y donde **n** es extremadamente grande, crea una representación de la serie de tiempo que contenga los elementos esenciales y que tenga una pequeña representación, donde pequeño significa que esté en una sola pantalla, una sola gráfica, etc.

5. **Detección de anomalías**: Dada un serie de tiempo y un modelo de comportamiento ‘normal’, encontrar todas la serie de tiempo que contienen anomalías o comportamiento ‘no normal’.

3.3 Representación de series de tiempo

Como cualquier otro problema de las ciencias de la computación, la representación de los datos puede afectar directamente en los resultado que se obtengan del análisis y mas aún tratando de representa a las series de tiempo. En la Fig. 6 se proporciona una clasificación de la representación de las series de tiempo para su análisis.

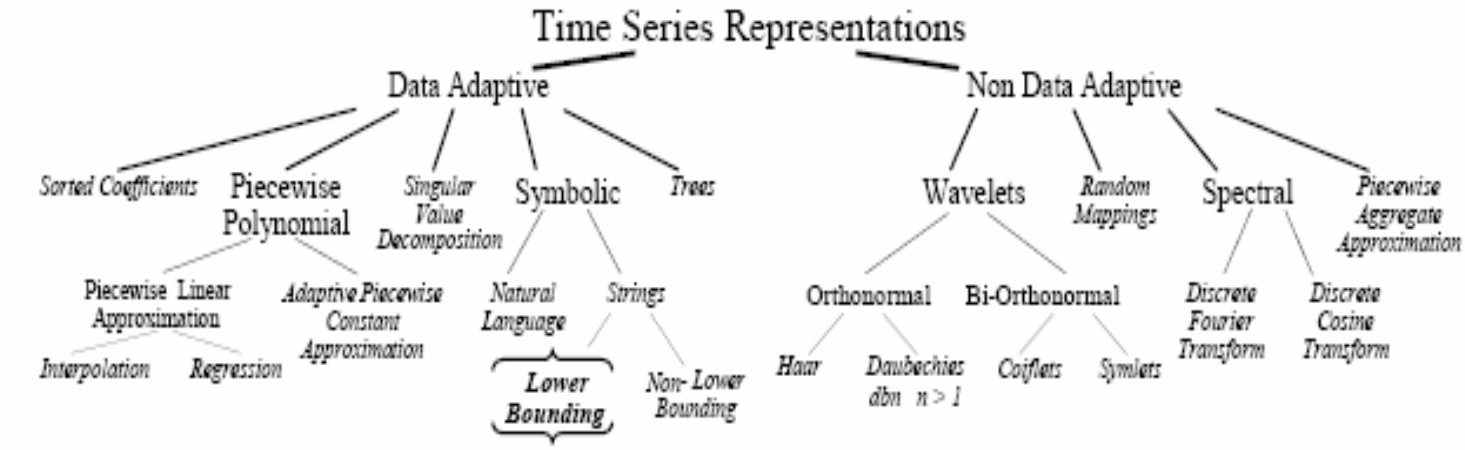


Fig. 9. Representación de las series de tiempo. [9]

Teniendo las diferentes posibles representaciones presente, han surgido muchas representaciones de las series de tiempo de las cuales mencionaremos, que cada una puede ser aplicada según sea el caso o mejor dicho, el tipo de análisis a realizar a las series de tiempo.

3.3.1 Transformada discreta de Fourier. (TDF)

En matemáticas la transformada discreta de Fourier, ampliamente usada en el procesamiento de señales y áreas relacionadas para realizar análisis de frecuencia sobre la señal.

Y para que se pueda utilizar Fourier se requiere de las siguientes características *teóricas* y que no siempre se tienen en la práctica para análisis no teóricos*:*

1. Número infinito de datos
2. Estacionalidad de la serie
3. Linealidad.

La secuencia de ***n*** números complejos $x_0, \ldots, x_{n-1}$ son transformados en otra secuencia de ***n*** números complejos $f_0, \ldots, f_{n-1}$ por la TDF de acuerdo con la siguiente fórmula:

Def. 3.2 Transformada Discreta de Fourier

$$f_j = \sum_{k=0}^{n-1} x_k e^{-\frac{2\pi i}{n} jk} \qquad j = 0, \ldots, n-1$$

La transformada discreta de Fourier inversa está definida por la siguiente fórmula:

Def. 3.3 Transformada Discreta de Fourier Inversa.

$$x_k = \frac{1}{n}\sum_{j=0}^{n-1} f_j e^{\frac{2\pi i}{n}jk} \qquad k = 0, \ldots, n-1.$$

Nótese que el factor multiplicativo de normalización de TDF y la TDFI son (**1** y **1/*n***) y que los signos de los exponentes, son convencionales y difieren en su aplicación. Lo relevante con ellas es que tienen signos opuestos entre si, y esto es dado por una normalización con el factor 1/*n*. La convención de tener signos negativos en el exponente es conveniente ya que significa que f_j es la amplitud de una "frecuencia positiva" 2π*j* / *n*.

Una normalización con $1/\sqrt{n}$ para ambas transformadas las hace unitarias, lo cual tiene muchas ventajas teóricas y mas prácticamente ya que pueden realizarse cambios de escala.

Dada una serie de tiempo podemos aplicarle la TDF para obtener los coeficientes de esta serie y así obtener una nueva representación de la misma serie, la Fig. 10, muestra una serie de tiempo original ***X,*** y la serie de tiempo resultante aplicando la TDF ***X',*** además de demostrar los coeficientes bases de dicha serie.

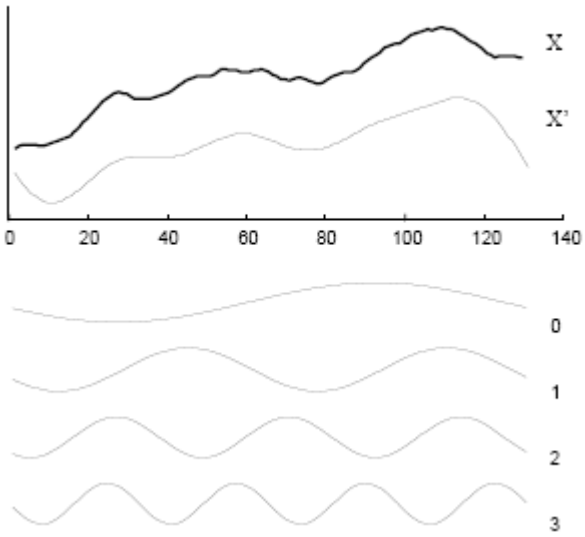


Fig. 10. Serie de tiempo representada con la TDF.

3.3.2 Transformada discreta de ondeletas. (TDO)

Las ondeletas son funciones matemáticas que representan datos u otras funciones en términos de sumas y restas de un prototipo de función, llamada la onda madre. Una importante característica en las ondeletas es que localizan el tiempo. Además

de que procesan datos en diversas escalas o resoluciones, en contraste con TDF donde solamente se consideran los componentes de la frecuencia.

El origen de las ondeletas se puede remontar al trabajo de Karl Weierstrass, en 1873. La construcción del primer sistema ortonormal es de Haar, una contribución importante. Las ondeletas tipo Haar tienen las siguientes propiedades: (1) aproximación con un subconjunto de coeficientes, (2) calculadas rápidamente y fácilmente, requieren de tiempo lineal y la codificación es simple, y (3) preserva distancia euclidiana.

Def. 3.4 Las ondeletas HAAR son definidas de la siguiente forma:

$$\psi_i^j = \psi(2^j x - i) \quad \mathrm{i} = 0, \ldots, 2^{\mathrm{j}} - 1$$

Donde: $$\psi(t) = \begin{cases} 1 & 0 < \mathrm{t} < 0.5 \\ -1 & 0.5 < \mathrm{t} < 1 \\ 0 & \text{otro caso} \end{cases}$$

Además de utilizar su función de escalamiento para ambas:

$$\varphi(t) = \begin{cases} 1 & 0 < \mathrm{t} < 1 \\ 0 & \text{otro caso} \end{cases}$$

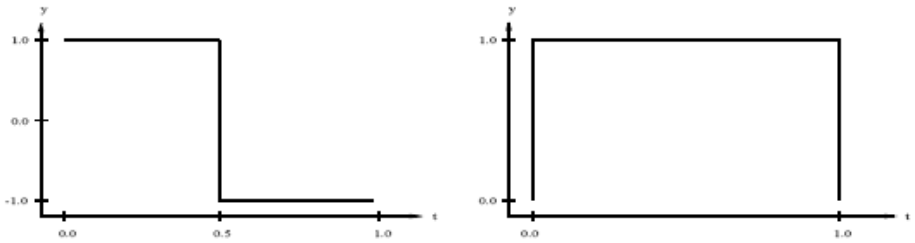


Fig. 11. Ondeleta Haar para $\psi_0^0(t)$ y su función de escalamiento.

La ventaja de usar la TDO es la representación de las señales con multiresolución, ésta tiene propiedades de localización de tiempo y frecuencia, además de que la representación de señales con ésta, pose más información.

Dada una serie de tiempo podemos aplicarle la TDO, para obtener los coeficientes de esta serie y así obtener una nueva representación de la misma serie, la Fig. 12, muestra una serie de tiempo original y la serie de tiempo resultante aplicando la TDO además de demostrar los coeficientes bases de dicha serie.

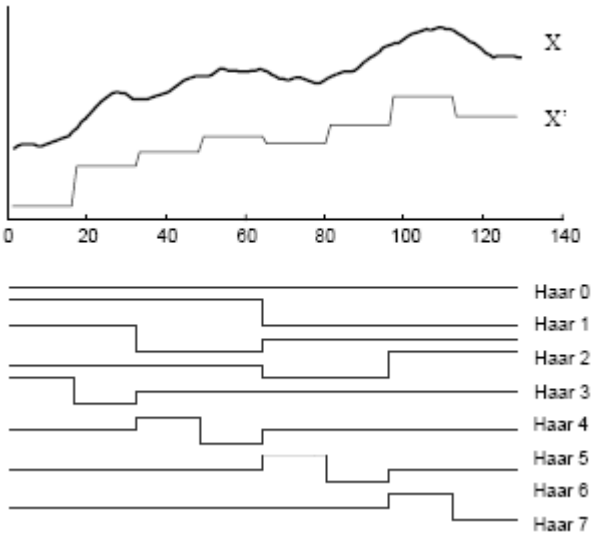


Fig. 12. Serie de tiempo representada con la TDO.

3.3.3 Modelo de porciones lineales (MPL)

El modelo de Porciones lineales proporciona la representación de una serie de tiempo en *segmentos.*

Def. 3.5 Una serie de tiempo X, de tamaño n, su representación N-dimensional será un vector $\overline{X} = \overline{x}_1, \ldots, \overline{x}_N$ donde el i-ésimo elemento es calculado como sigue:

$$x_i = \frac{N}{n} \sum_{j=\frac{n}{N}(i-1)+1}^{\frac{n}{N}i} (x_i - y_i)^2$$

En forma generalizada esta representación también puede producir porciones de la serie original. Entonces dada una serie de tiempo podemos aplicarle la ecuación anterior, para obtener una nueva representación de la misma serie, la Fig. 13, muestra una serie de tiempo original y la serie de tiempo resultante, además demostrar las bases de dicha serie.

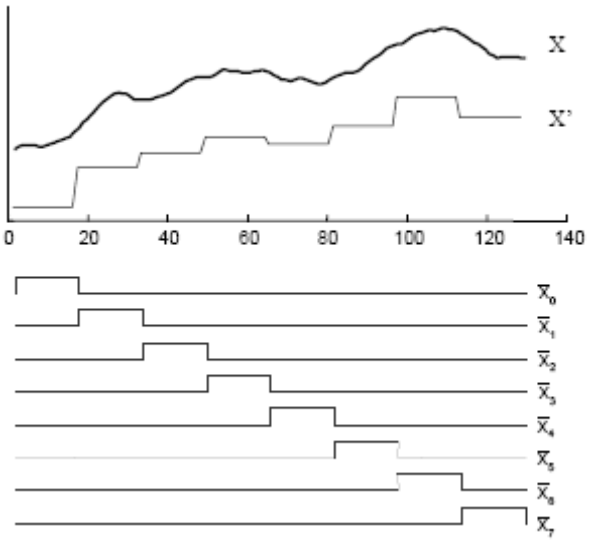


Fig. 13. Serie de tiempo representada por Porciones Lineales.

3.3.4 Descomposición de valores singulares. (DVS)

La descomposición de valores singulares proporciona una técnica muy poderosa para resolver problemas matriciales así como dar soluciones numéricas también.

Def. 3.6 Sea ***A*** una matriz real de *m por n*. La descomposición de valores singulares (DVS) de ***A*** es la factorización $A = U\Sigma V^T$, donde ***U*** y ***V*** son ortogonales y $\Sigma = \mathrm{diag}(\sigma_1, \ldots, \sigma_r)$ $r = \min(m, n)$, tal que $\sigma_1 \geq \cdots \geq \sigma_r \geq 0$.

Si ***A*** tiene números complejos entonces la DVS es $A = U\Sigma V^H$ donde ***U*** y ***V*** son unitarias y Σ, sigue siendo como la anterior, la diagonal con números reales.

Los σ_i, son llamados ***valores singulares***, las primeras ***D*** columnas de ***V*** son los valores singulares derechos y las primeras ***I*** columnas de ***U*** son los valores singulares izquierdos de ***U***.

La descomposición singular de valores difiere de las otras tres previamente señaladas, ya que las transformaciones son locales y examinan un dato o un objeto en cada tiempo que se aplica.

La DVS es una transformación global, el conjunto de datos es examinado completamente. Entonces dada una serie de tiempo podemos aplicarle la DVS, para obtener una nueva representación de la misma serie, en la Fig. 14, se muestra una serie de tiempo original y la serie de tiempo resultante, además de mostrar las bases de dicha serie.

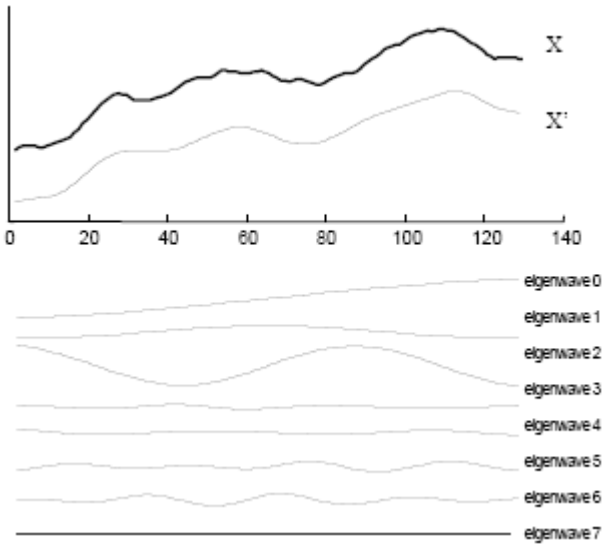


Fig. 14. Serie de tiempo representada por DSV.

3.3.5 Representación simbólica. (RS)

Una Serie de tiempo normalizada, es decir, que tenga una distribución Gaussiana, de tamaño arbitrario ***n*** puede ser discretizada a una cadena de tamaño arbitrario ***w***, donde ***w* < *n*** y ***w* << *n***. Para ello se determinan los ***Puntos Límite o de Quiebra***, que son una lista ordenada de números,
$\beta = \beta_1, \cdots, \beta_{a-1}$, tal que el área entre dos puntos es: $\beta_i \text{ y } \beta_{i+1} = 1/a$.

Def. 3.7 Una serie de tiempo ***C*** de tamaño ***n***, puede ser representada como una palabra $\hat{C} = \hat{c}_1, \cdots;, \hat{c}_w$ donde el ***i-ésimo*** elemento del alfabeto es:

$$\hat{c}_i = alfa_j, \text{sii } \beta_{j-1} \leq \hat{c}_i \leq \beta_j$$

Y el tamaño del alfabeto, usado en la cadena, es un entero arbitrario ***a*** donde ***a*** > 2.

Una serie de tiempo con representación simbólica se muestra en la Fig. 15, una serie de tiempo original y la serie de tiempo resultante.

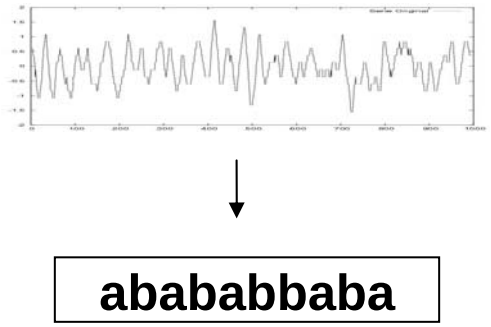


Fig. 15. Serie de tiempo y su representación simbólica.

Capítulo 4
Método experimental

En este capítulo se especifican las técnicas utilizadas así como el método experimental para el análisis y clasificación de series de tiempo, utilizando diferentes técnicas para integrarse en un método de análisis experimental.

4.1 Preprocesamiento de las series de tiempo.

Las series de tiempo contienen mucha información relevante a un fenómeno y comportamiento observado por un sistema, como pueden ser la bolsa de valores, un sistema de control de producción, un ECG, el clima, los sismos, etc. La extracción y comparación de dicha información es fundamental para realizar futuros análisis de extracción de modelos (reglas).

El análisis de series de tiempo requiere de la búsqueda e identificación de *elementos o componentes* que conforman a la serie de tiempo, estos proporcionan información sobre la dinámica del sistema que representa la serie de tiempo.

La extracción y comparación de información, sean elementos o componentes de la serie de tiempo, forman parte de un conjunto de técnicas de minería de datos. La minería de datos consiste en la extracción automatizada de información predictiva o descriptiva a partir de grandes bases de datos.

Como hemos mencionado, pueden tenerse diferentes tipos de representación de las series de tiempo y dependiendo de ésta, pueden variar los resultados del análisis, o mejor dicho dependiendo del tipo de series a analizar podemos utilizar alguna representación para estas series de tiempo.

Para generar un “preprocesamiento” y sin pérdida de generalidad, se sugieren las siguientes prácticas, para alguien más y que esté interesado para aplicar minería de datos a series de tiempo:

a) **Utilizar un número relativamente “grande” de series de tiempo**. Al referirnos a un número “grande” de series de tiempo involucra dos partes y fundamentado en conceptos tradicionales de minería de datos; 1) La longitud de números de datos u observaciones de cada serie de tiempo. 2) La cantidad de series de tiempo a analizar, es decir, para hacer minería de datos se debe tener mucha, pero mucha información para aplicar la minería, sino que caso tendría aplicarla.

b) **Tratar de estandarizar el conjunto de datos**, esto hace referencia a los datos faltantes de los fenómenos observados. En términos generales se sugiere poner un “valor conveniente” a los datos faltantes de la serie de tiempo en cuestión, esta selección puede ser desde lo trivial, utilizar el valor de cero, hasta lo más complejo, que un experto proporcione dicho valor.

c) **Utilizar un conjunto de datos de entrenamiento**, ya que se han hecho previamente los incisos anteriores, pueden proceder a usarlo, aunque muchas veces olvidamos estos puntos.

4.2 Algoritmos

A continuación proporcionamos la descripción de los algoritmos de minería de datos de series de tiempo, utilizados para la generación de la clasificación de series de tiempo, dichos algoritmos son validados por una técnica clásica de análisis de series de tiempo llamada mapas recurrentes.

4.2.1 Indexación

Como hemos mencionado previamente una de las líneas de investigación de la minería de datos de las series de tiempo es la indexación. El problema principal de la indexación es el diseñar métodos de búsqueda rápidos para localizar subsecuencias (query's) en forma exacta o aproximada dentro de alguna base de datos de series de tiempo.

La búsqueda de similitud de patrones en la base de datos es esencial ya que puede ayudar en la predictibilidad, pruebas de hipótesis, descubrimiento de reglas y en general a la minería de datos.

Los query's pueden ser clasificados en dos formas:

- **Búsqueda completa:** Dada una base de datos de series de tiempo ***DB***, con ***N*** número de series de tiempo, un query ***Q*** y un factor de similitud ***e*** se desea encontrar toda las series de tiempo que estén dentro del rango de distancia de ***e*** de ***Q***. Cabe mencionar que todas las series de tiempo de ***DB*** y el ***Q*** deben tener la misma longitud.
- **Búsqueda de subsecuencias:** Dada una base de datos de series de tiempo ***DB***, con ***N*** número de series de tiempo con tamaño arbitrario ***n***, un query ***Q*** y un factor de similitud ***e***, se desea encontrar toda las subsecuencias, tal que ***1<= i <= n***, que estén a distancia ***d,*** tal que: ***d <= e*** de ***q***.

Los métodos de indexación deben de proporcionar las siguientes propiedades (Faloutsos et. Al [19]):

1. Debe ser mucho más rápido que una búsqueda secuencial.
2. El método debe requerir de un pequeño espacio de trabajo.
3. Debe ser capaz de manejar query's de diferentes longitudes.
4. El método debe permitir inserciones y borrados sin tener que regenerar el índice.
5. Debe ser correcto, por ejemplo, no debe haber falsos positivos.
6. Debe ser posible construir el índice, "En tiempo razonable".
7. Debe ser capaz de manejar diferentes métricas apropiadamente.

Una serie de tiempo ***X,*** puede ser considerada como un punto en un espacio *n*-

dimensional. Como consecuencia inmediata se puede hacer uso de las técnicas de indexación y búsqueda, conocidos como *Métodos Espaciales de Acceso (MEA)* que hacen uso de estructuras de datos llamadas ***R-Tree*** **[6, 7].**

Sin embargo muchos de los métodos de acceso espacial se degradan rápidamente en dimensiones mayores de 8 a 12 y típicamente una consulta o query, contiene entre 20 y 1000 puntos de información, entonces para poder utilizar un método de acceso espacial es necesario realizar una **reducción de dimensión** de la serie de tiempo, estas *reducciones* de dimensión pueden ser cualquiera de las representaciones de las series de tiempo mencionadas en el capítulo 3**.**

Para la indexación es necesario lo siguiente [3]:

• Generar una técnica de reducción de dimensión de los datos de tamaño ***n*** a ***N***, donde ***N***, puede ser manejado *eficientemente* por algún MEA.

• Producir una métrica de distancia definida en la representación ***N-dimensional*** y respete que: ***DespacionIndexado(A, B) <= DespacioOriginal(A, B***) [3]

El algoritmo 4.1 muestra la indexación de alguna base de datos de serie de tiempo, utilizando técnica de reducción de dimensión para después agregarla a algún MEA, que maneje la serie de tiempo. La Fig. 16 muestra gráficamente el algoritmo de indexación.

```
  Algoritmo 4.1
  Entrada:
  DBST    //BD que contiene N Series de Tiempo

  Salida:
  Rtree   //Árbol que contiene la BD indexada

  Index:
  Rtree := crearMEA(); //Crear e iniciar Rtree
  for i := 1 to size(DBST) do{
    TSo := Tsi      //Serie de tiempo original
  //Aplicar la Transformación de Reducción
    Tsi:= transform(Tsi);
  //Insertar  Tsi y un apuntador a la Tso
    Rtree.insert(Tsi,Tso);
}
```

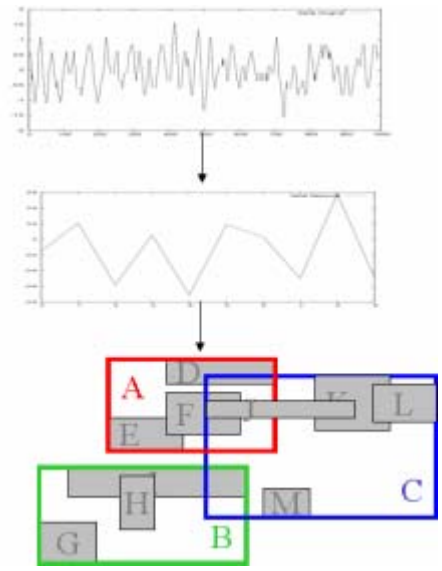


Fig. 16. Representación del algoritmo de Indexación de series de tiempo utilizando *MEA*.

El algoritmo muestra que para todas las series de tiempo en la base de datos, son reducidas y colocadas dentro de cualquier MEA. Se dice que una sola transformación de cualquier serie de tiempo es de orden ***O(n)***, entonces la indexación completa de ***K*** series de tiempo será en ***O (Kn)***.

El algoritmo 4.2 muestra la búsqueda de un Query ***Q,*** y alguna serie de tiempo Indexada ***Ti***, utilizando una métrica de similitud y un factor de similitud ***e***, tal que ***D (Q, Ti) <= e***.

```
Algoritmo 4.2
Entrada:
 Q            //Query a buscar
 e            //Factor de similitud
Salida:
 TSencontrada   //Serie de tiempo cercana
                //basado en factor de similitud
SearchQuery:
//Transformar Q al mismo espacio
Qi := transform(Q);
//Obtenemos los potenciales candidatos
candidate[] := Rtree.buscaCandidatos(Qi);

for i := 1 to candidate.length do{
  //por cada candidato
  ci := candidate[i];
  //Recuperamos la serie de tiempo original
  TSo := ci.getTsOriginal();
  //calculamos la distancia usando alguna métrica
  if D(TSo, Q) <= e then{
    TSencontrada = TSo;
    finalizar;
 }
}
```

El algoritmo 4.3 muestra la generación del K-vecino más cercano dado un Query.

```
Algoritmo 4.3

Entrada:
Rtree   //BD que contiene N Series de Tiempo
          //transformada
Q     //Query para generar su vecindario

Salida:
K-vecindario  //Subconjunto de la BDST

K-vecindario:
//Poner en el Query en el mismo espacio
Qi := transform(Q)
//Obtener los vecinos del árbol indexado
vecinos:= Rtree.buscaKvecinos(Qi);
//Calcular el valor máximo de distancia en vecinos
eMax = máximo(vecinos);

//Descartamos los falsos positivos
//Utilizando el Algoritmo 4.2
for i := 1 to vecinos.size() do{
  kVecinos.add(SearchQuery(vecinos[i], eMax));
}
```

La siguiente figura muestra la idea de búsqueda de un Query en la base de datos indexada.

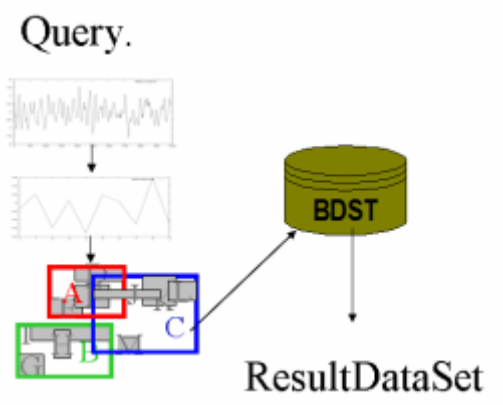


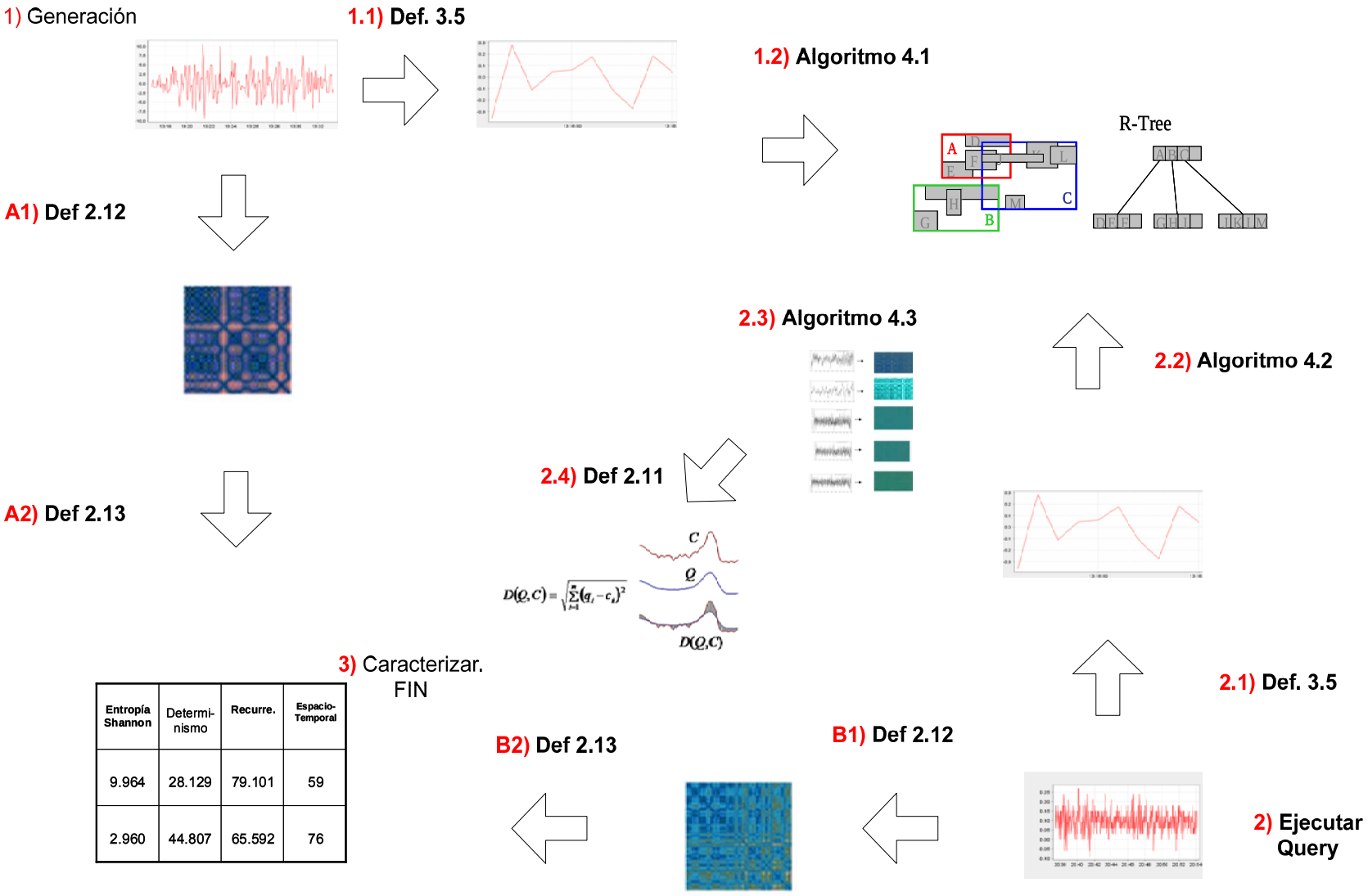
Fig. 17. Representación gráfica de la búsqueda de un query en la BDST.

Cabe mencionar que el ***manejo del tamaño de query,*** está basado en las series de tiempo indexadas, es decir, se toma como referencia la longitud de los números de elementos en la serie de tiempo de la base de datos y con las siguientes características:

1. Menores a las indexadas: En teoría se recomienda utilizar una métrica capaz de preservar las distancias entre el espacio de búsqueda, las indexadas y el espacio original de la serie de tiempo, es decir, definir una métrica utilizando algún tipo de norma, topológicamente hablando.

2. Mayores a las indexadas: Aquí simplemente se acota la serie de tiempo a buscar con las indexadas y se procede normalmente con el proceso de búsqueda.

Mapa de Indexación.



Entropía Shannon	Determi-nismo	Recurre.	Espacio-Temporal
9.964	28.129	79.101	59
2.960	44.807	65.592	76

Fig. 18. Gráfica del método de indexación.

4.2.2 Agrupamiento

Como mencionamos en el capítulo 2 el agrupamiento de series de tiempo, es una técnica parecida a la clasificación. Para la generación del método de agrupamiento de series de tiempo, haremos lo siguiente:

1. Utilizar la representación de las series de tiempo utilizando ondeletas, Cap3. sección 3.3.2, y utilizar sus coeficientes para realizar el clustering.

2. Extender el algoritmo K-means, y utilizar los componentes de la descomposición de ondeletas Haar, como medias de los cluster y sucesivamente utilizar los centroides de la iteración pasada.

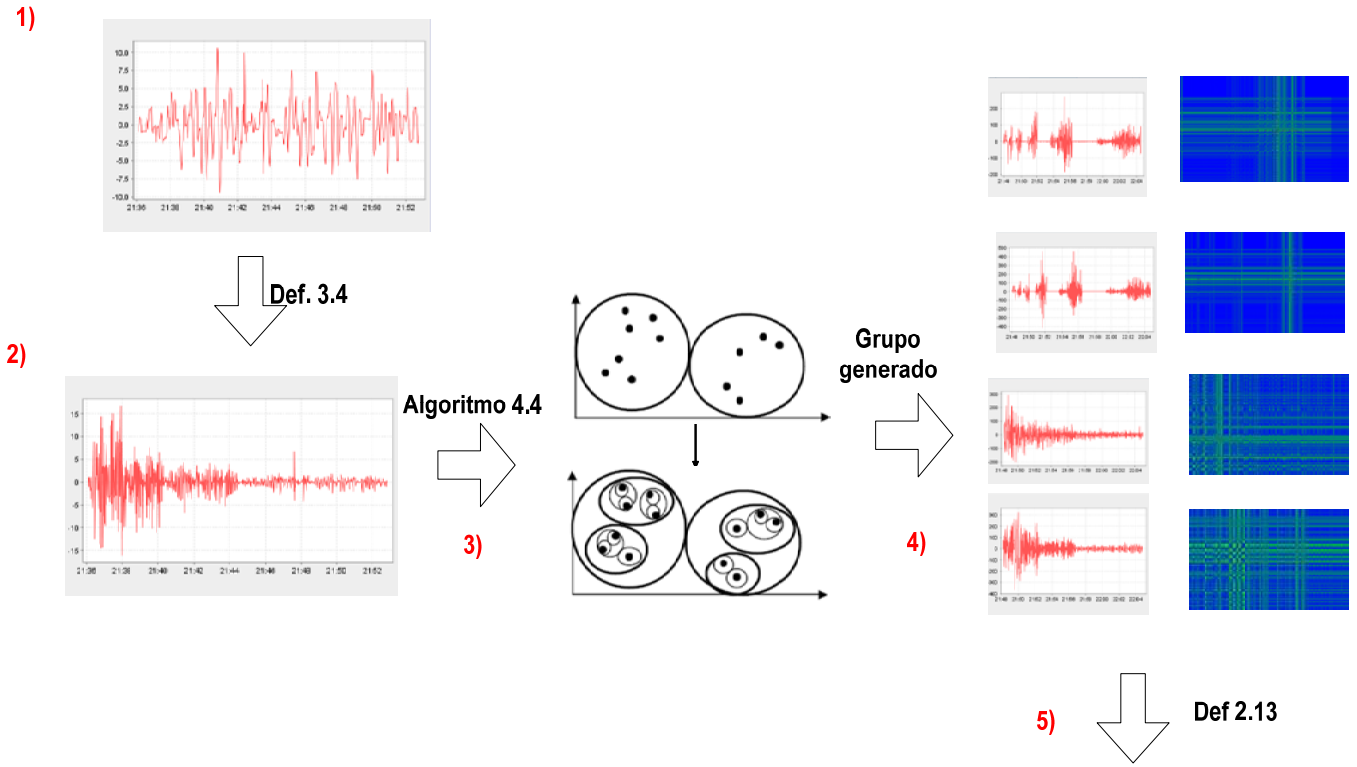
El algoritmo 4.4 muestra la extensión del K-means, utilizando las como medias los coeficientes de la transformada de ondeletas tipo HAAR.

Algoritmo 4.4
Entrada:
$D = \{t_1, t_2, \ldots, t_n\}$ //Conjunto Series de Tiempo
k //Número de clusters deseado
Salida:
K //Conjunto de clusters

Wavelets K-means:

```
//Series de tiempo usando ondeletas HAAR
 Dw:= transform (D);
 asignar medias iniciales m1,m2,...,mk;
 repetir
   asignar twi al cluster con media cercana;
   asignar nueva media con centroides de ti-1;
 hasta conocer el criterio de convergencia;
```

La complejidad del algoritmo K-mean, se dice que es de orden O(kNrD), donde D es la dimensión de la serie de tiempo. La complejidad de la transformación Haar es lineal para cada serie de tiempo, cabe mencionar que para bases de datos grandes con muchos puntos de información puede impactar el orden de ejecución por lo antes mencionado.



Entropía Shannon	Determinismo	Recurrencia	Entropía esp-tem
1.842	12.126	13.495	8.7
1.000	5.422	10.061	8.4
-1.00	0.050	0.949	8.0
-1.00	0.077	0.849	8.3

Fig. 19. Gráfica del método de agrupamientos.

4.2.3 Clasificación simbólica

Para la realización de la clasificación, utilizaremos la representación simbólica de las series de tiempo, mostrada en el capítulo tres. El algoritmo 4.5 muestra dicha clasificación de las series de tiempo.

Algoritmo 4.5
Entrada:
$D = \{t_1, t_2, \ldots, t_n\}$ //Conjunto Series de Tiempo
a //cardinalidad del alfabeto
w //longitud de la palabra
Salida:
C //Clasificación simbólica

STRSS:

```
alfabet[] := generaAlfabeto(a);
for k in D {
    TSk := transform(TSorik);
    WRD := generateWord(TSk,alfabet, w);
    C.addToClasification(WRD, TSk);
}
regresa C;
```

La siguiente figura muestra el proceso de la generación de una palabra, dada alguna serie de tiempo a clasificar.

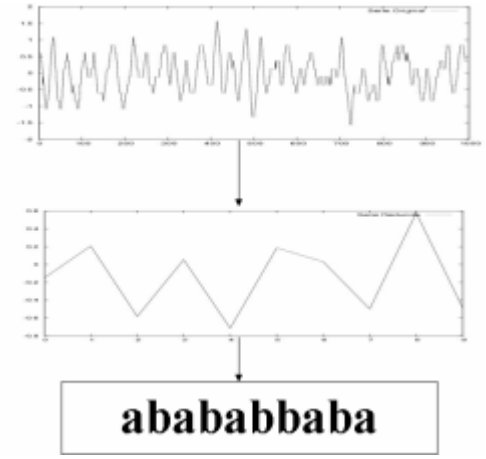


Fig. 20. Representación de la generación simbólica de alguna serie de tiempo.

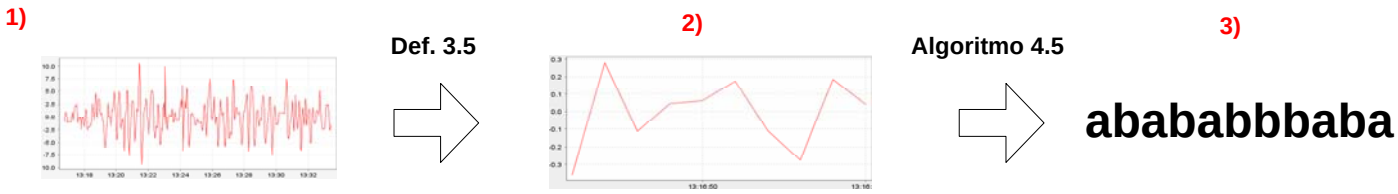


Fig. 21. Gráfica del método de clasificación simbólica.

4.3 Desarrollo del método

El desarrollo del método se muestra en la siguiente figura y se presenta en forma general.

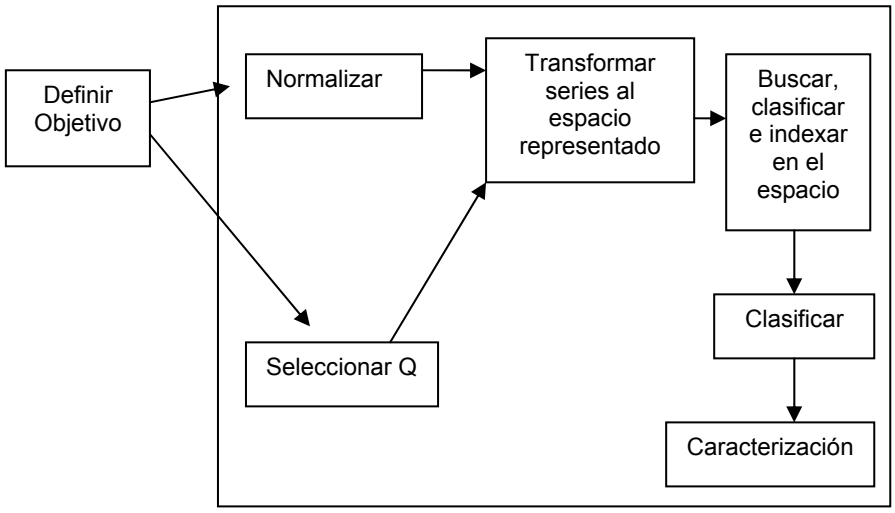


Fig. 22. Método experimental.

A continuación se enlista los pasos del método:

a) Definir el objetivo de búsqueda, esto significa definir lo que se va a clasificar y en base a que se puede clasificar.

b) Generar una representación "normalizada" de las series de tiempo para su análisis, es decir, si se piensan clasificar personas, trate de no mezclar algún tipo de animal o fruta en dicha clasificación, ya que pueden generar datos sin algún sentido científico o experimental, aunque si es el caso, sea consistente en las series de tiempo.

c) Genere el criterio de búsqueda o el criterio de clasificación, esto es, dada la base de datos de series de tiempo proporcione algún dato para su clasificación. Dicho criterio o query, puede estar o no en la misma

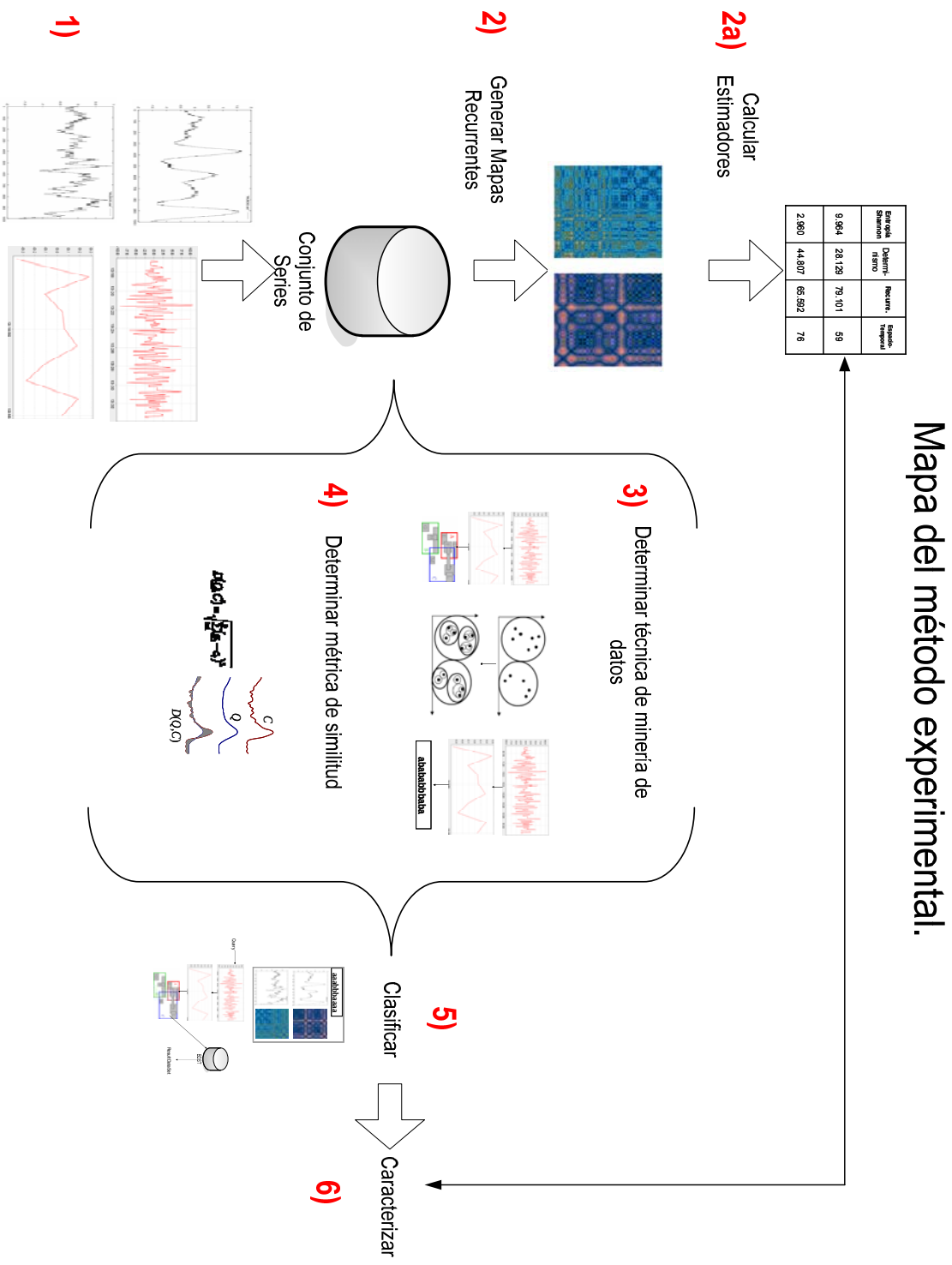
base de datos.

d) Transforme todas las series de búsqueda a un espacio de trabajo más manejable computacionalmente, para poder generar un mejor análisis de las series de tiempo.

e) Dentro del espacio, dado que es una representación conveniente, realice las operaciones necesarias previas a la clasificación de las series.

f) Después utilizando el espacio y el criterio de búsqueda, simplemente proceda a clasificar a las series de tiempo, de alguna forma, si se puede.

4.3.1 Mapa del método experimental



:

Fig. 23. Mapa del método experimental.

Capítulo 5
Resultados

El presente capítulo muestra algunos de los resultados generados por el método experimental presentado en el capítulo anterior. Así como los trabajos generados de esta investigación, las conclusiones de la misma y así como los trabajos futuros que se pueden generar.

5.1 Discusión de resultados obtenidos

Los siguientes resultados fueron obtenidos usando el método propuesto y haciendo una concreta instancia del método, es decir, se utilizo el siguiente experimento de indexación, fundamentado en el método:

1. Las series de tiempo para probar el método son tomadas de la base de datos de sismos fuertes [15], la cual contiene cuatro bancos de datos, de los cuales, el banco principal consta de 13,545 sismos.
2. Selección de 3 estaciones, de diferentes tipos de suelos, así como los sismos correspondientes al día 19 de septiembre de 1985 de diferentes estaciones, esto con el fin de tener un mayor criterio de búsquedas y diferentes fenómenos con diferentes tipos de suelos que implícitamente tienen diferentes características físicas de los sismos registrados.
3. Cada uno de los sismos fue truncado a mil puntos de información.
4. Generación de los mapas recurrentes de cada serie de tiempo obteniendo los siguientes parámetros: Entropía de Shannon, Determinismo, Recurrencia y Entropía Espacio-Temporal [14].
5. Se utilizaron cuatro tamaños de dimensiones: N = 5, 10, 20, 50.

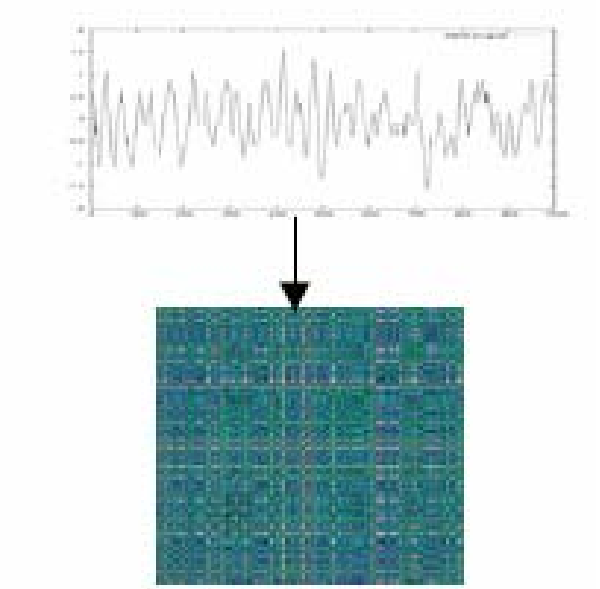


Fig. 24. Serie de tiempo original indexada con su mapa recurrente.

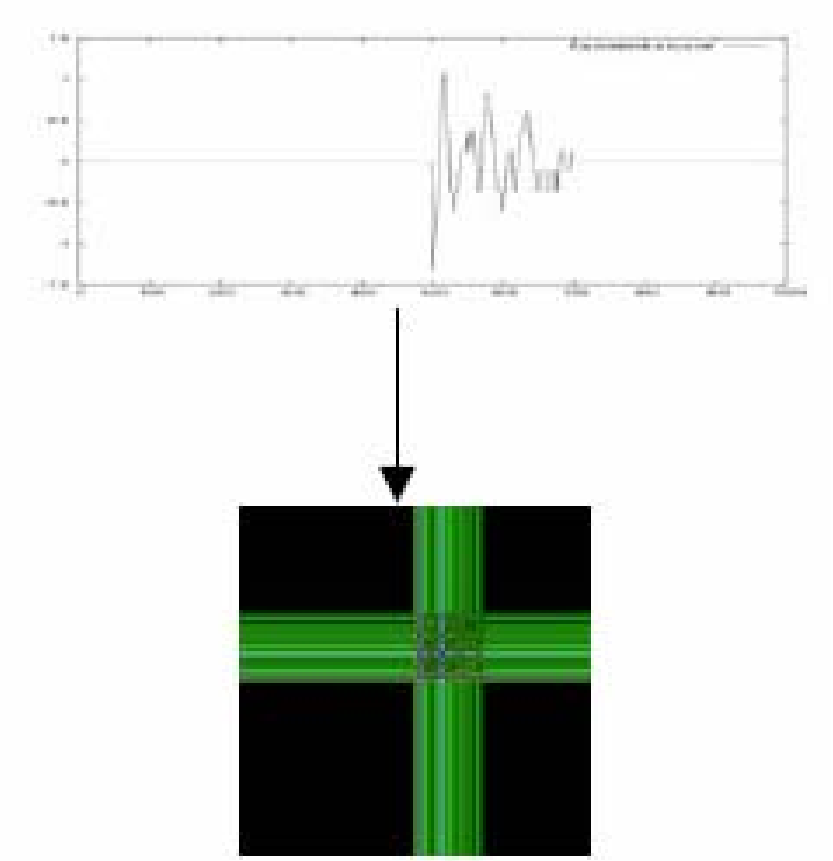


Fig. 25. Serie de tiempo a buscar y su mapa recurrente.

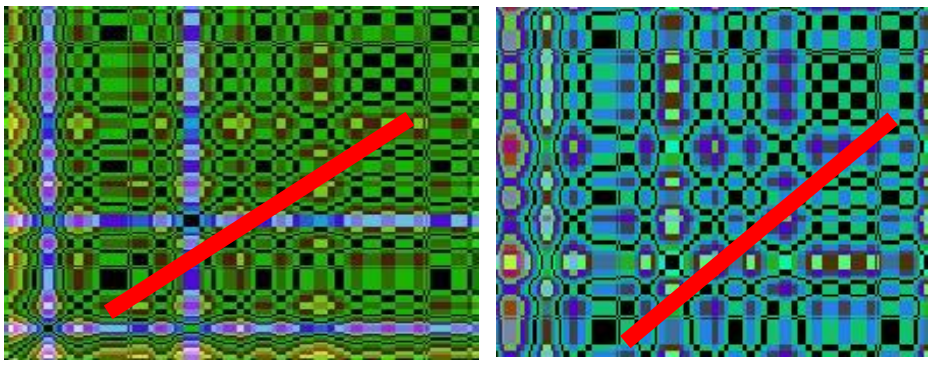


Fig. 26. Similitud en los mapas recurrentes.

Shannon	Determinismo	Recurrencia	Espacio-Temporal
9.849	39.396	34.082	65%
9.814	89.508	80.777	85%

Tabla 1. Parámetros de los mapas recurrentes del query y la encontrada

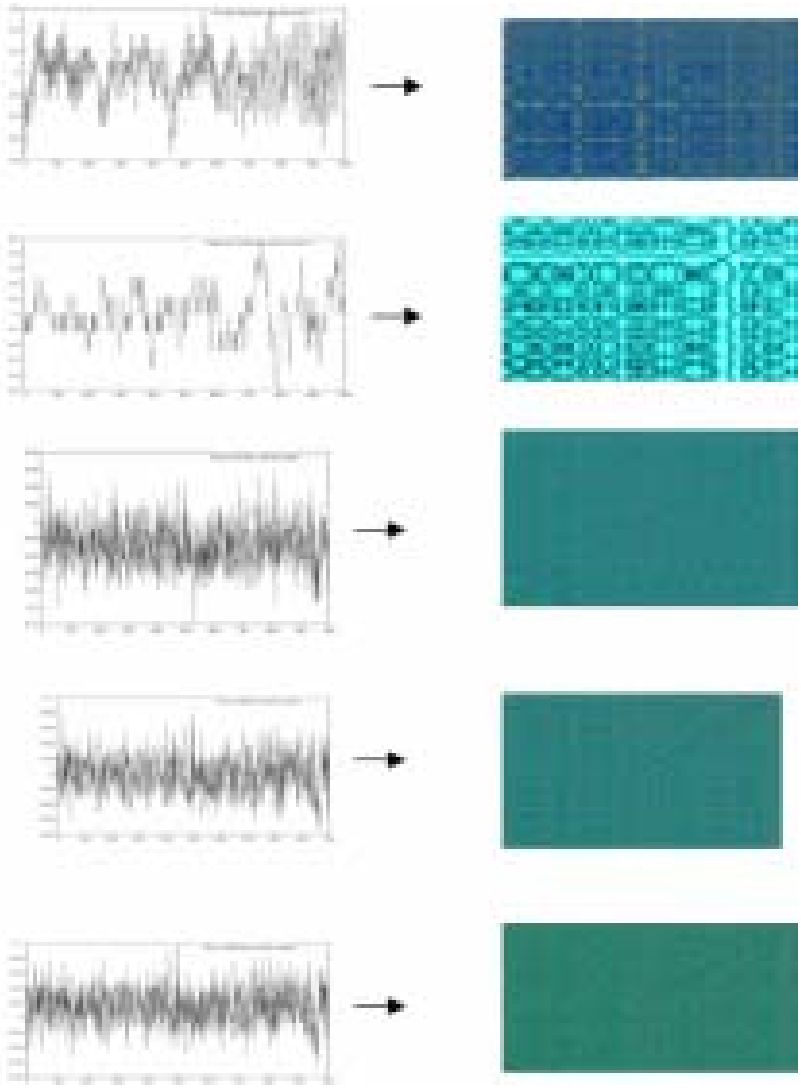


Fig. 27. Representación de algún K-vecindario obtenido.

SHN	DET	REC	E-T
9.964	3.812	42.945	91%
9.814	1.652	22.890	88%
7.311	0.235	36.654	84%
8.413	0.163	33.378	84%
9.715	0.070	33.835	84%

Tabla 2. Parámetros de los mapas recurrentes del vecindario

Los siguientes resultados fueron obtenidos usando el método propuesto y haciendo una concreta instancia del método, es decir, se utilizo el siguiente experimento de agrupación, fundamentado en el método:

1. Las series de tiempo para probar el método son tomadas de la base de datos de sismos fuertes [15], la cual contiene cuatro bancos de datos, de los cuales, el banco principal consta de 13,545 sismos.
2. Selección de 3 estaciones, de diferentes tipos de suelos, así como los sismos correspondientes del día 19 de septiembre de 1985 de diferentes estaciones, esto con el fin de tener un mayor criterio de búsquedas y diferentes fenómenos con diferentes tipos de suelos que implícitamente tienen diferentes características físicas de los sismos registrados.
3. Cada uno de los sismos fue truncado a mil puntos de información.
4. Generación de los mapas recurrentes de cada serie de tiempo obteniendo los siguientes parámetros: Entropía de Shannon, Determinismo, Recurrencia y Entropía Espacio-Temporal [14]
5. Se utilizaron representación de las series de tiempo por ondeletas, tipo fast haar.

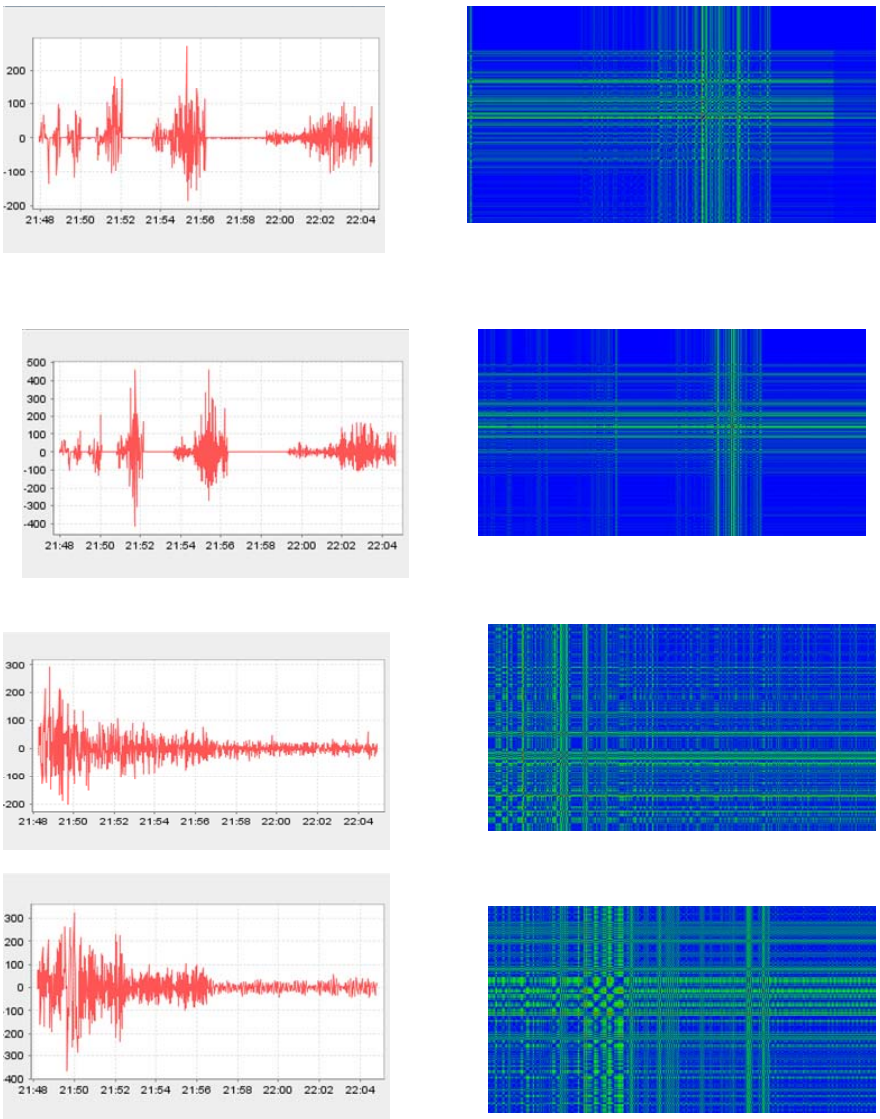


Fig. 28. Cluster generado con sus elementos.

Entropía Shannon	Determinismo	Recurrencia	Entropía esp-tem
1.842	12.126	13.495	8.7
1.000	5.422	10.061	8.4
-1.00	0.050	0.949	8.0
-1.00	0.077	0.849	8.3

Tabla 3. Parámetros de los mapas recurrentes de la Fig. 28

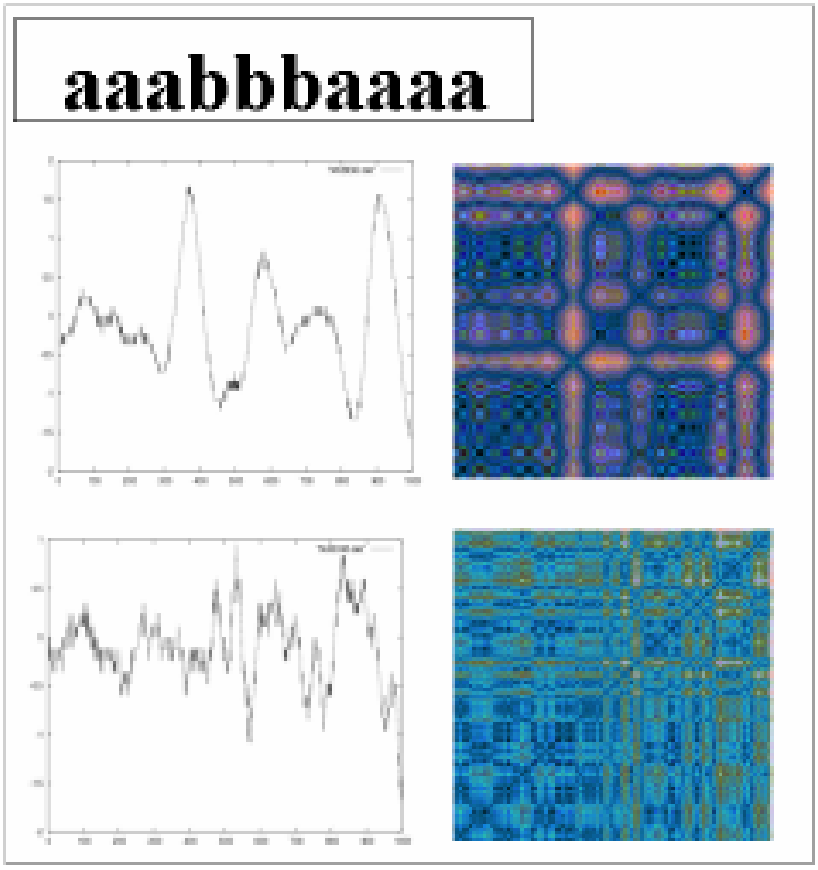


Fig. 29. Clasificación generada con sus elementos aaabbbaaaa.

Entropía Shannon	Determi-nismo	Recurre.	Espacio-Temporal
9.964	28.129	79.101	59
2.960	44.807	65.592	76

Tabla 4. Parámetros de los mapas recurrentes de la Fig. 29

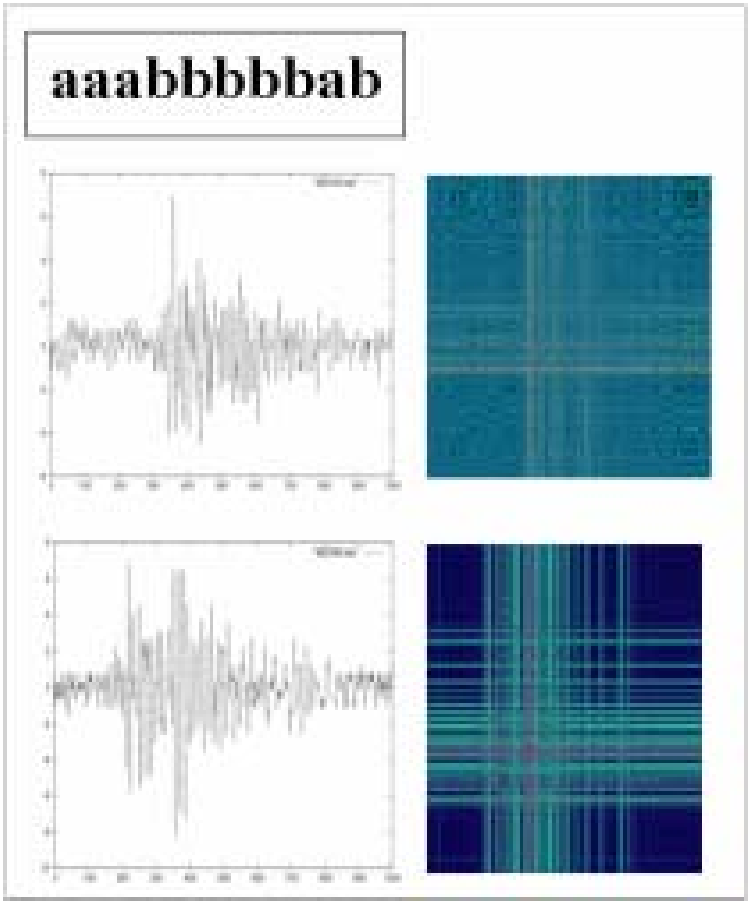


Fig. 30. Clasificación generada con sus elementos aabbbbbab.

Entropía Shannon	Determi-nismo	Recurre.	Espacio-Temporal
9.627	63.467	54.831	80
8.434	62.434	56.407	82

Tabla 5. Parámetros de los mapas recurrentes de la Fig. 30

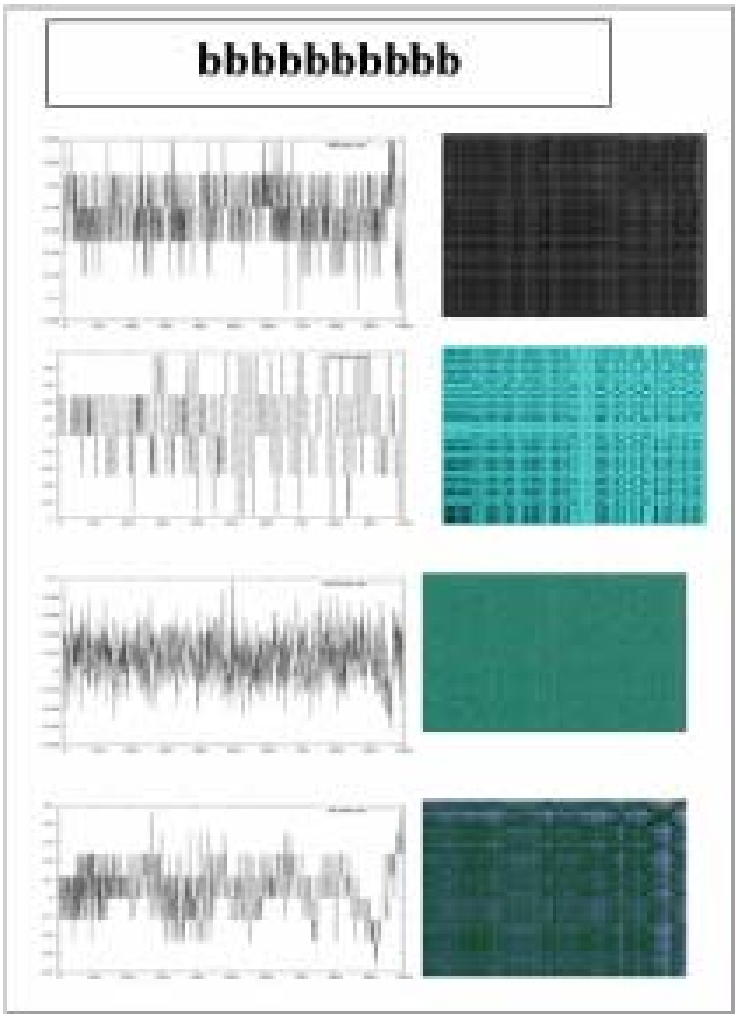


Fig. 31. Elementos de la clasificación bbbbbbbbbb.

Entropía Shannon	Determi-nismo	Recurre.	Espacio-Temporal
9.964	32.633	0.059	93
4.513	32.617	1.734	92
2.960	25.639	0.055	92
4.434	33.835	0.002	84

Tabla 6. Parámetros de los mapas recurrentes de la Fig. 31

Capítulo 6
Conclusiones

A continuación se presentan las conclusiones de este trabajo de tesis.

6.1 Contribuciones y conclusiones, puntuales, obtenidas.

1. Se generó una clasificación de series de tiempo, utilizando los estimadores de los mapas recurrentes. Además de que con ellos se proporcionó su caracterización y que puede ser utilizado como factor de similitud entre las series de tiempo.

2. Se proporcionó una herramienta computacional que utiliza técnicas de minería de datos para la generación de clasificación y caracterización de las series de tiempo. Además de que dicha herramienta proporciona un método para el almacenamiento y recuperación de series de tiempo.

3. Se mostró la utilización de la búsqueda de similitud de series de tiempo por indexación utilizando métodos espaciales de acceso. Así como la agrupación de series de tiempo, utilizando los componentes obtenidos por la descomposición de ondeletas. También se proporcionó una técnica de clasificación de series de tiempo con representación simbólica.

4. Además se mostró, que con fundamento en la similitud se pueden generar clasificaciones de las series de tiempo que representan diferentes fenómenos, con características semejantes.

6.2 Trabajos publicados.

1. ***Earthquakes Classifications using Data Mining Techniques*** [28].

2. ***Clustering Time Series with a Symbolic Representation*** [29].

3. ***Quiron: Similarity Search on Time Series Database*** [30].

6.3 Aplicación y extensión generadas del trabajo

El trabajo propuesto ha tenido aplicaciones directas para la ingeniería sísmica, en particular para el análisis de series de tiempo que representan sismos, de la tesis se han derivado de los siguientes trabajos publicados y que a continuación se enlistan:

Nacional de física 2004:

- *Similitud De Señales De Sísmicas Por La Técnica De Mapa Recurrente*. [33]

- *Clasificación Y Reconocimiento de Señales Sísmicas Empleando la Técnica de Mapa Recurrente Y Eigenfaces*. [34]

Congreso Nacional de Ingeniería Sísmica, 2005:

- *Clasificación y Semejanza de Espectros de Respuesta de Aceleración.* [35]

6.4 Trabajos futuros

De los trabajos futuros que se derivan directamente podemos mencionar los siguientes:

1. Generación de una caracterización de series de tiempo utilizando los parámetros obtenidos de los mapas recurrentes.

2. Aplicación de algoritmos de ***bioinformática*** para el descubrimiento de *biosecuencias* para el análisis y búsqueda de patrones de series de tiempo.

3. Generación de nuevas métricas de similitud utilizando la dinámica del fenómeno observado.

4. La extensión y generación de nuevos algoritmos de bases de datos para aplicaciones particulares, como son, la bolsa de valores.

Anexo A (Diseño de Artemiza)

En este anexo se proporciona el diseño de una herramienta prototipo llamada "***Artemiza**: Herramienta de minería de datos para series de tiempo*". La organización del capítulo muestra el diagrama principal de la aplicación, y después muestra los subcomponentes principales de cada componente.

6.1 Componentes principales

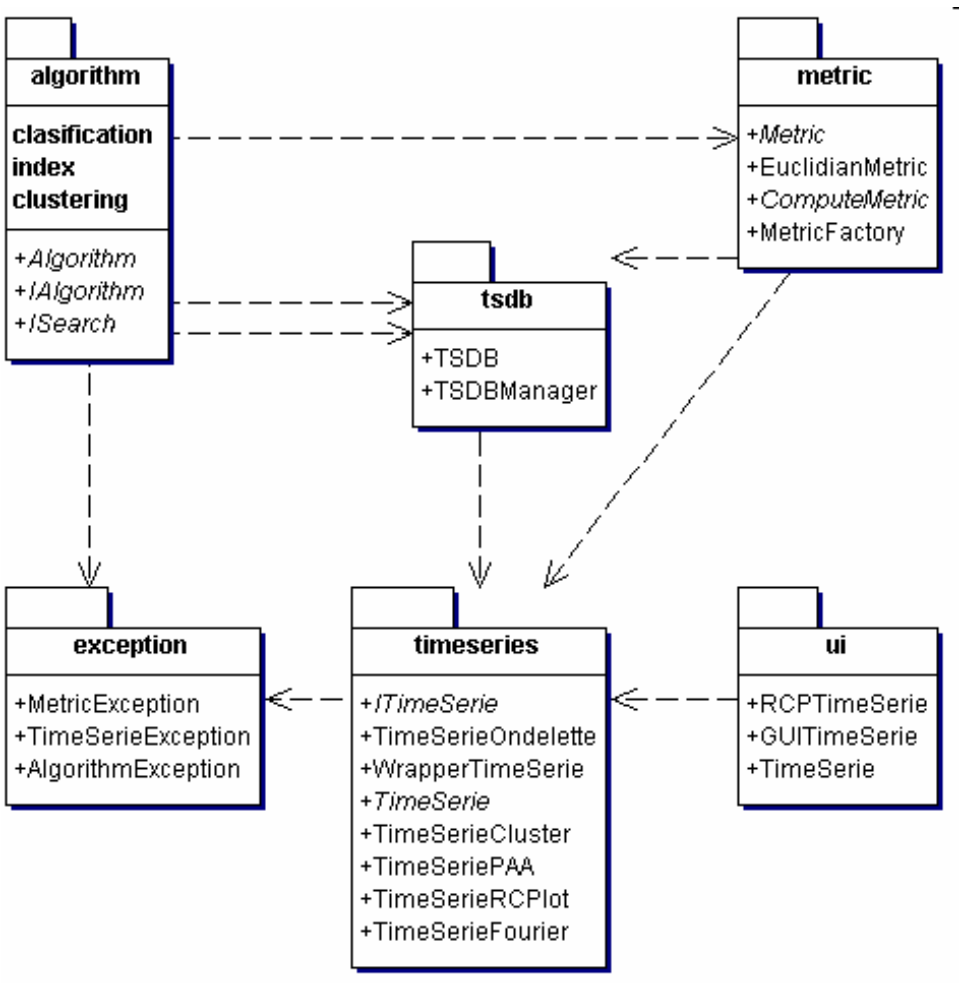


Fig. 32. Diagrama de componentes Artemiza.

6.2 Componentes para los algoritmos.

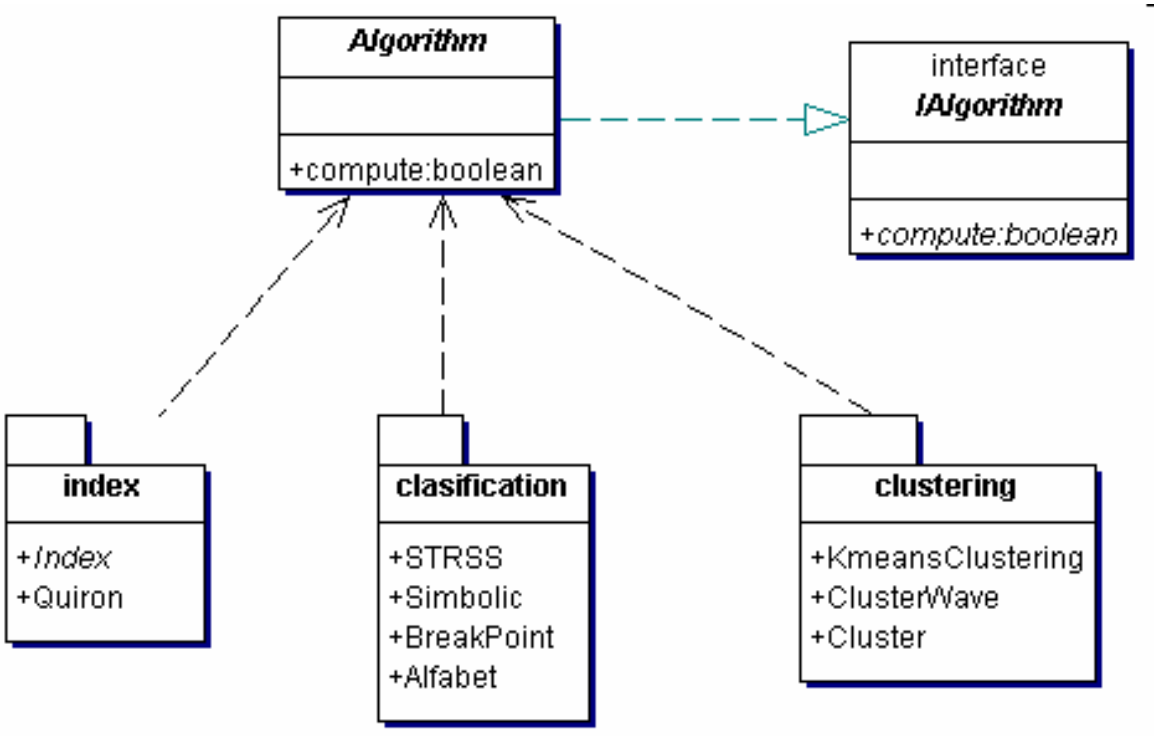


Fig. 33. Componentes para los algoritmos.

6.3 Componentes del manejador y la base de datos de series de tiempo.

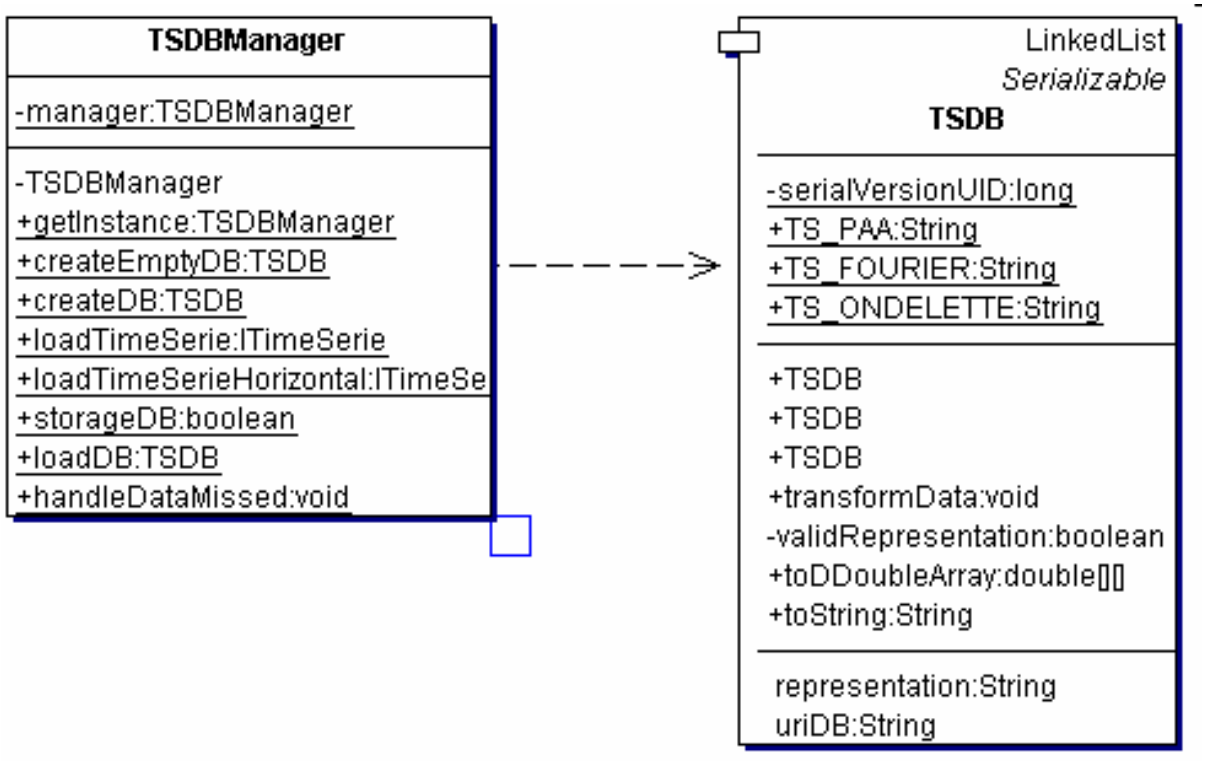


Fig. 34.Componentes del manejador y la base de datos de series de tiempo.

6.4 Diagrama del componente generador de métricas.

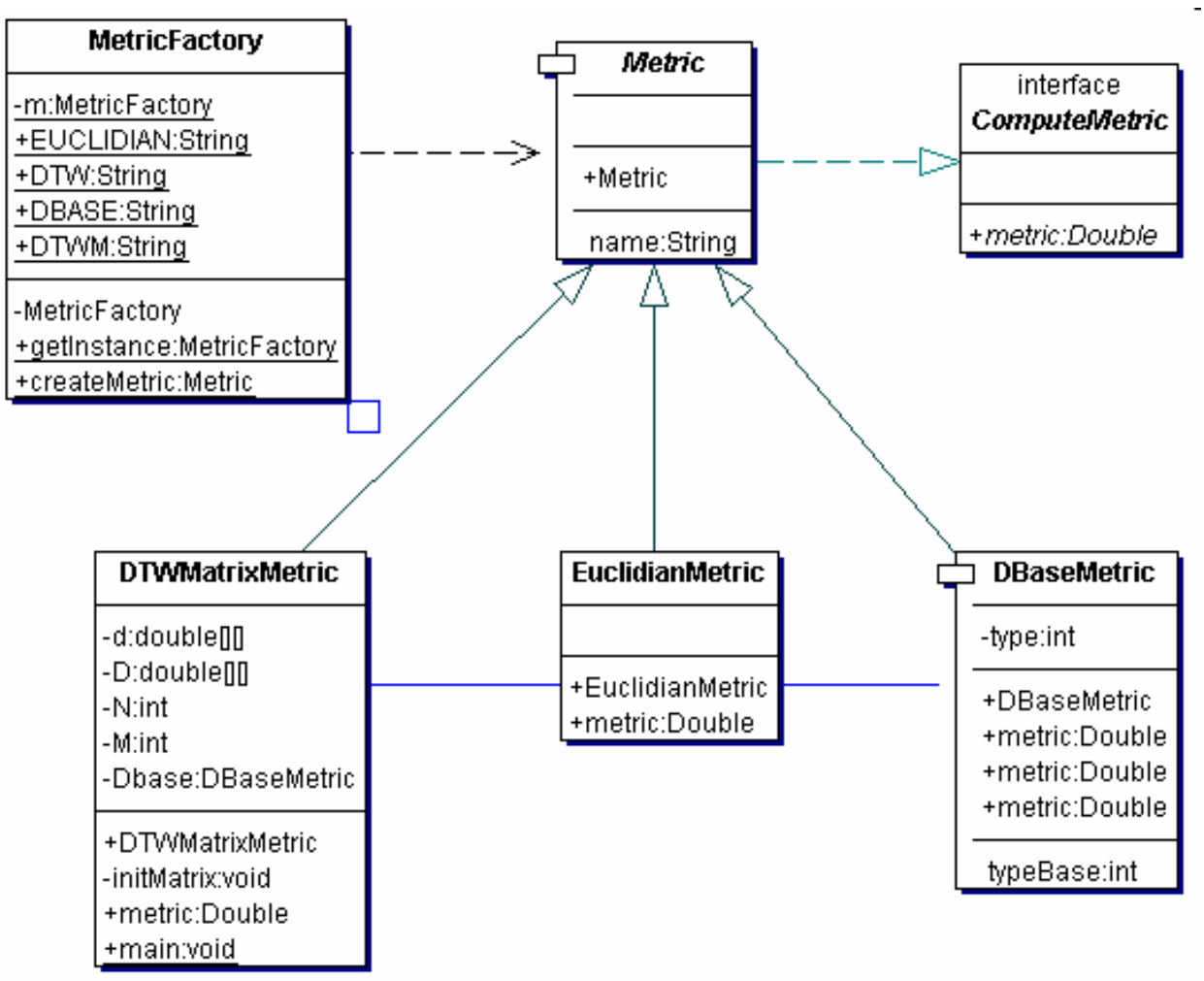


Fig. 35. Componente generador de métricas.

6.5 Diagrama de componentes para la representación de series de tiempo.

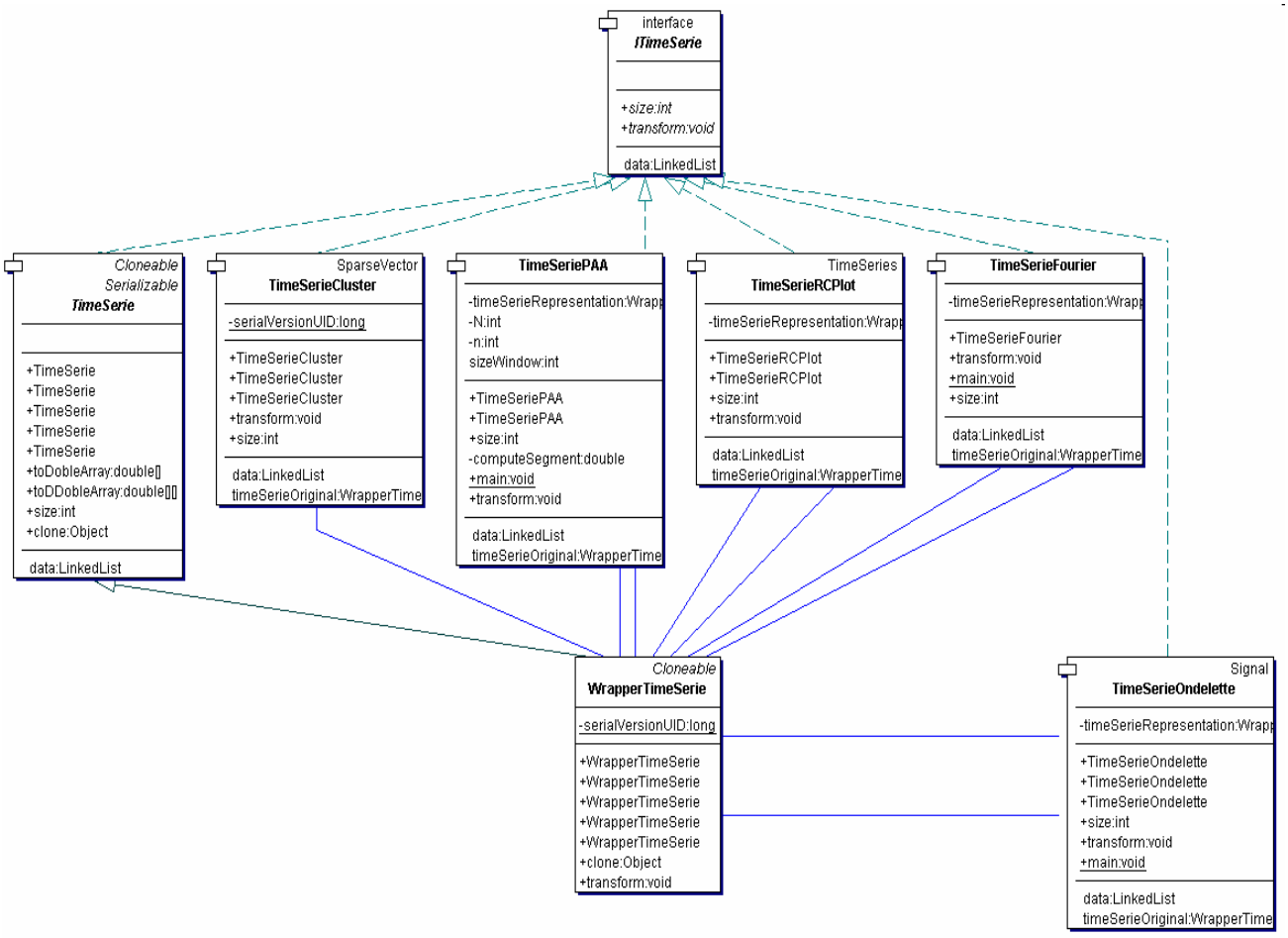


Fig. 36. Componentes para representación de series de tiempo.

Anexo B (Instalación, descripción y uso de Artemiza)

A continuación se describe la instalación, descripción; así como un ejemplo de ejecución de la aplicación.

7.1 Requerimientos

Los requerimientos necesarios para que funcione Artemiza son los siguientes:

- Espacio en disco de 15 MB.
- JDK 1.4.0 o superior, ya que la herramienta ha sido desarrollada con él.

7.2 Instalación.

Para instalar el software se necesita de los pasos siguientes:

1. Archivo llamado “artemiza.zip”.

Fig. 37. Archivo de distribución Artemiza.

2. Descomprima el archivo en donde se instalará Artemiza, en nuestro ejemplo lo haremos en el directorio llamado: “My Documentos\artemiza”.

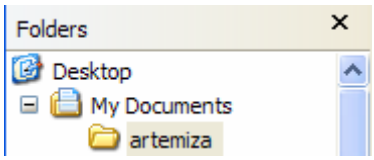


Fig. 38. Directorio de instalación.

3. Extraído el contenido del archivo se habrá creado una carpeta llamada “eclipse”, la cual contiene todo lo necesario y suficiente para la ejecución. Incluyendo dependencias de la aplicación, librerías y archivos de configuración. Ya todo listo para usar.

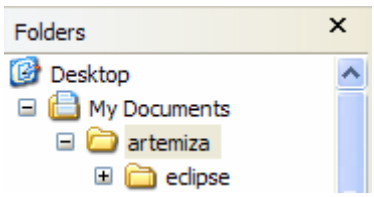


Fig. 39. Contenido del archivo descomprimido.

4. Ejecutar el archivo llamado “artemiza.exe” para su uso, que esta dentro del directorio llamado eclipse. Hasta este punto se ha realizado el proceso de instalación de la herramienta llamada Artemiza.

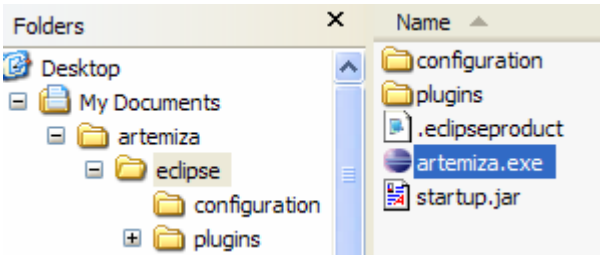


Fig. 40. Ejecutar archivo llamado “artemiza.exe”.

5. Adicionalmente se necesita la creación de un directorio de trabajo que se necesitará, por conveniencia, dicho directorio será llamando “artemizawork” y deberá de estar en la unidad de disco “c:\”.

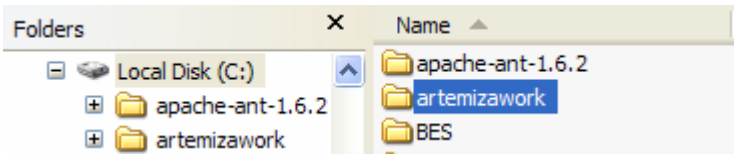


Fig. 41. Directorio de trabajo “artemizawork”.

7.3 Descripción.

Al momento de ejecutar el archivo “artemiza.exe”, se desplegará la pantalla de inicio de la aplicación.

Fig. 42. Pantalla de inicio de Artemiza.

Una vez iniciada la aplicación se muestra la aplicación con la pantalla de bienvenida así como mostrar la ayuda inicial de la aplicación.

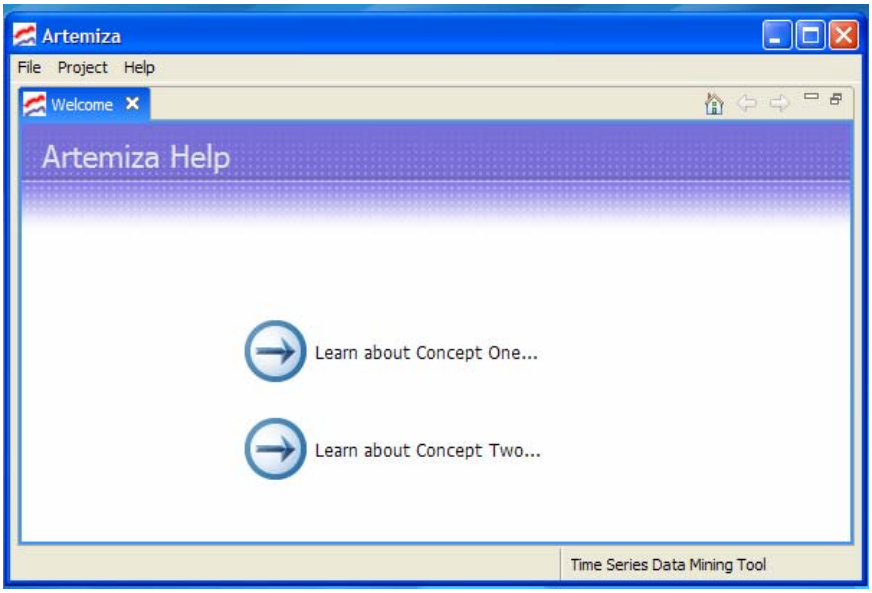


Fig. 43. Vista de Ayuda de Artemiza.

La aplicación consta de una barra de menú con tres menús principales llamados “File”, “Project” y “Help” respectivamente y que a continuación se describen.

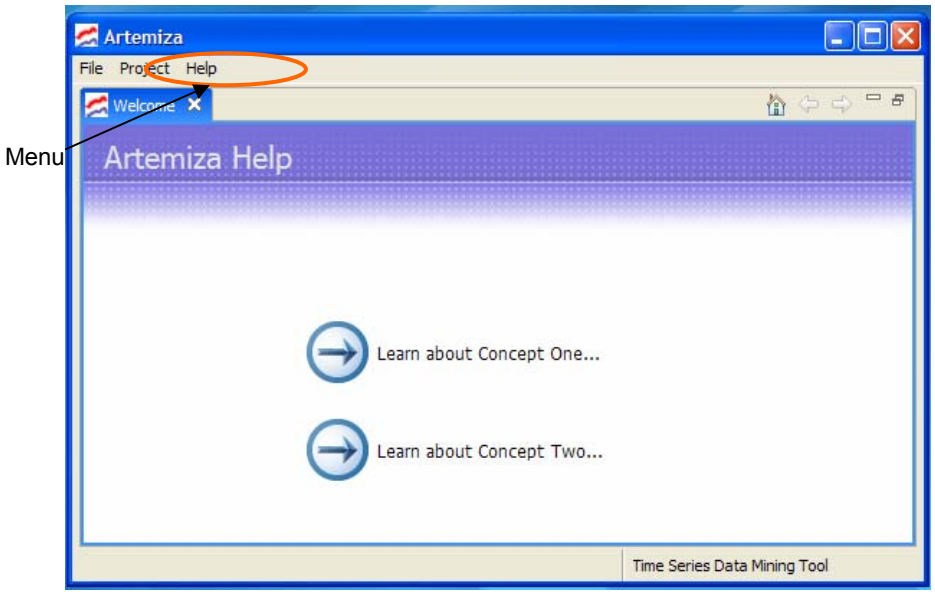


Fig. 44. Menú Principal de Artemiza.

El Menú llamado “File” contiene los siguientes elementos:

- “New Artemiza Window”: Este elemento del menú proporciona una nueva instancia de la aplicación.

- “Exit”: Este elemento, simplemente termina la ejecución de la aplicación.

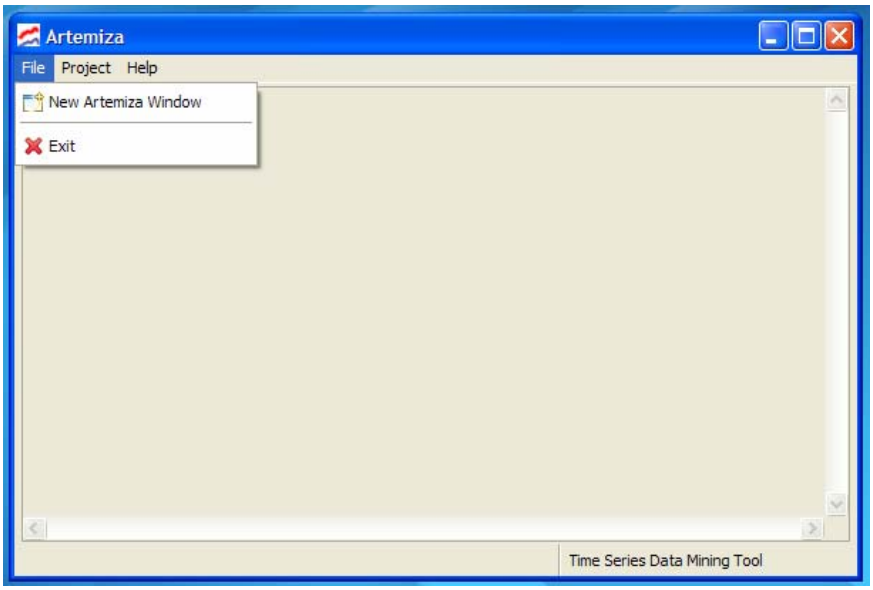


Fig. 45. Elementos del menú “File”.

El contenido del menú llamado “Project” contiene los siguientes elementos:

- “Configure”: El cual proporciona una funcionalidad para configurar la aplicación, es decir, aquí se especifican datos como: el nombre del proyecto,

la ruta donde se encuentran las series de tiempo a analizar, así como especificar también el tipo de algoritmo y métrica a utilizar para la ejecución.

- "Execute": Esta opción simplemente ejecuta la configuración realizada previamente.

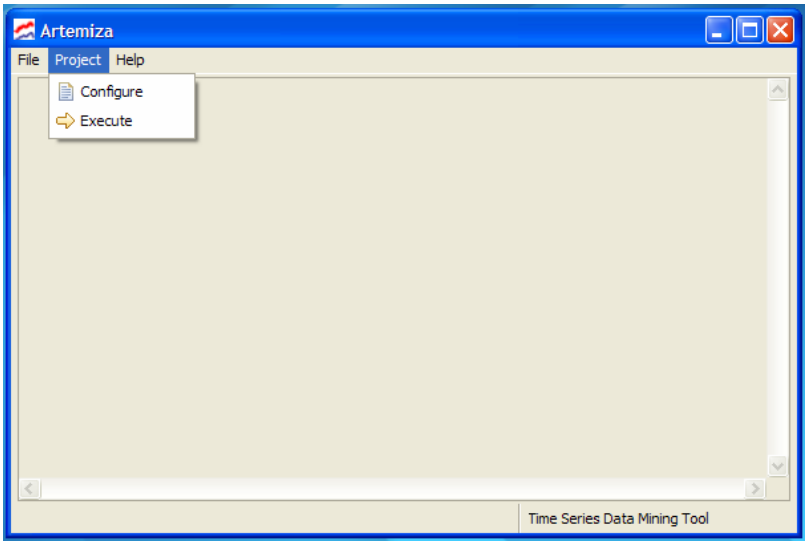


Fig. 46. Elementos del menú "Project".

El contenido del menú llamado "Help" contiene los siguientes elementos:

- "Welcome": Proporciona la ayuda de la herramienta.
- "About": Proporciona la información "*acerca de Artemiza".*

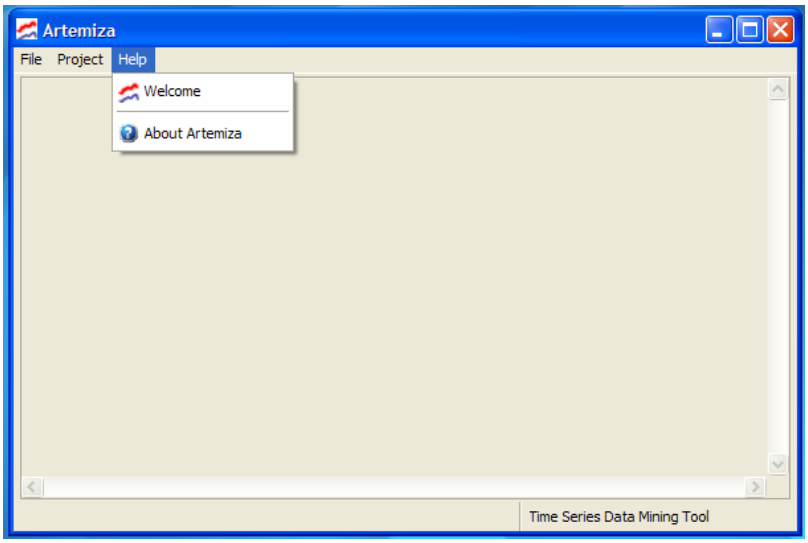


Fig. 47. Elementos del menú "Help".

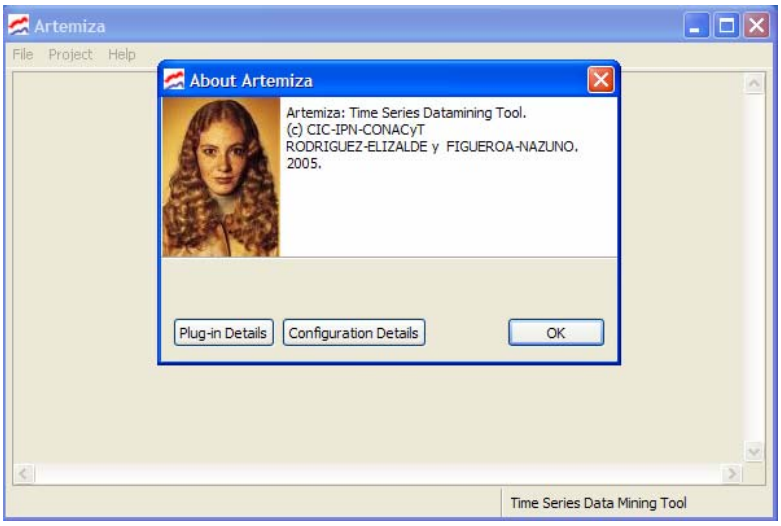


Fig. 48. Diálogo "Acerca de Artemiza".

Hasta el momento se ha descrito en forma general a la aplicación y en la siguiente sección se proporcionan ejemplos de ejecuciones utilizando la herramienta.

7.4 Ejemplo de indexación.

Para realizar la indexación simplemente configure la herramienta como se describe a continuación:

1. Utilice la opción "Configure" del menú llamado "Project" y aparecerá un dialogo en el cual proporcionará los datos de la configuración para la indexación.

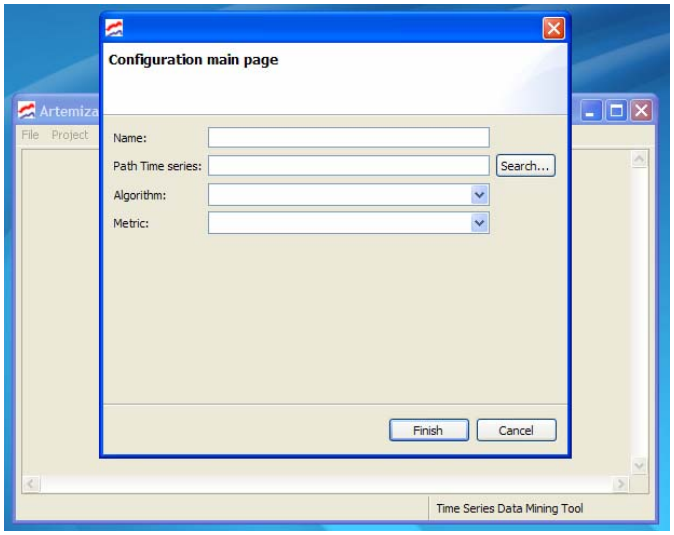


Fig. 49. Diálogo de configuración.

2. Proporcione todos los datos correspondientes y en la opción llamada “Algorithm”, seleccione “Index”.

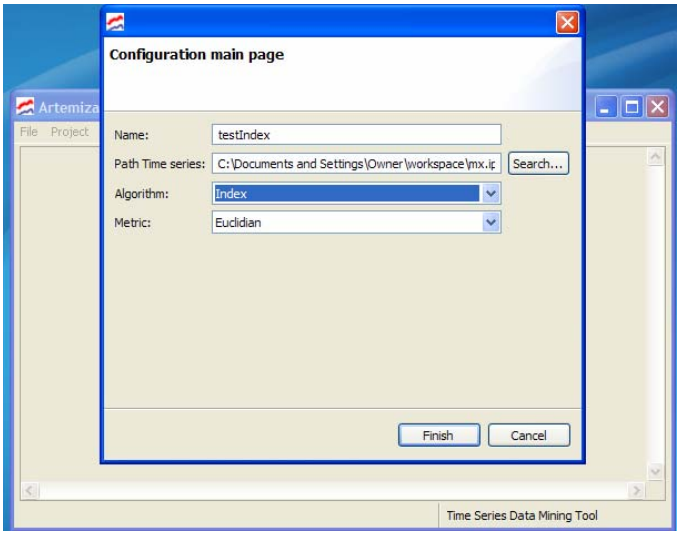


Fig. 50. Diálogo que especifica el algoritmo de indexación.

- Una vez configurada la aplicación mostrará los datos a analizar.

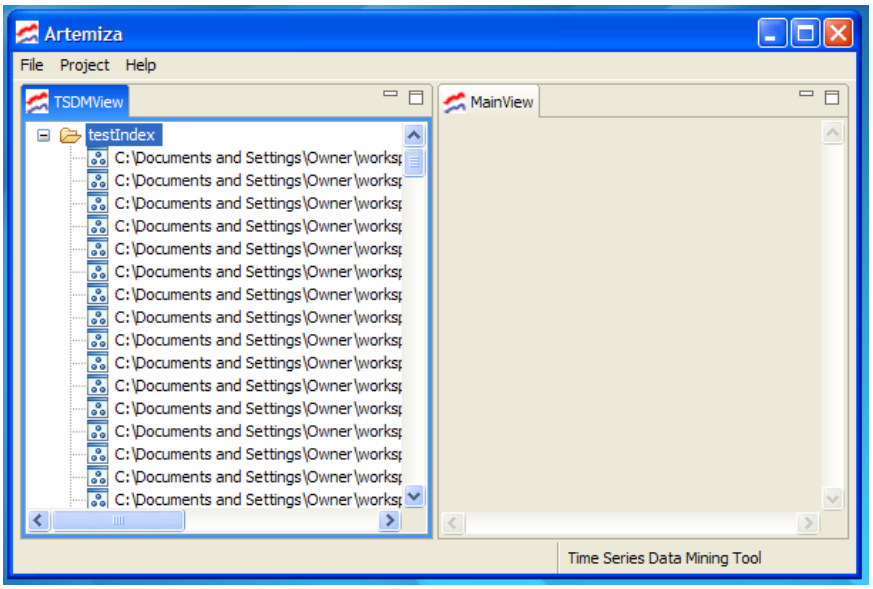


Fig. 51. Vista de datos Artemiza.

- Si selección alguna de las series de tiempo, que han sido mostradas, se proporcionará su mapa recurrente, así como la gráfica de la serie de tiempo.

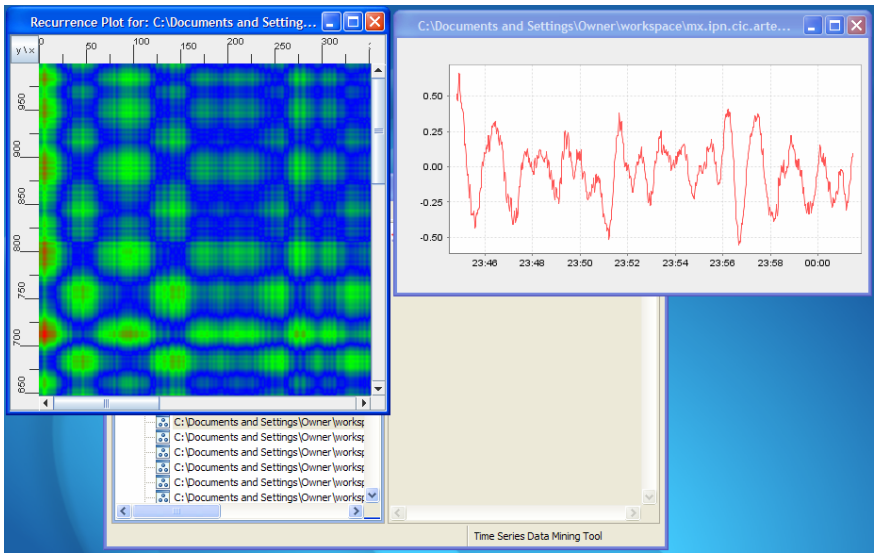


Fig. 52. Gráfica y mapa recurrente de la serie de tiempo.

3. Utilice la opción “Execute” del menú “Project” para ejecutar la configuración dada. Después de ejecutar se pedirá un dato adicional para la indexación, el *query* o consulta a buscar, proporcionando dicha consulta se mostrarán los resultados obtenidos de dicha búsqueda y proporcionando el vecindario más cercano a la consulta utilizando la métrica seleccionada.

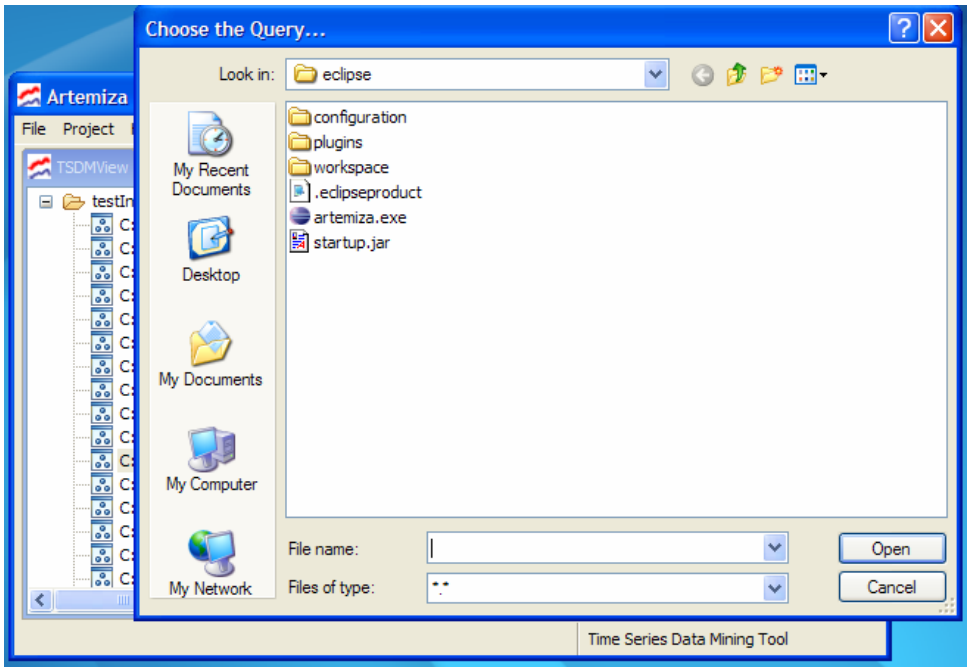


Fig. 53. Diálogo para seleccionar la consulta o query.

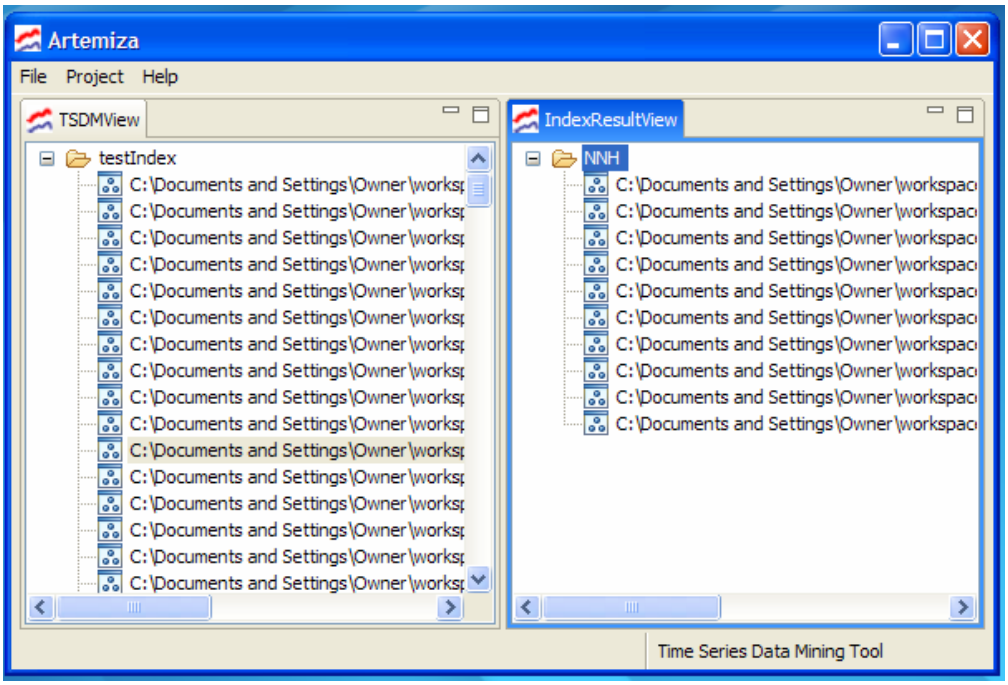


Fig. 54. Vista de vecindario más cercano de la consulta.

- Una vez más si selecciona algún elemento de los resultados se proporcionará la gráfica y mapa recurrente de la serie de tiempo.

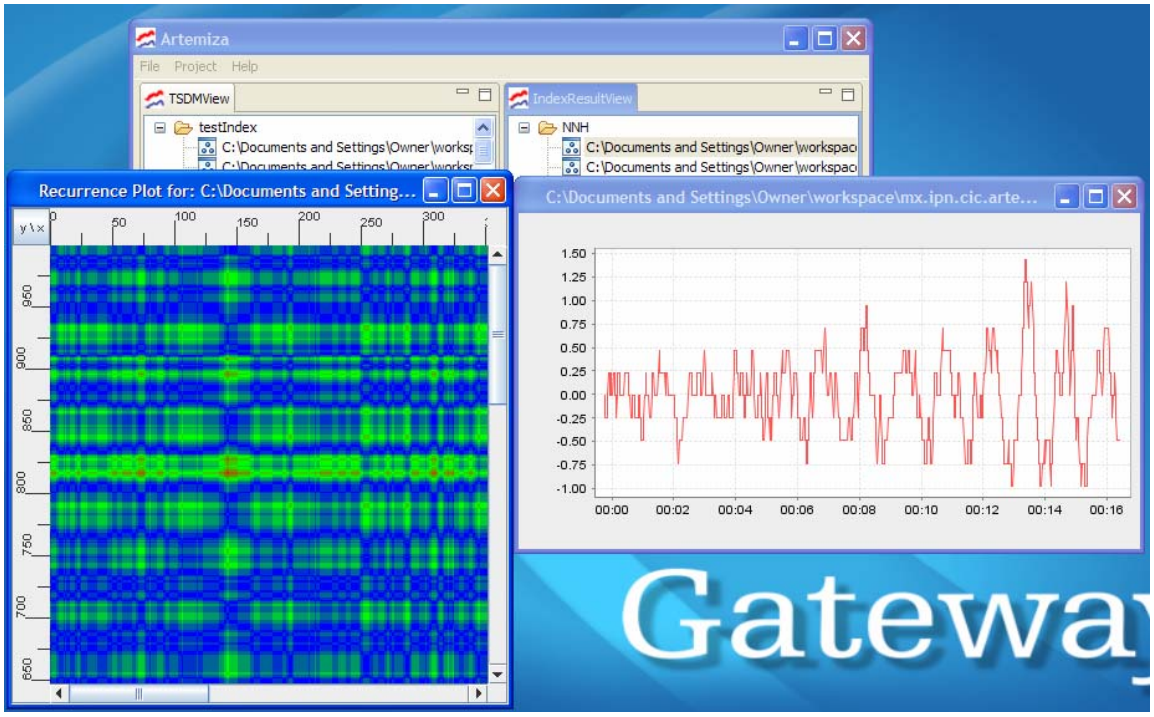


Fig. 55. Gráfica y mapa recurrente de la vista de resultados.

Análogamente se pueden configurar la aplicación para generar los otros diferentes tipos de análisis de series de tiempo utilizando las técnicas de minería de datos.

Referencias

[1] Keogh, E. & Pazzani, M. (1998). ***An enhanced representation of time series which allows fast and accurate classification, clustering and relevance feedback.*** In proceedings of the 4th Int'l Conference on Knowledge Discovery and Data -Mining. New York, NY, Aug 27-31. pp 239-241.

[2] Keogh, E. & Smyth, P. (1997). ***A probabilistic approach to fast pattern matching in time series databases.*** In proceedings of the 3rd Int'l Conference on Knowledge Discovery and Data Mining. Newport Beach, CA, Aug 14-17. pp 24-20.

[3] Keogh, E., Chakrabarti, K., Pazzani, M. & Mehrotra, S. (2001). ***Locally adaptive dimensionality reduction for indexing large time series databases.*** In proceedings of ACM SIGMOD Conference on Management of Data. Santa Barbara, CA, May 21-24. pp 151-162.

[4] Agrawal, R., Faloutsos, C. & Swami, A. (1993). ***Efficiente similarity search in sequence databases.*** In proceedings of the 4th Int'l Conference on Foundations of Data Oraganization and Algorithms. Chicago, Il, Oct 13-15. pp 69-84.

[5] Agrawal, R., Lin, K. I., Sawhney, H. S. & Shim, K. (1995). ***Fast Similarity search in the presence of noise, scaling, and translation in time-series databases.*** In proceedings of th 21st Int'l Conference on Very Large Databases. Zurich, Switzerland, Sept. 11-15 pp 490-501.

[6] Chakrabarti, K & Mehrotra, S. (1999). ***The Hybrid Tree: An Index Structure for High Dimensional Feature Space.*** Proc of the IEEE International Conference on Data Engineering.

[7] Guttman, A (1984). ***R-trees: A dynamic index structure for spatial searching.*** In Proc. ACM SIGMOD Conf., pp 47-57.

[8] Lin, J., Keogh, E., Patel, P. & Lonardi, S (2002). ***Finding motif in time series.*** In the 2nd Workshop on Temporal Data Mining, at the 8th ACM SIGKDD, International Conference on Knowledge Discovery and Data Mining. Edmonton, Alberta, Canada.

[9] Lin, J., Keogh, E., Lonardi, S. & Chui, B. (2003). ***A Novel Symbolic Representation of Time Series, with Applications to Classification, Clustering, Query by Content and Anomaly Detection.*** In SIGKDD'03, August 24-27, 2003, Washington, DC.

[10] Chan, K. & Fu, A. W. (1999). ***Efficient time Series Matching by Wavelets.*** In Proceedings of the 15th IEEE Int'l Conference on Data Engineering. Sydney, Australia, Mar 23-26. pp 126-133.

[11] Geurts, P. (2001). ***Pattern Extraction for Time Series Classification***. In Proceedings of the 5th European Conference on Principles of Data Mining and Knowledge Discovery. Sep 3-7, Freiburg, Germany. Pp 115-127.

[12] Larsern,R. J. & Marx, M. L, (1986) ***An introduction to Mathematical Statistics and its Applications***. Prentice Hall, Englewood, cliff, N. J. 2nd Edition.

[13] Bautista-Thompson y Figueroa-Nazuno. ***Matriz de Conocimiento sobre la Complejidad de Predicción en Series de Tiempo.*** VII Congreso Iberoamericano de Reconocimiento de Patrones (CIARP), 2002.

[14] J. M. Medina-Apodaca and J. Figueroa-Nazuno y S. García-Benitez. ***Time Series Modeling using Recurrence Plots and Face Recognition Techniques,*** VII Congreso Iberoamericano de Reconocimiento de Patrones (CIARP), 2002.

[15] ***Base Mexicana de Datos de Sismos Fuertes***, Sociedad Mexicana de Ingeniería Sísmica A.C.

[16] Agrawal, R., Psaila, G., Wimmers, E. L. & zait, M. (1995). ***Querying shapes of histories***. In proceedings of th 21st Int'l Conference on Very Large Databases. Zurich, Switzerland, Sept. 11-15 pp 502-514.

[17] Berndt, D. J. & Clifford, J. (1996). ***Finding patterns in time series: a dynamic programming approach***. Advances in Knowledge Discovery and Data Mining. AAAI/MIT Press, Menlo Park, CA. pp 229-248.

[18] Bozkaya, T., Yazdani, N. & Ozsoyoglu, Z. M. (1997). ***Matching and indexing sequences of different lengths.*** In proceedings of the 6th Int'l Conference on Information and Knowledge Management. Las Vegas, NV, Nov 10-14. pp 128-135.

[19] Faloutsos, C., Ranganathan, M. & Manolopoulos, Y. (1994). ***Fast subsequence matching in time-series databases.*** In proceedings of the ACM SIGMOD Int'l Conference on Management of Data. Minneapolis, MN, May 25-27. pp 419-429.

[20] Takens, F. (1986): ***Detecting strange attractors in turbulence.*** D.A. Rand y L.-S. Young (eds.): Lecture Notes in Mathematics: Dynamical Systems and Turbulence. Springer-Verlag, pp. 366-381

[21] Rafiei, D. & Mendelzon, A. O. (1998). ***Efficient retrieval of similar time sequences using dft.*** In proceedings of the 5th Int'l Conference on Foundations of Data Organization and Algorithms. Kobe, Japan, Nov 12-13.

[22] Struzik, Z. & Siebes, A. (1999). ***The haar wavelet transform in the time series similarity paradigm.*** In proceedings of Principles of Data Mining and

Knowledge Discovery, 3rd European Conference. Prague, Czech Republic, Sept 15-18. pp 12-22.

[23] Wu, Y., Agrawal, D. & El Abbadi, A. (2000). ***A comparison of dft and dwt based similarity search in time-series databases.*** In proceedings of the 9th ACM CIKM Int'l Conference on Information and Knowledge Management. McLean, VA, Nov 6-11. pp 488-495.

[24] Yi, B. & Faloutsos, C. (2000). ***Fast time sequence indexing for arbitrary lp norms.*** In proceedings of the 26th Int'l Conference on Very Large Databases. Cairo, Egypt, Sept 10-14. pp 385-394.

[25] Yi, B., Jagadish, H. & Faloutsos, C. (1998). ***Efficient retrieval of similar time sequences under time warping.*** In proceedings of the 14th Int'l Conference on Data Engineering. Orlando, FL, Feb 23-27. pp 201-208.

[26] M. T. Rosenstein and P. R. Cohen, ***Continuous Categories for a Mobile Robot***, Proceedings of sixteenth National Conference on Artificial Intelligence, 1999.

[27] T. Sauer, J. A. Yorke, and M. Casdagli, ***Embedology,*** Journal of statistical Physics, vol. 65, pp. 579-616, 1991.

[28] Rodríguez-Elizalde, J., & Figueroa-Nazuno J., ***Earthquakes Classifications using Data Mining Techniques.*** The 8th World Multi-Conference on Systemics, Cybernetics and Informatics. July 18-21, 2004 - Orlando, Florida, USA.

[29] Rodríguez-Elizalde, J., & Figueroa-Nazuno J., ***Clustering Time Series with a Symbolic Representation,*** IEEE ROC&C, Nov 2003, Acapulco, Gro. México.

[30] Rodríguez-Elizalde, J., & Figueroa-Nazuno J., ***Quiron: Similarity Search on Time Series Database,*** IEEE ROC&C, Nov 2003, Acapulco, Gro. México.

[31] D. H. Wolpert and W. G. MacReady. ***No free lunch theorems for optimization***. IEEE Transactions on Evolutionary Computation, Vol. 1, No. 1, April, 1997, pags. 67-82

[32] A. Pal and S.K. Pal. ***Patter recognition: Evolution of methodologies and data mining.*** In S.K. Pal and A. Pal, editors, *Pattern Recognition: From Classical to Modern Approaches, pages 1-23*, World Scientific, Singapure, 2001.

[33] ***Similitud De Señales De Sísmicas Por La Técnica De Mapa Recurrente***, R. Islas-Barrera, S. García-Benítez, J. Figueroa-Nazuno, XLVII Congreso Nacional de Física, Memorias en CD, Octubre 2004, ISSN: 01874713

[34] ***Clasificación Y Reconocimiento de Señales Sísmicas Empleando la Técnica de Mapa Recurrente Y Eigenfaces***, E. Meléndez Montes de Oca, A. Barajas Rodríguez, H. Jiménez Hernández, J. Figueroa Nazuno, XLVII Congreso Nacional de Física, Memorias en CD, Octubre 2004, ISSN: 01874713

[35] ***Clasificación y Semejanza de Espectros de Respuesta de Aceleración***, A. Angeles-Yreta, S. García, J. Figueroa-Nazuno, F. Correa, M. Ortega, H. Solís-Estrella y K. Ramírez-Amaro. XV congreso nacional de ingeniería sísmica, Septiembre 2005.