TESIS DOCTORAL

Reducción del volumen y complejidad de datos utilizando redes complejas con atributos temporales

Autor:

Sergio Iglesias Pérez

Directores:

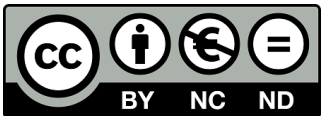
Regino Criado Herrero

Santiago Moral Rubio

Programa de Doctorado en Ciencias

Escuela Internacional de Doctorado

2023



Pon aquí tu dedicatoria.

AGRADECIMIENTOS

Gracias a todos los que me han ayudado durante este proceso, tanto en el mundo profesional como en el personal para poder llevar a cabo esta tesis.

En especial, a mis directores de tesís Regino y Santiago, que sin su paciencia y seguimiento esta tesis no podría haber sido posible.

Gracias a mi familia por el apoyo ofrecido durante estos casi cuatro años de trabajo nocturno.

CONTENIDOS PUBLICADOS Y PRESENTADOS

Durante la realización de esta tesis doctoral se han llevado a cabo investigaciones que han sido publicadas en diferentes revistas científicas de interés.

- "A new approach to combine multiplex networks and time series attributes: Building Intrusion Detection Systems (IDS) in cibersecurity", Chaos, Solitons&Fractals 150 (2021), 111143, JCR-2021 Impact Factor: 9.922. Este artículo está incluido parcialmente en la realización de esta tesis. El material de esta fuente incluido en la tesis no está señalado por medios tipográficos ni referencia

- "Combining multiplex networks, time series attributes and Big Data: A new way to optimize real estate forecasting in New York from cab rides", Physica A: Statistical Mechanics and Its Applications 609 (2023),128306, https://doi.org/10.1016/j.physa.2022.128306, JCR-2021 Impact factor: 3.778. Este artículo está incluido parcialmente en la realización de esta tesis. El material de esta fuente incluido en la tesis no está señalado por medios tipográficos ni referencia

- "Increasing the Effectiveness of Network Intrusion Detection Systems (NIDSs) by Using Multiplex Networks and Visibility Graphs", Mathematics 2023, 11 (1), 107 DOI: 10.3390/math11010107 , MDPI, Open Access, JCR-2021 Impact factor: 2.592.Este artículo está incluido parcialmente en la realización de esta tesis. El material de esta fuente incluido en la tesis no está señalado por medios tipográficos ni referencia

OTROS MÉRITOS DE INVESTIGACIÓN

Durante el transcurso de esta tesis doctoral se han realizado presentaciones de póster en diferentes simponiums internacionales:

- "Temporal behaviour complex network: Combining multiplex networks, time series attributes and Big Data: A new way to optimize real estate forecasting in New York from cab rides": XVIII Workshop on Instabilities and Nonequilibrium Structures 2021
- "Graph Dimension Reduction: Time series features give us the ability to reduce the graph dimensions": Manifesting Intelligence 2020

ÍNDICE GENERAL

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

1. INTRODUCCIÓN GENERAL

Durante los últimos años se dispone, cada vez más, de nuevos dispositivos y tecnologías para el procesamiento de datos de la realidad. Esta evolución se ve reflejada en el aumento exponencial de la información que reside en Internet. Como se aprecia en la figura 1.1 basada en la información de [1] y 1.2 basada en [2]. Este aumento, permite conocer con más detalle la realidad que nos rodea haciendo que los sistemas informacionales generen un entorno donde los datos nos proporcionan una imagen cada vez más parecida a la realidad.

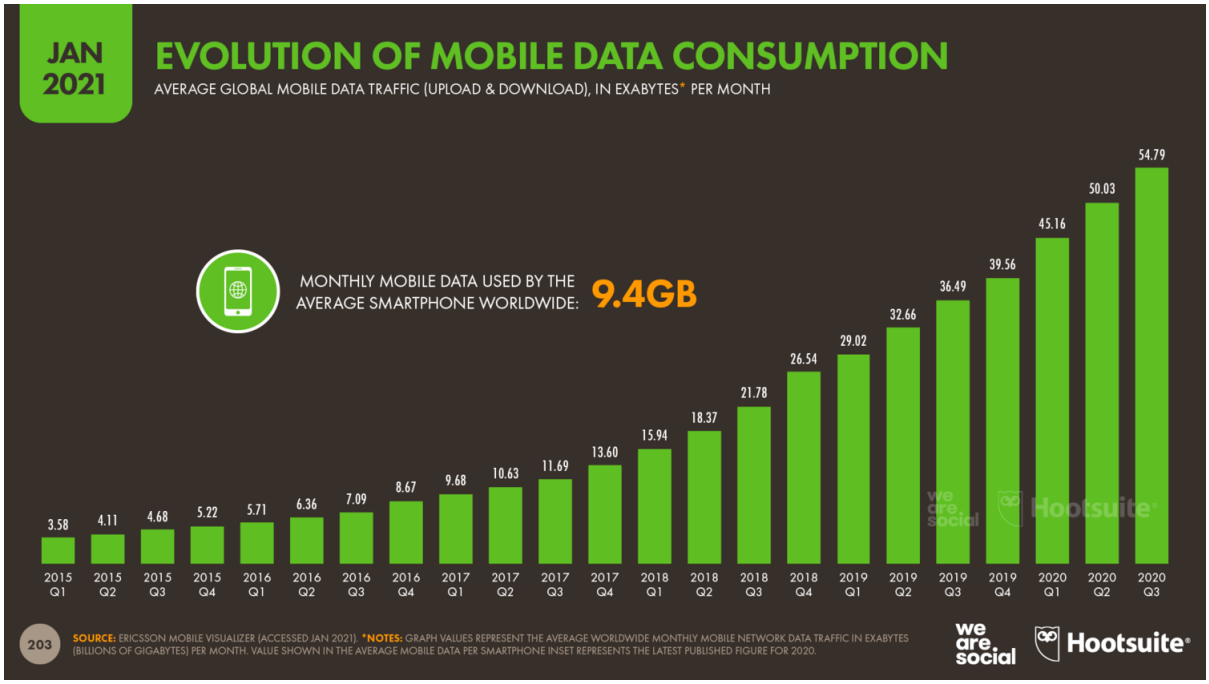


Figura 1.1: Evolución del consumo de datos móviles

Toda esta nueva información supone un nuevo reto en el manejo y obtención de resultados. Este reto se centra en las novedosas técnicas conocidas dentro del Big Data, donde se pretende dar una respuesta al mismo. Dotarse de nuevas técnicas de almacenamiento, computación y análisis de cantidades elevadas de datos. A pesar de todas estas prácticas, es necesario afrontar nuevas aproximaciones matemáticas y de representación de la información para poder adaptarse a las actuales cantidades de información de las que se dispone en cada uno de los desafíos actuales.

En esta tesis, presentaremos una diferente aproximación para poder realizar la representación y el cómputo de grandes cantidades de información usando técnicas utilizadas anteriormente de forma independiente que permiten, en conjunto, dar la respuesta a esta nueva necesidad. En concreto, presentaremos las capacidades de los grafos para poder dar solución a las modificadas necesidades que se nos presentan

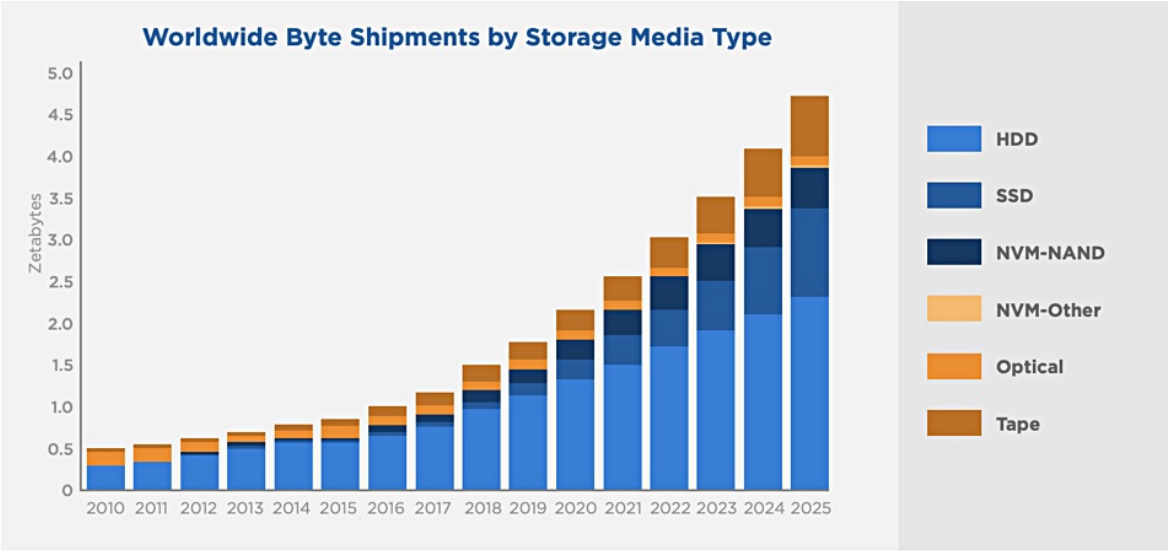


Figura 1.2: Evolución de los datos almacenados en la nube

en la actualidad. Los grafos son excelentes herramientas para la representación de la realidad. Son una gran herramienta para plasmar las interacciones entre comunidades muy grandes de elementos. También se utilizarán técnicas de análisis temporal para poder obtener atributos temporales de la información de la que disponemos.

Durante las últimas décadas, ha habido gran cantidad de investigaciones que proponen la utilización de grafos para poder representar la complejidad de la realidad [3]-[5]. Desde el siglo XVIII, donde aparecieron las primeras menciones a los grafos hasta la actualidad, han ido dando respuesta a entornos donde los activos o nodos tenían relaciones más complejas entre ellos. A partir de grafos simples se ha llegado en las últimas décadas a la creación de redes complejas entre los nodos y la creación de capas, diferentes tipos de grafos y atributos altamente complejos que representan una visión más veraz de las interacciones que existen en la realidad, como pueden ser los grafos multiplex o multilayer.

Sin embargo, estos estudios sobre los grafos no se han visto acompañados de investigaciones en las que se pueda incluir una evolución temporal de dichos grafos. Lo que pretendemos en esta tesis es crear nuevos algoritmos de grafos donde podamos recolectar y relacionar las iteraciones temporales entre los diferentes activos o nodos del grafo. Estas interacciones podrán incluirse dentro del grafo, con el fin de obtener información no existente hasta la fecha.

Por otro lado, la evolución de la información a lo largo del tiempo nos proporciona una capacidad adicional que puede proporcionarnos explicaciones clave sobre la forma en que las entidades almacenadas se relacionan entre sí. Para ello, profundizaremos en los diferentes métodos que se utilizan en la caracterización temporal de las relaciones entre entidades, como pueden ser métodos de clasificación de series

temporales o sus formas de representación con grafos.

1.1. Objetivos generales de la tesis

La presente tesis tiene como punto de partida los retos señalados en la introducción y aporta nuevos enfoques fundamentados en las posibilidades de los grafos y las redes temporales para dar una respuesta que contribuya a su solución. Para llevar a cabo esta tarea, la tesis se basa en el objetivo principal de dichos desafíos que se han presentado en los últimos tiempos e intenta aportar nuevas soluciones dentro de las posibilidades de los grafos y series temporales para resolverlos. Para ello, esta tesis se centra en diseñar una novedosa técnica para incluir en un grafo las cualidades temporales de la relación entre los diferentes nodos. Tal y como hemos planteado anteriormente, la complejidad del grafo crece exponencialmente cuando se incrementa el número de nodos y sus interacciones a través de las aristas. En esta tesis pretendemos agrupar la información sobre las relaciones temporales de los nodos dentro de atributos complejos en los grafos permitiendo así reducir la complejidad de éstos y proporcionando información compleja.

En este trabajo proponemos, como primer paso, crear nuevos atributos temporales en los grafos que permitan agrupar las interacciones entre los nodos, disminuyendo la información del nodo a un conjunto reducido de atributos de las aristas. Para ello, es necesario identificar las mejores técnicas para adquisición de los atributos temporales. En un segundo paso, incluiremos estos atributos de forma óptima para que sean explotables dentro de un grafo de alta dimensionalidad.

1.2. Marco conceptual

La utilización de los grafos para la representación de las interacciones entre diferentes activos ha sido empleada en múltiples ocasiones y en ámbitos muy diferentes desde su aparición en el siglo XVIII por parte del matemático suizo Leonhard Euler en 1736. Desde sus primeros momentos, los grafos han sido utilizados para la visualización y resolución de problemas complejos en los que las relaciones entre las entidades se simplificaban en gran medida por su representación en un formato gráfico, respecto a otras representaciones de los datos.

Hasta la fecha, los grafos preferentemente presentaban fotos fijas de las relaciones entre los diferentes nodos permitiendo describir las relaciones entre ellos así como interfiriendo su existencia y múltiples funciones adicionales, tal y como se puede apreciar en estudios como [6] y [7].

En las últimas décadas, se han iniciado diferentes estudios para poder representar la evolución de las interacciones entre los diferentes nodos de los grafos a lo largo del tiempo. Estos estudios se han centrado en la representación de las interacciones en

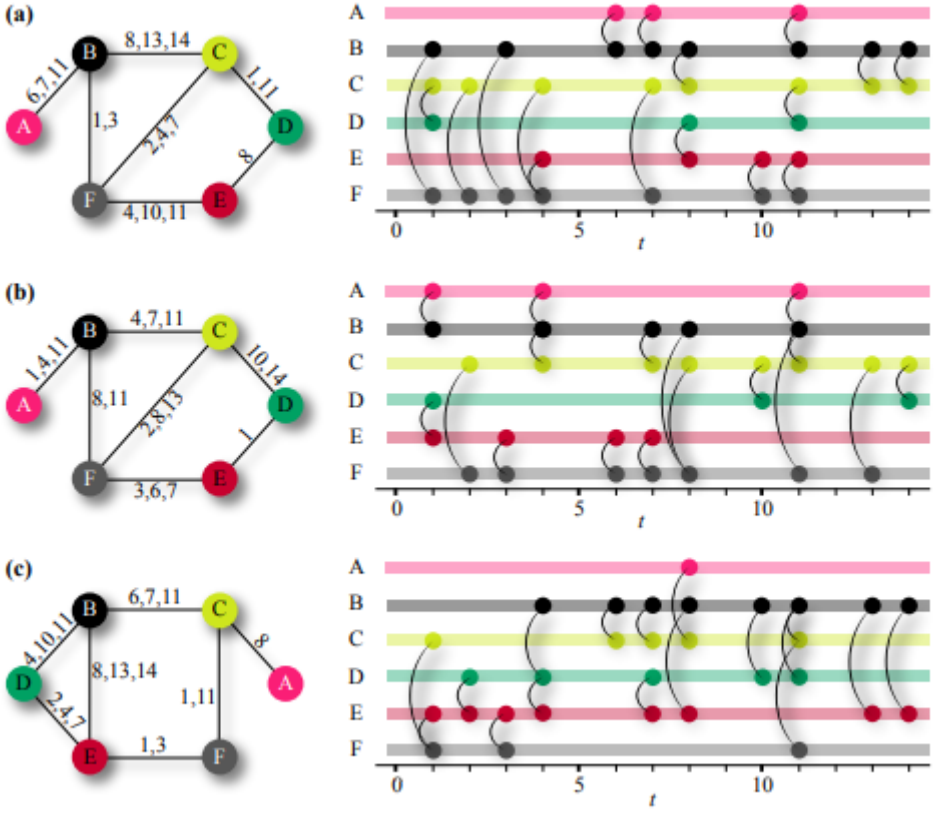


Figura 1.3: Grafos temporales

el tiempo para su posterior estudio. Entre estas investigaciones que están definiendo la línea de trabajo de grafos temporales podemos mencionar estudios como los siguientes [8], [9], [10] y [11].

Todos estos estudios realizados en las últimas décadas se centran principalmente en la representación de interacciones entre los nodos incluyendo un modo de representar la variable temporal. Como podemos ver en la figura 1.3 extraída y reproducida del trabajo [8], desde 2012 se han llevado a cabo diferentes aproximaciones para representar interacciones de distinta naturaleza de forma discreta.

A lo largo del desarrollo de la tesis se pretende presentar una forma alternativa para la adquisición de características temporales de las relaciones de los nodos, no centrándose en sus interacciones puntuales, sino focalizando el trabajo en la detección de comportamientos entre los nodos y la importancia que dicho comportamiento nos facilita para obtener información sobre la relación entre los diferentes nodos de un grafo.

Esta nueva aproximación, trata de utilizar técnicas de representación de series temporales para la adquisición de características temporales e introducir en el grafo estos atributos en lugar de representar la evolución de cada una de las relaciones entre los dos nodos que conforman el origen y el destino de cada arista.

Para tal fin, una parte importante de la investigación se centra en la representación

de las series temporales, sus características y capacidades de detección basadas en predicción futura, detección de anomalías y clasificación de series temporales. El objetivo de este análisis es la selección de los mejores atributos temporales de las series temporales que reflejan las relaciones entre los diferentes nodos.

Una vez obtenidos estos atributos temporales, se propone una solución para incluirla dentro del grafo. En esta tesis se propone como solución un grafo multiplex donde las capas reflejen los diferentes comportamientos entre los nodos y no una visión fija de la relación entre ellos. Uno de los resultados de la tesis, por tanto, es la definición de los grafos temporales con atributos temporales.

Estos nuevos atributos aportan al grafo unas características y una nueva perspectiva que difícilmente sería posible conseguir de otra manera.

1.3. Metodología

La metodología que vamos a seguir para desarrollar esta tesis está contenida en tres capítulos en los que presentaremos el uso de la propuesta de reducción de dimensionalidad de los grafos en diferentes entornos, utilizando para ello diferentes tipos de representación de las series temporales.

En los tres capítulos utilizaremos la propuesta de grafos multiplex con atributos temporales para condensar los datos temporales que relacionan las entidades en un número pequeño de atributos de cada entidad.

En este primer apartado dedicado a la introducción haremos una revisión del estado del arte de las dos técnicas que vamos a utilizar en el marco de esta tesis. Presentaremos en primer lugar el estado del arte en la representación de las series temporales, su clasificación y los diferentes tipos de grafos. En segundo lugar, presentaremos las tipologías de grafos y sus aproximaciones a los grafos temporales hasta la actualidad.

A continuación, focalizaremos nuestra atención en el entorno de la ciberseguridad. En el primer capítulo nos centraremos en la representación matemática de un entorno altamente complejo como es una red de ordenadores y sus interacciones a lo largo del tiempo. Dentro de este capítulo proponemos su caracterización mediante un grafo multiplex con atributos temporales desde el cual seremos capaces de obtener atributos complejos sobre los diferentes ordenadores que conforman la red. Estos atributos nos permitirán poder detectar posibles atacantes dentro de un entorno muy complejo.

El segundo capítulo se centra en la utilización de los grafos multiplex con características temporales en el entorno de las smart cities. En este entorno la cantidad de nodos e interacciones entre ellos ha presentado un reto que esta aún a mucha distancia de estar completamente solucionado. Propondremos la creación de un grafo

multiplex con la información de los viajes de taxis entre diferentes zonas de la ciudad de Nueva York para obtener características complejas de las zonas de dicha ciudad. Estas características nos permitirán predecir la tendencia en el precio del mercado inmobiliario en el futuro.

El tercer y último capítulo se centra en la utilización de una segunda técnica de clasificación de las series temporales para la creación de los grafos multiplex con características temporales. En este capítulo, utilizaremos métodos de clasificación de series temporales basados en grafos de visibilidad en vez de sistemas de clasificación de series temporales basados en sus formas o frecuencias utilizados en los dos primeros capítulos de la tesis.

1.4. Representación de series temporales

1.4.1. Introducción

En este apartado nos centraremos en los diferentes técnicas para poder analizar las series temporales de forma agregada y obtener una información sobre sus características.

En la actualidad, hay diversas aproximaciones para la adquisición y agrupación de las series temporales en diferentes grupos sobre los que trabajar. Por ello, a lo largo de este capítulo re-visitaremos las diferentes técnicas que se usan en la actualidad para la agregación de series temporales. Durante este proceso empezaremos por los algoritmos, que denominaremos tradicionales, como pueden ser todos los fundamentados en técnicas basadas en distancias como, por ejemplo, k-means. Continuaremos con los algoritmos basados en sistemas de redes neuronales y terminaremos por aproximaciones en los que utilizan sistemas de grafos para la representación de las series temporales.

1.4.2. Representación de series temporales

El primer reto es la definición de una serie temporal. Como se define en el libro [12] una serie temporal es una serie de observaciones x_t, donde cada una de las observaciones x se corresponde con un tiempo t específico. Aunque las series temporales se pueden dividir en series continuas o discretas, a partir de este momento, nos centraremos siempre en series discretas. Las series discretas tienen como característica básica que se construyen a partir de un conjunto discreto de observaciones, formando un conjunto de observaciones continuo sobre un intervalo de tiempo predefinido que puede ser $t = [t_0, t_1, t_2, ..., t_n]$ donde n es el número de observaciones de la serie temporal.

La separación entre las diferentes observaciones puede ser constante o aleatoria. A lo largo de este trabajo vamos a considerar siempre series temporales discretas con un espacio temporal constante entre las diferentes observaciones adquiridas.

Este tipo de series temporales son las comunes y las más utilizadas para la adquisición de datos en entornos del ámbito de la ingeniería para su posterior análisis.

El primer reto de toda serie temporal es la elección de una representación matemática adecuada para poder ser tratada posteriormente. Con respecto a este tema existe una amplia bibliografía, como puede ser [13], [14], [15], [16] que describe técnicas muy dispares que vamos a presentar a continuación:

- Transformadas de Fourier
- Wavelets
- Mapeos simbólicos

- Grafos

Transformadas de Fourier

Las transformadas de Fourier son una técnica utilizada muy frecuentemente en entornos de telecomunicaciones y de señales para la descomposición de una serie temporal en las diferentes frecuencias que forman parte de la serie. Esta descomposición es muy útil en entornos de telefonía o de envío de señales donde se pueden aislar los diferentes componentes de la señal y su posterior tratamiento de forma independiente.

Podemos definir la transformada de Fourier como una forma de pasar una señal de un entorno temporal a un entorno de frecuencias y viceversa. Para esta transformación se utilizan las siguientes fórmulas:

$$F(w) = \int_{-\infty}^{\infty} f(t)e^{iwt}\,dt$$

$$f(t) = \frac{1}{2\pi}\int_{-\infty}^{\infty} F(w)e^{iwt}\,dw$$

Wavelets

Las wavelets permiten descomponer la información compleja, como la música, el habla, las imágenes y los patrones, en formas elementales, en diferentes posiciones y escalas y reconstruirlas, posteriormente, con gran precisión. Tal y como se comenta en diferentes referencias como [17], la transformada Wavelet de una función es una versión mejorada de la transformada de Fourier. La transformada de Fourier es una potente herramienta para analizar los componentes de una señal estacionaria. Sin embargo, no se adapta tan sencillamente a las señales no estacionarias, mientras que la transformada de Wavelet permite analizar los componentes de una señal no estacionaria.

La primera alusión específica del término wavelet se produce en 1982 cuando Jean Morlet, un geofísico francés, hace su primera mención. La describe como una pequeña onda e inicia la transformada de wavelet, centrado su uso en el análisis de señales para seismos. Tras esta primera referencia, y con la ayuda de Alex Grossmann, físico teórico, definen la transformada inversa de wavelet.

Aunque es una técnica que sólo tiene treinta años, ha sido utilizada en gran número de casos de uso por su utilidad para la detección de cambios bruscos en las series temporales.

Para determinarla podemos suponer un conjunto de funciones $f_0, f_1, f_2, ..., f_n$ de forma que una serie temporal se puede definir como:

$$f_{(x)} = \sum_{n=0}^{\infty} c_n f_n(x)$$

expresión en la que existe un número n de coeficientes definidos como c_n. Gracias a esta representación, se puede utilizar y describir una serie temporal en entornos muy diversos.

Así, es posible definir wavelet como una pequeña onda que decae rápidamente en el tiempo, al contrario de las señales sinuidales que se utilizan en los estudios de Fourier. Este ligero matiz en la descripción, permite que sea más rápida para la detección de cambios abruptos en el tiempo.

La transformada de Wavelet tal y como propone [17] se puede definir como una familia de funciones construidas a partir de traslaciones y dilataciones de una función única llamada "wavelet madre"$\psi(t)$. Se definen como

$$\psi_{a,b}(t) = \frac{1}{\sqrt{|a|}} \psi\left(\frac{t-b}{a}\right)$$

donde $a, b \epsilon R, a \neq 0$.

El parámetro a es el parámetro de escalado o escala, y mide el grado de compresión. El parámetro b es el parámetro de traslación que determina la ubicación temporal de la wavelet. Si $|a| < 1$, la wavelet es la versión comprimida de la wavelet madre y corresponde principalmente a las frecuencias más altas. Por otro lado, cuando $|a| > 1$, entonces $\psi_{a,b}(t)$ tiene un mayor anchura de tiempo que $\psi(t)$ y corresponde a frecuencias más bajas. Por tanto, las wavelets tienen anchos de tiempo adaptados a sus frecuencias. Esta es la principal razón del éxito de las wavelet en el procesamiento de señales y en el análisis de señales de tiempo-frecuencia.

Mapeos simbólicos

La simbolización de las series temporales permite transformar una serie de datos que representen una serie de observaciones en una serie de símbolos discretizados que representan la evolución de la serie temporal inicial.

Esta aproximación permite reducir el grado de complejidad de la serie original en gran medida y permite almacenar la serie temporal en una cantidad más pequeña de información. Más aún, este mapeo simbólico a series de números también puede ir mas allá creando series de códigos que pueden ser la combinación de los símbolos iniciales de codificación.

El primer paso, como proponen en [18] tras la toma de la información, es la definición del conjunto de símbolos que se va a utilizar para la grabación de los datos medidos. En este apartado, se deberá definir el rango de la digitalización a realizar

para transformar el dato analógico en datos digitales. El número de símbolos posibles, n, se denomina tamaño del conjunto de símbolos que puede empezar desde el menor de los tamaños de $n = 2$ donde sólo se dispone de dos valores posibles: 0 y 1 a valores mucho más elevados de n. Cuando los valores son más altos corresponden a una discriminación más refinada de los detalles de la medición, incluyendo los efectos de cualquier ruido de medición.

Esta definición de símbolos posibles puede llegar al límite donde hay tantos símbolos como valores tiene la serie temporal. En este caso, se considera que no se pierde ningún aspecto de la medida ya que cada una de ellas se adquiere en su totalidad. Existen varias para seleccionar el número de símbolos disponibles ante una medida como puede ser [19].

Tras la transformación a un conjunto de símbolos, el siguiente paso para la identificación de patrones temporales es la construcción de secuencias de símbolos, que pueden ser descritas como palabras, a partir de las series de símbolos, reuniendo grupos de símbolos juntos en orden temporal.

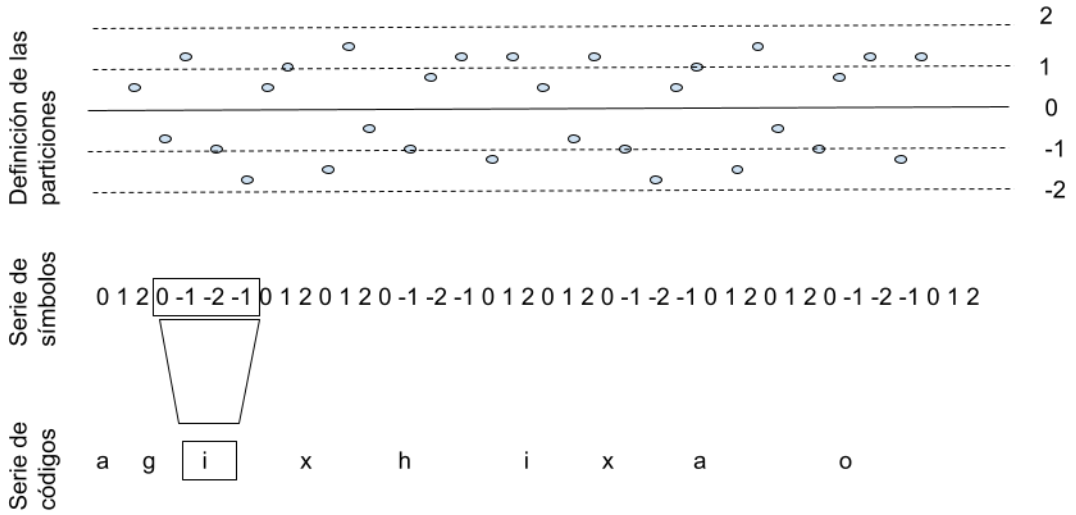


Figura 1.4: Transformación de serie temporal a símbolos y de aquí a códigos

En la figura 1.4 podemos ver los pasos que se siguen en la transformación de una serie temporal en una serie de códigos que reducen la cantidad de información a almacenar para representar la serie analizada. En el primer paso, se definen las particiones de los datos en segmentos que se transforman en el mismo símbolo. Una vez definidas las particiones, se traduce la serie al conjunto de símbolos que hayamos definido. Como puede apreciarse en la figura hay valores diferentes que se transforman en el mismo símbolo. Esto propicia una pérdida de información que en la mayoría de los casos será despreciable.

Una vez que se ha realizado la creación de la serie de símbolos, se buscan patrones a lo largo del tiempo que permitan traducir un conjunto de símbolos que tengan

el mismo orden temporal en todas sus ocurrencias por un código que reduce la cantidad de información almacenada. En este caso, los códigos que hemos utilizado son letras que representan conjuntos de símbolos (números).

Grafos de visibilidad naturales

Desde la primera vez que se definió el Grafo de Visibilidad Natural [20], este concepto se ha utilizado en varias investigaciones para transformar las series temporales en redes. Esta transformación nos da la posibilidad de utilizar la teoría de redes en eventos temporales, caracterizando una serie temporal como una red.

Para definir un grafo de Visibilidad Natural dibujamos una serie temporal como lo hacemos en la Figura 1.5. Considerándolo como un paisaje, vinculamos cada punto de la serie temporal (cada registro) con cualquier otro registro que pueda verse desde él. Si se pueden ver dos nodos, creamos una arista entre ellos. En caso contrario, no aparece ninguna arista entre ellos. De manera formal, podemos definir la visibilidad de dos registros (t_1, y_1) y (t_2, y_2) si no hay ningún otro registro definido como (t_3, y_3) de manera que $t_1 < t_3 < t_2$ y tal que

$$y_2 + (y_1 - y_2)\frac{t_2 - t_3}{t_2 - t_1} < y_3.$$

Básicamente, utilizando estas técnicas el grafo (o red) creado tiene las siguientes características:

- Conectado: cada nodo tiene al menos dos conexiones: nodo izquierdo y derecho en la serie temporal.
- No dirigido: no hay dirección entre los nodos.
- Invariante: la red es la misma aunque se re-escalen o trasladen. Cualquier cambio de escala en el eje horizontal o vertical no modifica la red. Del mismo modo, cualquier traslación horizontal o vertical tampoco afecta al grafo creado.

Como resultado de este procesamiento, la serie temporal que hemos utilizado como ejemplo se transforma en un grafo de visibilidad que se muestra en la Figura 4.14.

Grafos de visibilidad horizontales

Basado en los mismos principios que el grafo de visibilidad natural, un año después de su definición se propuso un nuevo grafo de visibilidad llamado grafo de visibilidad horizontal. Ambos gráficos fueron propuestos por el mismo equipo en un corto período de tiempo, basándose en la evolución de sus teorías.

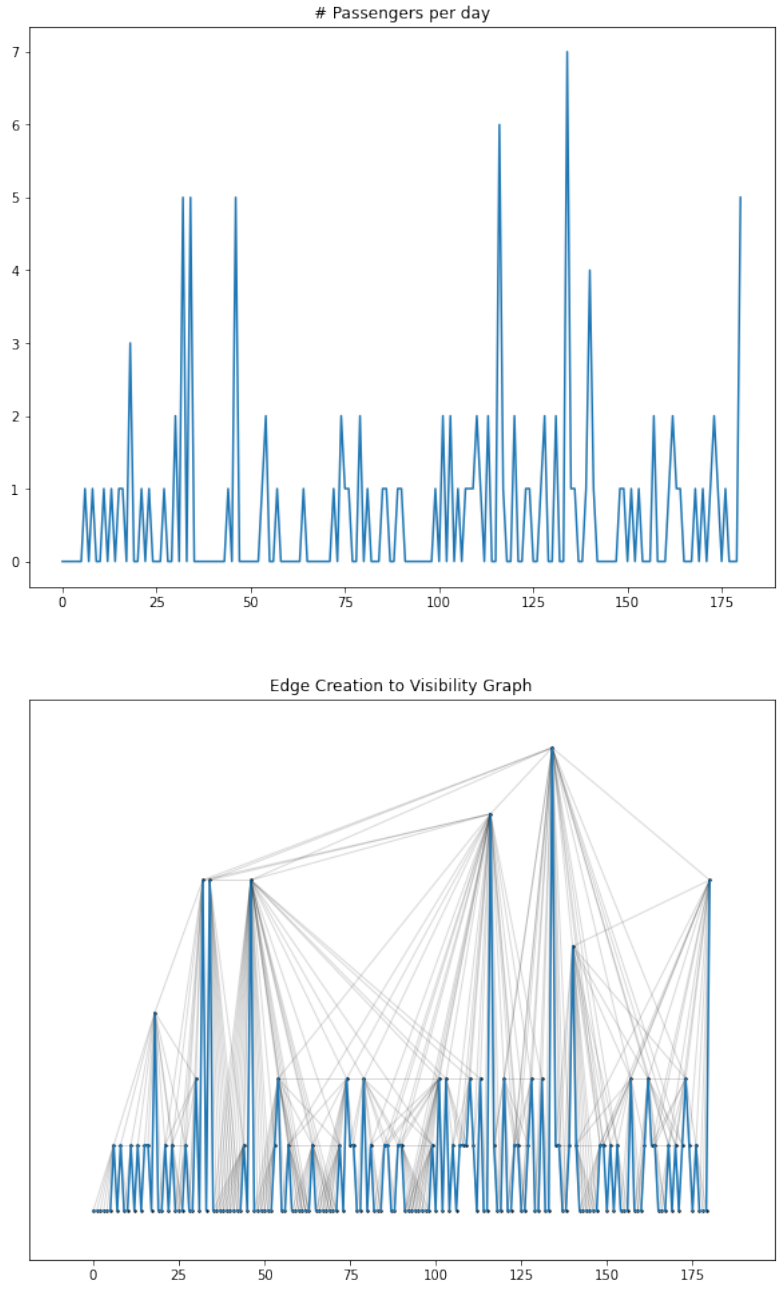


Figura 1.5: Series temporales a grafos de visibilidad

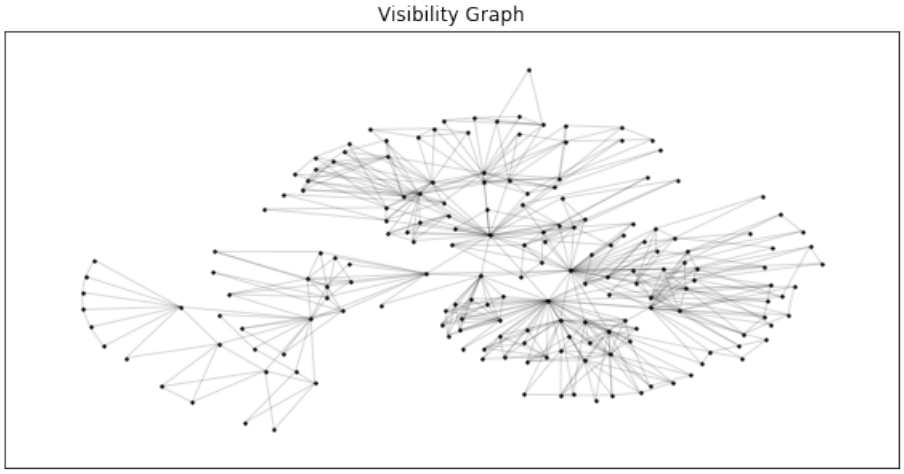


Figura 1.6: Grafo de visibilidad de una serie temporal

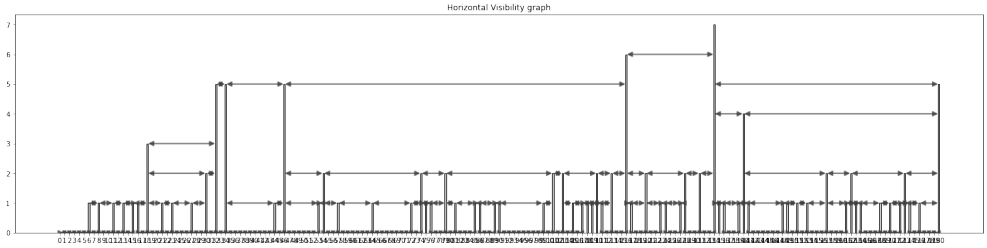


Figura 1.7: Creación de un grafo de visibilidad horizontal desde una serie temporal

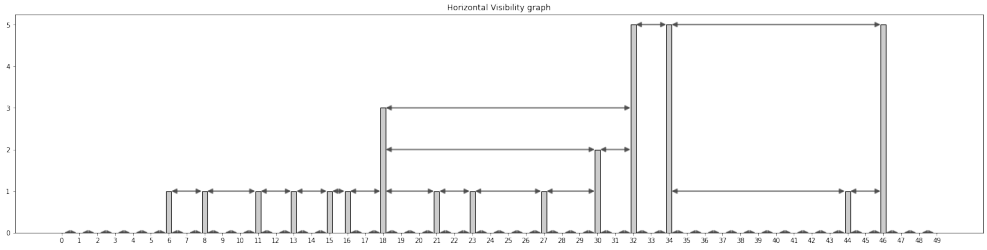


Figura 1.8: Creación de un grafo de visibilidad horizontal desde una serie temporal. Zoom de los 50 primeros puntos

Al igual que el gráfico anterior, este nuevo enfoque se basa también en la visibilidad de los nodos. Como podemos ver en las Figuras 4.19 y 4.17 la base de la creación de este tipo de gráfico es la misma: cada registro de la serie temporal se define como un nodo en el grafo de visibilidad. Dos nodos definidos como (t_1, y_1) y (t_2, y_2) están conectados si los dos nodos están conectados horizontalmente, es decir, podemos trazar una línea entre ellos sin que ninguna otra altura de registro limite su visibilidad.

$$y_1, y_2 > y_n \quad \text{for all n where } 1 < n < 2$$

Como el anterior grafo de visibilidad natural, este nuevo tipo de grafo es:

- Conectado, como el NVG.
- Invariante frente a cualquier traslación o re-escalamiento.
- Irreversible: utilizando el grafo de visibilidad horizontal varias series temporales pueden crear el mismo grafo de visibilidad horizontal, por lo que es imposible recuperar la serie temporal a partir del gráfico. En la mayoría de las ocasiones esto no es un problema porque al realizar esta operación nuestro propósito es captar las propiedades estructurales de la serie temporal. En el caso de que la reversibilidad sea necesaria, necesitamos utilizar un grafo ponderado. En este caso, es factible definir un grafo reversible.
- Grafo no dirigido: Básicamente, no hay dirección entre dos nodos. Sin embargo, es posible crear un grafo dirigido utilizando la evolución temporal de la serie temporal, es decir, la dirección de las aristas es la dirección en la que aumenta el tiempo en la serie temporal.
- Finalmente hay que señalar que el grafo de visibilidad natural está más conectado que el HVG.

Como podemos ver en el ejemplo, consideramos la misma serie temporal que en el grafo de visibilidad natural, conectando los nodos adecuados. En la Figura 4.19 podemos ver los 181 registros de la serie temporal. Nos centramos en los 50 primeros registros en la Figura 4.17. En el ejemplo, el registro 18 está conectado a 4 nodos en el gráfico horizontal, que es mucho menor que la conectividad que tenía en el grafo de visibilidad natural.

1.4.3. Resumen

En este apartado hemos presentado las diferentes formas de representar una serie temporal para su posterior análisis. Según se puede apreciar, hay diferentes representaciones que aportan un valor diferencial. Cada una de ellas añade unas ventajas diferenciales.

- Transformada de Fourier: esta transformación nos facilita una descripción de las frecuencias que componen la serie temporal, por lo que será una de las más utilizadas en entornos que describan las características temporales de una serie
- Wavelets: como las transformadas de Fourier se basa principalmente en la detección del conjunto de frecuencias de una serie. La ventaja de los wavelets es la detección de cambios bruscos en las series temporales. En nuestro caso, no queremos centrarnos en cambios bruscos, sino en tendencias a lo largo del tiempo, por lo que la dejaremos a un lado en posteriores capítulos.
- Representación simbólica: es una representación que se usa ampliamente para la detección de patrones dentro de las series temporales. Permite la fácil detección de patrones y repeticiones escondidos dentro de series temporales muy largas. Como la anterior, no se centra en la generación de características temporales sino que se basa en la detección de patrones, por lo que también, tras haberla analizado y utilizado, vamos a dejarla de lado en posteriores capítulos.
- Representaciones basadas en grafos: estas representaciones nos permiten la detección de patrones no fácilmente visibles a simple vista. Por ello y por su rapidez de cómputo será una de las representaciones que utilizaremos a partir de ahora dentro de los siguientes capítulos.

En los siguientes capítulos que componen el desarrollo de esta memoria utilizaremos las representaciones basadas en grafos y las transformadas de Fourier para la clasificación de las mismas.

En ambos casos las ventajas que nos aportan son elevadas para poder transformar cualquier serie de datos a lo largo del tiempo en una agregación donde las características principales de la serie puedan ser utilizadas en los siguientes pasos, para la construcción y utilización de los grafos multiplex con características temporales.

1.5. Clasificación de series temporales

1.5.1. Introducción

En las últimas décadas la utilidad de las series temporales se ha incrementado debido al aumento de los datos recolectados y almacenados. El engrosamiento de estos datos permite poder evaluar el comportamiento de las observaciones a lo largo del tiempo. Esto nos permite poder identificar más y más series temporales en la información recolectada. Como el número de series temporales se incrementa, aumenta también la importancia de poder compararlas y clasificarlas.

Una vez que en el apartado anterior hemos analizado las diferentes opciones para poder representar las series temporales, en este nuevo capítulo vamos a centrarnos en diferentes técnicas que permiten clasificar las series temporales. Para tal fin, utilizaremos varias formas de representación como fuente para la agrupación y compararemos las ventajas y desventajas de cada una de las aproximaciones propuestas.

Entrando en materia, hay muchas aproximaciones en la clasificación de series temporales. Podemos agrupar las técnicas en tres grandes grupos que analizaremos posteriormente:

- clasificaciones basadas en características
- clasificaciones basadas en modelos
- clasificaciones basadas en distancias
- clasificaciones basadas en deep learning [21], [22], [23], [24]
- clasificaciones basadas en otras técnicas como los grafos.

1.5.2. Basadas en características

En los métodos de clasificación basados en características, las series temporales se transforman en vectores de características y luego se clasifican mediante un clasificador convencional, como una red neuronal o un árbol de decisión.

Esta aproximación se relaciona directamente con las técnicas de feature engineering en los que se crean, de forma automática, múltiples características de una serie temporal. Podemos poner como ejemplo la aproximación de [25] y [26] y [27] que implementa hasta más de 700 características de una serie temporal. Todo ello se ha concentrado en una librería en python que se ha instaurado como líder en esta técnica [28]. Como puede verse en la figura 1.9 que está obtenida en la web de [28] muestra las características que se pueden extraer de una serie temporal.

Una vez obtenidas estas características se puede hacer cualquier tipo de clasificación supervisada o no supervisada en cada una de las situaciones a analizar. Como cualquier situación de machine learning se deberá realizar un estudio de los diferentes algoritmos que pueden hacer la clasificación de los datos.

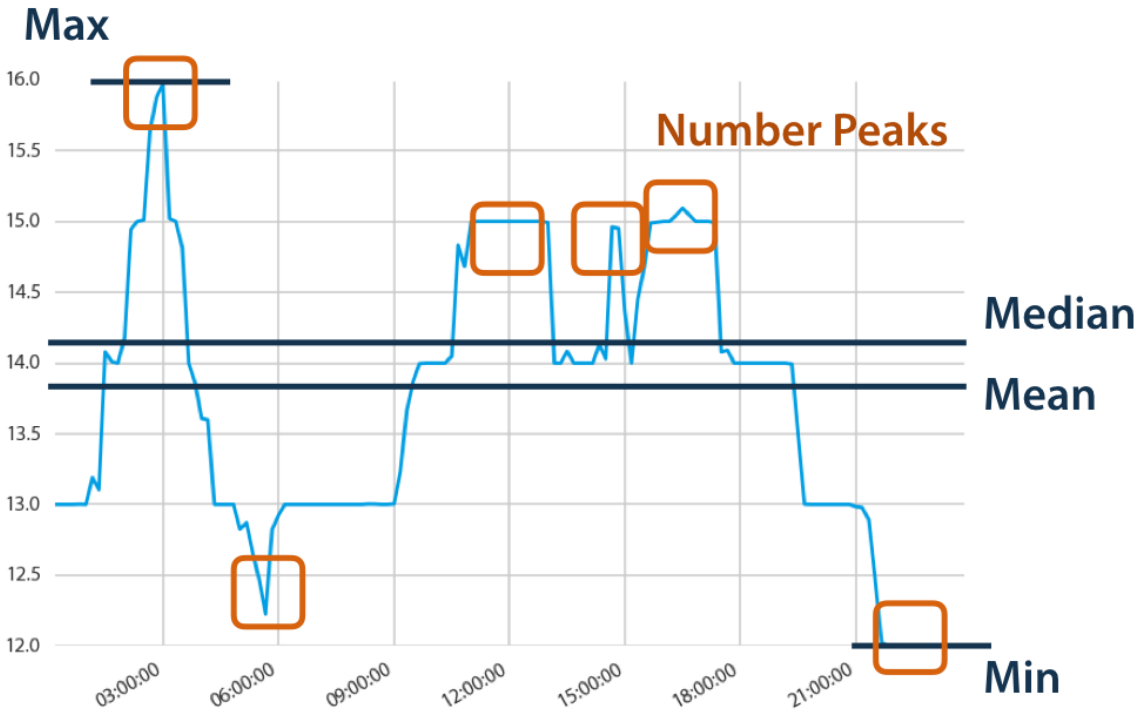


Figura 1.9: características de serie temporal from tsfresh

Podemos poner como referencia los algoritmos implementados en la librería sklearn [29] como ejemplos de clasificadores a utilizar. Los podemos dividir en:

- modelos lineales
- Support Vector Machines
- Stochastic Gradient Descent
- Nearest Neighbors
- Gaussian Processes
- Naive Bayes
- Decision Trees
- Ensemble methods
- Neural network models (supervised)

La utilización de estos u otros algoritmos permitirá la clasificación de las series temporales a través de sus características.

Algunos métodos de extracción de características incluyen métodos espectrales como transformada de Fourier discreta [30] o la transformada wavelet discreta , [31] donde se consideran características de frecuencia, o la descomposición de valores singulares [32], en la que se lleva a cabo para encontrar un conjunto óptimo de características. No nos centraremos en más detalle en este tipo de clasificación de series temporales al no ser objetivo de esta tesis.

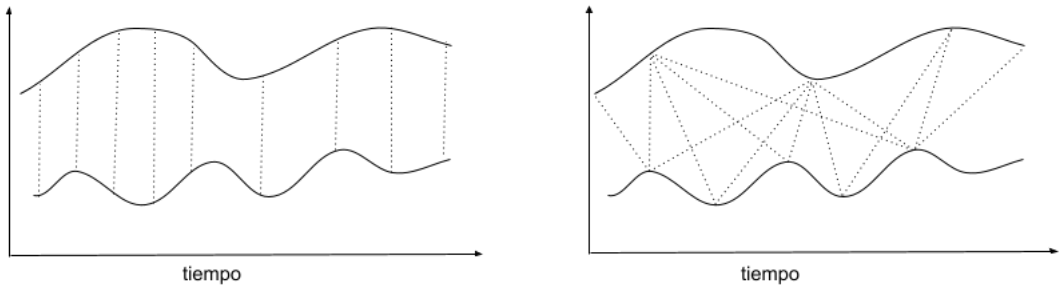


Figura 1.10: Medición de distancias en series temporales

Basadas en modelos

Este tipo de clasificación de series temporales se basa en que todas las series de un tipo, son creada con el mismo modelo y por tanto, todas las series que se adecuen a ese modelo formarán parte de ese grupo.

Algunos de los modelos utilizados para la realización de esos modelos son los modelos auto-regresivos o los modelos de Markov [33], [34] y [35] y [36] y [37]

Para tal fin, se define el modelo de Markov escondido como un conjunto de estados y un alfabeto de símbolos de salida. Cada uno de los estados se puede definir como un par de distribuciones: la distribución de transición sobre los estados y la distribución de emisión sobre los símbolos de salida. El matiz es que la serie pasa de un estado a otro según la distribución de emisión del estado actual definido.

Basadas en distancias

Estas técnicas son las más conocidas y actualizadas hasta la fecha. Se basan en aproximaciones tradicionales a los retos de las series temporales y han sido ampliamente utilizadas en la mayoría de los casos actuales. Destacan por su estabilidad, su sencillez de uso y su velocidad de cómputo en comparación con las otras propuestas que mencionaremos a continuación.

Como podemos ver en la siguiente figura 1.11 dentro de la clasificación de series temporales por distancias hay diferentes técnicas ya citadas en diversas investigaciones como [38] donde se hacen aproximaciones diferentes a la clasificación de series temporales.

El punto en común de todas ellas es que se basan en usar un método clásico de clasificación de datos donde la única diferencia se centra en la utilización de distintas definiciones de la distancia. En el resto de escenarios, la distancia euclídea entre diferentes valores de la misma o diferentes variables permitía cuantificar las similitudes entre los elementos para realizar la clasificación, tal y como exponemos

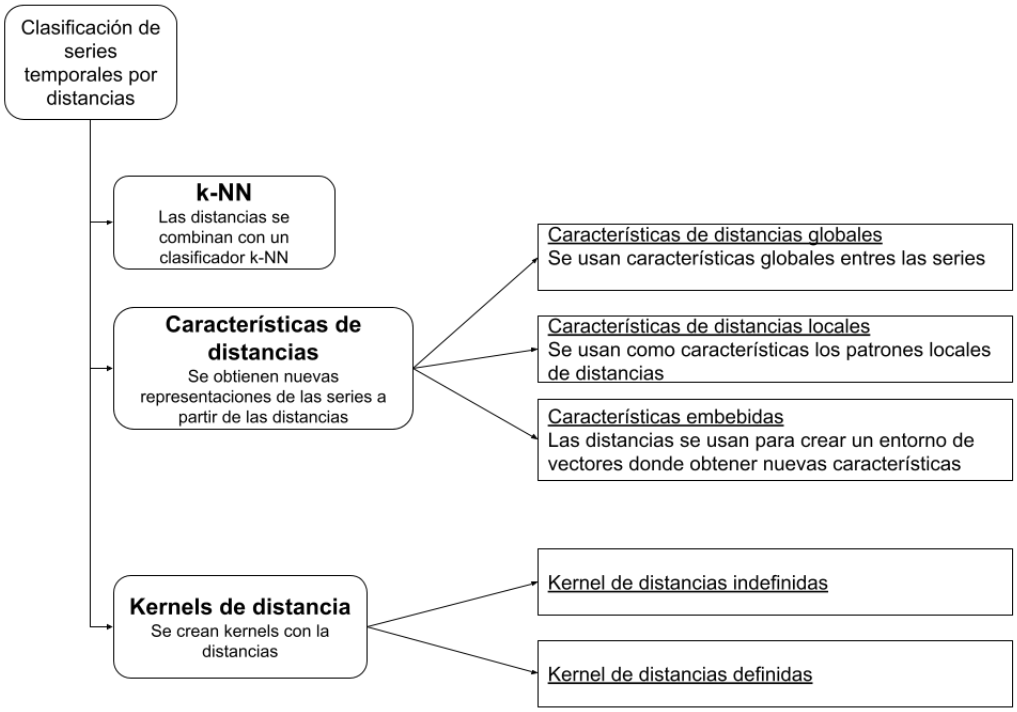


Figura 1.11: Taxonomía de series temporales

en la figura 1.10 . En el caso de las series temporales, se utilizarán otras formas de cuantificar las distancias entre los valores.

Como se presenta en la publicación [38] y reproducimos en la figura 1.11 se pueden definir tres grandes grupos dentro de las clasificaciones por distancia en series temporales.

A continuación entramos a definir cada uno de estos grupos, centrando nuestra mención en aquellos algoritmos que hemos utilizado en los casos de uso que vamos a describir más adelante.

k-NN

Durante las últimas décadas se han realizado muchos estudios comparando los diferentes algoritmos para la clasificación de series temporales como los siguientes estudios [39] y [40] y [41].

Entre los algoritmos de clasificaciones puede partir desde el 1-NN que ya se presentó hace unas décadas. Entre los muchos métodos que pueden utilizarse en la clasificación de series temporales, el clasificador de 1 vecino más cercano (1NN) ha resultado ser a menudo preciso en la práctica. El método de clasificación 1NN es muy sencillo de utilizar ya que no tiene parámetros de entrada y no requiere la selección y discretización de características para su clasificación. Además, como se manifestó anteriormente, se ha demostrado que la tasa de error del clasificador 1NN

es como máximo el doble de la probabilidad óptima de Bayes cuando se utiliza un conjunto de muestras infinito [42].

También se usa otro tipo de distancias. La primera de las distancias a analizar es la distancia Euclídea. Es la distancia más comúnmente utilizada en la mayoría de los usos para medir distancias [43]. La distancia Euclídea compara dos series temporales

$$\overrightarrow{x} = (x_1, ..., x_m)$$

y

$$\overrightarrow{y} = (y_1, ..., y_m)$$

de una longitud m de la forma siguiente:

$$ED(\overrightarrow{x}, \overrightarrow{y}) = \sqrt{\sum_{i=1}^{m} (x_i - y_i)^2}$$

1.5.3. Dinamic Time Wrapping

DTW[44] , [45], [46], [47] puede verse como una extensión de ED que ofrece una alineación local (no lineal).

Para ello, se construye una matriz m por m, con la distancia euclidiana entre dos puntos cualesquiera $\overrightarrow{x}$ y $\overrightarrow{y}$, es decir, cada elemento de la matriz tiene la distancia $d(x_i, y_j)$ entre los dos puntos x_i y y_j usando la siguiente fórmula: $d(x_i, y_j) = (x_i - y_j)^2$. Para modelar un alineamiento global entre los elementos de las secuencias $\overrightarrow{x}$ y $\overrightarrow{y}$, la idea es considerar una secuencia de pares de índices que cumplan ciertas restricciones.

Una ruta de deformación $W = w_1, w_2, ..., w_k$ es un conjunto contiguo de elementos de la matriz que define un mapeo entre $\overrightarrow{x}$ y $\overrightarrow{y}$ con varias limitaciones:

- **Condiciones límite:** $w_1 = (1, 1)$ and $w_K = (m, n)$. Esto requiere que la trayectoria de deformación comience y termine en celdas de esquina diagonalmente opuestas de la matriz.
- **Continuidad:** Dado $w_k = (a, b)$,entonces $w_{k-1} \leftarrow (a', b')$,where $a - a' \leq 1$ y $b - b' \leq 1$. Esto restringe los pasos permitidos en la ruta de deformación a las celdas adyacentes (incluyendo las celdas adyacentes en diagonal)
- **Monotonicidad:** Dado $w_k = (a, b)$,entonces $w_{k-1} \leftarrow (a', b')$,donde $a - a' \geq 0$ y $b - b' \geq 0$. Esto obliga a que los puntos de W estén espaciados monótonamente en el tiempo

Hay un número exponencial de trayectorias de alabeo que satisfacen las condiciones anteriores. Sin embargo, sólo nos interesa el camino que minimiza el coste de

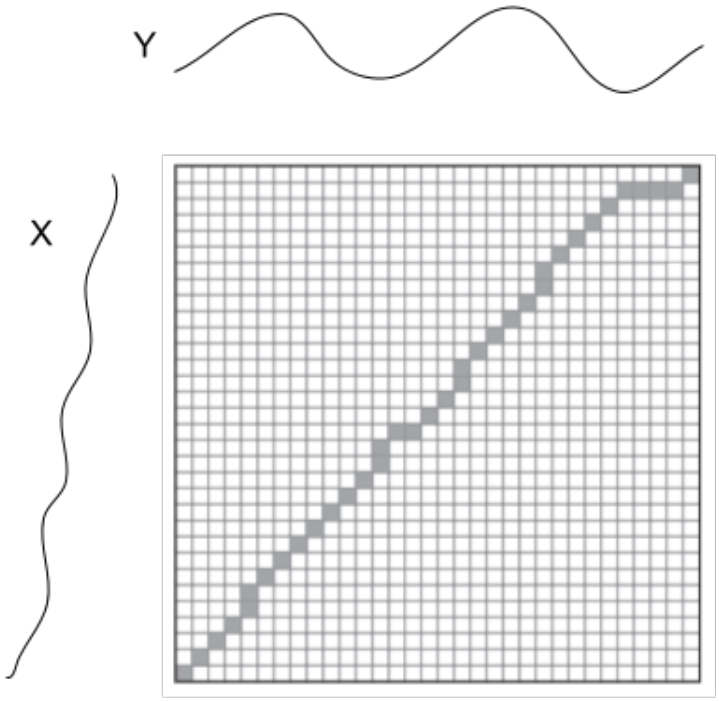


Figura 1.12: búsqueda de la trayectoria óptima con condición de contorno

deformación:

$$DTW(\overrightarrow{x}, \overrightarrow{y}) = min \sqrt{\sum_{i=1}^{k} w_i}$$

Para encontrar este camino hay que evaluar la siguiente Recurrencia, que define la distancia acumulada $\gamma(i, j)$ como la distancia $d(i, j)$ encontrado en la celda actual y el mínimo de las distancias acumuladas de los elementos adyacentes:

$$\gamma(i, j) = d(x_i, y_j) + \text{mín}\{(\gamma(i-1, j-1), \gamma(i-1, j), \gamma(i, j-1))\}.$$

Distancias basadas en formas

La idea clave del algoritmo presentado es encontrar segmentos coincidentes dentro de toda la serie temporal, denominados patrones, permitiendo el desplazamiento y el escalado en las dimensiones temporal y de amplitud. El problema de calcular el valor de la similitud entre las series temporales se transforma entonces en el de encontrar el conjunto más similar de patrones coincidentes. Existen varios algoritmos basados en la detección de estas formas, como SpADe, Una peculiaridad de SpADe es que requiere afinar una serie de parámetros, como el factor de escala temporal, el factor de escala de amplitud, la longitud del patrón, el tamaño del paso de deslizamiento, etc. Recientemente, se ha postulado un nuevo algoritmo basado en formas, denominado k-shape, que permite el cálculo de estas comparaciones de formas de forma rápida.

Nuestro objetivo, como se ha mencionado anteriormente, es agrupar las series temporales con la intención de aglutinar en un mismo grupo los tipos de comportamiento de las comunicaciones entre dos equipos similares aunque estén desplazados en el tiempo en cada uno de ellos. Hay que buscar los patrones de los atacantes y por tanto, la distancia seleccionada es la basada en las formas, ya que hay que comparar el perfil de cada uno de los ataques. Con esta solución no supervisada, el objetivo es que los comportamientos, independientemente del desplazamiento en el tiempo, se agrupen en un mismo cluster y nos proporcionen un valor fundamental.

Haciendo estos clusters en las series temporales, podemos finalmente clasificar cada una de las comunicaciones de un ordenador con cada una de las máquinas con las que se relaciona en diferentes grupos de forma que la combinación de sus flujos nos permita definir el comportamiento de la dirección IP.

Hemos seleccionado la nueva forma de clustering denominada k-shape [48] por su rapidez de cálculo ya que en entornos corporativos el número de flujos entre máquinas puede ser de millones, por lo que nos basamos en uno de los aspectos más relevantes de este tipo. Este algoritmo se basa en un procedimiento de refinamiento iterativo escalable, que crea clusters homogéneos y bien separados. Para ello, como medida de distancia, k-Shape utiliza una versión normalizada de la medida

de correlación cruzada para considerar las formas de las series temporales al compararlas. Basándose en las propiedades de esa medida de distancia, concretamente en las propiedades de la versión normalizada de la correlación cruzada, es posible desarrollar un método para calcular los centroides de los clusters, que se utilizan en cada iteración para actualizar la asignación de las series temporales a los clusters. Además ha sido evaluado por muchas investigaciones como [49], [50]

k-shape

El algoritmo k-shape realiza una agrupación no supervisada de las series temporales introducidas para el cálculo. Los principios en los que se basa son similares a los de K-means, donde se realiza una iteración de los clusters de forma repetitiva hasta que no se cambie ninguna serie temporal de grupo o cuando se alcance el número máximo de iteraciones.

Cada una de estas iteraciones intenta agrupar todas las series temporales en grupos. Termina cuando se alcanza el número máximo de iteraciones o cuando los agrupamientos han permanecido constantes en una iteración, como se describe en el Algoritmo 1.

El primer paso es encontrar un centroide para cada cluster. Para conseguirlo, se utiliza el algoritmo "shapeExtraction", donde nos da el centroide para cada k cluster dentro de un vector de 1 por m utilizando el Algoritmo 2.

Después de seleccionar un centroide para cada cluster k, cada serie temporal se asigna a un cluster, utilizando el Algoritmo 3.

Algorithm 1 [IDX,C] = k-Shape(X,k)

Entrada: X is an n-by-m matrix containing n time series of length m that are initially z-normalized. k is the number of clusters to produce.

Salida: IDX is an n-by-1 vector containing the assignment of n time series to k clusters (initialized randomly). C is a k-by-m matrix containing k centroids of length m (initialized as vectors with all zeros).

$iter \leftarrow 0$
$IDX^{'} \leftarrow []$
while $IDX \neq IDX^{'}$ **and** $iter \leq 100$ **do**
    $IDX^{'} \leftarrow IDX$
    // Refinement step
    for $j \leftarrow 1$ **to** k **do** //Create centroids
        $X^{'} \leftarrow []$
        for $i \leftarrow 1$ **to** n **do**
            if $IDX_{(i)} = j$ **then**
                $X^{'} \leftarrow [X^{'}; X_{(i)}]$
        $C_{(j)} \leftarrow ShapeExtraction(X^{'}, C_{(j)})$
    for $i \leftarrow 1$ **to** n **do** // Assign time series to a cluster
        $mindist \leftarrow \infty$
        for $j \leftarrow 1$ **to** k **do**
            $[dist, x_0] \leftarrow Shape - basedDistance(C_{(j)}, X_{(i)})$
            if $dist \leq mindist$ **then**
                $mindist \leftarrow dist$
                $IDX(i) \leftarrow j$
    $iter \leftarrow iter + 1$

Algorithm 2 $C^{'} = ShapeExtraction(X, C)$

Entrada: X is an n-by-m matrix containing n time series of length m that are initially z-normalized. k is the number of clusters to produce.

Salida: C0 is a 1-by-m vector with the centroid.

$X^{'} \leftarrow []$
for $i1$ **to** n **do**
    $dist, x^{'} \leftarrow Shape - basedDistance(C, X(i))$
    $X^{'} \leftarrow [X^{'}; x^{'}]$
$S \leftarrow X^{'T} \cdot X^{'}$
$Q \leftarrow I - \frac{1}{m} \cdot O$
$M \leftarrow Q^{T} \cdot S \cdot Q$
$C^{'} \leftarrow Eig(M, 1)$

Algorithm 3 $[dist, y^{'}] \leftarrow Shape - basedDistance(x, y)$

Entrada: Two z-normalized sequences x and y

Salida: Dissimilarity dist of x and y Aligned sequence y0 of y towards x

$length \leftarrow 2^{nextpower2(2 \cdot length(x)-1)}$

$CC = IFFT\{FFT(x, length) \cdot FFT(y, length)\}//Equation12$

$NCC_c = \frac{CC}{\|x\|\|y\|}$

$[value, index] \leftarrow max(NCCc)$

$dist \leftarrow 1 - value$

$shift \leftarrow index - length(x)$

if $shift \geq 0$ **then**

    $y^{'} \leftarrow [zeros(1, shift), y(1 : end - shift)]$

else

    $y^{'} \leftarrow [y(1 - shift : end), zeros(1, -shift)]$

1.6. Grafos

En este apartado utilizaremos las descripciones y definiciones mencionados en los apartados anteriores para presentar un grafo multiplex con características temporales.

Durante este apartado presentaremos la teoría básica de grafos, las características principales de los grafos temporales y de los grafos multicapa y cómo proponemos la unión de ambos para la creación de los grafos multiplex con características temporales. El concepto de grafo o red es el mismo, pero se suele denominar red o red compleja a los grafos que, como son los casos que se estudiarán, tienen muchos nodos y aristas.

En los posteriores capítulos planteamos su utilización para solventar casos de uso donde su empleo aporta un valor diferencial para la obtención de características temporales a los grafos.

Durante las últimas décadas la utilización de las representaciones basadas en grafos de las relaciones entre diferentes activos ha visto incrementarse debido a la necesidad de relacionar gran cantidad de elementos de una forma eficiente.

Desde que en 1736 el matemático suizo Leonhard Euler citase los grafos por primera vez, al intentar responder a la siguiente pregunta en la ciudad de Königsberg (actual Kaliningrado): "¿Es posible dar un paseo comenzando desde cualquiera de estas regiones, pasando por todos los puentes, recorriendo solo una vez cada uno y regresando al mismo punto de partida?"los usos de los grafos se han diversificado hasta pasar a ser en la actualidad uno de los campos más relevantes.

1.6.1. Material y Métodos

Los grafos son una buena herramienta para poder modelar y representar las interacciones en gran cantidad de entornos altamente complejos simplificando su representación [51], [52] y [53] y [54].

La evolución de los grafos ha sido un arduo y complejo camino desde las primeras representaciones de los grafos del siglo XVIII hasta nuestros días. Las primeras representaciones son simples imágenes donde se relacionaban entidades conocidas como nodos (o vértices) que se conectaban de forma directa. Estas primeras representaciones no hacían hincapié en las características de estas relaciones, sino que simplemente indicaban una relación entre ambos nodos. Cada uno de los nodos podía representar cualquier tipo de entidad o activo de forma estática.

No hay que menospreciar estas primeras definiciones de grafos, ya que son altamente utilizadas en gran cantidad de casos de uso como puede ser en la detección de comunidades dentro de ellas tal y como se aprecia en [55], [56] y [57].

Tras estas primeras definiciones se pasó a la descripción de una forma más compleja de las interacciones entre los nodos como lo son los grafos directos [58], [59], [60], los pesados [61], [62], temporales [8], [10] o bipartidos [63] en los que las aristas inician a dotarse de información adicional sobre las relaciones entre los nodos.

Como punto de partida podemos definir un grafo G como un conjunto de vértices o nodos N y un conjunto de aristas A, cada una de las cuales une un nodo con otro:

$$G = (N, A)$$

Dentro de los grafos existen diferentes tipos. Si las aristas tienen una dirección, el grafo se llama grafo dirigido u orientado. En otros ambientes también se conocen como digrafos como puede ser la librería que desarrolla soluciones de grafos en python [64]. Un grafo dirigido se representa por medio de un par ordenado que incluye el nodo origen y nodo origen de la arista definida.

Métricas de grafos

Orden de un grafo

El orden de un grafo es el número de nodos que existen dentro del grafo y se representa:

$$ord(g) = |N(G)| = N$$

La talla de un grafo es el número de aristas que contiene ese grafo y se representa tal que:

$$talla(G) = |A(G)|$$

Otra métrica también usada es el grado de un nodo. El grado de un nodo es el número de aristas incidentes en el nodo. Así, el grado del nodo "i" se denota por grad(i).

Grafos Multicapa

Nos centraremos durante esta tesis en la utilización de grafos multicapa para poder agrupar los atributos temporales de las aristas.

Lo primero es definir lo que se entiende como red multicapa. Una red multicapa es un par $\mathcal{M} = (\mathcal{G}, C)$ donde $\mathcal{G} = \{G_\alpha;\ \ \alpha \in \{1, \cdots, M\}\}$ es una familia de grafos (dirigidos o no, pesados o no) $G_\alpha = (X_\alpha, E_\alpha)$ (llamados capas de $\mathcal{M}$) y

$$C = \{E_{\alpha\beta} \subseteq X_\alpha \times X_\beta;\ \ \alpha, \beta \in \{1, \cdots, M\},\ \ \alpha \neq \beta\} \tag{1.1}$$

es el conjunto de interconexiones (aristas) entre nodos de diferentes capas G_α y G_β con $\alpha \neq \beta$. A los elementos de E_α se les denomina conexiones *intracapa* de $\mathcal{M}$ y a los elementos de cada $E_{\alpha\beta}$ ($\alpha \neq \beta$) se les denomina conexiones *intercapa* .

Tipos de grafos multicapas

Redes Multiplex

Un grafo multiplex $\mathcal{M}$, con m capas es un conjunto de capas $\{G_\alpha;\ \ \alpha \in \{1, \cdots, m\}\}$, donde cada capa es un grafo (dirigido o no dirigido, pesado o no pesado) $G_\alpha = (X_\alpha, E_\alpha)$, con $X_\alpha = \{x_1, \cdots, x_N\}$. Como todas las capas tienen los mismos nodos, una red multiplex se puede interpretar como una red multicapa en la que $X_1 = \cdots = X_M = X$ y $E_{\alpha\beta} = \{(x, x);\ \ x \in X\}$ para cualquier $1 \leq \alpha \neq \beta \leq M$.

Redes temporales

Los grafos temporales puede representarse con un grafo multicapa con un conjunto de capas $\{G_1, \cdots, G_T\}$ donde $G_t = G(t)$, $E_{\alpha\beta} = \emptyset$ si $\beta \neq \alpha + 1$, mientras

$$E_{\alpha,\alpha+1} = \{(x, x);\ \ x \in X_\alpha \cap X_{\alpha+1}\} \tag{1.2}$$

Hay otros tipos de redes muy relacionadas con las redes multicapa como son las redes interconectadas, las redes multidimensionales, las multinivel o los hipergrafos, pero estos son conceptos en los que no vamos a entrar a describir en esta tesis.

2. DEFINICIÓN DE GRAFO MULTIPLEX CON CARACTERÍSTICAS TEMPORALES

2.1. Introducción

En este segundo capítulo, se presenta la definición y construcción de un grafo multiplex con características temporales que en posteriores capítulos se utilizará para la resolución de problemas complejos de agregación de grandes cantidades de datos con evoluciones temporales de las relaciones entre los diferentes activos que interactúan.

Este tipo de grafos multiplex con características temporales pretende recolectar grandes cantidades de datos temporales dentro de un grafo que permita obtener características sobre los diferentes activos existentes y su interacciones con el resto.

Principalmente, los grafos, hasta la fecha, son una representación estática sobre las interacciones sobre diferentes activos. Podemos mencionar como ejemplo en entornos financieros el número de transacciones entre dos clientes, en entornos de transportes el número de viajes entre dos ciudades, ... pero siempre referenciado a un espacio de tiempo concreto y específico.

Hasta la fecha, las iniciativas propuestas de los grafos temporales se centraban en la recolección de las interacciones temporales entre los activos de una forma directa, es decir, cada una de las interacciones entre dos nodos se representaban con una arista o atributo que relacionaba esos dos nodos en un período de tiempo definido.

Este nuevo tipo de grafo denominado grafo multiplex de atributos temporales lo que pretende es recolectar atributos temporales de las interacciones entre los nodos de una forma compleja, en vez de representar sus interacciones de una en una. Para ello, se utilizarán análisis temporales de cada una de las interacciones de los nodos, visualizando todas esas interacciones dentro de un contexto entre ambos nodos. Para ello, se analizarán todas las interacciones entre cada uno de los nodos de forma conjunta.

Esta combinación de las interacciones, proporcionará unos atributos complejos sobre la evolución de las interacciones de una forma global, no asociado explícitamente a un tiempo t, sino a un rango de tiempo.

Debido a esta condición de temporalidad, el grafo multiplex con atributos temporales se definirá dentro de un período de tiempo y una frecuencia de muestreo exacta. Por tanto, para el mismo grupo de nodos que representan un grafo, pueden existir tantos grafo multiplex con atributos temporales como períodos de tiempo y frecuencias existan.

Esta capacidad de disponer de múltiples grafos para el análisis del comportamiento de los diferentes nodos a lo largo del tiempo, nos posibilita aumentar las capacidades de estudio. Nos permite poder hacer el análisis de las interacciones a corto, medio o largo plazo, en función del período de tiempo analizado.

También la determinación de una frecuencia de análisis nos permite poder analizar diferentes comportamientos temporales. El poder analizar la temporalidad con frecuencias muy altas (segundos), medias (horas) o bajas (días) nos permitirá la detección de diferentes patrones de comportamiento temporal entre los nodos que hasta la fecha no era posible identificar con otro tipo de grafos temporales.

2.2. Definición de grafo multiplex con características temporales

En esta sección se presentan los pasos para la creación de un grafo multiplex con características. En concreto, la evolución desde una representación básica con un grafo simple, hasta la creación de este grafo multiplex.

Para ello, suponemos un conjunto finito de interacciones A entre un conjunto de activos N, donde cada una de estas interacciones estará definida por los siguientes atributos:

- Activo origen, desde ahora lo definiremos como nodo origen
- Activo destino, desde ahora lo definiremos como nodo destino
- Tiempo donde se ha producido la interacción
- Valor que define la interacción entre ambos nodos en ese momento de tiempo.

A continuación, mostraremos la evolución de nuestra investigación desde las primeras representaciones de grafos hasta el objetivo final del grafo multiplex con características temporales. Cada uno de estos pasos en la evolución ha sido debido a una necesidad adicional de agrupar de una forma más precisa la información temporal existente. La evolución puede mostrarse en los siguientes pasos:

- grafo simple
- grafo orientado
- grafo pesado simple
- grafo pesado en función del tiempo
- grafo multiplex

Estas fases irán incrementando la complejidad de la solución. Sin embargo, obtendrán cada vez un nivel de atributos temporales superior, pudiendo crear finalmente atributos temporales complejos de las interacciones.

2.2.1. Grafo simple

La primera forma de poder incluir atributos temporales en un grafo se basa en la creación de aristas entre los nodos en los que exista algún tipo de interacción entre ellos.

Un grafo simple puede definirse como aquel que no presenta lazos en sus vértices ni más que una arista entre cualquier par de vértices como se puede apreciar en la figure 2.1.

Podemos definir el grafo como

$$G =< N, A >$$

donde N es la lista de nodos y A la lista de aristas que unen los nodos de N. Para que un grafo sea simple, se debe cumplir que:

$$\mid A_{n_1,n_2} \mid < 2$$

y si

$$n_1 = n_2, \mid A_{n_1,n_2} \mid = 0$$

La arista que une los nodos n_1 y n_2 existirá si existe algún tipo de interacción entre ambos nodos dentro del espacio temporal definido. De lo contrario, no existirá arista entre ambos nodos.

Esta primera representación de las interacciones sólo nos permite detectar la existencia de una relación entre dos nodos. No permite incluir ningún tipo de temporalidad entre ellos, por lo que sólo puede representar el estado de conexión entre los nodos en el período de tiempo estudiado, sin proporcionar atributos adicionales de la caracterización de las interacciones.

2.2.2. Grafo orientado

Una versión más compleja para la definición de las interacciones entre los nodos, puede ser realizada con los grafos orientados. Estos grafos aumentan la complejidad respecto a los grafos simples, permitiendo indicar la dirección de la interacción de los nodos indicando el origen y el destino como se aprecia en la figura 2.2.

Se definen los grafos orientados como un grafo

$$G = (N, A)$$

donde:

- N es un conjunto de pares ordenados nodos.
- A es un conjunto de combinación de relaciones de los nodos

$$A \subseteq (a, b) \in N \times N : a \neq b\}$$

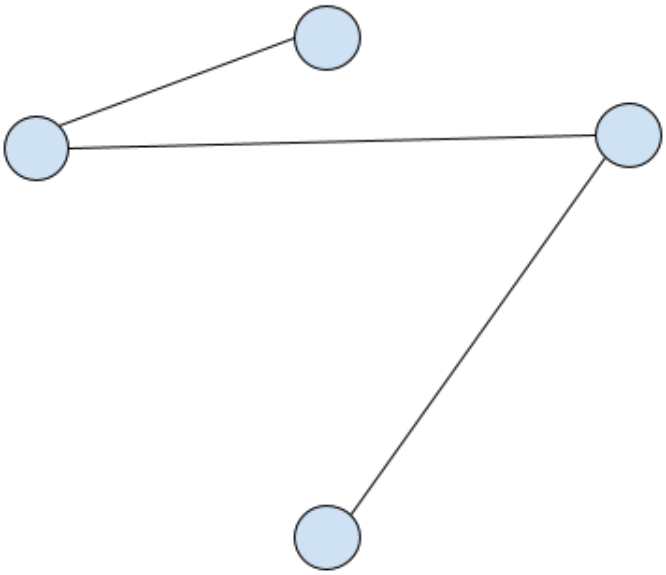


Figura 2.1: Grafo simple

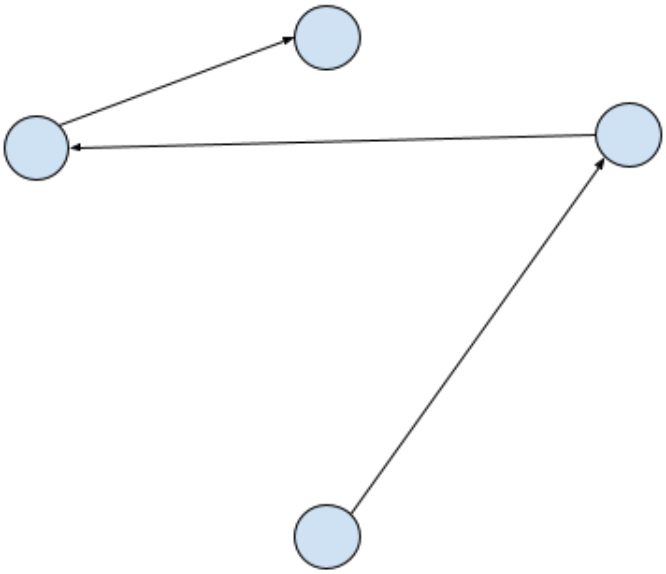


Figura 2.2: Grafo orientado

En este grafo, existirá una arista que une dos nodos n_1 y n_2 en una dirección si existe alguna interacción entre estos nodos en ese sentido dentro del período de tiempo estudiado.

El uso de este grafo para la representación de las interacciones entre los nodos aporta mejoras respecto al grafo simple. En este caso, nos permite indicar el sentido de la interacción entre ambos nodos. Es muy útil para representar flujos entre los nodos. Como ejemplo, puede representar los flujos migratorios entre ciudades o países, flujos financieros entre entidades o flujos de datos entre ordenadores.

Este valor adicional de indicar el flujo ya nos permite conocer un atributo más de la relación entre cada uno de los nodos.

2.2.3. Grafo pesado dirigido simple

Como siguiente evolución para la representación de la temporalidad en un grafo, nos encontramos con el grafo pesado donde el atributo W de cada arista no es un simple valor numérico como en el apartado anterior, sino que puede ser una función basada en el tiempo que muestre de una forma más precisa algún atributo temporal.

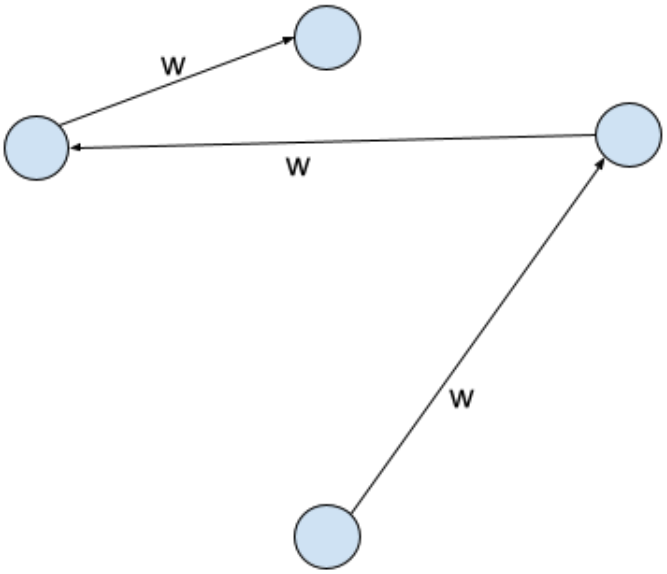


Figura 2.3: Grafo pesado

El grafo pesado puede definirse como un trío ordenado,

$$G = (N, A, W)$$

donde

$$N = \{n_1, \ldots, n_n\}$$

es su conjunto de nodos,

$$A = \{a_1, \ldots, a_m\}$$

es su conjunto de aristas, y

$$W = \{w_1, \ldots, w_m\}$$

es el conjunto de pesos asociados a cada arista.

Este peso asociado a cada arista puede ser un valor entero que puede definirse como el número de interacciones entre ambos nodos en el período de tiempo analizado. Como ejemplo, podemos indicar el número de vehículos a motor que circulan entre dos ciudades, el número de llamadas telefónicas que ha ocurrido entre dos terminales móviles o la suma total de importaciones realizadas entre dos países.

La capacidad de asociar un valor o atributo a cada una de las aristas nos abre un nuevo campo en el que podemos incluir valores, que pueden ser desde muy simples

hasta muy complejos, para representar la interacción entre dos nodos. Estos atributos pueden ser desde el número de llamadas telefónicas entre dos móviles hasta el beneficio esperado para la compañía telefónica de cada interacción.

2.2.4. Grafo pesado en función del tiempo

Como siguiente evolución en la inclusión de atributos temporales dentro del grafo podemos mencionar los grafos pesados donde el peso se define como una función de tiempo. De esta forma, cada una de las aristas será una función del tiempo y no un número entero, sino un valor dependiente del análisis que se va a realizar y será el resultado complejo de una interacción con el tiempo.

El grafo pesado puede definirse como un trío ordenado,

$$G = (N, A, W)$$

donde

$$N = \{n_1, \ldots, n_n\}$$

es su conjunto de nodos,

$$A = \{a_1, \ldots, a_m\}$$

es su conjunto de aristas, y

$$W = \{w_1, \ldots, w_m\}$$

es el conjunto de pesos asociados a cada arista. En este caso, no como en la anterior cada w_m se define como una función del tiempo.

$$w_m = f_{(t)}$$

Este nuevo grafo tendrá información compleja en cada arista en función del tiempo t que se utilice en ella.

Como puede apreciarse este valor dependerá del tiempo, obedeciendo a una función que puede considerarse discreta, por lo que el grafo obtendrá diferentes valores en sus aristas dependiendo del tiempo en el que se analice.

En este caso, al considerar el tiempo como una función discreta, hay dos valores para estimar la representación del grafo:

- período de tiempo en el que se analiza la función de tiempo. Diferentes rangos de tiempos como son períodos de segundos, días o años harán que las aristas de este grafo pesado tengan valores diferentes
- frecuencia de muestreo para el análisis de la muestra temporal. Para el mismo período de tiempo, la representación de la evolución temporal de la función como un muestreo de la señal temporal nos permite analizar el mismo período de tiempo con diferentes frecuencias. Es decir, podemos analizar un período de 5 años, con frecuencias de muestreo horaria, diaria, mensual,...

Existirán tantas representaciones del grafo pesado con funciones de tiempo como períodos de tiempo analizados y frecuencias de muestreo se realicen en la muestra.

Esta nueva representación de la temporalidad dentro de un grafo nos abre grandes posibilidades para poder simplificar en el peso de una arista funciones complejas que determinen la relación a lo largo del tiempo de los nodos. Esto es posible no sólo modificando el valor de la función que determina la relación entre los nodos, sino también estudiando el mismo grafo en diferentes períodos de tiempo o en distintas frecuencias de análisis.

2.2.5. Grafo multiplex con características temporales

En las aproximaciones anteriores hemos ido aumentando la complejidad en las relaciones entre los nodos, incrementando la complejidad de las aristas. Esta dilatación de la complejidad se ve reflejada en un aumento de la información existente dentro del grafo que describe la relación entre los diferentes nodos.

Tanto los grafos simples como los dirigidos y pesados, van incluyendo un valor más complejo sobre la relación entre los nodos a medida que se hace más compleja la relación. Esto nos permite incorporar cada vez un valor que proporciona una información más compleja para la descripción de la relación entre los dos nodos a lo largo de un período de tiempo.

Sin embargo, estas aproximaciones como el resto de propuestas de grafos temporales [65] se centran en la extracción de una relación entre ambos nodos de forma aislada.

En esta tesis proponemos la utilización de técnicas de series temporales para la obtención de atributos temporales de todos los nodos y su interacción global, no centrándonos sólo en la relación entre dos nodos de forma directa.

Para tal fin, nos hemos basado en las aproximaciones de teoría de las señales y algoritmia de series temporales. Estas aproximaciones nos permiten analizar de forma conjunta todas las relaciones entre todos los nodos y no aislarnos a la actividad entre dos nodos de forma independiente. Nuestra premisa es que el comportamiento entre dos nodos se ve afectado por el comportamiento de todos los nodos en su conjunto. Como ejemplo, no podemos indicar si el atributo de una arista es muy elevado sin compararlo al resto de aristas. Esta nueva aproximación nos permite interpretar las interacciones de los nodos de una forma completa y no independiente.

En el estudio de los grafos, todos los cálculos se realizan tras disponer de todos los datos dentro del grafo de una forma aislada. En nuestra aproximación, pretendemos realizar cálculos anteriores que comparen y compensen variaciones entre todas las aristas que representan las relaciones a lo largo del tiempo entre los nodos.

Para ello, nos basamos en definir cada una de las interacciones entre dos nodos como una serie temporal, concretamente como una secuencia finita y discreta de valores que representan la relación entre los nodos.

Así pues, suponiendo que tenemos N nodos y un conjunto de aristas A que unen direccionalmente dos nodos a y b.

$$A \subseteq \{(a, b) \in N \times N : a \neq b\}$$

en nuestro caso cada arista a_j se puede describir como una serie temporal discreta con los valores a lo largo del tiempo de la interacción entre ambos nodos (a y b).

Por tanto, la arista a_j podría definirse como un conjunto de valores cuya longitud se determina por tres valores:

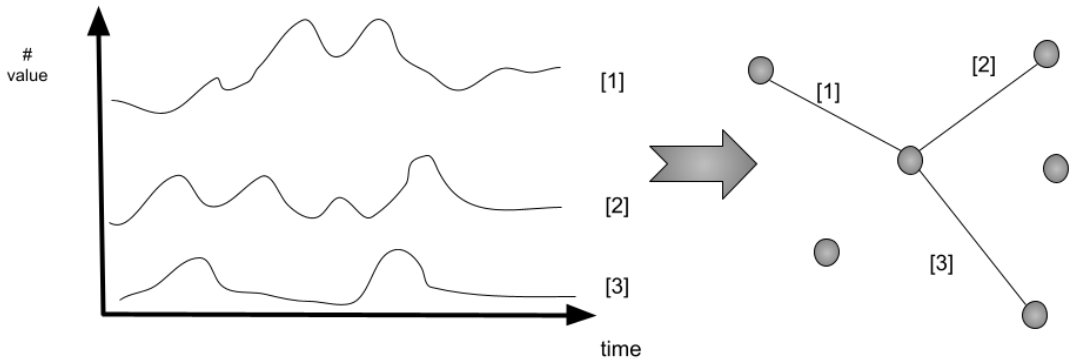


Figura 2.4: Combinación series temporales y grafos

- Tiempo origen: T_{ini}
- Tiempo final: t_{fin}
- Frecuencia de muestreo: $frec$

Lo que nos permite indicar que la longitud de la serie discreta que conforma la arista estará formada por el número de veces que la frecuencia ocurra entre el tiempo final y el tiempo inicial:

$$longitud(a_j) = \frac{T_{fin} - T_{ini}}{frec}$$

El valor de cada uno de estos pasos será un número entero que indique el valor de la interacción entre los dos nodos durante este período de tiempo. Puede ser número de viajes, número de transferencias bancarias, número de pasajeros, ... cualquier objeto medible entre los dos nodos o activos analizados.

Como puede apreciarse en el gráfico 2.4 se crearán tantas series temporales como aristas A. Estas series temporales se analizarán de forma conjunta para poder incluir dentro del grafo atributos temporales que no sólo indiquen la relación entre los dos nodos, sino que también tengan una relación sobre las interacciones del resto de nodos.

Como se ha mencionado en la introducción de esta tesis doctoral, la algoritmia de las series temporales se ha focalizado en varias de las acciones que pueden realizarse en ellas. En la gráfica siguiente mostramos algunas de las características más utilizadas dentro de las series temporales 2.5 donde se visualizan las predicciones del futuro y las anomalías de la serie. En resumen, podríamos referirnos a las más utilizadas que son :

- Predicción del futuro: desde hace muchos años la representación de mediciones reales representadas por medio de series temporales nos permite la predicción del comportamiento futuro de éstas. Como puede constatar en diferentes

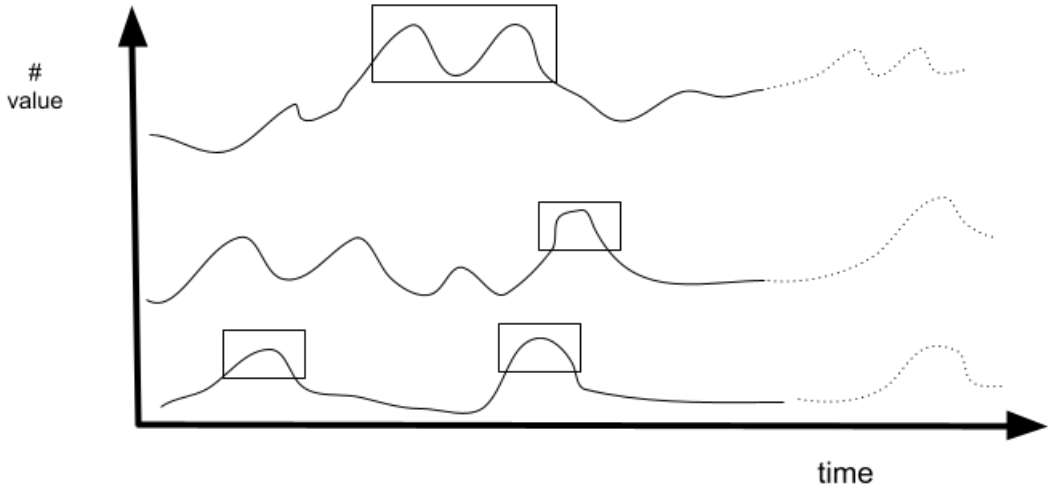


Figura 2.5: Predicción y anomalías en series temporales

investigaciones como [66], [67] y otros la evolución de las técnicas para la mejora de su predicción y reducción computacional.

- Detección de anomalías: disponer de un elevado número de series temporales o secuencias discretas de valores permite la detección de aquellas donde se producen comportamientos no esperados en su evolución y que nos posibilitan focalizar nuestra atención en esos puntos concretos. Para este tipo de análisis existen muchas técnicas y aproximaciones tal y como se menciona en [68], [69] y [70] que recoge un estudio de las técnicas más usadas en el mundo de la investigación.
- Agrupación de series temporales: existe una rama de la algoritmia de series temporales que se centra en la comparación de series temporales y su clasificación en diferentes grupos. Este acercamiento, como se ha presentado anteriormente, puede realizarse con diferentes aproximaciones y por tanto, su aplicación y efectividad va muy de la mano del reto al que se enfrente.

Como hemos comentado anteriormente, se realizará este preprocesamiento de las series temporales que caracterizan las aristas, antes de incluir esta información temporal en el grafo multiplex.

Tal y como indicábamos previamente, disponemos de A series temporales que representan como series dicretas cada una de la interacciones entre dos nodos dentro del rango de tiempo estudiado y con la frecuencia escogida.

Agrupación de series temporales

Todas estas series temporales, las vamos a analizar de forma conjunta para poder estudiar el comportamiento del conjunto de interacciones del grafo de forma unificada.

Vamos a realizar la agrupación de las series temporales en N grupos que posteriormente incluiremos dentro del grafo multiplex. Esta agregación se realizará con una técnica que busque la mayor homogeneidad en cada uno de los N grupos propuestos. Dentro de las múltiples aproximaciones que pueden utilizarse para esa agrupación de series temporales, nos hemos centrado en dos que destacan por su tiempo de cómputo y eficacia.

Agrupación de series temporales con K-Shape

La primera propuesta que realizamos en esta tesis doctoral es la utilización de técnicas de series temporales donde se utiliza su representación como serie discreta de números enteros. Esta primera aproximación es la más tradicional que puede tenerse al representar cada serie temporal como una serie discreta y a través de esta representación realizar la agrupación.

Para tal fin, se utilizará una metodología que compare dichas series discretas de números aglutinándolas en diferentes grupos. En esta primera aproximación se utiliza la técnica ya mencionada anteriormente y presentada por la investigación [71] como mejor método para la realización de series temporales basadas en distancias de tiempo.

Como hemos indicado en epígrafes precedentes, en el campo científico las técnicas basadas en distancias son ampliamente utilizadas. Como todas las demás, K-shape se basa en una aproximación basada en distancias como puede apreciarse en la descripción de este algoritmo realizado anteriormente 1, 3 y 2. Como anteriormente mencionamos esta aproximación se basa en la forma que dispone la serie temporal para crear la distancia entre dos series temporales y posteriormente ir realizando una interacción entre todas las series temporales para encontrar a qué grupo asociar cada una.

Tal y como se aprecia en la figura 2.6, en los grupos realizados en un caso de uso empleando la técnica Kshape se puede evaluar que las formas de cada uno de los grupos es consistente y por tanto, referencia para la selección del grupo de cada uno.

Agrupación de series temporales con grafos de visibilidad

Esta segunda aproximación para la agrupación que hemos seleccionado para la realización de los grafos multiplex con características temporales se basa en la utilización de otra representación de las series temporales. En la primera, como hemos propuesto, nos hemos basado en una representación de cada serie temporal como

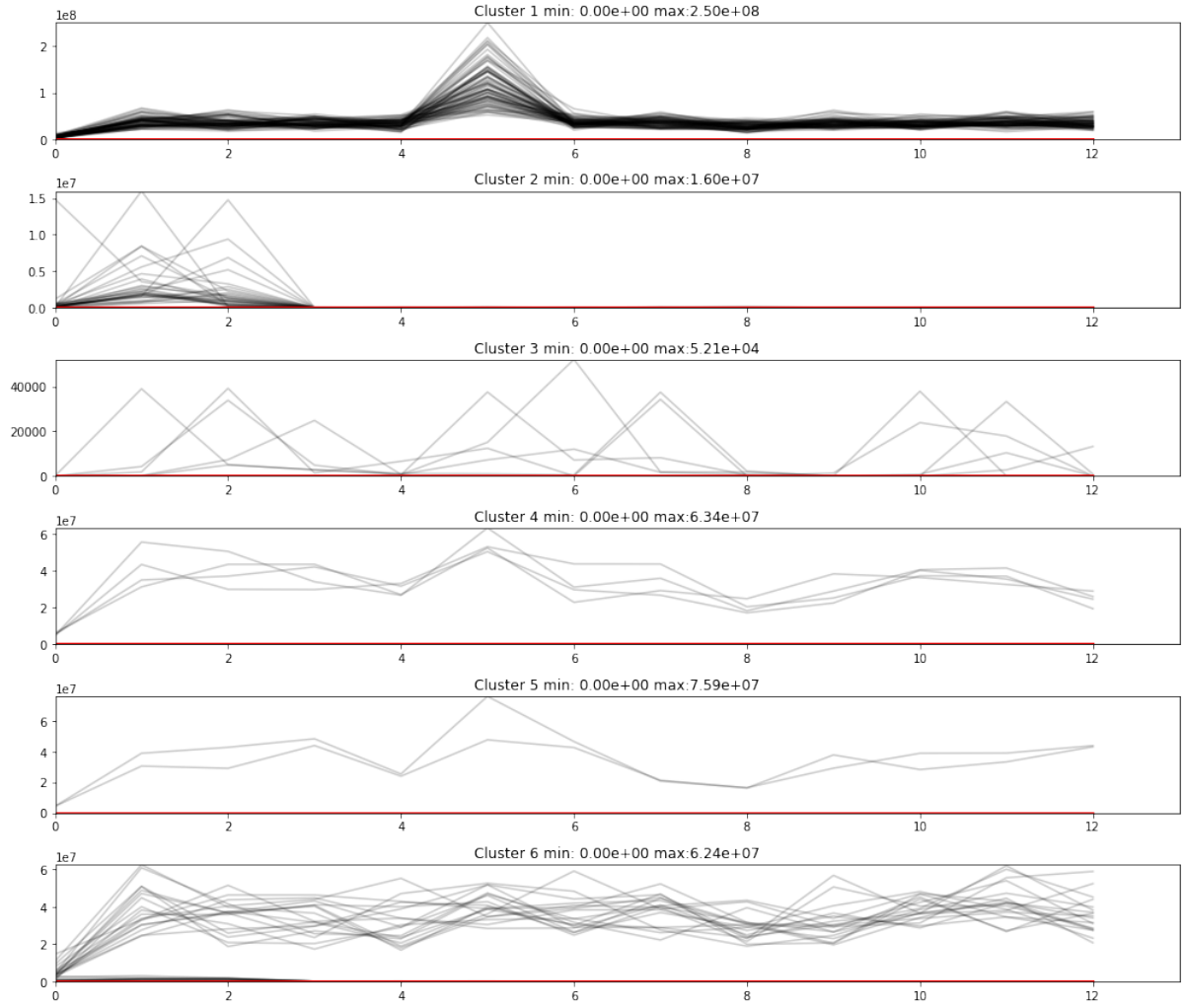


Figura 2.6: Clustering con K-shape

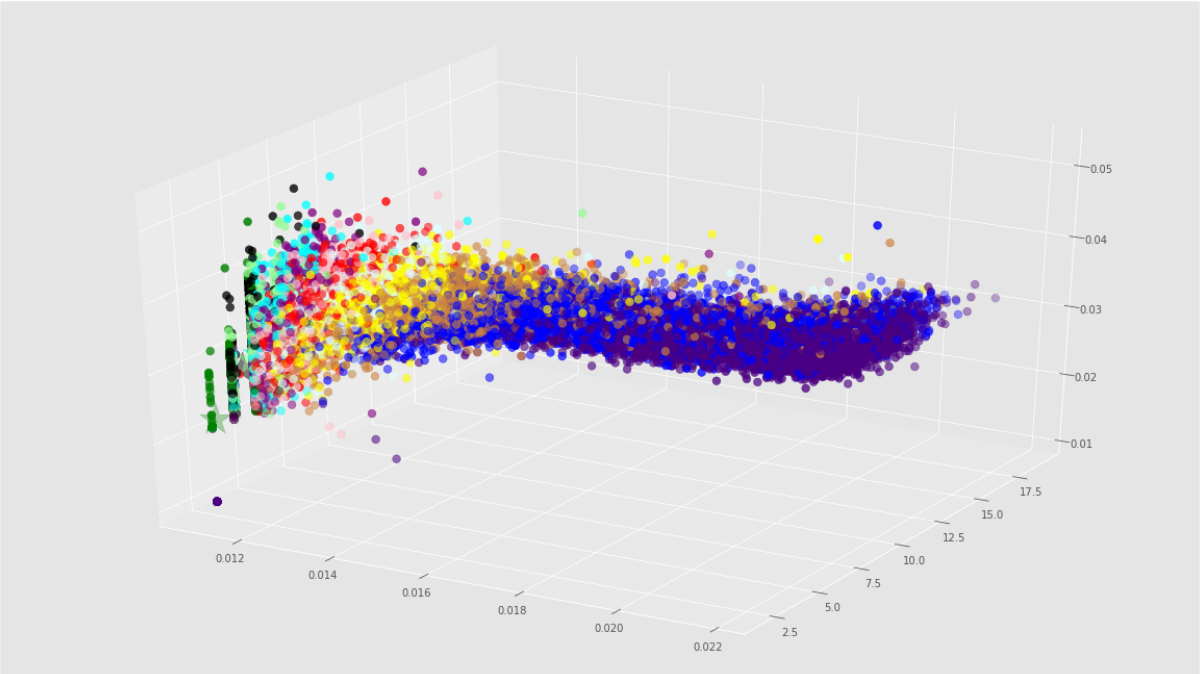


Figura 2.7: Clasificación de las series temporales usando grafos de visibilidad

una secuencia discreta de números enteros. En esta segunda aproximación, proponemos una aproximación diferente basada en la representación por medio de grafos de las series temporales.

Para esta aproximación, nos hemos basado, tal y como hemos presentado en el capítulo anterior en la utilización de la nueva técnica en la que una serie temporal se puede representar como un grafo, que se denomina grafo de visibilidad.

Estos grafos permiten, con una capacidad de computación muy eficiente, poder obtener características de los grafos que usando técnicas de series temporales sería mucho más complejo. Estos grafos de visibilidad se presentaron en la última década en la investigación [20].

Tal y como comentamos previamente y se apreciaba en la gráfica 1.5 los grafos de visibilidad permiten representar las series temporales por medio de un grafo donde las aristas posibilitan obtener datos sobre la serie temporal como puede ser las frecuencias, las tendencias, ...

A partir de esta primera exposición [20], en el año 2009 también se propuso el grafo de visibilidad horizontal [72] que aportaba complementariedad al grafo de visibilidad natural.

Tal y como se ha descrito en el capítulo anterior la agrupación de estas series temporales pasa por la creación del grafo de visibilidad natural y el horizontal para cada una de las series temporales obtenidas en las aristas.

De cada uno de estos dos grafos obtenemos dos atributos del grafo:

- grado máximo del grafo
- densidad del grafo

El grado máximo del grafo se basa en el cálculo del grado de cada nodo. El grado de cada nodo lo podemos definir como la cantidad de aristas que inciden en él.

La densidad del grafo es una propiedad de los grafos que determina la proporción de aristas que posee. Suponiendo que un grafo está definido como un par:

$$G = (N, A)$$

la densidad sería:

$$Densidad = \frac{2|A|}{N(N-1)}$$

Como extremos de un grafo en función de la densidad, se puede definir un grafo denso como aquel que tiene un elevado número de aristas, cercano al número máximo de aristas. Por el contrario, se puede definir como un grafo disperso aquel que tiene un número muy bajo de aristas

Finalmente, por cada arista (serie temporal) dispondremos de dos grafos:

- grafo de visibilidad natural
- grafo de visibilidad horizontal

Y por cada uno de ellos, las dos propiedades anteriormente explicadas:

- densidad
- grado máximo del grafo

Por lo que cada arista(serie temporal) dispondremos de 4 atributos que describen de una forma compleja la evolución temporal entre los dos nodos que relacionan la arista.

Al disponer de cuatro atributos por arista, podemos usar una técnica para la creación de grupos partiendo desde estos cuatro atributos. Usaremos un algoritmo de clasificación para poder agrupar las aristas que tengan atributos similares.

Para la clasificación, proponemos el uso de un algoritmo de KMeans, ampliamente usado, para poder realizar la clasificación de las series temporales.

En resumen, ambas técnicas de clasificación de las aristas nos permiten asociar cada arista a un cluster que describe los atributos del grupo. Este centroide nos permitirá describir cada uno de los grupos con unas características temporales bien definidas.

Grafo multiplex

Ahora vamos a presentar el método para combinar el análisis de series temporales y las redes multiplex que permitirá mejorar las capacidades de ambas técnicas en la representación de entornos altamente complejos que varían en su comportamiento a lo largo del tiempo. Todos estos atributos nos permitirán tener una representación más real de las interrelaciones que se producen en el mundo real, utilizando los nuevos enfoques que nos proporciona la teoría y el análisis de redes complejas relacionadas con las redes multiplex, espaciales y temporales [65], [73]-[75].

Consideraremos una red multiplexada ponderada y dirigida [76] M, con n capas

$$N = \ell_\alpha; \;\; \alpha \; en\{1, ..., n\}$$

Consideraremos n capas, tantas como agrupaciones se hayan realizado en la fase de preparación de las series temporales descritas anteriormente. Se define el grafo multiplex con un conjunto de nodos $X = 1, ..., N$, donde cada capa es un grafo dirigido ponderado $\ell_\alpha = (X_\alpha, E_\alpha)$ sobre un conjunto de nodos $X_\alpha N \subset X$ y con un conjunto de aristas:

$$E_\alpha = \{e^\alpha_{i,j}; \alpha \in \{1, ..., n\}\},$$

donde $e^\alpha_{i,j}$ representa el enlace que conecta los nodos i y j en ℓ_α, y w_α es una función $w_\alpha : E_\alpha \longrightarrow [0, +\infty)$ tal que para cada arista $e^{alpha}_{i,j} enE_\alpha$, el coeficiente $w_\alpha(e^\alpha_{i,j})$ se llama *weight* de $e^\alpha_{i,j}$. Como en [74], suponemos que cada nodo está siempre conectado a sí mismo cuando se refleja en otra capa con un enlace bidireccional. Por comodidad esta red multiplexada $\mathcal{M}$ se denotará como un triplete $\mathcal{M} = (X, E, L)$ donde X, y L son como los anteriores, y para cada $\alpha \in \{1, ..., n\}$

$$E_\alpha \subset E \equiv \left(\bigcup_{\beta=1}^{n} E_\beta \right),$$

y donde la capa a la que pertenece el nodo o la arista considerada debe entenderse por el contexto, por lo que en la secuela denotaremos también las aristas, indistintamente, por (i, j) o iæ cuando sea conveniente.

Obsérvese que no es necesario que los nodos de X pertenezcan a todas las capas, mientras que cada nodo seguirá estando conectado a sí mismo cuando se refleje en otra capa.

A continuación, explicaremos cómo procesar la información y utilizar estas herramientas para nuestro propósito con tres casos de uso donde se utiliza este grafo multiplex con características temporales.

3. CREACIÓN DE IDS CON CARACTERÍSTICAS TEMPORALES

3.1. Resumen

En este capitulo pretendemos ofrecer un nuevo enfoque para reducir el número de alertas enviadas a un analista del Centro de Operaciones de Seguridad (SOC) analizando los flujos de red en los Sistemas de Detección de Intrusiones de Red y centrándonos más en el comportamiento de los flujos que en la red única que dispara la alerta gracias al uso de los grafos multiplex con características temporales que han sido expuestos en el anterior capítulo. Este nuevo enfoque se centra en el análisis de las direcciones IP de la red a lo largo del tiempo y, por lo tanto, la generación de alertas basadas en el comportamiento de las direcciones IP en ese periodo de tiempo específico. Esta similitud de los flujos de red se realiza mediante algoritmos de agrupación de series temporales basados en la forma de la comunicación de red entre los ordenadores. Los principales beneficios de este nuevo enfoque son reducir el número de alertas que se generan para su estudio por parte de los analistas del SOC, así como el procesamiento informático necesario al analizar todos los eventos de una dirección IP en franjas temporales, en lugar de cada uno de forma independiente, proporcionando una puntuación de que una dirección IP es un atacante en un periodo de tiempo. En base a estos beneficios, es una solución adecuada para las necesidades de las grandes empresas.

El análisis lo realizaremos comparando los dos tipos de clasificación de series temporales mencionadas anteriormente (k-shape y grafos de visibilidad) analizando las ventajas y desventajas de ambos.

IoT Endpoint Market by Segment, 2018-2020, Worldwide (Installed Base, Billions of Units)

Segment	2018	2019	2020
Utilities	0.98	1.17	1.37
Government	0.40	0.53	0.70
Building Automation	0.23	0.31	0.44
Physical Security	0.83	0.95	1.09
Manufacturing & Natural Resources	0.33	0.40	0.49
Automotive	0.27	0.36	0.47
Healthcare Providers	0.21	0.28	0.36
Retail & Wholesale Trade	0.29	0.36	0.44
Information	0.37	0.37	0.37
Transportation	0.06	0.07	0.08
Total	**3.96**	**4.81**	**5.81**

Source: Gartner (August 2019)

Figura 3.1: Evolución del mercado de endpoints IoT 2018-2020

3.2. Introducción

Hoy en día la cantidad de información generada en todo el mundo entre los dispositivos conectados crece exponencialmente. Esto nos lleva a un campo en el que la disponibilidad de la información en la forma adecuada aporta un valor diferencial para la detección de las evidencias adecuadas dentro de una cantidad creciente de datos. IoT es una de las disciplinas más importantes donde se está produciendo este incremento. Gartner, empresa líder mundial en investigación y asesoramiento, predice una gran cantidad de puntos finales de IoT interconectados. La ciberseguridad debe responder a los requisitos derivados de este aumento de nuevos dispositivos. El modelado de redes proporciona un marco conceptual para describir las relaciones entre sistemas y medirlas de forma significativa [74], [75], [77]-[79]. Las dos últimas décadas han visto nacer una rama de la ciencia conocida como teoría de redes complejas, que tiene como uno de sus objetivos explotar la disponibilidad actual de big data para extraer una representación de los sistemas y mecanismos complejos subyacentes [80]-[87]. La teoría de las redes complejas trata de analizar los sistemas complejos para encontrar una nueva forma de analizarlos [88]-[90]. En el modelo, se utiliza una red compleja para representar el sistema, donde los nodos son los diferentes componentes del sistema, mientras que las aristas representan los enlaces entre ellos. Así, el atacante navega a través de la red compleja para extraer y capturar información valiosa contenida en el sistema, aunque cada salto de un nodo a otro tiene su propio coste. En cualquier caso, esta miríada de puntos finales del IoT (**5,81 miles de millones de unidades de nuevos dispositivos IoT**) requiere procesos eficaces de ciberseguridad.

La ciberseguridad es una de las disciplinas que más ha crecido en los últimos años, tratando de cubrir las diversas necesidades surgidas en los nuevos entornos digitales y el IoT de las grandes corporaciones. La ciberdelincuencia cuesta a las empresas de Estados Unidos más de 3.500 millones de dólares al año en delitos y daños relacionados con Internet, según un informe del FBI de 2019. El creciente número de ordenadores y puntos finales conectados a Internet hace que sea un campo de juego más complejo para detectar a los atacantes. Estas pérdidas muestran la necesidad real de mejorar las herramientas de detección, protección y reacción en las grandes corporaciones, donde las soluciones actuales no resuelven sus necesidades reales de seguridad. El gasto en ciberseguridad alcanzó los 123.000 millones de dólares en 2020, y se prevé que la gestión de riesgos de la ciberseguridad crezca un 2,4 %, frente a la tasa de crecimiento prevista del 8,7 % a principios de este año, según las previsiones de gasto en seguridad y gestión de riesgos de Gartner.

En esta situación, uno de los retos más importantes es detectar cuanto antes cualquier brecha de seguridad en la red. Uno de los indicadores más utilizados para medir el tiempo que tardan los analistas del SOC en detectar a un atacante en la red es el llamado MTTD (Mean Time to Detect). Sin embargo, la evolución del MTTD está aumentando en los últimos años ya que Cost of a Data Breach Report 2020 describe. La tendencia actual es que el MTTD está aumentando hasta los 207 días en los últimos años, como podemos ver en la figura 3.2

Para reducir este indicador, se despliega un gran número de dispositivos de seguridad en las redes. Uno de ellos es el NIDS. Un sistema de detección de intrusiones en la red puede describirse como un dispositivo o una aplicación de software que supervisa una red o unos sistemas en busca de actividades maliciosas o violaciones de las políticas se seguridad. Los NIDS inspeccionan cada paquete de red y carga útil en función de un gran número de reglas. Estas reglas suelen encontrar alguna bandera o información de la carga útil, similar a un antivirus, detectando algún patrón de ataque como podemos ver en los ejemplos descritos en el Listado 1. Cada regla se realiza inspeccionando cada paquete generado en un ataque y encontrando algún comportamiento característico, por lo que debe ser desarrollada por expertos analistas de seguridad ubicados en laboratorios de seguridad 24x7. Una vez desarrolladas estas reglas, se envían a las grandes corporaciones mediante servicios de suscripción actualizando el conjunto de reglas periódicamente para detectar los nuevos patrones de ataque.

Listing 3.1: snort NIDS rule examples

```
alert tcp $EXTERNAL_NET any -> $TELNET_SERVERS 23 ( msg:"MALWARE-BACKDOOR w00w00 attempt";
flow:to_server,established; content:"w00w00"; metadata:ruleset community;
classtype:attempted-admin; sid:209; rev:9; )
alert tcp $EXTERNAL_NET any -> $TELNET_SERVERS 23 ( msg:"MALWARE-BACKDOOR attempt";
flow:to_server,established; content:"backdoor",nocase; metadata:ruleset community;
classtype:attempted-admin; sid:210; rev:7; )
alert tcp $EXTERNAL_NET any -> $TELNET_SERVERS 23 ( msg:"MALWARE-BACKDOOR MISC r00t attempt";
flow:to_server,established; content:"r00t"; metadata:ruleset community;
classtype:attempted-admin; sid:211; rev:7; )
alert tcp $EXTERNAL_NET any -> $TELNET_SERVERS 23 ( msg:"MALWARE-BACKDOOR MISC rewt attempt";
flow:to_server,established; content:"rewt"; metadata:ruleset community;
classtype:attempted-admin; sid:212; rev:7; )
```

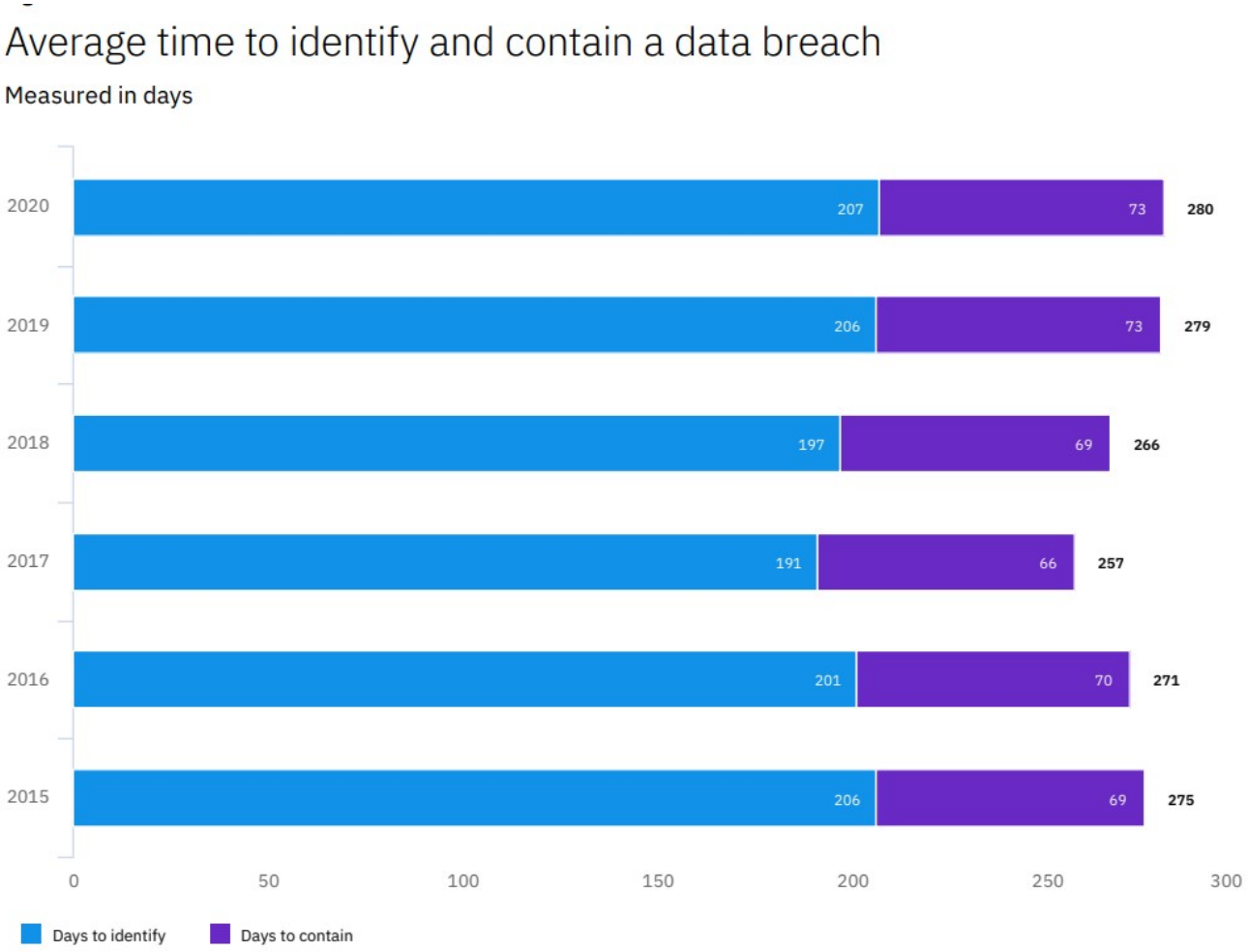


Figura 3.2: Tiempo medio de detección Promedio por año

Esta generación de conjuntos de reglas tiene tres desventajas principales:

- no hay detección de ataques de día cero
- muchas alertas de falsos positivos porque la regla sólo encuentra patrones limitados en los paquetes
- no hay información contextual sobre la alerta generada

Debido a que cualquier alerta generada debe ser analizada para detectar a los atacantes en las grandes corporaciones, cualquier actividad de intrusión o violación es normalmente reportada a un administrador o recolectada centralmente usando otra herramienta de ciberseguridad llamada Security Information and Event Management (SIEM). El SIEM intenta reducir una de las desventajas de los NIDS y otras herramientas de ciberseguridad, la información contextual centrándose en correlacionar la información existente dentro de las alertas generadas, dando al analista del SOC más información sobre el entorno en el que se disparó la alerta. El SIEM combina los resultados de múltiples fuentes y utiliza técnicas de filtrado de alarmas para distinguir la actividad maliciosa de las falsas alarmas. Estos sistemas de detección temprana deben enviar el menor número posible de alarmas al equipo de ciberseguridad del SOC. Una gran cantidad de falsos positivos puede reducir la eficacia de todo un departamento de Ciberseguridad al centrarse en investigaciones no críticas. Por lo tanto, un enfoque actual de inspeccionar cada paquete de red centrándose sólo en la carga útil del paquete y la información de la sesión de red no es suficiente para reducir la métrica MTTD.

En este trabajo, nuestro reto es reducir el número de días antes de la detección, centrándonos en el comportamiento del dispositivo conectado a la red en lugar de evaluar cada evento de red por separado, ya que no nos ayuda a detectar en etapas tempranas los ataques.

Por lo tanto, presentamos un enfoque que nos proporciona las ventajas de ambas técnicas (series temporales y redes) generando un gráfico contextualizado con la información extraída del comportamiento de las comunicaciones detallado mediante técnicas de (User Entity Behaviour Analytics) UEBA.

3.3. Material y Métodos

3.3.1. Sistemas de detección de intrusos basados en el aprendizaje automático

Hoy en día, el reto más importante de los sistemas de detección de intrusos es aumentar su precisión en redes más complejas, con más ordenadores y más comunicaciones. A medida que las técnicas de ataque se vuelven más sofisticadas y más extendidas en el tiempo, el tiempo medio para detectar al atacante aumenta debido a la dificultad de filtrar las alarmas falsas positivas de las alertas reales generadas por la actividad del atacante.

En la actualidad, el tipo más común de NIDS se basa en las técnicas de conjunto de reglas y violación de protocolos. También existen nuevos enfoques basados en técnicas de aprendizaje automático para identificar los eventos de los atacantes. Estos nuevos enfoques tratan de resolver dos de las desventajas de los NIDS: la detección de ataques de día cero y la reducción de falsos positivos en las grandes empresas, tratando de dar al analista del SOC alertas más precisas para detectar en etapas tempranas cualquier ataque.

Hay muchas formas de analizar los paquetes de red dentro de los sistemas de detección de intrusos basados en el aprendizaje automático, por ejemplo, nuevas técnicas como el aprendizaje por refuerzo o las redes convolucionales-LSTM. En general, cualquier enfoque supervisado o no supervisado es válido para aumentar la detección NIDS basada en soluciones de aprendizaje automático. Sin embargo, todos estos enfoques se centran en los mismos retos, es decir, aumentar la precisión del modelo.

3.3.2. Trabajos relacionados

Básicamente, existen diferentes enfoques para encontrar el mejor algoritmo de detección o combinación de algoritmos [91] y [92]. La principal clasificación de las técnicas puede reducirse a:

- Clasificación de patrones
- Clasificadores simples: Vecino más cercano a K, máquinas de vectores de apoyo, redes neuronales artificiales,...
- Clasificadores híbridos
- Conjunto de clasificadores

Algunos de estos enfoques se realizan con series temporales, como [93]-[95], así como con redes [96] y [97]. Sin embargo, ninguno de estos enfoques proporcionó una solución única a este problema.

Algunas investigaciones se centran en obtener más información sobre el comportamiento de los ordenadores con varias técnicas de ingeniería de características. Por ejemplo, utilizando la media, la mediana y otras métricas que comparan el flujo actual con los flujos anteriores del ordenador para detectar comportamientos anómalos.

En nuestro caso, proponemos una técnica más profunda para adquirir información sobre el comportamiento.

3.4. Aproximación con grafo multiplex y k-shape con comportamiento temporal

Nuestro novedoso enfoque de IDS propuesto trata de centrarse en todas las desventajas actuales del IDS dando menos alertas al analista del SOC, basándose en un algoritmo de aprendizaje automático para detectar ataques de día cero y la generación de alertas basadas en el comportamiento del ordenador, no sólo en el último paquete de red como presentamos en un trabajo anterior [98].

El análisis del comportamiento de las comunicaciones de red del ordenador puede proporcionar información sobre la relación pasada entre dos direcciones IP específicas. En este trabajo, describimos un nuevo enfoque para simplificar una técnica de detección que se centra en el comportamiento de una dirección IP en lugar de analizar todos y cada uno de los eventos.

Nuestro enfoque trata de analizar los comportamientos de las direcciones IP durante un periodo de tiempo para encontrar ataques a la red. Utilizamos técnicas de redes multiplexadas y series temporales para crear un nuevo contexto avanzado para cada dirección IP y, por tanto, predecir el comportamiento de la dirección IP que revelaría uno o más ataques a la vez. De este modo, los operadores de los SOC reciben menos alertas y más precisas.

Este enfoque cambia la perspectiva de la predicción basada en eventos a la predicción basada en el comportamiento de las direcciones IP durante un periodo de tiempo específico.

Para lograr este objetivo, creamos una red multiplexada con atributos dependientes del tiempo, lo que nos da la oportunidad de entender la relación entre las direcciones IP.

En esta sección describimos la metodología empleada para crear un nuevo algoritmo de aprendizaje automático supervisado para predecir cuál es la dirección IP del atacante, basado en el conjunto de datos reconocido UNSW-NB15[99] utilizado por otros investigadores para crear otros enfoques de aprendizaje automático. Para ello, utilizaremos una red para extraer la información necesaria para detectar las direcciones atacantes y luego analizaremos estas características con un modelo Random Forest para detectar las direcciones IP. Esta predicción se basará en un rango de tiempo: si analizamos la red cada hora, este algoritmo IDS nos da una predicción de cada dirección IP cada hora, reduciendo la cantidad de información proporcionada a los operadores del SOC.

3.4.1. Data set

El primer reto de la mayoría de los casos reales es que los datos están dispuestos en archivos de registro que almacenan la información de forma secuencial. Estos

archivos tienen poca información sobre cada evento, por lo que su agregación en una red nos permitirá obtener una mayor cantidad de información.

Como tenemos que crear series temporales con las interacciones entre cada dirección IP, es necesario que haya suficientes eventos relevantes para cada dirección IP para poder utilizar esta técnica.

Una de las mayores dificultades es encontrar un conjunto de datos que nos permita comparar nuestras ideas con las publicadas en otros trabajos. Para ello, hemos decidido utilizar un conjunto de datos bien conocido en el campo de los sistemas de detección de intrusos. En nuestro caso, vamos a utilizar el conjunto de datos UNSW-NB15 [99] porque cumple los requisitos que necesitamos para nuestro enfoque:

- Formato del conjunto de datos basado en los flujos de la red y sus características
- Observación a largo plazo: para conocer el comportamiento de cada comunicación de la red necesitamos información a lo largo del tiempo
- Conjunto de datos etiquetados
- Ampliamente utilizado en investigaciones anteriores para ser comparado. Se han publicado más de 30 investigaciones sobre soluciones NIDS de aprendizaje automático basadas en este conjunto de datos.

Este conjunto de datos cumple todos nuestros requisitos:

1. El Data set contiene 2.540.047 eventos en el fichero recolectando todos los flujos de la red
2. etiqueta cuáles de estas comunicaciones son ataques
3. nos proporciona 49 características que lo describen de cada flujo de red entre dos direcciones IP como podemos ver en la figura 3.11

Por otro lado, este conjunto de datos es bien conocido para evaluar IDS de aprendizaje automático, ya que varios trabajos han validado su enfoque con los mismos datos, como **otro**, [100] y **otro**, como referencias principales. Nuestro principal objetivo es mantener la misma tasa de detección media con menos esfuerzo computacional. Como línea de base, nos centramos en este estado del arte sobre los enfoques de IDS de aprendizaje automático con este conjunto de datos [101] donde se recogen más de 40 artículos que describen diferentes enfoques para implementar técnicas de aprendizaje automático en entornos de IDS. Basado en esta colección, la precisión media en la mayoría de los casos es de alrededor del 80 %-95 % de precisión. Por lo tanto, nuestro enfoque debe tener la misma precisión analizando el mismo conjunto de datos.

srcip	sport
dstip	dsport
proto	state
dur	sbytes
dbytes	dttl
sloss	dloss
service	Sload
Dload	Spkts
Dpkts	swin
dwin	stcpb
dtcpb	smeansz
dmeansz	trans_depth
res_bdy_len	Sjit
Djit	Stime
Ltime	Sintpkt
Dintpkt	tcprtt
synack	ackdat
is_sm_ips_ports	ct_state_ttl
ct_flw_http_mthd	is_ftp_login
ct_ftp_cmd	ct_srv_src
ct_srv_dst	ct_src_dport_ltm
ct_dst_sport_ltm	attack_cat
Label	

Tabla 3.1: Lista de características analizadas

3.4.2. Arquitectura IDS

Proponemos un nuevo método para la generación de alertas. Los NIDS actuales comprueban cada paquete de red con un conjunto de reglas o un algoritmo de aprendizaje automático, mientras que nuestro método recopila el tráfico de red en un periodo de tiempo. Después de recoger toda esta información, se activan las alertas con las posibles direcciones IP de los atacantes en el período de tiempo. En la figura 3.3, mostramos la arquitectura básica para recoger el tráfico de red y a partir de él activar las alertas.

Con este enfoque tomamos más información sobre el comportamiento de los ordenadores en lugar de comprobar cada paquete de red a la vez.

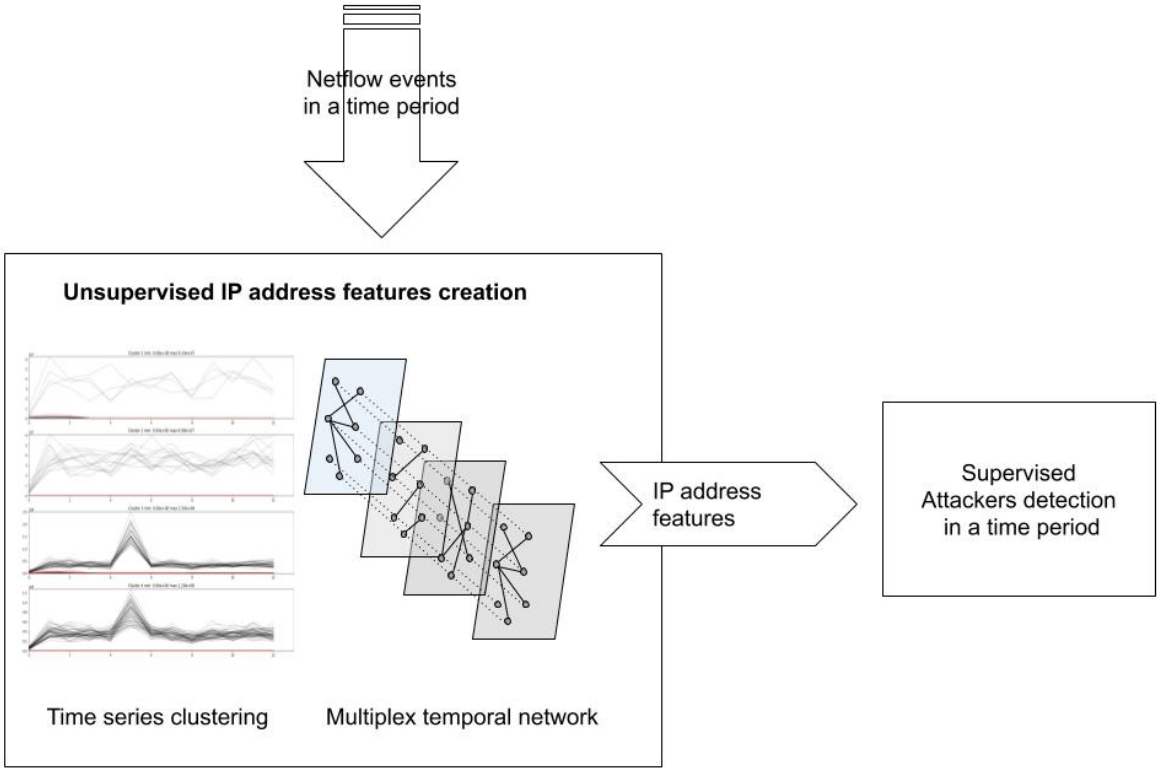


Figura 3.3: Arquitectura IDS propuesta

3.4.3. Grafo multiplex basado en series temporales para la detección NIDS con k-shape

Para centrarnos en el comportamiento de la comunicación de red entre direcciones IP, utilizamos una combinación de dos técnicas conocidas seleccionando la mejor de cada una. En primer lugar, intentamos agrupar las comunicaciones de red con el mismo comportamiento; en las grandes empresas hay varias actividades conocidas que dan el mismo patrón en los flujos de red, como la navegación por Internet, el acceso a la base de datos, el acceso a las carpetas compartidas, etc. Representar la comunicación de red entre dos direcciones IP como una serie temporal nos da la oportunidad de agrupar todo el tráfico de la red en varios servicios.

Sin embargo, la agrupación de todos los flujos de red sólo nos da información sobre uno de los flujos de red del ordenador y no sobre cómo están interactuando entre ellos. Tenemos que observar el comportamiento temporal de los flujos e intentar descubrir cómo están interactuando los nodos. Obviamente, un ordenador suele tener flujos de red con muchos otros ordenadores durante su actividad normal. Para recoger toda esta información utilizaremos gráficos de red en los que cada nodo representa una dirección IP de ordenador o dispositivo IoT y cada arista representa un flujo de red entre dos direcciones IP.

Con esta representación, cada ordenador o dispositivo IoT (nodo) tiene flujos de red (aristas) del grafo de red. Todos estos atributos permitirán tener una representación más veraz de las interrelaciones que tienen lugar en el mundo real.

En esta investigación, definimos la relación entre dos nodos a lo largo del tiempo como el número de bytes transferidos a la IP de destino. Agregamos todo el tráfico incluyendo todo tipo de servicios. En futuros trabajos, será posible analizar otras relaciones entre los nodos, es decir, diferentes aristas según el tipo de servicio (http, dns, telnet,...) .

Los campos necesarios para completar nuestra investigación son:

- Dirección IP de origen: srcip.
- Dirección IP de destino: dstip.
- Timestamp: Stime. Seleccionamos la hora de inicio porque estamos evaluando una dimensión temporal. En futuras investigaciones podemos seleccionar otra dimensión temporal.
- dbytes: número de byte transmitido de la dirección de destino.
- Label: Booleano que describe si el evento es un ataque.

A partir de este conjunto de datos, procesamos la información para obtener una red con las siguientes definiciones:

- Definiremos como nodos de la red todas las direcciones IP que existen en el conjunto de datos. En nuestro caso, hay 49 nodos.

- Crearemos una arista entre dos nodos, si hay algún evento que relacione ambas direcciones IP. En el caso de nuestro conjunto de datos, hay 311 aristas que tienen información.

Con este proceso hemos creado 311 marcos de datos con las relaciones entre dos direcciones IP que describen el comportamiento entre ambos nodos.

Así, hemos pasado de centrar nuestra atención en los eventos a centrarla en la dirección IP.

En este caso, el conjunto de datos seleccionado se define como un archivo donde se realiza el registro secuencial de la información en eventos consecutivos. Cada uno de los eventos de este registro debe tener unos requisitos mínimos para su uso:

- Detección de los nodos entre los que se realiza la comunicación como origen y destino.
- Todos los eventos deben tener la información del momento de su origen.
- Valor cuantitativo de la relación entre los dos nodos en ese momento.
- Para los estudios en la red de casos supervisados, también debe tener la clase del evento.

La creación de la red se hará a través de la librería de python networkx [64] , ya que es una de las librerías más utilizadas en python para el análisis de redes complejas. Los nodos serán cualquiera de los orígenes y destinos de los eventos procesados en el registro. Se creará una arista entre dos nodos con al menos un evento que los relacione. En el caso de la existencia de varios eventos, todos ellos se almacenan como un atributo en la arista, almacenando para cada evento el momento de su ocurrencia así como su valor y, en el caso de las clasificaciones, también su clase.

Cada arista tiene tantos eventos como flujos entre los nodos con la siguiente información sobre cada uno:

- timestamp: en este caso, la hora de inicio del flujo de red.
- value: dbytes de este evento.
- label: etiqueta del evento.

Al final de este proceso, tenemos una red compleja en la que cada nodo es una dirección IP y hay tantas aristas como comunicaciones entre las direcciones IP. Lo primero que podemos validar es la existencia de conectividad de los nodos, con nodos muy conectados en comparación con la mayoría.

Análisis dinámico de la red

Un estudio de la red en el tiempo nos proporcionará una visión de las relaciones entre los nodos con una visión basada en el tiempo. Para ello, el estudio debe realizarse en un determinado rango de tiempo y con una determinada frecuencia de muestreo.

Esta posibilidad nos permite analizar las relaciones en diferentes momentos creando un atributo temporal en la red según el inicio y el final de la muestra. Por otro lado, y siguiendo con las capacidades temporales, nos da la posibilidad de analizar la iteración de los nodos según una unidad temporal, como segundos, horas, días o años, de forma que podemos realizar estudios más acotados analizando las relaciones a largo, medio o corto plazo. Estas capacidades también nos permiten analizar un mismo espacio de tiempo en varias frecuencias de muestreo, lo que nos posibilita centrarnos en el comportamiento en el espacio de tiempo corto, medio y largo, que puede superponerse para aumentar la información extraída de los eventos serializados.

El resto de nuestro análisis se basa en la relación de los nodos en un periodo de tiempo y frecuencia de muestreo definidos para ello.

- Start time: Marca de tiempo a partir de la cual se crea la serie temporal.
- Finish time: Marca de tiempo a partir de la cual se termina la serie temporal.
- Sampling frequency: franja de tiempo para analizar la red.

Creación de series temporales

Los eventos que relacionan dos nodos serán los eventos que conforman las interacciones en el tiempo. Para poder comparar todas las interacciones entre nodos, tendremos que normalizar todas las interacciones por lo que crearemos una serie temporal con los eventos.

Teniendo en cuenta la fecha inicial, la fecha final y la frecuencia de muestreo, haremos que todas las series temporales tengan la misma longitud para una homogeneización del estudio a partir de ahora. Todas las series representarán el comportamiento en las mismas franjas horarias.

Cada franja coincidirá con la frecuencia de muestreo, si no hay eventos en esa franja tendrá un valor de cero y si hay varios eventos en ella se sumarán todos los eventos.

En nuestro caso, los eventos existentes en el conjunto de datos son muestras recogidas desde el 22 de enero de 2015 hasta el 18 de febrero de 2015. Tras realizar diferentes pruebas de muestreo, optamos por un muestreo horario que nos proporciona el detalle necesario en este caso. Esto nos proporciona 651 franjas horarias para comparar el comportamiento entre cada par de nodos.

Estas series temporales nos permiten realizar un estudio multivariante de las relaciones entre los nodos utilizando aproximaciones matemáticas orientadas a este fin.

Dentro del análisis de las series temporales podemos utilizar dos técnicas que podrán proporcionar un alto conocimiento de los patrones de interacción entre los nodos que son la creación de clusters y la detección de puntos anómalos dentro de las series. Ambas nos permiten obtener atributos sobre el comportamiento temporal aportando información adicional a la red. En este trabajo nos centramos en el uso de técnicas de clustering en un entorno de series temporales porque nos ofrecen una mayor tasa de precisión en la búsqueda de IPs de atacantes.

3.4.4. Clasificación de series temporales: k-shape

Uno de los principios que queremos demostrar en este trabajo es que el comportamiento de los atacantes es similar a lo largo del tiempo. La mejor manera de conseguirlo es agrupar todas las comunicaciones existentes en la red. En esta agrupación, intentamos acercar todos los comportamientos similares. Como ejemplo, podemos esperar que todos los empleados del departamento financiero de una empresa tengan los mismos patrones y por lo tanto acaben dentro del mismo cluster no supervisado, es decir, formas similares de navegar por Internet, accesos similares a los servidores corporativos, etc., sin embargo un atacante dentro del departamento financiero no coincide con el mismo comportamiento y por lo tanto será emparejado con ‘otras‘ direcciones IP con comportamientos anormales similares.

Cada serie temporal representa el comportamiento de la relación entre dos nodos de la red. El comportamiento se obtiene analizando el número de bytes enviados entre ambos ordenadores o dirección IP. Estos bytes nos proporcionan similitudes entre las series temporales y nos permiten encontrar comportamientos similares entre los nodos.

Para la agrupación de las series temporales se utiliza una técnica de medición de la distancia entre los puntos. Como en otros métodos de clustering, las entidades se agrupan en función de su proximidad a cada uno de los centros de atracción de cada uno de los clusters. El reto en las series temporales es que las distancias tradicionalmente utilizadas en estas técnicas, como la distancia euclidiana, no permiten una solución computacionalmente rápida al haber un gran número de puntos que comparar entre sí.

Uno de los enfoques más recientes para la agrupación de series temporales es el denominado k-shape [48]. Esta técnica permite la realización de un clustering homogéneo que permite una segmentación de las series temporales de forma rápida y sencilla. Este enfoque se basa en una técnica similar a k-means, pero su medida de distancia se basa en las formas.

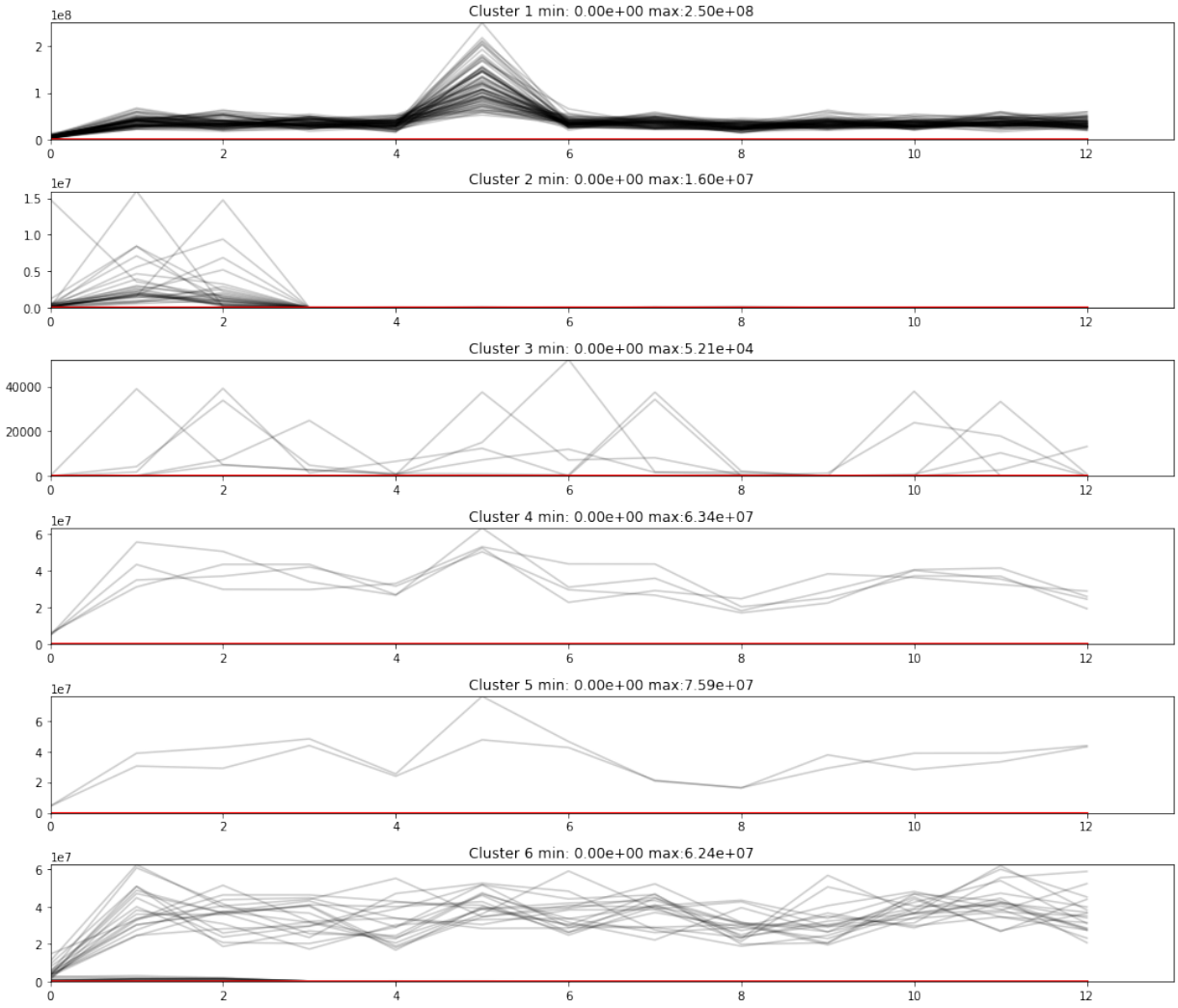


Figura 3.4: Agrupación de series temporales para 6 clusters

En este caso creamos un clustering con las 311 series temporales creadas con la comunicación entre los ordenadores. Con la técnica no supervisada k-means creamos 6 clusters clasificando cada serie temporal en uno de los clusters.

En nuestro caso, los clusters se distribuyen como podemos ver en la Tabla 3.2

Cluster	Número de aristas
0	106
1	14
2	37
3	102
4	38
5	11

Tabla 3.2: Distribución de las series temporales en N clusters

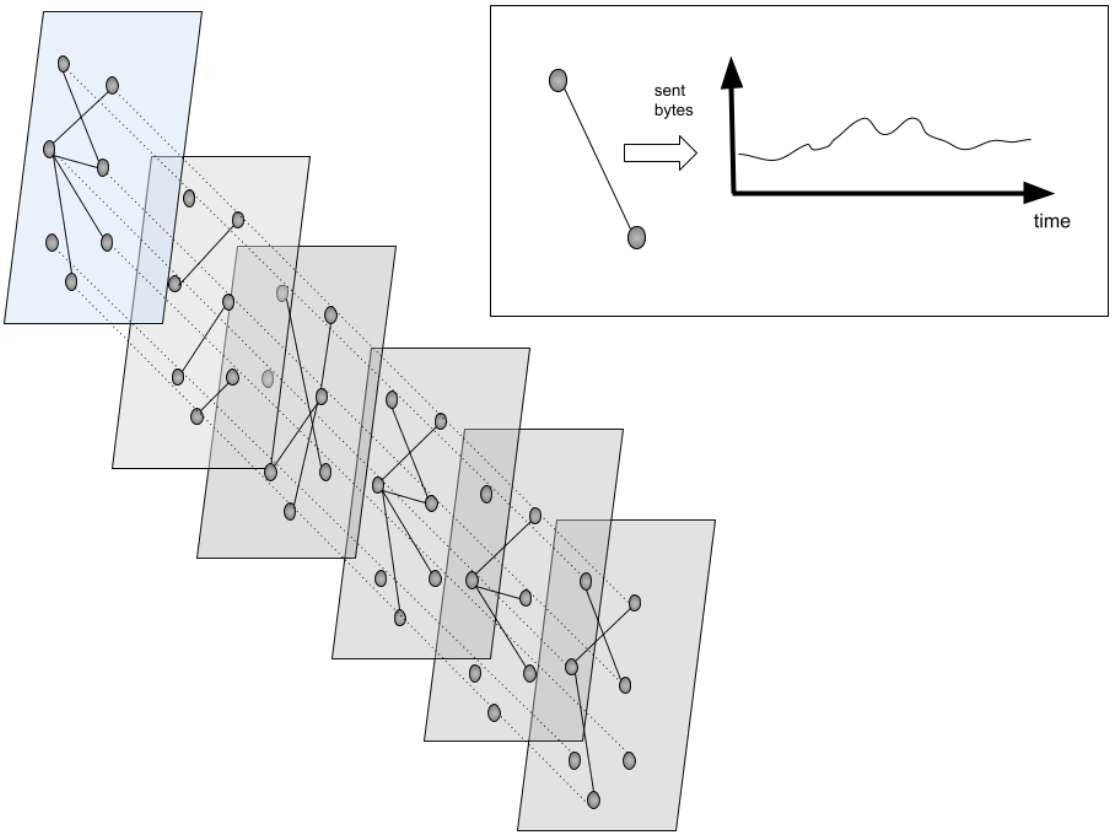


Figura 3.5: Ejemplo de red multiplexada con 4 capas

Atributos temporales incluidos en el grafo multiplex

En el apartado anterior, cada serie temporal se ha incluido en un cluster, etiquetando cada arista como parte de uno de los clusters. Este número de cluster es el nuevo atributo que incluiremos en la red, es decir, si la serie temporal está en el cluster número 3, incluimos un nuevo atributo de arista con el número de cluster: 3.

Ahora, creamos en el grafo tantas capas como clusters se hayan realizado en la serie temporal e incluimos cada arista en la capa correspondiente según el cluster en el que se haya asignado previamente. Finalmente, cada arista con el atributo cluster con valor 3, se ubicará en la capa 3 del grafo.

Como se puede ver en la figura 3.14, todos los nodos existirán en todas las capas pero las aristas sólo existirán en la capa definida en el clustering de la serie temporal.

Este nuevo atributo nos permite tener más información en la red, con esta nueva vista podemos analizar la relación de los nodos con las capas para detectar varios tipos de nodos. En un enfoque más complejo, si la relación entre dos nodos se puede describir con varias series temporales, es decir, bytes de origen y destino, la red multiplexada tendrá tantas dimensiones como tipos de relación.

A continuación, mostramos el número de nodos y aristas existentes en cada capa Tabla 4.1.

Cluster	Número de aristas	Número de nodos
0	106	46
1	14	17
2	37	20
3	102	35
4	18	20
5	11	17

Tabla 3.3: Distribución de nodos y aristas en capas

Como podemos ver, el número de aristas y nodos puede variar indicando diferentes comportamientos en cada capa. Por ejemplo, el clúster 1 describe nodos con más iteracciones que otros clústeres como el 2. En las tablas 3.13, 3.14, 3.15, 3.16, 3.17 y 3.18, mostramos los 5 principales nodos conectados en cada capa dándonos información compleja sobre la conectividad de cada nodo en cada capa.

Nodo	aristas conectadas
175.45.176.3	12
175.45.176.1	11
175.45.176.0	10
59.166.0.7	9
149.171.126.9	7

Tabla 3.4: Los 5 nodos con más aristas en la capa del cluster 0

Nodo	aristas conectadas
149.171.126.2	3
59.166.0.7	3
149.171.126.3	3
59.166.0.6	3
59.166.0.5	2

Tabla 3.5: Los 5 nodos con más aristas en la capa del cluster 1

Las capas permiten dotar a los nodos de información adicional a la previamente existente.

Cada una de ellas representa el comportamiento de la red de equipos analizados de forma diferente, por lo que aportará información adicional en el estudio.

Nodo	aristas conectadas
149.171.126.0	6
149.171.126.3	5
149.171.126.1	5
59.166.0.3	5
59.166.0.2	5

Tabla 3.6: Los 5 nodos con más aristas en la capa del cluster 2

Nodo	aristas conectadas
175.45.176.3	10
175.45.176.2	10
175.45.176.1	10
175.45.176.0	10
149.171.126.5	8

Tabla 3.7: Los 5 nodos con más aristas en la capa del cluster 3

Nodo	aristas conectadas
149.171.126.8	4
149.171.126.3	3
59.166.0.8	3
59.166.0.4	3
59.166.0.0	2

Tabla 3.8: Los 5 nodos con más aristas en la capa del cluster 4

Nodo	aristas conectadas
59.166.0.3	4
149.171.126.3	2
149.171.126.7	2
149.171.126.0	1
59.166.0.5	1

Tabla 3.9: Los 5 nodos con más aristas en la capa del cluster 5

Adquisición de características del nodo

Como hemos descrito anteriormente, el clúster en el que se encuentra cada arista se añade como un atributo a la red creada.

El siguiente paso es obtener la información necesaria de la red multiplexada para crear un sistema de detección de intrusos con aprendizaje automático. En este caso,

como necesitamos predecir si una dirección IP (nodo) es un atacante, obtenemos como métrica el número de nodos vecinos en cada capa.

Para cada una de las redes multiplexadas, obtendremos el número de aristas a las que está conectado el nodo en cada una de las capas de la red. En este ejemplo, en el caso de la red con seis segmentos, cada nodo tendrá el número de aristas a las que está conectado.

3.4.5. Detección de atacantes

En esta sección, describiremos cómo desarrollamos un algoritmo de aprendizaje automático supervisado utilizando las características de los nodos generadas anteriormente para clasificar cada nodo en una dirección IP atacante o no atacante.

Métricas de evaluación

Como mencionamos antes, una de las cosas más importantes para validar este nuevo enfoque es comparar nuestros resultados con los realizados anteriormente con el mismo conjunto de datos. Por lo tanto, centramos nuestras métricas de evaluación en las mismas métricas que los trabajos anteriores. En base a esto, la métrica seleccionada es la precisión (25 de los 29 enfoques analizados la tienen como métrica principal de evaluación).

A continuación, nuestro enfoque propuesto se evalúa en el conjunto de datos en términos de la precisión que se calcula como sigue: **Exactitud** identifica el número total de observaciones correctamente identificadas con respecto al número total de observaciones.

$$Precision = \frac{TP + TN}{TP + TN + FP + FN}.$$

Un verdadero positivo (TP) es un resultado en el que el modelo predice correctamente la clase positiva. Del mismo modo, un verdadero negativo (TN) es un resultado en el que el modelo predice correctamente la clase negativa.

Un falso positivo (FP) es un resultado en el que el modelo predice incorrectamente la clase positiva. Y un falso negativo (FN) es un resultado en el que el modelo predice incorrectamente la clase negativa.

Detección de direcciones IP de forma supervisada

Como paso final, creamos un algoritmo Random Forest para predecir si una dirección IP es un atacante. Para ello, utilizamos las características de los nodos creadas anteriormente para resolver un sencillo problema de clasificación. Tenemos varias

características para cada dirección IP, y sabemos cuáles son las direcciones IP atacantes.

Una de las ventajas de este enfoque es que las características extraídas de los gráficos nos permiten transformar los 2,5 millones de eventos en una predicción analítica en la que las características analizadas son los atributos de los gráficos temporales, proporcionando información sobre las relaciones entre las entidades analizadas en un ámbito temporal. El beneficio final es que cambiamos el reto de predecir 2,5 millones de eventos por un reto más sencillo de predecir un atacante en 49 direcciones IP.

A continuación, describimos los pasos para crear un algoritmo Random Forest y obtener la precisión del algoritmo de clasificación.

El primer paso es crear un entorno de validación cruzada en el que se pueda comprobar el resultado con varios experimentos, para confirmar nuestro estudio.

El segundo paso consistirá en dividir el conjunto de datos en dos partes: entrenamiento y prueba. Esta división nos ofrece dos conjuntos de datos: uno de entrenamiento con el 60 % del conjunto de datos original y otro con el 40 % de las filas originales para probar la precisión del algoritmo de clasificación.

Una vez hecho esto, utilizamos un modelo Random Forest con 100 estimadores para hacer un modelo predictivo.

Este modelo predice la detección de direcciones IP de atacantes con una red multiplexada de 4 capas con una precisión superior al 70 %.

Búsqueda óptima de capas en la red de multiplexación temporal

La red multiplex nos da información sobre una clusterización no supervisada. En este trabajo queremos validar que esta red multiplex puede darnos información compleja y útil. Intentamos validar que cuantos más clusters de series temporales obtenemos más datos logramos para obtener información sobre las IPs atacantes en el mundo real. Aumentaremos el número de clusters y trataremos de ver si a medida que se crean más clusters se obtiene más información.

Como el número de capas de la red multiplexada es el elemento más importante, queremos saber si al aumentar el número de capas obtenemos resultados más precisos. En este ejercicio lo vamos a hacer seis veces, con diferentes números de cluster:

- 2
- 4
- 6
- 8

- 10
- 12

que nos permiten clasificar las series temporales en diferentes segmentos proporcionando en cada una de las iteraciones información sobre el comportamiento de un nodo.

Número de clusters	Precisión
6	0.91
8	0.92
10	0.94
12	0.98

Tabla 3.10: Número de clusters vs precisión

A lo largo de este trabajo, hemos realizado el ejercicio con diferente número de clusters y podemos comprobar que el incremento de clusters aumenta la información existente en la red, y por tanto la exactitud del modelo de predicción para predecir los atacantes, como se puede ver en el gráfico (Tabla **??**).

3.4.6. Comparación de enfoques anteriores

Como se ha mencionado anteriormente, la razón principal para utilizar el conjunto de datos UNSW-NB15 es comparar con investigaciones anteriores para validar si este nuevo enfoque puede mantener la precisión reduciendo la complejidad del enfoque.

Utilizando el trabajo anterior de [101] podemos obtener una gran máquina de aprendizaje IDS. En esta investigación, se recogen 29 artículos que utilizan algún enfoque de aprendizaje automático para desarrollar un IDS basado en este conjunto de datos.

Después de analizar los artículos, decidimos emplear como métrica de evaluación la más utilizada en los enfoques anteriores. Esta métrica es la precisión (25 de los 29 enfoques la tienen como métrica principal de evaluación).

En la Figura 3.6 recogemos la precisión de cada enfoque y la combinamos con los resultados de nuestro enfoque.

Sólo el 24 % de los enfoques anteriores tienen mejor precisión que nuestro enfoque actual, y todos ellos tienen un mayor coste computacional para lograrlo.

3.5. Aproximación con grafo multiplex y grafos de visibilidad

Esta segunda parte del capítulo se centrará en la utilización de un segundo Dataset para validar el funcionamiento de los grafos multiplex con características tempora-

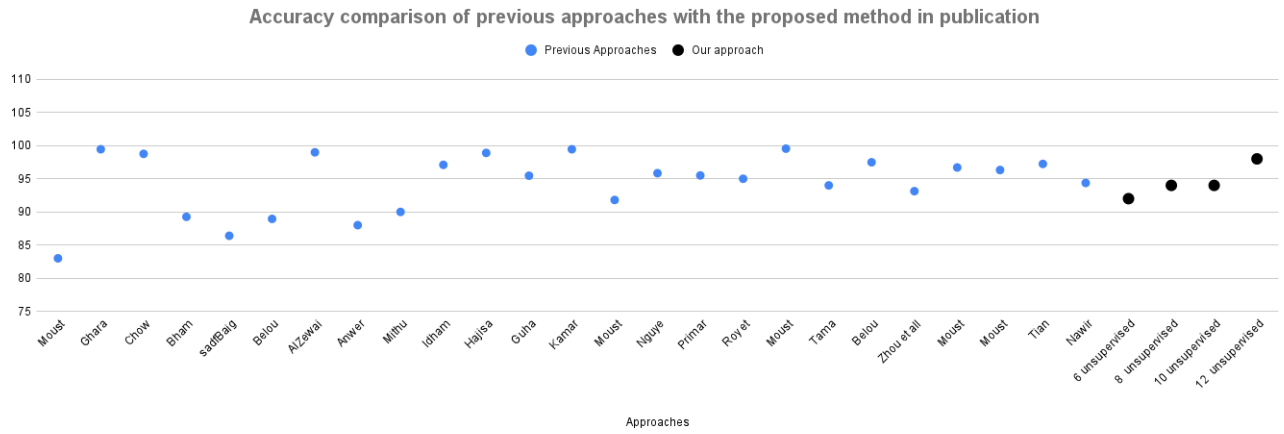


Figura 3.6: Comparación de precisión de las distintas aproximaciones

les. Utilizaremos otro Dataset ampliamente extendido y comparable en estructura al anterior.

Por otro lado, utilizaremos los grafos de visibilidad como capacidad para poder hacer la clasificación de las series temporales, en comparación con el caso anterior usado el k-shape.

3.5.1. Dataset

El primer reto de la mayoría de los casos reales es que los datos se disponen en archivos de registro que almacenan la información de forma secuencial. Estos ficheros tienen poca información sobre cada evento, por lo que su agregación en una red nos permitirá obtener una mayor cantidad de información.

En este nuevo caso, vamos a utilizar el conjunto de datos Bot-IoT porque cumple con los requisitos que necesitamos para nuestro enfoque:

- Un formato de conjunto de datos basado en flujos de red y sus características,
- Monitorización a largo plazo: para comprender el comportamiento de cada comunicación de red, necesitamos información a lo largo del tiempo, Un conjunto de datos etiquetados,
- Amplio uso en investigaciones anteriores con las que comparar.

Este conjunto de datos cumple todos nuestros requisitos:

- Recoge todos los flujos de una red. El conjunto de datos contiene más de 72.000.000 de registros;
- Etiqueta qué comunicación es un ataque;

- Nos proporciona un gran número de características que lo describen de cada flujo de red entre dos direcciones IP, como podemos ver en la Tabla 3.11.

ts	src_ip
src_port	dst_ip
dst_port	proto
service	duration
src_bytes	dst_bytes
conn_state	missed_bytes
src_pkts	src_ip_bytes
dst_pkts	dst_ip_bytes
dns_query	ns_qclass
dns_qtype	dns_rcode
dns_AA	dns_RD
dns_RA	dns_rejected
ssl_version	ssl_cipher
ssl_resumed	ssl_established
ssl_subject	ssl_issuer
http_trans_depth	http_method
http_uri	http_version
http_request_body_len	
http_response_body_len	http_status_code
http_user_agent	http_orig_mime_types
http_resp_mime_types	weird_name
weird_addl	weird_notice
label	type

Tabla 3.11: Lista de características localizadas en el conjunto de datos propuesto.

Además, no hay que olvidar que se trata de una referencia para la evaluación de los IDS.

3.5.2. Grafo multiplex con características temporales

En este capítulo, proponemos un enfoque mejorado para crear una red multiplexada de comportamiento temporal para prever cualquier caso de uso basado en el tiempo, basado en la técnica anteriormente propuesta. Sin embargo, este nuevo enfoque utiliza una técnica diferente para adquirir los comportamientos temporales de los activos. En este caso, proponemos utilizar grafos de visibilidad.

La razón principal para modificar la técnica propuesta en un documento anterior es encontrar más precisión y menos esfuerzo computacional en la optimización de la técnica propuesta anteriormente, como discutiremos a continuación.

Seguimos el mismo flujo de trabajo descrito en la investigación anterior Figura 3.7. En primer lugar, tenemos que recopilar toda la información de los paquetes de red dentro de la red multiplexada de comportamiento temporal, y después, extraer características complejas de la red para predecir si una dirección IP es un atacante o no, utilizando un algoritmo de aprendizaje automático controlado. Sin embargo, en este nuevo enfoque, proponemos utilizar los gráficos de visibilidad, k-means para la clasificación de series temporales, en lugar del algoritmo k-shape .

Para crear la red múltiplex de comportamiento temporal, en primer lugar, creamos una serie temporal, recogiendo las iteraciones entre dos nodos. Esta serie temporal nos da información sobre la evolución en el tiempo de la relación entre ellos, utilizando como ranura el número de bytes por hora, como podemos ver en la primera parte de la Figura 3.8.

El segundo paso consiste en crear una clasificación de estas series temporales. Para ello, decidimos utilizar una combinación de gráficos de visibilidad y el algoritmo no supervisado k-means. Con esto, creamos clusters de series temporales. Cada grupo describe una relación similar entre dos ordenadores.

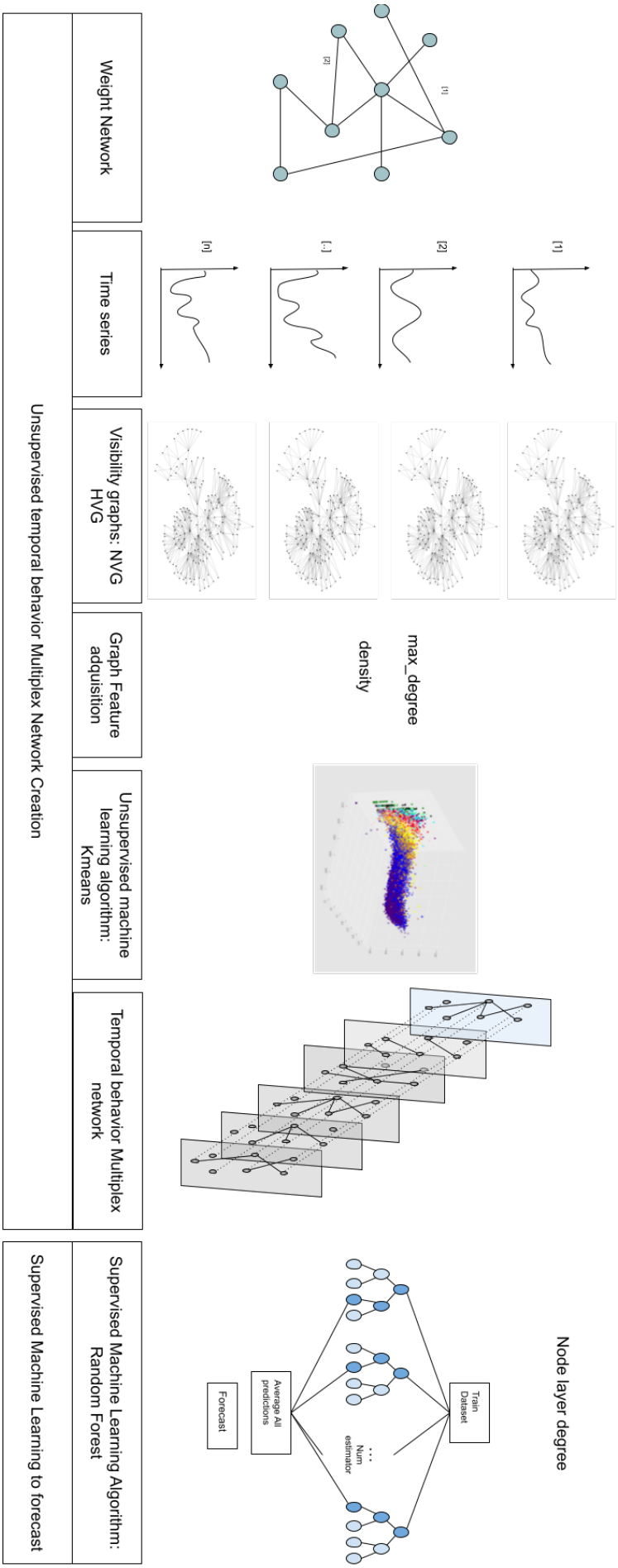


Figura 3.7: Research workflow.

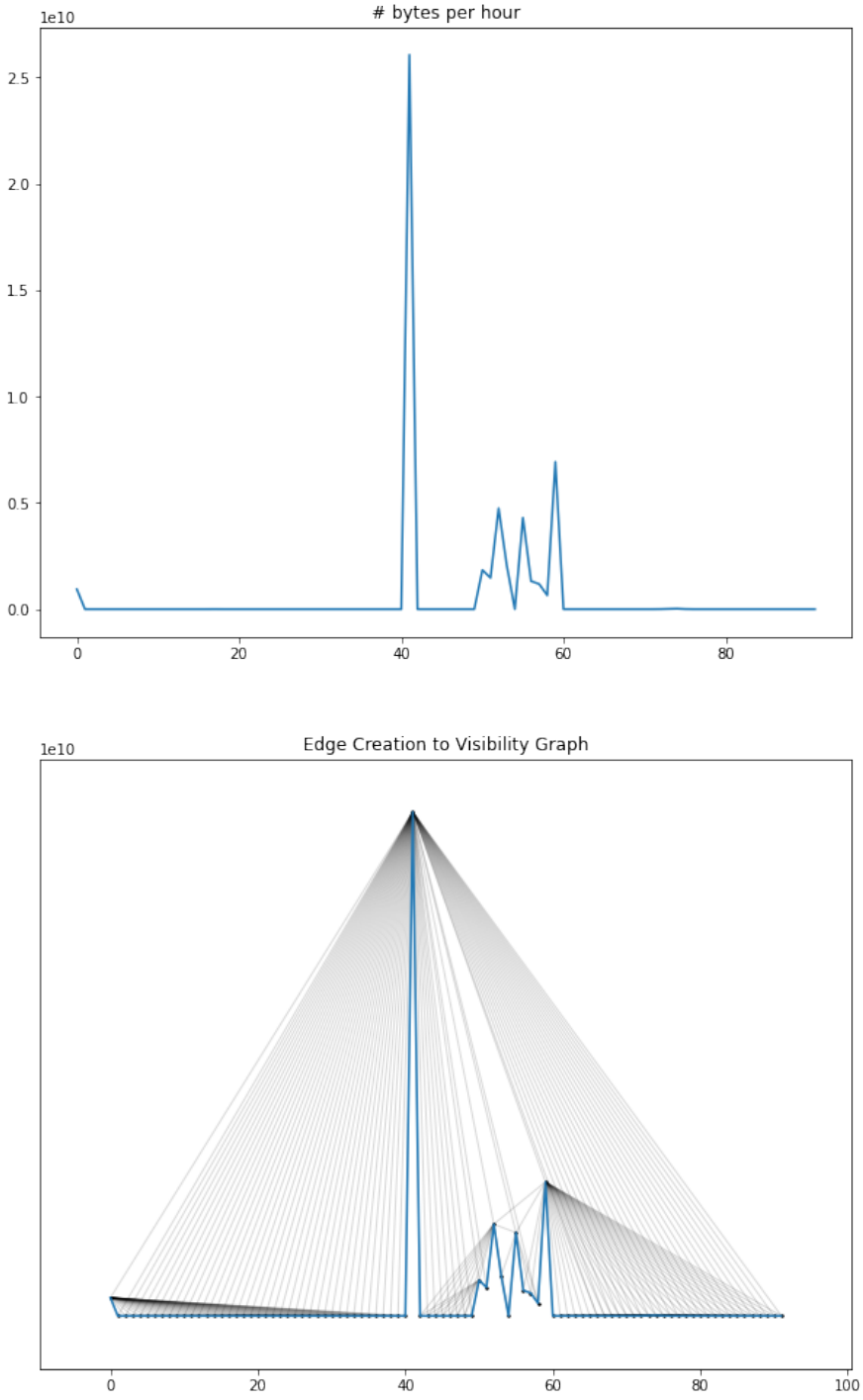


Figura 3.8: Time series to visibility graph conversion.

El último paso es rellenar la red multiplex. Esta red multiplex tendrá muchas capas como clusters de las series temporales que tenemos. Cada borde que conecte dos ordenadores se colocará en la capa, dependiendo del cluster de la serie temporal creada a partir de la información del borde. Por ejemplo, si la serie temporal creada con los bytes por hora entre dos ordenadores se encuentra en el cluster #5, la arista entre estos dos nodos se colocará en la capa #5 de la red múltiplex.

El objetivo final de la creación de esta red multiplex es poner en la misma capa con

patrones de tráfico similares. Es decir, todos los ordenadores que intenten navegar por Internet estarán situados en la misma capa. Cada nodo tendrá aristas en una o más capas que describirán su relación con el resto de la red.

El primer punto es recoger toda la información temporal sobre la relación entre los activos dentro de una red compleja de comportamiento temporal. En los párrafos siguientes describimos cómo crearla.

En primer lugar, consideramos una red ponderada como una red monocapa utilizando la forma

$$G = (N, E, W)$$

donde N se define como un conjunto de nodos $\mathcal{N} = \{n_1, n_2, .., n_i\}$ y E el conjunto de aristas que conectan los nodos como $\mathcal{N} = \{n_1, n_2, .., n_i\}$, y W se define como el peso de cada arista según una función discreta basada en el tiempo $\mathcal{W} = \{w_1, w_2, .., w_i\}$.

Como enfoque temporal, cada arista $E \in e$ puede definirse como una función discreta basada en t $W_t = \{w_{1(t)}, w_{2(t)}, .., w_{i(t)}\}$.

La serie temporal descrita como una función discreta necesita tres características para ser definida:

- Hora de inicio: Primera fecha definida para crear la serie temporal.
- Hora de finalización: Última fecha tenida en cuenta para la creación de la serie temporal.
- Frecuencia: Tiempo de intervalo en el que se utiliza la función discreta. Normalmente, la frecuencia se basa en días, semanas, segundos o años.

Basándonos en las características anteriores, describimos el peso de cada arista como:

$$w_(t) = \bigcup_{i=fecha\_inicio}^{fecha\_fin} \sum_{j=0}^{n} numero\_eventos$$

Cada arista puede definirse como una serie temporal o una función discreta con un número de elementos igual al número de periodos dentro de la referencia temporal entre la hora de inicio y la hora final. Por ejemplo, en nuestro caso, basándonos en la hora de inicio (1 de enero de 2019), la hora de finalización (30 de junio de 2019) y la frecuencia diaria, consideramos 181 elementos en cada arista. Cada elemento es la suma de ocurrencias en la franja horaria analizada, como podemos ver en la Figura 3.8.

3.5.3. Grafos de visibilidad en la obtención de atributos de temporalidad

En este mismo capítulo, anteriormente hemos presentado redes complejas de comportamiento temporal [102], [103], utilizando el algoritmo k-shape para clasificar

todas las series temporales creadas en cada arista. Este algoritmo se basa en la ”forma” de la serie temporal. Utilizando la transformación rápida de Fourier, determina la clasificación de las series temporales en función de la distancia entre ellas. Este enfoque nos permitió utilizar una capacidad compleja para clasificar una serie temporal de un modo que las redes no pueden.

El proceso k-shape es muy similar a la técnica k-means. En ambos casos, el algoritmo itera encontrando el mejor enfoque para clasificar los activos. En cada iteración, el algoritmo intenta encontrar un punto medio, denominado centroide, para cada conglomerado. A continuación, intenta determinar el mejor conglomerado para cada serie temporal. Uno de los principales inconvenientes de los algoritmos de clasificación de series temporales basados en una comparación de series temporales es el deterioro del rendimiento a medida que aumenta el número de series temporales que hay que analizar.

Para resolver este problema, hemos buscado soluciones alternativas que nos permitan resolver el mismo problema pero sin las limitaciones de complejidad existentes. En este caso, como el reto consiste en clasificar series temporales de una forma computacionalmente más eficiente, hemos buscado otras formas de representar las series temporales. Con nuevas formas de representación, es más fácil utilizar otras técnicas de clasificación que las ya empleadas hasta la fecha (k-shape).

En esta investigación hemos propuesto el uso de grafos de visibilidad introducidos por Lacasa, Luque et al. en [104] (grafos naturales de visibilidad y grafos horizontales de visibilidad) para obtener una nueva forma de representar las series temporales. Cada serie temporal se define como un grafo donde cada paso es un nodo y está conectado a todos los pasos que son visibles desde él. Sin embargo, si el valor de un paso es mayor que el de los pasos no visibles que le siguen, no aparecerá ninguna arista entre ellos, como puede verse en la figura 3.8.

En esta figura, nos centramos en los bytes enviados entre dos ordenadores. Cada registro es el número de bytes entre dos ordenadores de la red, comenzando el 23 de abril de 2019 a las 13:00:0 y terminando el 27 de abril de 2019 a las 08:00:00. Este enfoque nos da unos 100 registros como serie temporal.

Grafos naturales de visibilidad

En 2008 aparece la primera referencia sobre grafos de visibilidad natural. Se definieron en el artículo **Lacasa** y se utilizaron muy rápidamente en varias investigaciones para transformar series temporales en grafos. Esta transformación es muy útil para introducir la teoría de redes en sucesos temporales como hacemos en este trabajo, caracterizando una serie temporal como una red.

En las secciones siguientes, describiremos el método para la creación de un gráfico de visibilidad natural, tal como se presenta en el artículo mencionado. Para ello, nos basaremos en primer lugar en la representación de una serie temporal consistente en

un número limitado de enteros que representan el valor de la serie temporal en cada uno de los puntos temporales en los que se analiza la serie temporal. El grafo de visibilidad natural se basará en la creación de aristas entre todos los valores de la serie que sean visibles entre sí. Concretamente, serán visibles si no hay un valor mayor de la serie entre ellos. De una forma matemática, partiendo de una serie temporal con un número de valores en varios tiempos: $(t_1, y_1), (t_2, y_2), (t_3, y_3), ..., (t_n, y_n)$, el nodo 1 y el nodo 2 están conectados si no hay un nodo 3 entre ellos, de la siguiente manera:

$$y_3 < y_2 + (y_1 - y_2)\frac{t_2 - t_3}{t_2 - t_1}$$

Utilizando este enfoque, podemos obtener una red que represente la evolución de una serie temporal para analizar el comportamiento temporal utilizando un grafo, en lugar de una representación tradicional de series temporales.

Esta red, como comenta la investigación anterior, es:

- Conectada: basado en la definición del grafo de visibilidad. Cada nodo está conectado con el nodo izquierdo y derecho, como mínimo.
- Sin dirección.
- Invariante: ningún escalado o traducción puede afectar al grafo de visibilidad generado.

Según las series temporales de la figura 3.8, dos puntos de la serie temporal están conectados a través de una arista si es posible conectarlos; es decir, no hay picos entre ellos. Este enfoque nos da información sobre la frecuencia y el comportamiento de las series temporales con las ventajas de utilizar grafos.

Como resultado final, cada serie temporal puede describirse como un grafo de visibilidad natural, como describimos en la Figura 4.14.

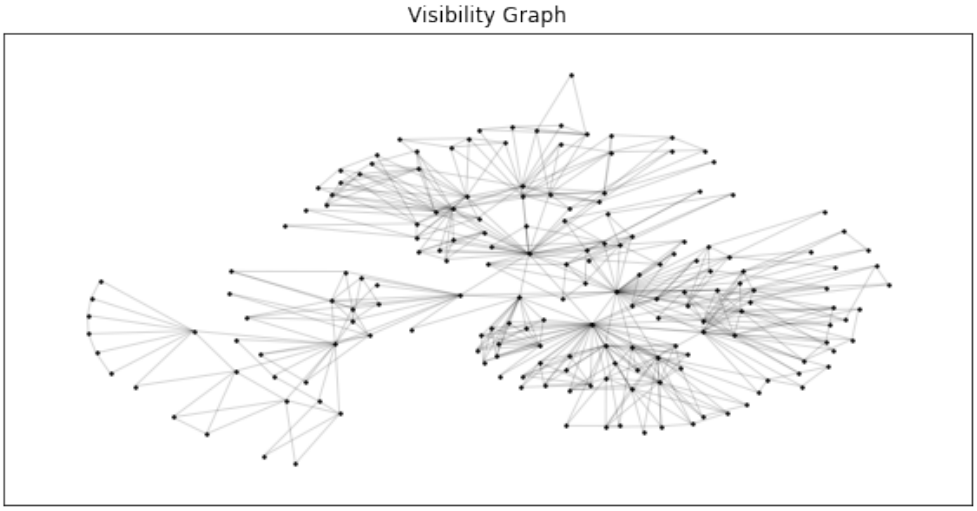


Figura 3.9: Time series to visibility graph conversion.

Grafos de visibilidad horizontales

Basado en los mismos principios que el gráfico de visibilidad natural, un año después de su definición se propuso un nuevo gráfico de visibilidad denominado gráfico de visibilidad horizontal. Ambos gráficos fueron propuestos por el mismo equipo en un breve periodo de tiempo, basándose en la evolución de sus teorías **Luque**.

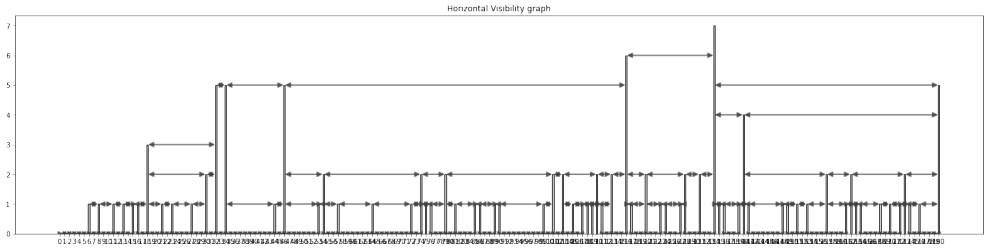


Figura 3.10: Time series to visibility graph conversion.

Al igual que el gráfico anterior, este nuevo enfoque también se basa en la visibilidad de los nodos. Como podemos ver en las figuras 4.17 y 4.19, la base de la creación del gráfico es la misma: cada registro de la serie temporal se define como un nodo en el gráfico de visibilidad. Dos nodos definidos como t_1, y_1 y t_2, y_2 están conectados si tienen conexiones horizontales, es decir, si podemos trazar una línea entre ellos sin que ninguna otra altura de registro limite su visibilidad.

$$y_1, y_2 > y_n \quad \text{for all n where } 1 \ < n < 2$$

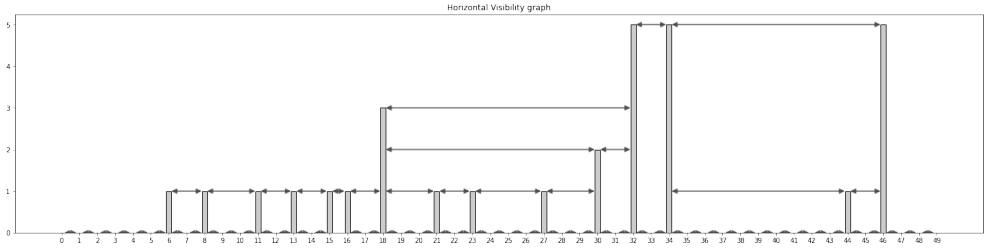


Figura 3.11: Time series to visibility graph conversion. First 50 days.

Como en el anterior gráfico de visibilidad natural, este nuevo tipo de gráfico es:

- Conectado, como el NVG.
- Invariante a cualquier traslación o reescalada.
- Irreversible: utilizando el HVG, varias series temporales pueden crear el mismo HVG, de modo que es imposible volver del gráfico a la serie temporal. Esto casi nunca es un problema porque nuestro propósito en esta operación es captar las propiedades estructurales de las series temporales. En el caso en que se necesite reversibilidad, tenemos que utilizar un grafo ponderado, y definir un grafo reversible es factible.

- Grafo no dirigido. Básicamente, no se establece ninguna dirección entre los dos nodos. Sin embargo, es posible crear un grafo dirigido utilizando la evolución temporal de la serie temporal, es decir, la dirección es la dirección en la que aumenta el tiempo en la serie temporal.
- El grafo de visibilidad natural es un grafo más conectado que el HVG.

Como podemos ver en el ejemplo siguiente, creamos las mismas series temporales que de costumbre en el gráfico de visibilidad natural, y conectamos los nodos adecuados. En la figura 4.19, podemos ver los 181 registros de la serie temporal, y se centran en los 50 primeros registros de la Figura 4.17. En el ejemplo, el registro 18 está conectado a cuatro nodos en el gráfico horizontal, una conectividad muy inferior a la que tenía en el grafo de visibilidad natural.

Agrupación de aristas

En este momento, en una red ponderada, cada arista tiene dos grafos de red: el grafo de visibilidad natural y un grafo horizontal. Estos dos grafos nos dan la información necesaria sobre el comportamiento temporal de la red 3.12.

Para crear una agrupación de aristas, tomamos dos características para cada grafo:

- max_degree: Definimos el grado de la red como el número de aristas adyacentes al nodo. Si definimos una red como un conjunto de nodos $N = \{n_1, n_2, .., n_i\}$ y E el conjunto de aristas que conectan los nodos como $E = \{e_1, e_2, .., e_i\}$, con una matriz de adyacencia A, podemos definir el grado máximo del grafo como

$$grado\_max = \text{máx} \sum_j A_{ij}$$

- densidad: Podemos definir el valor de la densidad como 0 cuando no existe ninguna arista en el grafo. Por otro lado, el valor es igual a 1 si estamos describiendo un grafo completo.

$$densidad = \frac{2m}{n(n-1)'}$$

donde n es el número de nodos y m es el número de aristas en G.

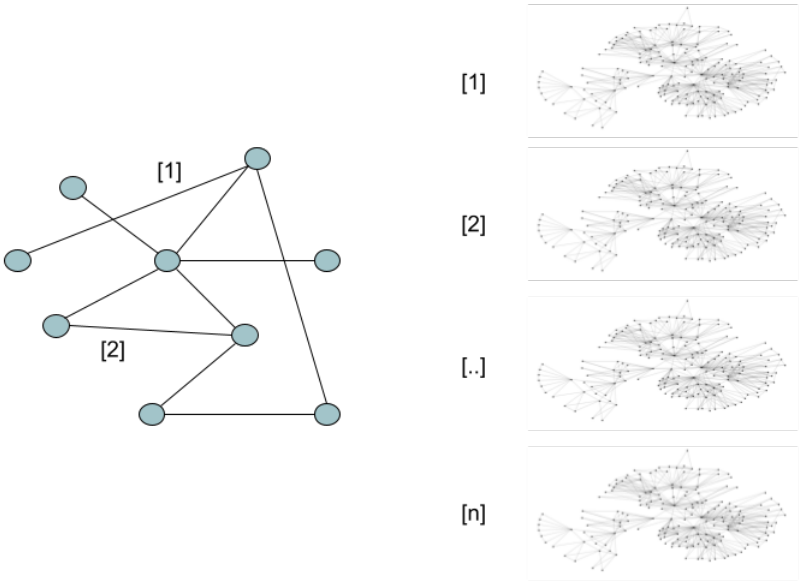


Figura 3.12: Visibility graph as edge.

Para cada arista, tenemos cuatro características:

- Gráfico de visibilidad natural grado_max,
- Densidad del gráfico de visibilidad natural,
- Gráfico de visibilidad horizontal grado_max,
- Densidad del gráfico de visibilidad horizontal.

Después de partir de las series temporales creadas con los eventos a lo largo del tiempo entre los nodos, transformarlas en ambos gráficos de visibilidad y obtener las dos características de cada uno de los gráficos de visibilidad, estamos listos para las agrupaciones de estas series temporales en grupos similares, que nos darán atributos temporales similares en cada uno de ellos.

Como no sabemos cuál es la mejor forma de agruparlos, decidimos utilizar un algoritmo de aprendizaje automático no supervisado para encontrar el procedimiento más conveniente para agruparlos. Uno de los algoritmos más famosos para este fin es el k-means, que puede clasificar los ensayos como podemos ver en la Figura 4.3.

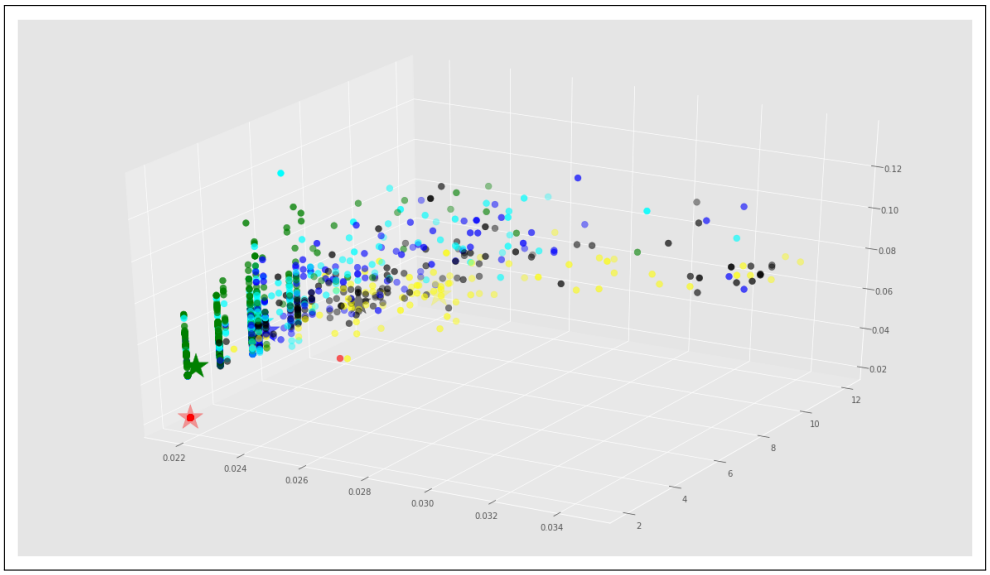


Figura 3.13: Classification in 6 clusters.

k-means es un algoritmo de clasificación no supervisado que organiza los objetos en k grupos en función de sus características. La agrupación se realiza minimizando la suma de distancias entre cada objeto y el centroide de su grupo o cluster. Se suele utilizar la distancia cuadrática. El algoritmo consta de tres pasos:

- Inicialización: una vez elegido el número de grupos, k, se establecen k centroides en el espacio de datos; por ejemplo, eligiéndolos aleatoriamente.
- Asignación de objetos a centroides: cada objeto de los datos se asigna a su centroide más cercano.
- Actualización de los centroides: la posición del centroide de cada grupo se actualiza tomando como nuevo centroide la posición de la media de los objetos pertenecientes a ese grupo.

Los pasos 2 y 3 se repiten hasta que los centroides no se mueven, o hasta que se mueven por debajo de una distancia umbral en cada paso. El algoritmo k-means resuelve un problema de optimización, siendo la función a optimizar (minimizar) la suma de las distancias cuadráticas de cada objeto al centroide de su cluster. Los objetos se representan mediante vectores reales d-dimensionales $x_1, x_2, .., x_i$, y el algoritmo k-means construye k grupos donde la suma de las distancias de los objetos, dentro de cada grupo $G = G_1, G_2, .., G_i$ a su centroide. El problema puede formularse del siguiente modo:

$$\min_{G} E(\mu_i) = \min_{G} \sum_{i=1}^{k} \sum_{x_j \in G_i} ||x_j - \mu_i||^2$$

donde G es el conjunto de datos cuyos elementos son los objetos x_j representados por vectores, donde cada uno de sus elementos representa una característica o atributo. Tendremos k grupos o clusters con su correspondiente centroide μ_i .

En cada actualización del centroide, desde un punto de vista matemático, imponemos la condición de extremo necesario a la función $G(\mu_i)$ que, para la función cuadrática anterior, es

$$\frac{\partial E}{\partial \mu_i} = 0 \Rightarrow \mu^{(t+1)} = \frac{1}{\mid G_i^{(t)} \mid} \sum_{x_j \in G_i^{(t)}} x_j$$

y la media de los elementos de cada grupo se toma como nuevo centroide.

Las principales ventajas del método k-means son que es un método sencillo y rápido. Sin embargo, es necesario decidir el valor de k, y el resultado final depende de la inicialización de los centroides.

Con este algoritmo, podemos asignar a cada serie temporal el mejor cluster, y por lo tanto, los bordes del cluster en varios clusters.

Cuantos más clusters, más información almacenaremos sobre las características temporales, ya que realizaremos una división más precisa de los comportamientos.

3.5.4. Creación de grafo multiplex

Hasta este punto, hemos definido un grafo ponderado en el que las aristas son funciones discretas basadas en el tiempo. Estas funciones pueden describirse como series temporales a partir de las cuales hemos creado dos grafos: el grafo de visibilidad natural y el grafo de visibilidad horizontal. A partir de estos gráficos, tomamos dos características de cada uno: el grado máximo y la densidad de las redes.

Con estas características, hemos asignado finalmente a cada serie temporal un cluster que agrupa las series temporales que tienen las mismas características mediante un sistema de clasificación no supervisado (k-means).

En esta última fase, vamos a transformar nuestro grafo ponderado en un grafo multiplexado a partir de la información que hemos obtenido en los pasos anteriores. Como hemos comentado al principio, los grafos son la herramienta adecuada para agrupar mucha información en ellos. Sin embargo, en la actualidad, la recopilación de características temporales dentro de los grafos ha sido bastante limitada.

En este caso, desplegaremos nuestra red de pesos en N capas, donde las capas $N = (n_1....n_n)$ en $N \in k-means(n)$. En cada una de estas N capas, incluiremos las aristas que formen parte del mismo cluster no supervisado realizado en pasos anteriores.

Por último, obtenemos una red multiplexada ponderada y dirigida G, con n capas $L = \{\ell_\alpha;\ \alpha \in \{1, ..., n\}\}$ en un conjunto de nodos $T = \{1, ..., N\}$. Cada capa se define como un grafo dirigido ponderado $\ell_\alpha = (T_\alpha, V_\alpha)$ en el mismo grupo de nodos $T_\alpha \subset T$, y con un conjunto de aristas:

$$V_\alpha = \{v_{i,j}^\alpha; \alpha \in \{1, ..., n\}\},$$

donde $v^{\alpha}_{i,j}$ representa el enlace que conecta los nodos x and y in ℓ_{α}, y w_{α} es una función $w_{\alpha} : V_{\alpha} \longrightarrow [0, +\infty)$, tal que para cada arista $e^{\alpha}_{i,j} \in E_{\alpha}$, el coeficiente $w_{\alpha}(v^{\alpha}_{x,y})$ se llama *peso* of $v^{\alpha}_{x,y}$.

3.5.5. Predicción con Random Forest

Después de crear una red multiplex de comportamiento temporal utilizando la biblioteca de python networkx [64], utilizaremos toda la información dentro de la red para extraer características complejas sobre la evolución temporal de las interacciones de los nodos.

Todas las características obtenidas del grafo son interpretaciones complejas de las interacciones entre los nodos de forma condensada con una pequeña cantidad de información.

La forma más adecuada es encontrar cuáles son las mejores características para extraer de la red multiplex de comportamiento temporal. Tras varios experimentos, decidimos seleccionar el grado de cualquier nodo de cada capa de la red. Con este enfoque, cuantas más capas tenga la red, más características podremos obtener para la realización de las fases siguientes.

Con la información extraída sobre los nodos, podemos utilizar cualquier algoritmo de aprendizaje automático supervisado para pronosticar cualquier característica de los nodos. En nuestro caso, hemos podido detectar cualquier atacante dentro de los ordenadores de la red analizada.

Para conseguirlo, tras varias investigaciones con distintos algoritmos de aprendizaje automático, decidimos utilizar Random Forest. Se trata de un algoritmo rápido y muy preciso en casi todos los casos.

3.5.6. Resultados

En primer lugar, como hemos mencionado antes, creamos las series temporales con la información temporal sobre los flujos de red ocurridos entre dos nodos. Como describimos en la sección del conjunto de datos, creamos una serie temporal entre los slots temporales existentes:

- Hora de inicio: 23 de abril de 2019 13:00:00,
- Hora de finalización: 27 de abril de 2019 08:00:00,
- Frecuencia: Cada hora.

Con estos criterios, obtenemos una serie temporal con 91 valores donde se sitúa el número de bytes enviados entre ordenadores de la red.

Para cada serie temporal, creamos un gráfico de visibilidad natural y un gráfico de visibilidad horizontal. A partir de ellos, tomamos el grado máximo y la densidad

de la red. Con esta información, clasificamos las aristas en seis grupos en los que podría aparecer cada arista.

Ahora, creamos en la red tantas capas como clusters se hayan realizado en la serie temporal, e incluimos cada arista en la capa adecuada según el cluster en el que haya sido previamente asignada. Finalmente, cada arista con el atributo cluster de valor 3 se situará en la capa 3 de la red.

Como puede verse en la figura 3.14, todos los nodos existirán en todas las capas, pero las aristas sólo existirán en la capa definida en la agrupación de las series temporales.

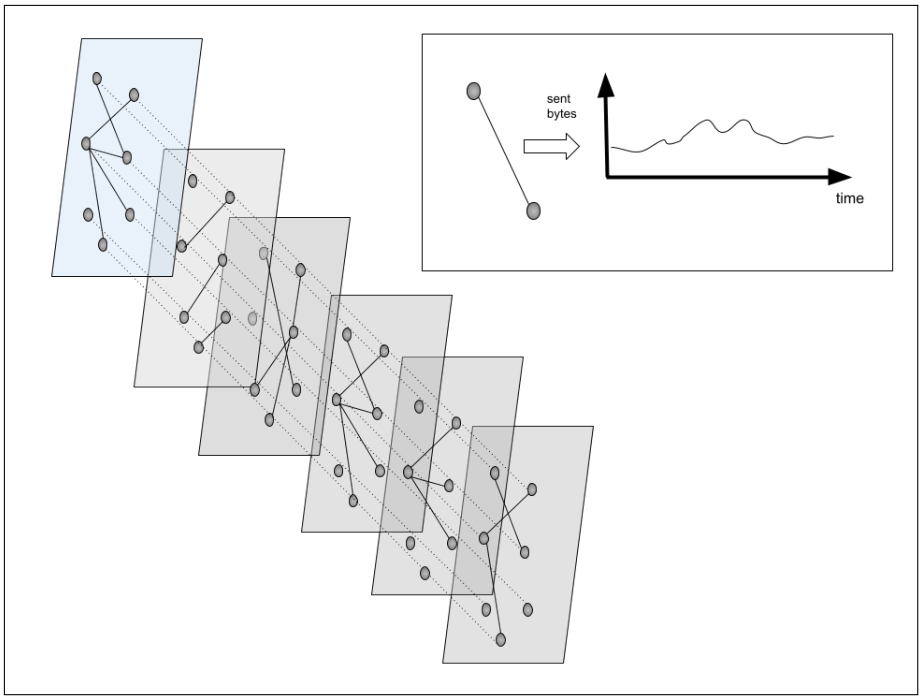


Figura 3.14: Example of multiplex network with 6 layers.

Este nuevo atributo nos permite disponer de más información en la red. Con esta nueva visión, podemos analizar la relación de los nodos con las capas para detectar varios tipos de nodos. En un enfoque más complejo, si la relación entre dos nodos puede describirse con varias series temporales, es decir, los bytes de origen y destino, la red multiplex tendrá tantas dimensiones como tipos de relación.

A continuación, mostramos el número de nodos y aristas existentes en cada capa (Tabla 3.12).

Tabla 3.12: Distribución de nodos y aristas en capas.

Cluster	Número de bordes	Número de nodos
0	25,093	24,336
1	4207	3952
2	166	166
3	392	377
4	409	411
5	102	106

Como podemos ver, el número de aristas y nodos puede variar, indicando diferentes comportamientos en cada capa. Por ejemplo, el cluster 1 describe nodos con más interacciones que otros clusters como 2. En las Tablas 3.13–3.18, mostramos los cinco nodos mejor conectados de cada capa, lo que nos proporciona información compleja sobre la conectividad de cada nodo en cada capa.

Tabla 3.13: Top 5 nodos en la capa de Cluster 0.

Nodo	Aristas conectadas
192.168.1.194	9257
192.168.1.190	5728
192.168.1.152	5284
192.168.1.184	3358
192.168.1.2	256

Tabla 3.14: Top 5 nodos en la capa de Cluster 1.

Nodo	Aristas conectadas
192.168.1.190	3503
192.168.1.195	138
192.168.1.180	123
192.168.1.30	101
192.168.1.31	89

Tabla 3.15: Top 5 nodos en la capa de Cluster 2.

Nodo	Aristas conectadas
192.168.1.190	113
192.168.1.195	20
192.168.1.180	18
192.168.1.193	9
192.168.1.30	4

Tabla 3.16: Top 5 nodos en la capa de Cluster 3.

Nodo	**Aristas conectadas**
192.168.1.190	296
192.168.1.195	25
192.168.1.180	24
192.168.1.30	14
192.168.1.31	13

Tabla 3.17: Top 5 nodos en la capa de Cluster 4.

Nodo	**Aristas conectadas**
192.168.1.190	354
192.168.1.195	18
192.168.1.180	14
192.168.1.193	8
192.168.1.30	5

Tabla 3.18: Top 5 nodos en la capa de Cluster 5.

Nodo	**Aristas conectadas**
192.168.1.190	71
192.168.1.195	21
192.168.1.31	3
192.168.1.180	3
192.168.1.1	3

Las capas permiten dotar a los nodos de información adicional a la previamente existente.

Cada uno de los grafos representa de forma diferente el comportamiento de la red de equipos analizada, por lo que aportará información adicional al estudio.

La principal ventaja del uso de grafos de visibilidad es la reducción del tiempo de proceso, como podemos ver en la Figura 4.25. k-shape [48] basado en la comparación de matrices: cuanto más larga sea la serie temporal, más tiempo de cálculo. Por otro lado, el nuevo enfoque con grafos de visibilidad mantiene el tiempo independientemente del número de clusters que tengamos.

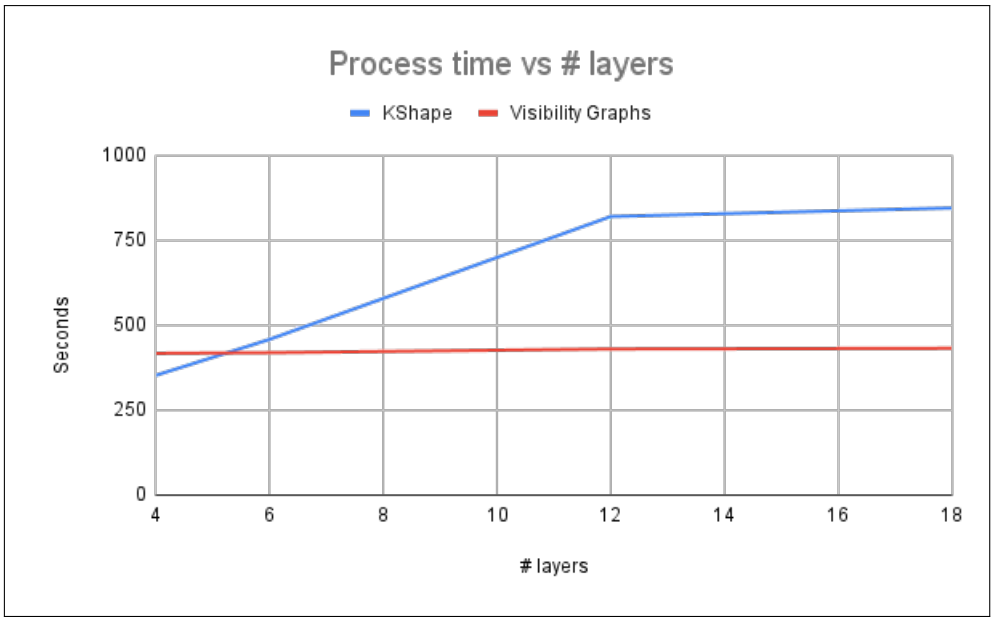


Figura 3.15: Process time comparison between k-shape and visibility graphs.

3.5.7. Adquisición de características de los nodos

Como hemos descrito anteriormente, el cluster en el que se encuentra cada arista se añade como atributo a la red creada.

El siguiente paso consiste en obtener la información necesaria de la red multiplexada para crear un sistema de detección de intrusiones de aprendizaje automático. En este caso, como necesitamos predecir si una dirección IP (nodo) es un atacante, obtenemos el número de nodos vecinos en cada capa como métrica.

Para cada una de las redes multiplexadas, obtendremos el número de aristas a las que está conectado el nodo en cada una de las capas de la red. En nuestro ejemplo, en el caso de la red con seis segmentos, cada nodo tendrá un número de aristas a las que está conectado.

3.5.8. Detección de los atacantes

En esta sección final, basándonos en las características generadas sobre el comportamiento temporal complejo de la red, intentamos detectar la dirección IP del atacante con un algoritmo supervisado.

En primer lugar, presentamos la métrica de evaluación que seleccionamos para validar el éxito de nuestro modelo. Se trata de una decisión muy relevante porque es importante compararla con proyectos de investigación anteriores. Utilizando la misma métrica de evaluación, podemos comparar nuestra técnica con otros enfoques realizados hasta el momento.

En la segunda sección, describiremos cómo desarrollamos un algoritmo de aprendizaje automático supervisado utilizando las características de nodo generadas anteriormente para clasificar cada nodo en una dirección IP de atacante o de no atacante.

Nuestro enfoque propuesto se evalúa en el conjunto de datos en términos de precisión. Esta métrica se utiliza ampliamente para la validación de modelos en muchos proyectos de investigación, por lo que podemos compararla con ellos para ratificar nuestro enfoque. La precisión se define como el número total de observaciones correctamente definidas en relación con el número total de observaciones.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}.$$

Para entender la precisión, hay que mencionar otras dos definiciones. En nuestro caso, un verdadero positivo es un resultado en el que el modelo predice correctamente la dirección IP del atacante. Del mismo modo, un verdadero negativo es un resultado en el que el modelo predice correctamente la dirección IP del no atacante.

Detección de las IP atacantes a través de un modelo supervisado

Para predecir las direcciones IP de los atacantes, creamos un algoritmo supervisado. Como sabemos cuáles son las direcciones IP de los atacantes, tomamos todas las variables obtenidas en la red compleja de comportamiento temporal y creamos un algoritmo supervisado con ellas.

Decidimos utilizar el algoritmo Random Forest para resolver este problema de clasificación. Lo seleccionamos porque el coste computacional y la eficacia están muy equilibrados. En nuestro caso, este equilibrio nos proporciona una forma sencilla de comparar varias selecciones de hiperparámetros sin incurrir en elevados costes computacionales.

En nuestro caso, este enfoque nos da la oportunidad de reducir los 72 millones de eventos originales a un algoritmo de clasificación con un menor número de características para cada nodo. Esta enorme disminución de la dimensión se debe al comportamiento temporal complejo de la red.

Como en cualquier problema de clasificación, seguimos las fases estándar para ajustar el modelo al conjunto de datos real.

- **Validación cruzada**: para evitar problemas de sobreajuste. Es muy importante utilizar la validación cruzada para obtener una precisión estable en el modelo. La CV consiste en dividir aleatoriamente los datos en N grupos; todos los grupos excepto uno entrenan el modelo, y el último lo prueba. Este proceso se repite N veces, y su precisión media se selecciona como precisión final. En nuestro caso, utilizamos la clase StratifiedKFold del paquete sklearn de python. Utilizamos esta clase en lugar de KFold porque conserva el porcentaje

de muestras para cada clase. Nuestro caso es un conjunto de datos de alto desequilibrio, por lo que es muy relevante mantener el equilibrio de clases. Por defecto, el número de pliegues se establece en 5, pero decidimos aumentarlo a 10 para obtener una predicción más precisa.

- **Train-test:** Dividimos el conjunto de datos en dos partes diferentes. La primera (entrenamiento) se utiliza para ajustar el modelo, y la otra (prueba) se utiliza para validar el modelo ajustado. La clase StratifiedKFold nos proporciona, en nuestro caso, 10 conjuntos de datos de entrenamiento y 10 conjuntos de datos de prueba para crear un modelo para cada par.
- **Random Forest** tiene varios hiperparámetros que seleccionar. Utilizamos una técnica de búsqueda en cuadrícula para seleccionar el mejor resultado para nuestro proyecto. Uno de los parámetros más relevantes a afinar es el número de estimadores. En nuestro caso, tras comparar entre varios valores, decidimos utilizar 100 como el mejor número de estimadores.

Una vez conseguido esto, utilizamos un modelo Random Forest con 100 estimadores para hacer un modelo predictivo. Como tenemos 10 modelos diferentes, utilizamos su precisión media para validar la precisión media del modelo. Todos estos pasos se describen en el algoritmo 4.

Algorithm 4 Cross Validation.

$num_splits \leftarrow 10$
$skf \leftarrow StratifiedKFold(n\_splits = num\_splits)$
for train_index, test_index in skf.split(X, y) **do**
  $X\_train, X\_test \leftarrow X[train\_index], X[test\_index]$
  $y\_train, y\_test \leftarrow y[train\_index], y[test\_index]$
  $clf \leftarrow RandomForestClassifier(n\_estimators = 100)$
  $clf.fit(X_train, y_train)$
  $y\_pred = clf.predict(X\_test)$
  $accuracy.append(metrics.accuracy\_score(y\_test, y\_pred))$

Este modelo predice la detección de direcciones IP atacantes con una red multiplexada de cuatro capas con una precisión del 99,8 %.

Comparación de enfoques

Podemos comparar nuestra precisión (99,8 %) con proyectos de investigación anteriores. Este conjunto de datos se utiliza ampliamente y por lo tanto hay una gran cantidad de referencias para comparar con, como **Koroniotis**, **Shafiq**, **Khraisat**, **Churcher**, **Zeeshan**. La siguiente tabla 3.19 nos ofrece la evaluación de estos proyectos de investigación para compararlos con nuestro resultado.

Tabla 3.19: Precisión de los enfoques anteriores.

Research	Accuracy
Koroniotis et al. **Koroniotis**	88.3/99.9
Shafiq et al. **Shafiq**	97.8/99.99
Khraisat et al. **Khraisat**	99.97
Churcher et al. **Churcher**	74/99
Zeeshan et al. **Zeeshan**	96.3

3.6. Discusión

En este trabajo, presentamos un nuevo enfoque para el despliegue de NIDS en grandes corporaciones basado en el análisis del comportamiento de la dirección IP de la red en lugar de las técnicas tradicionales de detección basadas en el análisis de cada paquete de eventos de la red.

Este enfoque se centra en la construcción de un grafo donde incluir los atributos temporales de las interacciones entre los diferentes ordenadores de una nueva forma que hemos denominado grafo multiplex de atributos temporales. Dentro de este grafo se agregan los atributos temporales obtenidos por las técnicas de clasificación de series temporales, pudiendo reducir el número de alertas generadas de millones a unos cientos de una forma computacionalmente sencilla.

Este nuevo enfoque nos proporciona dos ventajas en comparación con los NIDS tradicionales que hemos estudiado y comparado durante este capítulo:

- Reducir el número de alertas generadas por los NIDS para ser analizadas por el analista SOC: la precisión de los NIDS actuales y de los enfoques de aprendizaje automático es muy alta, sin embargo analizar cada paquete de red nos da una gran cantidad de alertas debido al gran número de eventos que cruzan las redes y en la mayoría de los casos, estas alertas con un contexto correcto sobre la dirección IP y el comportamiento de la relación entre ellos, hacen que sea más fácil para el analista SOC descartarlas. Sin embargo, esta acción redirige el foco hacia la alerta correcta donde el comportamiento nos da la razón para analizar en profundidad esta relación.
- Reducir los requisitos computacionales reduciendo el análisis a ranuras de tiempo en lugar de cada evento de red. Por ejemplo, con el análisis de comportamiento se hace cada 5 minutos, cambiamos la profundidad de análisis de cada paquete en esta ranura de tiempo para el análisis de comportamiento para las relaciones de la dirección IP existente en esta ranura de tiempo reduciendo la posible alerta de miles a cientos.

Además, la combinación de series temporales y redes complejas nos proporciona una nueva técnica para estudiar el comportamiento de cada relación entre dos di-

recciones IP y detectar el comportamiento malicioso que el analista del SOC debe analizar.

Trabajos futuros en la línea de la ciberseguridad

En este trabajo, describimos un nuevo enfoque para agregar eventos y obtener una red de multiplexación temporal y utilizamos un conjunto de datos conocidos para validar la eficacia del nuevo enfoque en un desafío real.

Sobre la base de este trabajo, hay varios estudios nuevos por hacer:

- Validación de este tipo de datos en grandes entornos corporativos.
- Cambiar las técnicas de agrupación de series temporales no supervisadas por técnicas supervisadas. Este enfoque nos permite comprender mejor la descripción de cada clúster.
- Utilizar otras técnicas de series temporales para obtener más información sobre el comportamiento de la relación entre los nodos.
- Utilizando este nuevo conjunto de datos en ciberseguridad con el mismo enfoque para confirmar la misma precisión obtenida en este trabajo.
- Utilizar este enfoque para resolver otros problemas del mundo real.
- Predicción con capacidades de redes complejas en lugar de bosques aleatorios.

4. PREDICCIÓN DE PRECIO DE LOS INMUEBLES EN NUEVA YORK BASÁNDOSE EN LOS MOVIMIENTOS DE TAXI

4.1. Introducción

La creación de modelos predictivos es un gran reto para el que las últimas técnicas de Big Data aportan soluciones para procesar grandes cantidades de información. En este trabajo presentamos una metodología basada en algoritmos de series temporales y redes multiplexadas enfocada a soluciones de Big Data. Esta metodología permite procesar una gran cantidad de información y obtener una forma de agrupar la información más efectiva y útil, permitiendo resolver problemas con un gran número de variables temporales de forma eficiente. Este enfoque ofrece la posibilidad de predecir la evolución de los precios de la vivienda a partir de los desplazamientos en taxi entre diferentes zonas de la ciudad. La metodología presentada para combinar toda esta información se basa tanto en el uso original de algunas técnicas de aprendizaje automático no supervisado como en la utilización de ciertos atributos de las series temporales y su representación como una red compleja multiplexada, consiguiendo una reducción muy significativa de la dimensionalidad de la representación de los datos obtenidos. El resultado es una previsión que reduce la representación de los viajes en taxi a un pequeño conjunto de datos para la previsión de los precios de la vivienda. Muchas situaciones reales pueden representarse con precisión mediante una red compleja con una estructura de capas en la que los enlaces de cada capa reflejan la función de los nodos en diferentes entornos [74], [75], [105], [106]. En este trabajo, intentamos predecir el precio de los inmuebles a partir de los viajes en taxi realizados en diversas zonas de Nueva York. Así, basándonos en más de 44 millones de viajes en este medio de transporte, intentamos derivar unas pocas variables por zona de taxi para simplificar las relaciones entre todos los viajes en taxi. Para crear nuestro modelo introducimos un grafo multiplexado en el que los nodos son las zonas de taxi y las aristas vienen determinadas por la existencia de viajes de taxi entre dos zonas, de forma que cada arista tiene asociada una serie temporal que describe las características de estos viajes durante un periodo determinado. A partir del tratamiento de estos datos mediante técnicas de Big Data, se define un grafo multiplexado en el que cada capa tiene asociada una

característica temporal mediante la agrupación de aristas con un comportamiento temporal similar en cada capa. Esta representación por capas permite obtener un modelo original que simplifica el problema mediante el uso de un grafo multiplex con atributos temporales. Cada capa puede explicarse como una topología temporal similar que conecta los nodos. Esta topología temporal se crea mediante técnicas de agrupación de señales no supervisadas, que describimos a continuación, cuyo uso en la resolución de otros problemas reales de características similares debe tener en cuenta dos atributos temporales para la creación de la red multiplex: el intervalo de tiempo considerado y la frecuencia de muestreo. Por lo tanto, en este tipo de problemas, se pueden crear un gran número de redes temporales multiplexadas en función de los atributos temporales seleccionados. Así, en nuestro caso, la red de taxis de Nueva York podría generar redes multiplex temporales analizando tendencias a corto, medio o largo plazo; analizando horas, días o años. Esta red de multiplexación temporal puede proporcionarnos nuevos atributos complejos para responder a retos que no hemos podido abordar hasta ahora.

Cabe destacar que en [102] acabamos de utilizar una nueva técnica para incluir nuevos atributos temporales dentro de una red y que en este trabajo, tratamos de validar y mejorar las técnicas descritas para aumentar su usabilidad en un nuevo caso de uso en el que se pueden validar los beneficios y capacidades de la red multiplex de comportamiento temporal. En concreto, basándonos en la combinación de series temporales y redes multiplex con herramientas de Big Data, en este trabajo introducimos una red multiplex en la que cada capa tiene asociada una característica temporal agrupando aristas con un comportamiento temporal similar en cada capa obteniendo unas pocas variables por zona de taxi para simplificar las relaciones entre todos los viajes de taxi. El modelo creado permite predecir el precio de los inmuebles a partir de los viajes en taxi realizados entre las zonas de taxi de Nueva York en un tiempo reciente determinado.

4.2. Contexto

Las ciudades inteligentes nos proporcionan nuevas capacidades para utilizar los datos con el fin de aumentar la precisión de los modelos que intentan modelar las características de la vida real. Las iniciativas de datos abiertos nos dan nuevas relaciones entre los activos de la ciudad que eran impensables hace unos años, abriendo nuevos casos de uso de esta información.

La aparición del Aprendizaje Automático ha llevado a la evolución de la Ciencia de los Datos y sus métodos [107]-[109] a un campo multidisciplinar cuyo objetivo es transformar los datos sobre un problema en información valiosa que pueda ser utilizada para definir modelos cuantitativos de sistemas complejos y entender su comportamiento. La idea clave de la Ciencia de Datos es que son los propios datos los que dan forma al modelo matemático. En concreto, en el campo de lo que se

conoce como aprendizaje no supervisado, se trata de diseñar una herramienta para explorar y clasificar los datos en grupos bien definidos en función de una serie de características que pueden ser conocidas o no, por lo que el objetivo es encontrar alguna estructura en ellos ('clustering'). Por otro lado, en los últimos treinta años, la teoría de redes complejas ha avanzado con notable éxito en la descripción y caracterización de las interacciones entre los diferentes elementos de los sistemas complejos y que los modelos basados en redes multicapa y multiplex se han extendido a prácticamente todas las áreas del conocimiento, es llamativo que esta teoría de modelización nos permita incorporar el parámetro tiempo, con un enfoque original desde un punto de vista diferente al utilizado en [110].

Por todo ello, el crecimiento y las aplicaciones de las redes complejas en las últimas décadas han llevado al descubrimiento de propiedades comunes a muchas redes reales, así como a una descripción y caracterización más profunda en las interacciones entre los diferentes elementos de los sistemas complejos [82], [83], [85], [111]-[113]. Así, se han ido seleccionando nuevos modelos basados en redes multicapa y multiplexadas que se están aplicando con notable éxito a situaciones reales [74], [75], [79], [82], [83], [85], [105], [114]-[116].

Actualmente, la cantidad de información que se genera en las ciudades inteligentes a partir de los dispositivos conectados está creciendo exponencialmente. Esto nos lleva a un campo en el que la disponibilidad de información en la forma adecuada proporciona un valor diferencial para la precisión de los modelos dentro de una cantidad de datos cada vez mayor. Según las previsiones de Gartner, la principal empresa de investigación y asesoramiento del mundo, este problema acaba de empezar. Según sus estimaciones, a finales de 2020 5.810 millones de unidades instaladas en todo el mundo estaban enviando información para recopilarla y analizarla, creando un nuevo problema, analizarla.

Por otro lado, es destacable que el análisis de series temporales se ha desarrollado [117],[118] en los últimos años dándonos nuevas técnicas de previsión, detección de anomalías y clustering, proporcionándonos más capacidades. En este trabajo, utilizamos el último algoritmo de clustering no supervisado de series temporales para obtener información de una gran cantidad de datos.

Para ello, construimos un modelo basado en una red multiplexada de comportamiento temporal [74], [75], [114] en la que los nodos son zonas de taxi en Nueva York, orígenes y destinos de los eventos procesados en la bitácora, y las aristas, que se establecen entre dos nodos para los que hay al menos un evento que los relaciona, tienen asociado un vector de características que nos permite analizar y agrupar las correspondientes a series temporales similares.

La estructura de este capítulo es la siguiente: tras esta introducción, presentamos las dos aproximaciones de clasificación a través de k-shape y con grafos de visibilidad.

4.3. Antecedentes y trabajos relacionados

Hasta el momento existe una enorme literatura sobre la previsión de los precios inmobiliarios. Es un reto en el que se ha investigado mucho. Sin embargo, hasta ahora, no existe un único enfoque propuesto para validar el precio de los inmuebles.

Uno de los enfoques más intensos, es comparar los bienes inmuebles como una actividad de inversión. En este enfoque, varios trabajos tratan de comparar la evolución de los precios inmobiliarios con los precios de las acciones en varias situaciones geográficas, como [119], [120], [121] y [122].

Sin embargo, algunas de ellas [119] no confirman que esta relación pueda validarse siempre.

Por otro lado, otros métodos de previsión de precios inmobiliarios **ambientales** se basan tanto en las características físicas internas como en las externas del entorno. Por ejemplo, varias características físicas internas que pueden influir en el precio son el número de habitaciones, los baños, la calidad de la construcción y los aparcamientos del edificio.

Como características ambientales externas, podemos mencionar la calidad del barrio, como la disponibilidad de transporte público, el acceso a los supermercados, ...

En base a estas variables para definir una unidad inmobiliaria, se encuentran diferentes enfoques para pronosticar el precio. Desde enfoques hedonistas y modelos de regresión hasta redes neuronales. Todos ellos son utilizados en diversas investigaciones para validar varias propuestas [123], [124] y [125].

4.4. Material y Métodos

Proponemos validar si el comportamiento temporal de los viajes en taxi puede ser relevante para predecir los precios de los inmuebles en la ciudad de Nueva York. Nuestra investigación trata de tomar nuevas características de las ciudades inteligentes (utilizando los viajes en taxi), incluirlas en una red de comportamiento temporal múltiple y utilizarlas para predecir los precios de los inmuebles.

Este enfoque ingenuo trata de utilizar un nuevo tercer tipo de datos para el precio de los inmuebles. Hasta ahora, como hemos mencionado antes, las características físicas internas y las del entorno externo son el tipo de características más utilizadas. Nuestro enfoque trata de darnos información sobre los desplazamientos temporales de los taxis. Pretendemos agrupar las relaciones entre las zonas de taxis para describir el tipo de relación entre ellas; intentamos distinguir los tipos de viajes: viajes de negocios y viajes de ocio.

La validación de los diferentes tipos de viajes nos da información relevante sobre la evolución futura de los precios de los inmuebles.

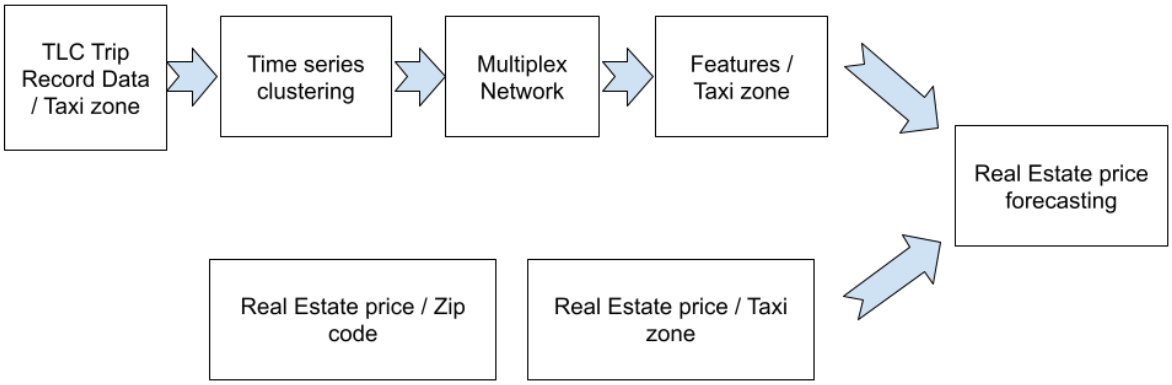


Figura 4.1: Flujo de investigación

4.5. Grafo con clasificación k-shape

En la primera parte de este capítulo, igual que en el anterior vamos a realizar el grafo multiplex con características temporales a través de la clasificación de series temporales k-shape.

4.5.1. Dataset

El primer reto al que hay que enfrentarse es fusionar todas las fuentes de datos para gestionar toda la información con la estructura adecuada. Toda la información sobre viajes en taxi se presenta por zonas de taxi. Sin embargo, toda la información inmobiliaria se presenta por códigos postales. Por lo tanto, nuestro primer reto es unir ambas fuentes para proporcionar toda la información en la misma distribución geográfica.

4.5.2. Precios inmobiliarios

Nuestra primera fuente de datos es la evolución de los datos inmobiliarios. Tras analizar varias fuentes, decidimos utilizar la información adquirida de Zillow [126]. Zillow es una empresa estadounidense de mercado inmobiliario online que se fundó en 2006. Zillow dispone de datos sobre aproximadamente 110 millones de viviendas en todo Estados Unidos. La empresa ofrece varias funciones, como estimaciones del valor de las viviendas, cambios en el valor de cada vivienda en un periodo de tiempo determinado, vistas aéreas de las viviendas y precios de viviendas comparables en la zona.

Con toda esta información, esta empresa dispone de un valor agregado de Datos de Vivienda donde tiene la evolución de las previsiones del valor de la vivienda. Sin embargo, esta información se proporciona por código postal.

A partir de este conjunto de datos, queremos obtener la tendencia de los precios durante seis meses:

- código postal
- Precio de julio de 2019
- Precio de agosto de 2019
- Precio de septiembre de 2019
- Precio de octubre de 2019
- Precio de diciembre de 2019

4.5.3. Viajes en taxi

Nuestra segunda fuente de datos son los viajes en taxi en la ciudad de Nueva York. Toda esta información se recoge durante los últimos años, dándonos una enorme información sobre el comportamiento de los viajes por toda la ciudad [127].

Nuestro reto es, sólo con esta información, poder predecir la tendencia de los precios de los inmuebles. Intentamos demostrar que el comportamiento de los viajes en taxi nos da información sobre la relación económica entre los códigos postales, por lo que podemos inferir la evolución de los precios de los inmuebles con esta información.

Para conseguirlo, obtenemos todos los Registros de viajes en Taxi Amarillo en los primeros seis meses de 2019. Tenemos 44.459.136 viajes que nos dan información exacta sobre el lugar de recogida, el número de pasajeros, el lugar de destino,...

En concreto, por cada viaje tenemos la siguiente información:

- VendorID: Un código que indica el proveedor de TPEP que facilitó el registro.
- tpep_pickup_datetime: La fecha y hora en que se activó el contador.
- tpep_dropoff_datetime: La fecha y hora en que se desconectó el contador.
- passenger_count: El número de pasajeros del vehículo.
- ride_distance: La distancia recorrida en millas indicada por el taxímetro.
- RatecodeID:
- store_and_fwd_flag
- PULocationID: Taxi TLC Zona en la que se contrató el taxímetro
- DOLocationID: Taxi TLC Zona en la que se desconectó el taxímetro
- payment_type: código numérico que indique cómo pagó el pasajero el viaje
- fare_amount: La tarifa de tiempo y distancia calculada por el contador
- extra: Extras y recargos varios. Actualmente, sólo incluye las tasas de 0,50 y1 por hora punta y noche.

- mta_tax: 0,50$ de impuesto ATM que se activa automáticamente en función de la en uso.
- tip_amount: Importe de la propina. Este campo se rellena automáticamente para las tarjeta de crédito. No se incluyen las propinas en efectivo.
- tolls_amount: Importe total de todos los peajes pagados en un viaje.
- improvement_surcharge: $0.30 recargo de mejora aplicado a las atracciones en la bajada de bandera. El recargo de mejora comenzó a recaudarse en 2015.
- total_amount: Importe total cobrado a los pasajeros. No incluye propinas en efectivo
- congestion_surcharge

Cruzando la información disponible en el conjunto de datos y la que necesitamos para nuestra investigación, decidimos obtener las siguientes columnas del conjunto de datos para cada viaje en taxi:

- tpep_pickup_datetime
- PULocationID
- DOLocationID
- passenger_count

4.5.4. Agregación de datos

Como podemos observar en la subsección anterior, ambas fuentes de datos describen la ciudad de Nueva York. Sin embargo, ambas fuentes de datos se basan en diferentes definiciones de zona. Por un lado, los precios de los inmuebles se basan en los códigos postales de la ciudad. Por otro lado, los viajes en taxi se basan en las zonas de taxi. Por lo tanto, necesitamos cruzar estos conjuntos de datos para fusionarlos en una misma zona geográfica en la que podamos prever los precios inmobiliarios.

Para resolverlo, decidimos utilizar la biblioteca Shapely para obtener el punto medio de cada zona de taxis. Después, encontramos dónde se encuentra este punto geográfico y tomamos el código postal relacionado con este punto.

4.5.5. Grafos multiplex basados en características temporales

En esta sección, describimos cómo incluir 44 millones de viajes en taxi dentro de la red. Nuestro objetivo es generar atributos temporales dentro de la red que puedan agregar todos los viajes en taxi con una visión compleja. Utilizamos el mismo enfoque que empleamos en un artículo anterior [102]. Pretendemos validar el uso de este enfoque a una situación más sofisticada, con más información para fusionar.

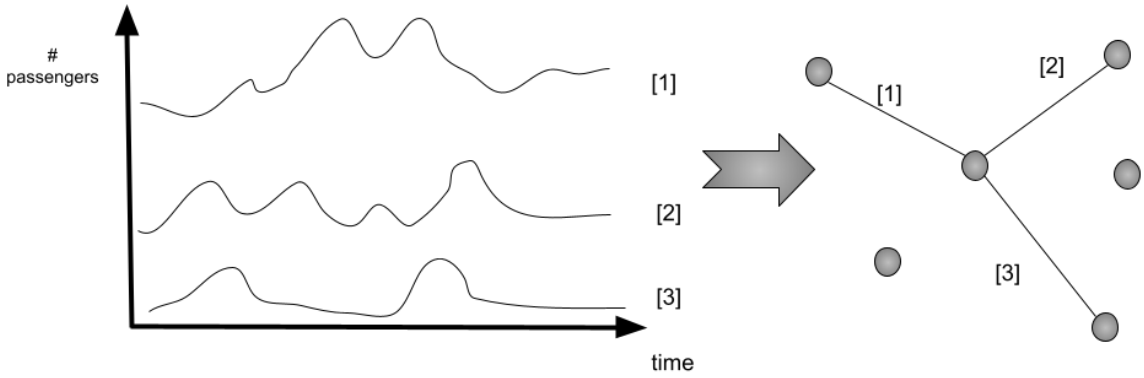


Figura 4.2: Network + timeseries combination

Pretendemos combinar algoritmos de series temporales con redes para enriquecer redes complejas con nuevos atributos temporales, imposibles de conseguir hasta ahora.

Como estamos hablando de una representación temporal de la realidad, tenemos que definir cuáles son los parámetros temporales para realizar nuestra investigación. Los mismos datos pueden darnos múltiples redes temporales en función de las franjas horarias utilizadas para la investigación.

A partir de este momento, vamos a analizar las relaciones entre las zonas de taxi en un espacio de tiempo y frecuencia de muestreo como el día, definido para este fin.

- Fecha inicial(01/01/2019): Fecha en la que la serie temporal se inicia.
- Fecha final(06/30/2019): Fecha en la que se termina la serie temporal.
- Frecuencia de muestreo: corresponde con la frecuencia con la que se toman las muestras en la serie temporal. En este caso se considera una frecuencia diaria para el actual análisis.

Red de una sola capa

El primer paso, es definir la estructura básica de la red multiplex de comportamiento temporal.

Para ello, el primer paso es determinar las definiciones básicas de la red.

- Definimos los nodos como cualquier zona de taxis de Nueva York.
- creamos una arista para dos zonas de taxis en las que se encuentra cualquier viaje en taxi en el tiempo utilizado para realizar esta investigación. En cada

arista, incluimos como atributo todos los viajes en taxi existentes entre estas dos zonas de taxi. En nuestro caso, tenemos 45.952 aristas existentes en nuestra red.

Esta es una de las ventajas más relevantes de la representación en red, tenemos tantos nodos como salidas de zonas de taxi. Todas las salidas de taxi del periodo nos darán más información sobre su relación, pero el número de nodos máximo es siempre el número de zonas de taxi, no más. En este caso, el número de nodos es de 264 nodos por cada zona de taxi en Nueva York.

Series temporales

En la subsección anterior, separamos los viajes en taxi en atributos de aristas. En cada arista, tenemos cualquier viaje en taxi entre dos zonas diferenciadas.

En este paso, tenemos que normalizar estos viajes en taxi, convirtiéndolos en series temporales similares antes de agruparlos.

A partir de los datos de cada arista, queremos crear una serie temporal. Si consideramos la función entre dos nodos llamados i y j, la función de la serie temporal es

$$f_{(i,j,t)} = \sum_{t=0}^{n} numero\_pasajeros_{(t)}; t \in \{1, ..., n\}$$

Primeramente tenemos que definir los parámetros básicos de la serie temporal. Como hemos comentado antes, decidimos analizar el comportamiento temporal con tres características temporales:

- Fecha inicial: 01/01/2019.
- Fecha final: 06/30/2019.
- Frecuencia: diaria.

Con estas características, podemos crear series temporales a partir de cada arista. La serie temporal se definirá como el número de pasajeros por hora entre la hora de inicio y la de finalización. Esta definición nos da una serie temporal con 181 puntos donde cada uno es el número de pasajeros de cada día.

Con esta definición, tenemos 45952 series temporales que describen el número de pasajeros por una hora entre zonas de taxi con la misma hora de inicio y final y la misma frecuencia.

El siguiente paso, es analizar y agrupar todas estas series temporales para obtener el comportamiento temporal dentro de la red. Para conseguirlo, tenemos que agruparlas para proporcionar un comportamiento temporal a la red. Para realizar este

clustering de comportamiento, evaluamos varias técnicas para conseguir el clustering no supervisado basado en estudios anteriores, como ejemplo [128]. En general, hay tres maneras diferentes de agrupar las series temporales. La primera está basada en la forma, la segunda en las características y la tercera en el modelo.

- En el **enfoque basado en la forma**. Este enfoque trata de hacer coincidir dos series temporales lo mejor posible en función de la "forma"de las mismas. Cualquier deslizamiento temporal o magnitud no se analiza en este tipo de enfoque. Este tipo de clustering se denomina enfoque basado en datos brutos porque como entrada utiliza datos de series temporales brutos sin modificaciones. Los algoritmos basados en la forma suelen utilizar métodos de clustering tradicionales, como k-means,..., que son compatibles con los datos estáticos con una pequeña modificación. Utilizan una medida de distancia diferente para adaptarlos a las series temporales. K-shape [48] es una de las implementaciones más famosas.
- En el enfoque textbf basado en características, las series temporales se transforman en un vector de características de menor dimensión. Como ejemplo de este enfoque, podemos mencionar una biblioteca de python llamada tsfresh [25]. Posteriormente, se aplica un algoritmo de clustering convencional a los vectores de características extraídos.
- En los métodos basados en el modelo, una serie temporal sin procesar se transforma en parámetros de modelo y, a continuación, se elige una distancia de modelo adecuada y un algoritmo de agrupación que se aplica a los parámetros de modelo extraídos. Sin embargo, se demuestra que, por lo general, los enfoques basados en el modelo tienen problemas de escalabilidad, y su rendimiento se reduce cuando los clusters están cerca unos de otros.

Como hemos mencionado antes, nuestro objetivo es comparar comportamientos similares, por lo que, tras varias pruebas, decidimos utilizar un enfoque basado en la forma. Es el mejor modo de comparar el comportamiento temporal entre zonas de taxis. Entre los enfoques basados en la forma, hemos seleccionado este nuevo modelo de clustering llamado k-shape [48] por su velocidad de cálculo y facilidad de uso ya que el número de zonas de taxi crea una gran cantidad de aristas entre ellas, por lo que nos basamos en uno de los aspectos más relevantes de este tipo. k-shape se basa en un procedimiento de refinamiento iterativo escalable, que crea clusters homogéneos y bien separados. Para ello, como medida de distancia, k-Shape tiene que utilizar una métrica de distancia. En este caso, utiliza una versión normalizada de la medida de correlación cruzada para considerar las formas de las series temporales al compararlas. A partir de las propiedades de esa medida de distancia, basada en las propiedades de la correlación cruzada, es posible desarrollar un método para calcular rápidamente los centroides de los clusters, que se utilizan en cada iteración para actualizar la asignación de las series temporales a los clusters.

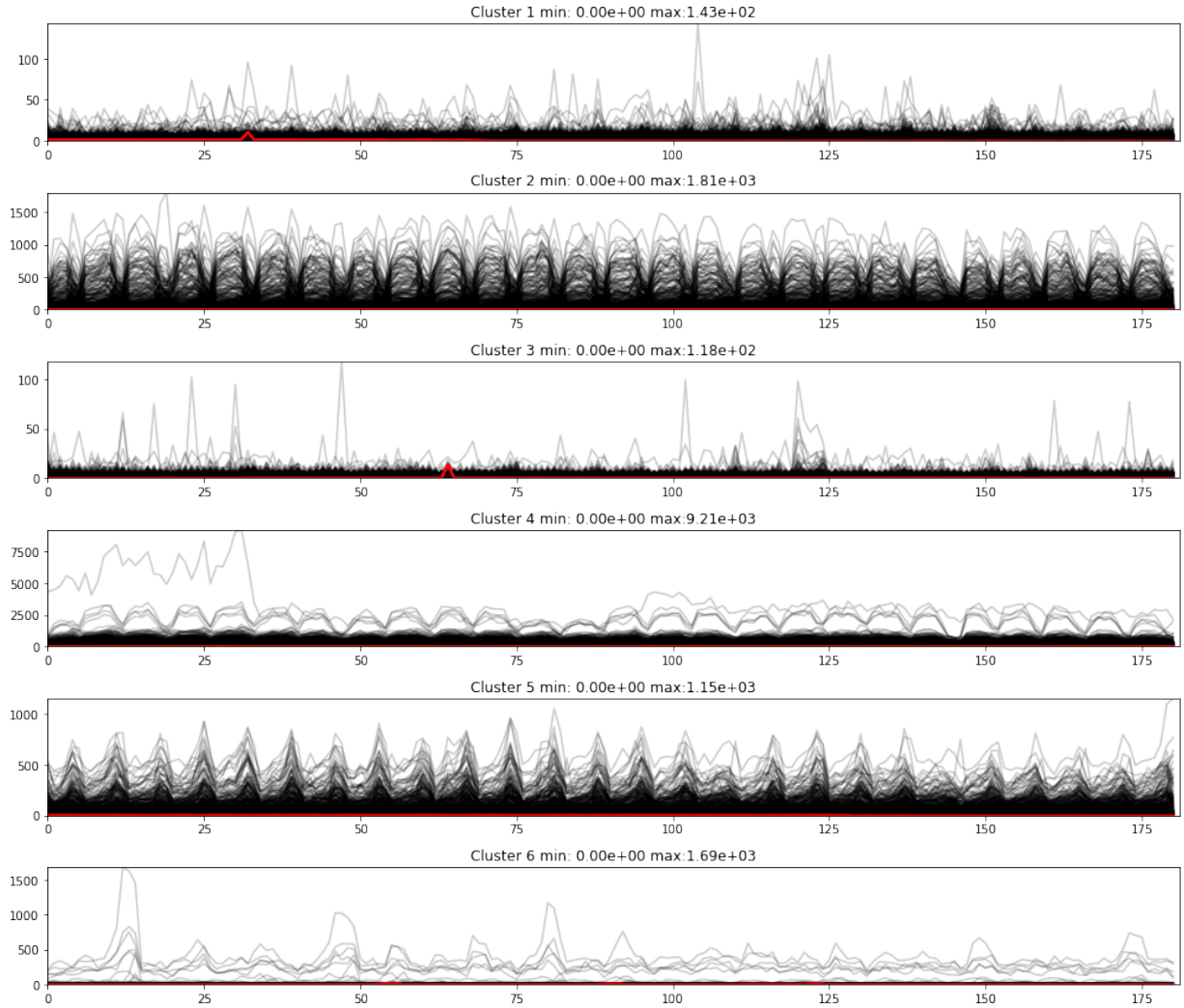


Figura 4.3: Agrupación no supervisada de series temporales

Como podemos ver en la figura 4.3, el cluster 2 muestra un comportamiento temporal basado en un mayor tráfico de lunes a viernes que los fines de semana, es decir, interacciones financieras. Por otro lado, el clúster 6 se basa en los pasajeros de fin de semana, podemos prever que este tipo de relación se basa en las actividades recreativas. Este es el comportamiento temporal que queremos incluir en la red monocapa que hemos creado antes.

Red multiplex

El siguiente paso es incluir toda esta nueva información dentro de la red. Para ello, definimos una red multiplexada de comportamiento temporal.

En las últimas décadas, hay muchos estudios relacionados con redes complejas que utilizan varias capas, como las redes multirelacionales en sociología y las redes interconectadas de diferentes subsistemas en ingeniería. Sin embargo, la primera definición de redes multicapa [74], [75] se desarrolló hace unos años con el objetivo principal de ayudar a estudiar muchos tipos de redes y recogerlas en una misma arquitectura. A pesar de su relativamente corta historia, el estudio de este tipo de grafos se ha vuelto muy prominente. En este documento, vamos a seguir principalmente la terminología y las convenciones del artículo de revisión [75].

En nuestro modelo, nos centramos en las redes multiplexadas, en las que a menudo se asume por conveniencia que todas las capas están formadas por el mismo conjunto de nodos, pero esto no es necesario.

Normalmente, este tipo de red se utiliza para dos tipos de redes:

- Redes multirelacionales: cada capa representa un tipo diferente de interacción. Por ejemplo, una red de tipos de transporte dentro de una gran ciudad. Cada capa es un tipo de transporte diferente: metro, autobús, tren...
- En las redes temporales, cada capa codifica el mismo tipo de interacciones durante diferentes puntos temporales o a lo largo de diferentes ventanas de tiempo. En la representación multiplex más común de una red temporal, las capas consecutivas están conectadas a través de bordes entre capas que vinculan a los individuos consigo mismos en diferentes momentos.

Las redes temporales se basan en la representación de la relación de cada nodo a lo largo del tiempo, como se describe en investigaciones como [65]. Sin embargo, este enfoque no nos da ningún atributo temporal para la red. En nuestro caso, el objetivo principal es obtener información temporal de la evolución de la relación entre los nodos en función del tiempo.

La red multiplexada de comportamiento temporal trata de desplegar una red compleja en varias capas de comportamiento temporal, dándonos la oportunidad de utilizar esta información temporal desplegada para obtener información sobre la

similitud entre las relaciones de los nodos alrededor de todas las redes. Para ello, incluimos en cada capa de la red multiplexada diferentes comportamientos temporales que hemos detectado antes con el algoritmo k-shape.

Al igual que en el apartado anterior, agrupamos las series temporales en 6 clusters, luego tenemos que definir 6 capas en nuestra red multiplex. En cada capa, incluimos las aristas que pertenecen al cluster seleccionado, es decir, en la primera capa incluimos cada arista cuya serie temporal forma parte del cluster 0, en la capa 1 cada arista pertenece al cluster 1, y así sucesivamente.

Después de esto, tenemos una red multiplexada de comportamiento temporal en la que en cada capa tenemos un comportamiento temporal diferente. El resultado de todo este proceso es una red multiplexada donde los atributos de cada viaje en taxi de Nueva York, es decir, incluimos los 44 millones de viajes en una red multiplexada con 264 nodos, uno por cada zona de taxi, y 45952 aristas incluidas en 6 capas que describen diferentes comportamientos temporales.

Este enfoque nos proporciona una poderosa herramienta para reducir el tiempo de procesamiento de una enorme cantidad de eventos temporales.

Grafo multiplex

El siguiente paso es incluir toda esta nueva información dentro de la red. Para ello, definimos una red multiplexada de comportamiento temporal.

En las últimas décadas, hay muchos estudios relacionados con redes complejas que utilizan varias capas, como las redes multirelacionales en sociología y las redes interconectadas de diferentes subsistemas en ingeniería. Sin embargo, la primera definición de redes multicapa [74], [75] se desarrolló hace unos años con el objetivo principal de ayudar a estudiar muchos tipos de redes y recogerlas en una misma arquitectura basada. A pesar de su relativamente corta historia, el estudio de este nuevo tipo de redes se ha vuelto muy prominente. Al presentar brevemente esta idea, seguimos principalmente la terminología y las convenciones del artículo de revisión [75].

Normalmente, este tipo de red se utiliza para dos tipos de redes:

- Redes multirelacionales: cada capa representa un tipo diferente de interacción. Por ejemplo, una red de tipos de transporte dentro de una gran ciudad. Cada capa es un tipo de transporte diferente: metro, autobús, tren...
- En las redes temporales, cada capa codifica el mismo tipo de interacciones durante diferentes puntos temporales o a lo largo de diferentes ventanas de tiempo. En la representación multiplex más común de una red temporal, las capas consecutivas están conectadas a través de bordes entre capas que vinculan a los individuos consigo mismos en diferentes momentos.

Las redes temporales se basan en la representación de la relación de cada nodo a lo largo del tiempo, como se describe en investigaciones como [65]. Sin embargo, este enfoque no nos da ningún atributo temporal para la red. En nuestro caso, el objetivo principal es obtener información temporal de la evolución de la relación entre los nodos en función del tiempo.

La red multiplexada de comportamiento temporal trata de desplegar una red compleja en varias capas de comportamiento temporal, dándonos la oportunidad de utilizar esta información temporal desplegada para obtener información sobre la similitud entre las relaciones de los nodos alrededor de todas las redes. Para ello, incluimos en cada capa de la red multiplexada diferentes comportamientos temporales que hemos detectado antes con el algoritmo k-shape.

Al igual que en el apartado anterior, agrupamos las series temporales en 6 clusters, luego tenemos que definir 6 capas en nuestra red multiplex. En cada capa, incluimos las aristas que pertenecen al cluster seleccionado, es decir, en la primera capa incluimos cada arista cuya serie temporal forma parte del cluster 0, en la capa 1 cada arista pertenece al cluster 1, y así sucesivamente.

Después de esto, tenemos una red multiplexada de comportamiento temporal en la que en cada capa tenemos un comportamiento temporal diferente. El resultado de todo este proceso es una red multiplexada en la que los atributos de cada viaje en taxi de Nueva York, es decir, incluimos los 44 millones de viajes en una red multiplexada con 264 nodos, uno por cada zona de taxi, y 45952 aristas incluidas en 6 capas que describen un comportamiento temporal diferente.

Este enfoque nos proporciona una poderosa herramienta para reducir el tiempo de procesamiento de una enorme cantidad de eventos temporales.

Si consideramos una *red multiplex directa y pesada* [76] M como una tripleta $M = (X, A, L)$ donde X es un conjunto de nodos $X = \{1, ..., N\}$, y L son n capas $L = \{\ell_\alpha;\ \ \alpha \in \{1, ..., n\}\}$ y A es un conjunto de aristas:

$$A_\alpha = \{a^\alpha_{i,j}; \alpha \in \{1, ..., n\}\},$$

y definimos k-shape como un conjunto de c clusters $C \in \{0, ..., c\}$

$$a_{(i,j),c} = k - shape(c)$$

En la tabla 4.1, podemos ver el número de nodos y aristas en cada capa después de ejecutar el algoritmo k-shape para todas las series temporales creadas con los viajes en taxi entre todas las zonas de taxis en Nueva York.

En las tablas 4.2, 4.3,4.4, 4.5, 4.6, y 4.7, describimos los 5 nodos superiores que más aristas conectadas tienen en cada capa dándonos información compleja sobre la conectividad de cada nodo en cada capa. Como podemos ver, el número de aristas y nodos puede variar indicando varios comportamientos en cada capa.

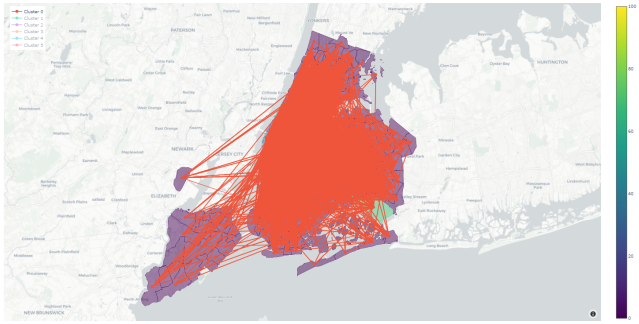


Figura 4.4: Cluster 0

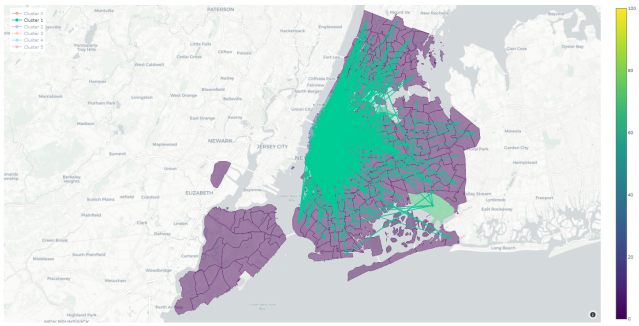


Figura 4.5: Cluster 1

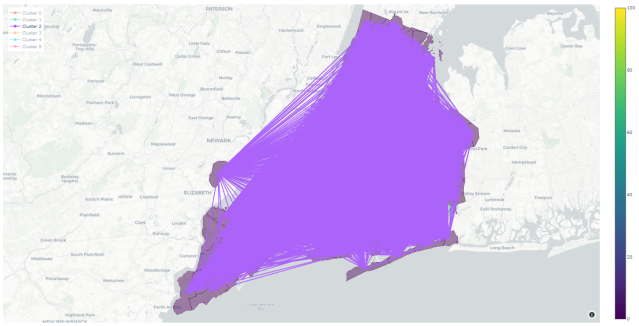


Figura 4.6: Cluster 2

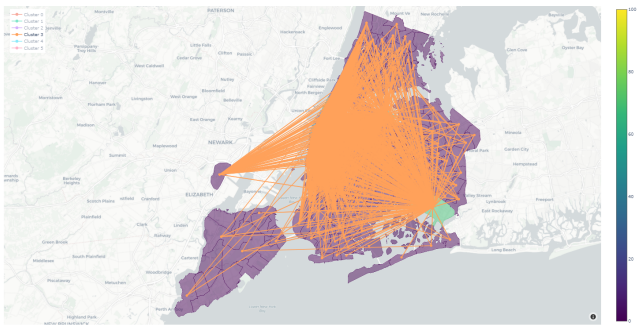


Figura 4.7: Cluster 3

Cluster	Número de aristas	Número de nodos
0	5014	258
1	972	173
2	22465	264
3	3073	231
4	973	162
5	173	137

Tabla 4.1: Distribution of nodes and edges in layers

Nodo	aristas conectadas
137	94
164	94
264	94
170	90
100	87

Tabla 4.2: Top 5 nodos con aristas en la capa del cluster 0 en NY

Nodo	aristas conectadas
148	82
79	60
161	54
140	51
162	45

Tabla 4.3: Top 5 nodos con aristas en la capa del cluster 1 en NY

Nodo	aristas conectadas
265	236
197	225
215	223
219	223
130	222

Tabla 4.4: Top 5 nodos con aristas en la capa del cluster 2 en NY

4.5.6. Regresor tipo Random Forest

Como paso final, tenemos que obtener características especiales de la red multiplexada para completar nuestro análisis. A partir del modelo que hemos creado, los atributos del comportamiento temporal se localizan en capas. Así, en nuestro caso

Nodo	aristas conectadas
132	219
138	186
186	100
100	97
130	97

Tabla 4.5: Top 5 nodos con aristas en la capa del cluster 3 en NY

Nodo	aristas conectadas
79	58
114	53
249	50
50	45
148	44

Tabla 4.6: Top 5 nodos con aristas en la capa del cluster 4 en NY

Nodo	aristas conectadas
264	26
246	9
69	8
76	8
42	7

Tabla 4.7: Top 5 nodos con aristas en la capa del cluster 5 en NY

y para obtener la información necesaria sobre cada nodo, utilizaremos el grado del nodo en cada una de las capas. Por lo tanto, si consideramos una red multiplexada pesada y dirigida M como en los apartados anteriores, el grado de un nodo es el número de aristas conectadas al nodo.

Ahora, consideraremos el análisis del grado de cada nodo en todas las capas, es decir, el grado multicapa de un nodo x_i, $grado_M(x_i)$ como la suma de todos los grados de i dentro de cada capa, es decir

$$grado_M(x_i) = \sum_{alfa=1}^{m} grado^{alfa}(x_i).$$

donde $\forall \alpha \in \{1, \cdots, m\}$, $deg^{\alpha}(x_i) = grado^{in}(x_i) + grado^{out}(x_i)$, es decir, la cardinalidad del conjunto de vecinos directos de x_i en la capa α.

Es importante tener en cuenta que en el caso que estamos analizando, cada nodo tiene tantos atributos diferentes como capas tenga la red multiplex.

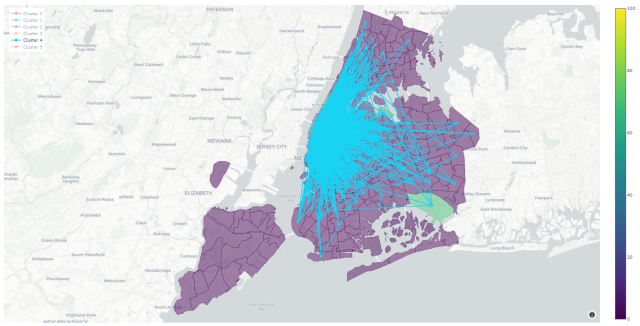


Figura 4.8: Cluster 4

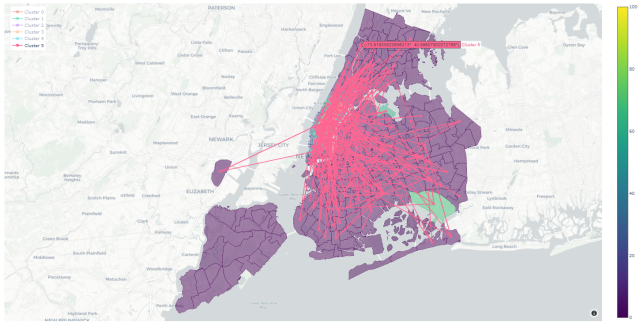


Figura 4.9: Cluster 5

Por ejemplo, en una red múltiplex con 12 capas tendremos 12 características para cada nodo, como podemos ver en la Figura 4.11.

Como siguiente paso, decidimos emplear un algoritmo de aprendizaje automático supervisado para predecir las tendencias futuras en los próximos seis meses. Después de analizar varios algoritmos para utilizar como regresor, elegimos Random Forest [129], [130], [131] porque nos ofrece la mejor precisión para nuestra investigación. El Random Forest es un algoritmo altamente utilizado en muchos casos de uso por su rapidez de cómputo y su precisión.

Utilizamos como conjunto de datos las 6 características obtenidas con el grado de red multiplexada de comportamiento temporal con 6 capas.

Una de las ventajas de este enfoque es que las características extraídas de las redes nos permiten transformar 44 millones de viajes en taxi en una predicción analítica donde las características analizadas son los atributos de los grafos temporales, proporcionando nuevas características para la predicción inmobiliaria.

Para validar nuestra idea, dividimos el conjunto de datos en dos partes:

- Dataset de entrenamiento: contiene 70 % de las zonas de taxis de NY.
- Dataset de test: contiene 30 % de las zonas de taxis de NY.

Con estos dos conjuntos de datos, vamos a validar la precisión de nuestro modelo de previsión. Intentamos crear un regresor para pronosticar la tendencia a 6 meses vista.

	1	2	3	4	5	6	7	8	9	10	11	12	taxi_zone	zip	trend
236	52	7	9	21	4	3	0	84	175	6	1	4	237	10065.0	-62215.085714
237	44	6	9	14	10	2	0	75	171	2	2	1	239	10024.0	-20472.200000
238	4	0	6	0	0	0	0	0	151	0	0	0	240	10470.0	2818.971429
239	27	3	4	15	2	1	0	37	191	0	2	5	243	10040.0	-4584.571429
240	17	0	10	0	1	0	0	2	182	0	1	0	252	11357.0	-2924.857143
241	40	7	5	8	49	0	0	20	160	2	21	1	255	11249.0	-12032.200000
242	15	0	11	2	0	1	0	2	183	1	0	1	259	10466.0	2059.600000
243	42	12	12	11	8	2	0	66	179	4	0	5	261	10006.0	-9012.485714
244	33	9	3	19	7	1	0	64	168	3	4	2	262	10128.0	-29596.771429

Figura 4.10: Zonas de taxi con 12 capas

Random Forest es un algoritmo de aprendizaje automático supervisado que utiliza un método de conjunto para realizar previsiones. Random Forest se basa en el siguiente proceso, representado en la figura **??**

- Tomar k elementos del conjunto de datos de entrenamiento.
- Crear un árbol de decisiones a partir de ellos.
- Crear tantos como el número del estimador se solicita.

Para utilizar el bosque aleatorio, lo primero es encontrar los mejores hiperparámetros del regresor del bosque aleatorio. Nos centramos sólo en el más relevante: el número de estimadores. El número de estimadores define el número de árboles del bosque.

Como podemos ver en la Figura 4.12, el mejor número de estimadores es alrededor de 100.

Existen otros parámetros de entrada para el regresor de Random Forest. Algunos de ellos son:

- criterion: para medir la calidad de una división.
- max_depth: profundidad máxima del árbol.
- max_leaf_nodes and min_leaf_nodes.

4.5.7. Resultados

Nuestro último paso es la evaluación del enfoque propuesto para validar que la red de multiplexación del comportamiento temporal puede recoger características temporales complejas en su interior. Para conseguirlo, tenemos que validar la precisión

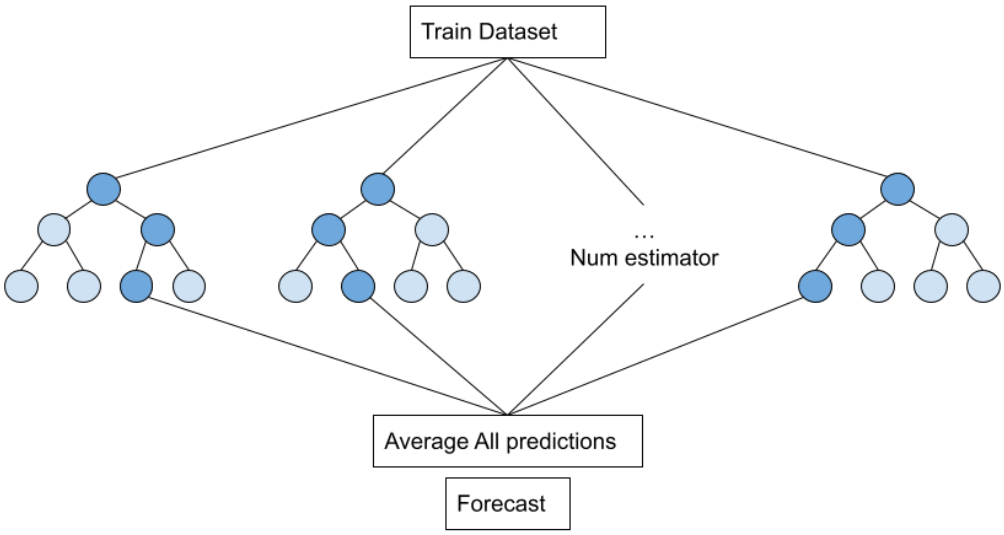


Figura 4.11: Random Forest

del Random Forest Regressor. Hay múltiples métricas que se pueden utilizar para este propósito. Entre ellas, podemos destacar:

- (MSE) – Error cuadrático medio.
- (RMSE) -raíz cuadrada del Error cuadrático medio.
- (MAE) - Error medio absoluto.
- (MSPE) – Porcentaje de error cuadrático medio.
- (MAPE) – Porcentaje medio de error absoluto.

Todas ellas tratan de evaluar la diferencia entre el valor propuesto y el valor real. La principal diferencia entre ellos es la forma de recoger todas estas diferencias, es decir, la forma de sumarlas.

Finalmente, hemos propuesto el MAPE para nuestro propósito. Nuestra principal razón es su capacidad para proporcionarnos una forma fácil e intuitiva de visualizar la precisión de los valores pronosticados. El MAPE puede calcularse como la media de los MAE (errores porcentuales absolutos) de las previsiones. El error se define como el resultado del valor observado menos el valor previsto. Los errores porcentuales se suman con el signo para calcular el MAPE. Esta medida es muy fácil de entender. La razón es que proporciona el error en términos de porcentajes.

$$MAPE = \frac{100}{n} \sum_{t=1}^{n} \frac{A_t + F_t}{A_t}.$$

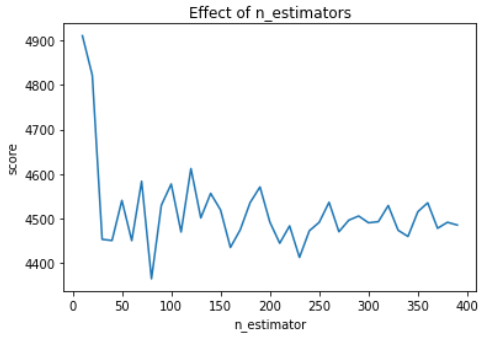


Figura 4.12: Score evolution by the number of estimators

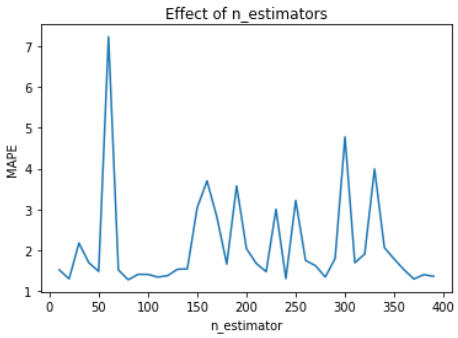


Figura 4.13: MAPE Evolution by the number of estimators

donde A_t es el valor real y F_t es el valor previsto.

La biblioteca de Python sklearn, nos da este valor. Sin embargo, la salida no es un porcentaje en el rango [0, 100]. Además, la salida puede ser arbitrariamente alta como en este caso. Cuanto más pequeño, mejor. Como podemos ver en la figura para un rango entre 1 y 400 el número de estimadores en el Regressor Random Forest.

4.6. Uso de grafos de visibilidad

En este trabajo, proponemos una nueva forma de crear una red multiplexada de comportamiento temporal y utilizarla para predecir cualquier caso de uso basado en el tiempo, como en los casos publicados previamente en los capítulos anteriores.

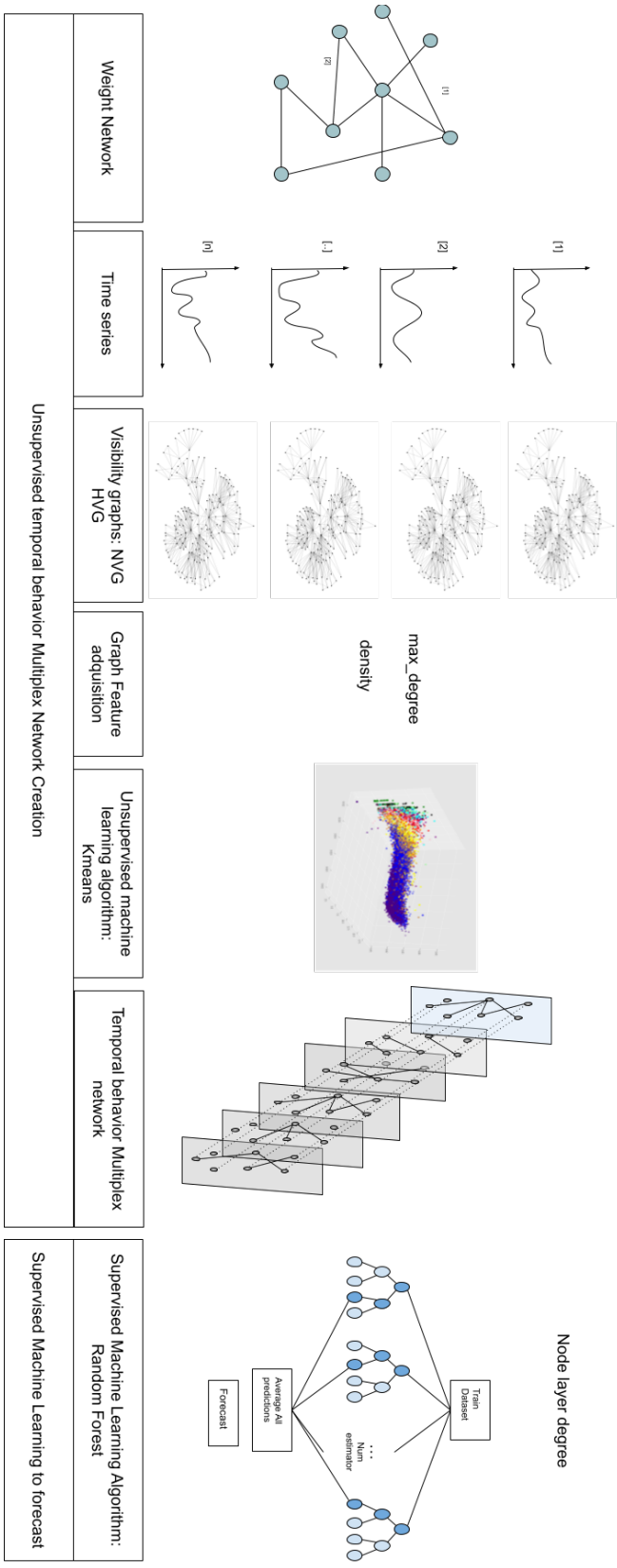


Figura 4.14: Flujo de reducción de dimensionalidad con grafos temporales

Esta solución consta de dos partes:

- La creación de la red múltiplex mediante una solución no supervisada.
- La utilización de una técnica de aprendizaje automático supervisado para proporcionar la solución a la pregunta del caso de uso. Este algoritmo emplea las características del gráfico múltiplex creado en la fase anterior.

Esta propuesta tiene ventajas debido a la combinación de varias técnicas que, normalmente, no suelen combinarse. Posibilita crear una estructura dentro del grafo que permite almacenar atributos temporales complejos y que se utilizan en la segunda fase para proporcionar atributos complejos que de otro modo no podrían generarse. Este nuevo enfoque se basa en el uso de gráficos de visibilidad en lugar de algoritmos de series temporales no supervisados para obtener las características de las series temporales para agrupar los bordes en las capas de la red multiplexada.

Como primer paso, tenemos que definir una red ponderada como red de una sola capa definida de la forma

$$G = (N, E, W)$$

donde N es el conjunto de nodos $\mathcal{N} = \{n_1, n_2, .., n_i\}$ y E el conjunto de aristas que conectan los nodos como $\mathcal{N} = \{n_1, n_2, .., n_i\}$ y W se define como el peso de cada arista basado en una función discreta basada en el tiempo $W = \{w_1, w_2, .., w_i\}$.

Como aproximación temporal, cada arista $E \in e$ tiene una función discreta diferente basada en t $W_t = \{w_{1(t)}, w_{2(t)}, .., w_{i(t)}\}$

Esta función discreta tiene tres características a evaluar:

- Start time: primer periodo de tiempo analizado en el estudio.
- Finish time: último periodo de tiempo analizado en el estudio
- Frecuencia: período mínimo de tiempo para analizar. Típicamente, esta frecuencia se define como diaria, semanal, anual,...

Con estas definiciones podemos describir el peso de cada arista como

$$w_(t) = \bigcup_{i=starttime}^{finishtime} \sum_{j=0}^{n} numero\_eventos$$

es decir, cada arista será una serie de elementos de longitud número de periodos de tiempo dentro de la referencia temporal entre la hora de inicio y la hora de finalización. Por ejemplo, en el caso que estamos analizando definimos la hora de inicio como 01/01/2019, la hora de finalización como 30/06/2019 y la frecuencia como diaria. Con estos valores, nuestra serie temporal contendrá 181 elementos. Cada elemento es la suma de ocurrencias en la franja horaria analizada como podemos ver en la Figura 4.15.

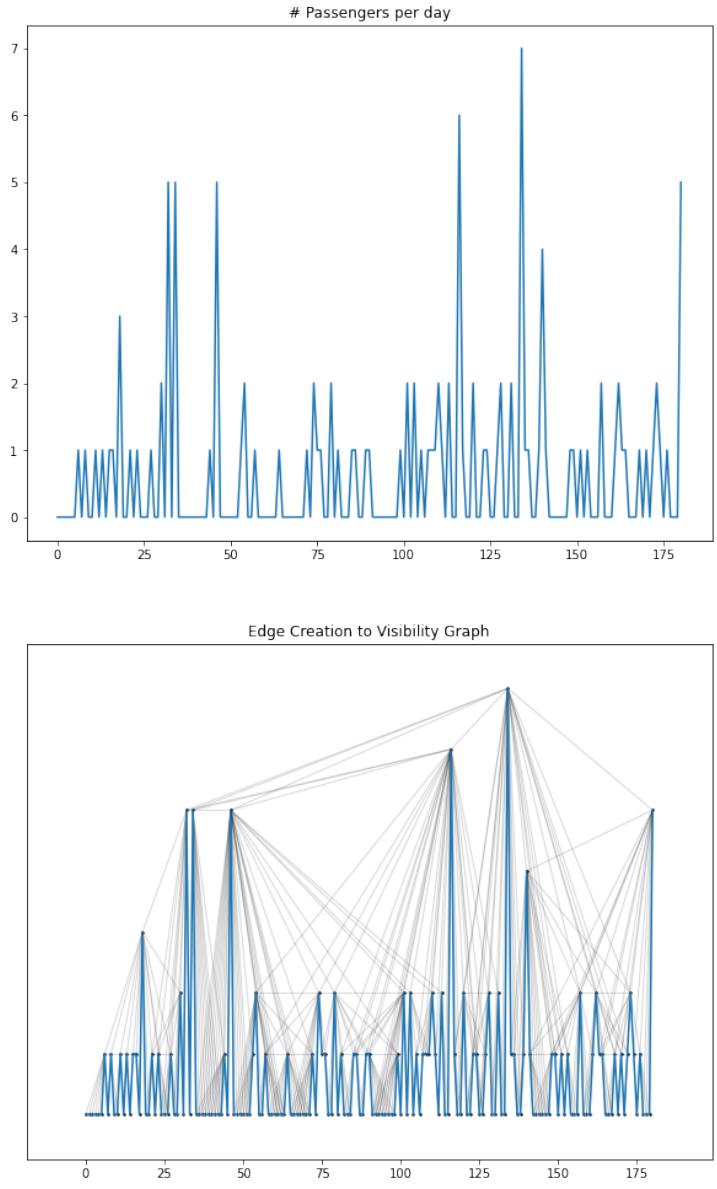


Figura 4.15: Conversión de series temporales a un grafo de visibildad

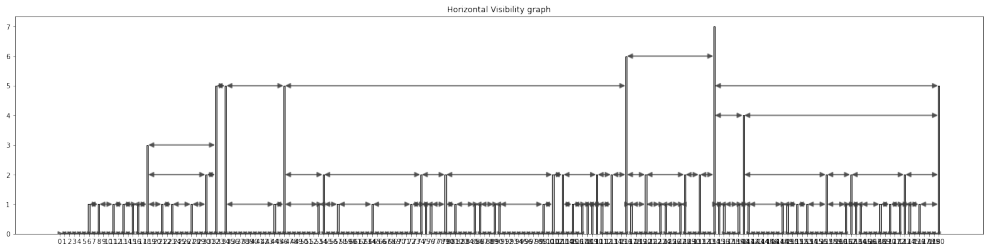


Figura 4.16: Conversión de series temporales a grafo de visibilidad

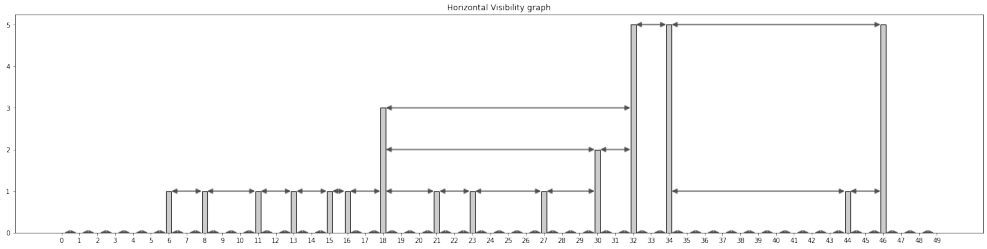


Figura 4.17: Zoom del grafo de visibilidad de la conversión de una serie temporal en los primeros 50 días de la serie

4.6.1. Clasificación con grafos de visibilidad

En este nuevo enfoque, utilizamos otra estrategia. Intentamos mapear cada serie temporal como un gráfico de visibilidad y un gráfico de visibilidad horizontal. Cada registro de la serie temporal se define como un nodo y se conecta a todos los nodos que son visibles desde él. Bajo la línea recta, todos los nodos están conectados. Sin embargo, si cualquier registro entre ellos es mayor que ellos, los nodos no son visibles entre ellos, entonces no aparecerá ninguna arista entre ellos como podemos ver en la Figura 4.17.

En esta figura, nos centramos en los pasajeros entre las zonas de taxi 137 y 112 de Nueva York. Cada registro es el número de pasajeros entre la zona de taxis por día, comenzando el 01/01/2019 y terminando el 30/06/2019. Este enfoque nos da 181 registros como serie temporal.

Al igual que el gráfico anterior, este nuevo enfoque se basa también en la visibilidad de los nodos. Como podemos ver en las Figuras 4.19 y 4.17 la base de la creación del gráfico es la misma: cada registro de la serie temporal se define como un nodo en el gráfico de visibilidad. Dos nodos definidos como t_1, y_1 y t_2, y_2 están conectados si los dos nodos están conectados horizontalmente, es decir, podemos trazar una línea entre ellos sin que ninguna otra altura de registro limite su visibilidad.

$$y_1, y_2 > y_n \quad \text{para todo n donde } 1 < n < 2$$

4.6.2. Clasificación de aristas

En este momento, una red de pesos donde cada arista tiene dos gráficos de red: Un gráfico de visibilidad natural y un gráfico horizontal. Estos dos gráficos nos dan la información necesaria sobre el comportamiento temporal de la red.

Para crear una clasificación de aristas tomamos dos características para cada gráfico:

- maximo grado Definimos el grado de la red como el número de aristas adyacentes al nodo. Si definimos un grafo como un conjunto de nodos $\mathcal{N} = \{n_1, n_2, .., n_i\}$ y E el conjunto de aristas que conectan los nodos como $\mathcal{E} = \{e_1, e_2, .., e_i\}$ y con una matriz de adyacencia A podemos definir el grado máximo del grafo como
$$maxgrado = \text{máx} \sum_j A_{ij}$$
- densidad: Podemos definir el valor de la densidad como 0 cuando no hay aristas en el gráfico. Por otro lado, el valor es igual a 1 si estamos describiendo un gráfico completo.
$$densidad = \frac{2m}{n(n-1)}$$
donde n es el número de nodos y m es el número de aristas en G.

Para cada arista, tenemos 4 características:

- Máximo grado del grafo de visibilidad natural.
- Densidad del grafo de visibilidad natural.
- Máximo grado del grafo de visibilidad horizontal.
- Densidad del grafo de visibilidad horizontal.

Después de partir de las series temporales creadas con los eventos a lo largo del tiempo entre los nodos, transformarlas en ambos gráficos de visibilidad y obtener las dos características de cada uno de los gráficos de visibilidad, estamos preparados para la agrupación de estas series temporales en grupos semejantes, que nos darán atributos temporales similares en cada uno de ellos.

Como no sabemos cuál es la mejor forma de agruparlas, decidimos utilizar un algoritmo de aprendizaje automático no supervisado para encontrar la mejor forma de agruparlas. Uno de los algoritmos más famosos para este propósito es el k-means.

K-means es un algoritmo de clasificación no supervisado que clasifica los objetos en k grupos en función de sus características. La agrupación se realiza minimizando la suma de las distancias entre cada objeto y el centroide de su grupo o cluster. Normalmente se utiliza la distancia cuadrática.

El algoritmo consta de tres pasos:

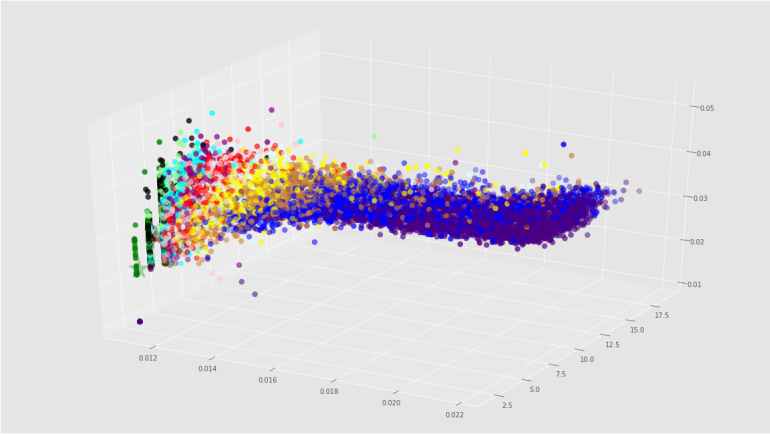


Figura 4.18: Clasificación de las series temporales usando grafos de visibilidad

- Inicialización: una vez elegido el número de clusters, k, se establecen k centroides en el espacio de datos, por ejemplo, eligiéndolos al azar.
- Asignación de objetos a los centroides: cada objeto de los datos se asigna a su centroide más cercano.
- Actualización de los centroides: la posición del centroide de cada grupo se actualiza tomando como nuevo centroide la posición de la media de los objetos pertenecientes a ese grupo.

Los pasos 2 y 3 se repiten hasta que los centroides no se mueven, o se mueven por debajo de un umbral de distancia en cada paso. El algoritmo k-means resuelve un problema de optimización, siendo la función a optimizar (minimizar) la suma de las distancias cuadráticas de cada objeto al centroide de su cluster. Los objetos están representados por vectores reales d-dimensionales $x_1, x_2, .., x_i$ y el algoritmo k-means construye k grupos donde la suma de las distancias de los objetos, dentro de cada grupo $G = G_1, G_2, .., G_i$ a su centroide. El problema se puede formular como sigue:

$$\min_{G} E(\mu_i) = \min_{G} \sum_{i=1}^{k} \sum_{x_j \in G_i} \|x_j - \mu_i\|^2$$

donde G es el conjunto de datos cuyos elementos son los objetos x_j representados por vectores, donde cada uno de sus elementos representa una característica o atributo. Tendremos k grupos o clusters con su correspondiente centroide μ_i . En cada actualización del centroide, desde el punto de vista matemático, imponemos la condición de extremo necesario en la función $G(\mu_i)$ que, para la función cuadrática anterior es

$$\frac{\partial E}{\partial \mu_i} = 0 \Rightarrow \mu^{(t+1)} = \frac{1}{\mid G_i^{(t)} \mid} \sum_{x_j \in G_i^{(t)}} x_j$$

y la media de los elementos de cada grupo se toma como nuevo centroide.

Las principales ventajas del método k-means son que es un método sencillo y rápido. Pero es necesario decidir el valor de k y el resultado final depende de la inicialización de los centroides.

Con este algoritmo, podemos asignar a cada serie temporal el mejor clúster y, por tanto, agrupar los bordes en varios clústeres.

Cuantos más clusters, más información almacenaremos sobre las características temporales, ya que haremos una división más precisa de los comportamientos.

4.6.3. Grafo Multiplex con atributos temporales

Hasta este punto, hemos definido un grafo de pesos donde las aristas son funciones discretas basadas en el tiempo. Estas funciones pueden describirse como series temporales con las que hemos creado dos gráficos: Gráfico de visibilidad natural y Gráfico de visibilidad horizontal. De estos gráficos, tomamos dos características de cada uno: el grado máximo y la densidad de las redes.

Con estas características, finalmente, hemos asignado a cada serie temporal un cluster que agrupa las series temporales con las mismas características mediante un sistema de clasificación no supervisado (k-means).

En esta última fase, vamos a transformar nuestro grafo ponderado en un grafo multiplexado con la información que hemos obtenido de los pasos anteriores. Como hemos comentado al principio, los grafos son la herramienta adecuada para agrupar mucha información dentro de ellos. Sin embargo, hasta la fecha, la recopilación de características temporales dentro de él ha sido bastante limitada.

En este caso, desplegaremos nuestra red de pesos en N capas donde las capas $N = (n_1....n_n)$ en $N \in k-means(n)$. En cada una de estas N capas incluiremos las aristas que forman parte del mismo cluster no supervisado realizado en pasos anteriores.

Finalmente obtenemos una red multiplexada ponderada y dirigida [76] G, con

$$n$$

capas

$$L = \{\ell_\alpha;\ \ \alpha \in \{1, ..., n\}\}$$

en un conjunto de nodos.

$$T = \{1, ..., N\}$$

Cada capa se define como un grafo dirigido ponderado

$$\ell_\alpha = (T_\alpha, V_\alpha)$$

sobre el mismo grupo de nodos

$$T_\alpha \in T$$

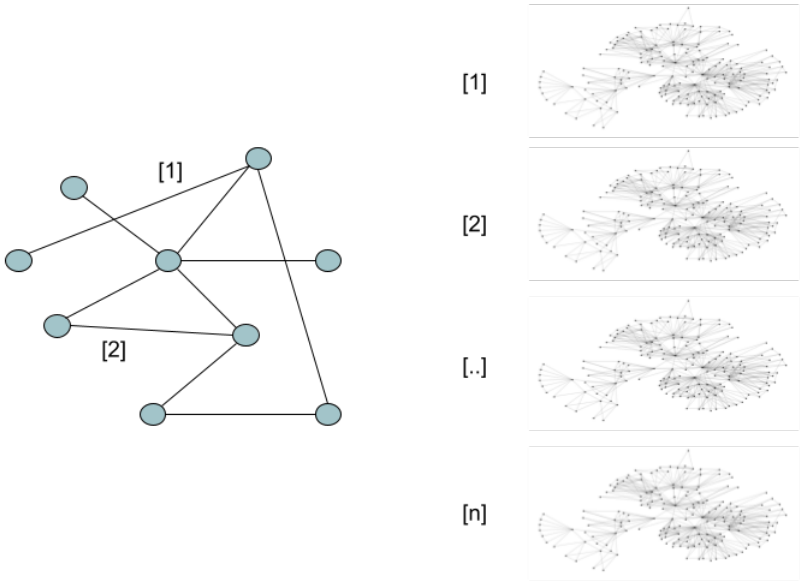


Figura 4.19: Translación de cada arista en un grafo de visibilidad

y con un conjunto de aristas:

$$V_\alpha = \{v_{i,j}^\alpha; \alpha \in \{1, ..., n\}\},$$

donde $v_{i,j}^\alpha$ representa el enlace que conecta los nodos x e y en ℓ_α, y w_{alpha} es una función $w_\alpha : V_\alpha \longrightarrow [0, +\infty)$ tal que para cada arista $e_{i,j}^{alpha} enE_{alpha}$, el coeficiente $w_\alpha(v_{x,y}^\alpha)$ se llama *peso* de $v_{x,y}^\alpha$.

4.6.4. Previsión mediante Random Forest

Después de crear el grafo de comportamiento temporal múltiple, utilizaremos toda la información dentro del grafo para extraer características complejas sobre la evolución temporal de las interacciones de los nodos.

Todas las características obtenidas del grafo son interpretaciones complejas de las interacciones entre los nodos de forma condensada en una pequeña cantidad de información.

La mejor cuestión es encontrar cuáles son las características óptimas para extraer de la red multiplexada de comportamiento temporal. Después de varios experimentos, deducimos que hay que seleccionar de cualquier nodo el grado del nodo en cada capa de la red. Con este enfoque, cuantas más capas tenga la red, más características podremos obtener para la realización de las siguientes fases.

Con la información extraída sobre los nodos, podemos utilizar cualquier algoritmo de aprendizaje automático supervisado para predecir cualquier característica sobre los nodos. En nuestro caso, podemos hacer una previsión a 6 meses vista con la evolución de los últimos 6 meses.

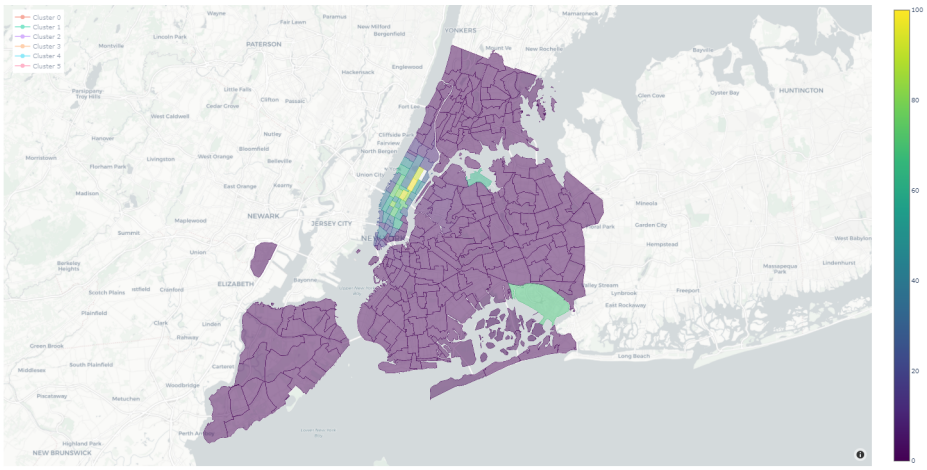


Figura 4.20: Distribution de las zonas de taxis de Nueva York y su concentración

Para ello, tras varias investigaciones con diferentes algoritmos de aprendizaje automático, decidimos utilizar Random Forest. Este algoritmo es rápido y muy preciso en casi todos los casos.

4.6.5. Resultados

Para evaluar la nueva propuesta, utilizamos el mismo caso de uso que el capítulo anterior y comparamos la precisión y el tiempo de proceso entre ellos. Nuestro reto es predecir la tendencia de los precios de los inmuebles con 6 meses de antelación en la ciudad de Nueva York, utilizando los viajes en taxi de 6 meses. Nuestro objetivo es utilizar el comportamiento temporal de los viajes de los pasajeros alrededor de Nueva York para encontrar si tiene alguna capacidad de previsión sobre la tendencia de los precios inmobiliarios.

Este nuevo enfoque para crear redes multiplexadas de comportamiento temporal nos permite reducir el tiempo de proceso de los enfoques anteriores, en los que la agrupación no supervisada de series temporales se realizaba mediante el algoritmo k-shape.

Fuente de datos y preparación

Utilizando las mismas técnicas que en el trabajo anterior se utilizarán dos fuentes de datos para fusionar ambas informaciones: los precios de los inmuebles y los viajes en taxi por la ciudad de Nueva York.

- La primera fuente de datos son los viajes en taxi de NYC durante 6 meses (01/01/2019 - 30/06/2019) [127]. Esta fuente de datos contiene 44.459.136 viajes en este periodo, lo que nos da información adecuada sobre el lugar de recogida, el número de pasajeros, el lugar de destino, etc. Toda la información

sobre los viajes en taxi se representa utilizando las zonas de taxi en Nueva York

- La segunda fuente de datos consiste en el precio de los inmuebles en Nueva York durante los próximos 6 meses (1/07/2019- 31/12/2019). Esta información se proporciona por código postal.

El primer paso es ajustar las zonas de taxi y la brecha de los códigos postales. Para ello, utilizamos el algoritmo de Shapley para encontrar el centroide de los códigos postales, para localizar cada código postal dentro de un código postal. Con este enfoque podemos comparar ambas informaciones utilizando un mismo espacio geográfico.

4.6.6. Preparación de las series temporales

Para crear series temporales, necesitamos tres parámetros para definirlas:

- Fecha de inicio: 01/01/2019.
- Fecha de finalización: 30/06/2019.
- Frecuencia: diaria.

Una vez que los datos están listos, se crean series temporales que describen el número de pasajeros que transitan entre dos zonas de taxis por día. La longitud de cada una es de 181 registros; cada registro es la suma de pasajeros entre dos zonas de taxi. En el caso de que no haya pasajeros entre dos zonas de taxi durante un día, se determina que el valor es 0.

Después de procesar todos los viajes, tenemos finalmente 45.952 series temporales que representan las combinaciones de zonas de taxi entre las que hay algún viaje.

Uso de los grafos de visibilidad y clasificación de tiempo

Esta sección es la característica distintiva de esta nueva técnica. En estudios anteriores para la creación de la red de multiplexación del comportamiento temporal se utilizó un sistema de clasificación no supervisado llamado k-shape. Este algoritmo se basa en la comparación de las sombras de las series temporales y su agrupación por ellas.

En cambio, en este nuevo trabajo, hemos propuesto la transformación de las series temporales en redes de visibilidad tal y como se describe en la propuesta. Para ello, hemos creado la red de visibilidad natural y horizontal para cada una de las 45.592 series temporales. Esto significa que hemos creado 91.904 redes de visibilidad que describen los trayectos en taxi entre zonas de taxi.

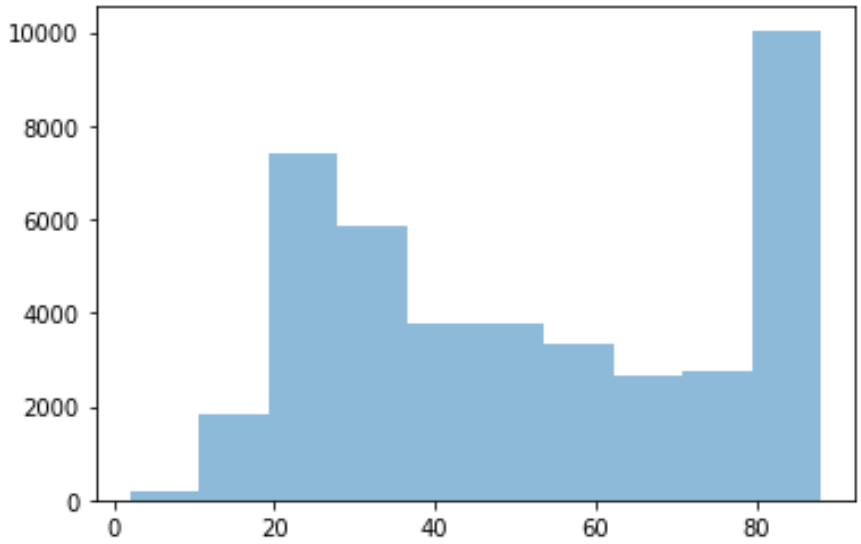


Figura 4.21: Distribución del grado máximo de los grafos naturales visibilidad

A partir de estas redes, tal y como se propone en el apartado anterior, hemos obtenido el grado y la densidad máxima de cada una de ellas.

De la generación de estos atributos para cada una de las redes se pueden extraer algunas conclusiones. En primer lugar, se observa que el grado máximo de las redes naturales de la figura 4.21 (entre 20-60) es mucho mayor que el de la red horizontal de la Figura 4.23 (<12), tal y como se menciona en la descripción de ambas redes. Lo mismo ocurre con la densidad de ambos tipos de redes. La densidad de las redes naturales figura 4.22 es mayor que la de las redes horizontales Figura 4.24.

La otra cosa que hay que mencionar es que el tiempo de construcción de las redes y su clustering no aumenta en función del número de clusters que queramos hacer. En el caso del algoritmo k-shape, el tiempo de cálculo aumenta en función del número de clusters que queramos hacer. En cambio, en el caso de la extracción de los atributos de las redes de visibilidad, el tiempo de creación de los clusters es independiente del número de capas o clusters que se especifiquen. Esta evolución se puede ver en la figura 4.25.

Uso de los atributos para la predicción de la tendencia

En los capítulos anteriores, se demostró que el aumento del número de capas de la red aumenta la precisión de los modelos algorítmicos. En nuestro caso, esta mejora también se refleja en incrementos de más del 20 % en la precisión de los modelos al pasar de 6 capas a 18. El uso de redes de visibilidad no reduce la precisión del modelo, sino que reduce los tiempos de procesamiento y, por tanto, permite escalar este tipo de redes multiplexadas a un nivel superior.

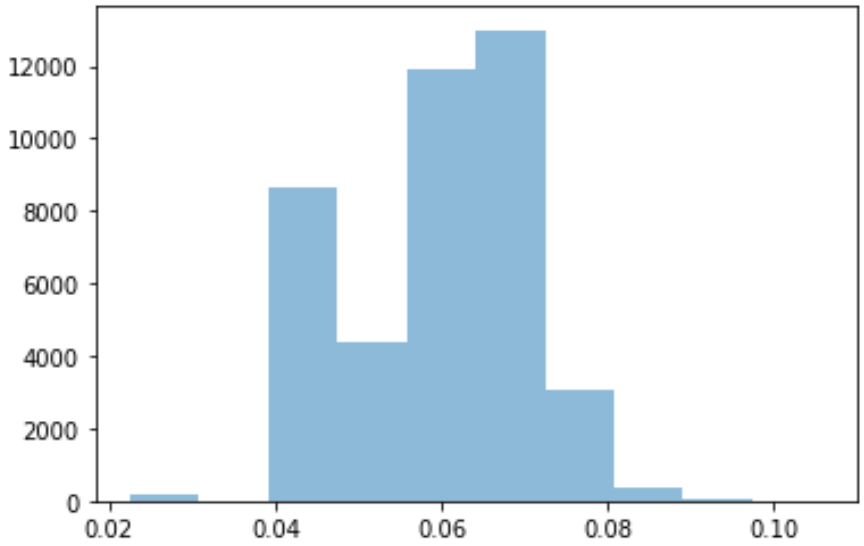


Figura 4.22: Distribución de la densidad de los grafos naturales visibilidad

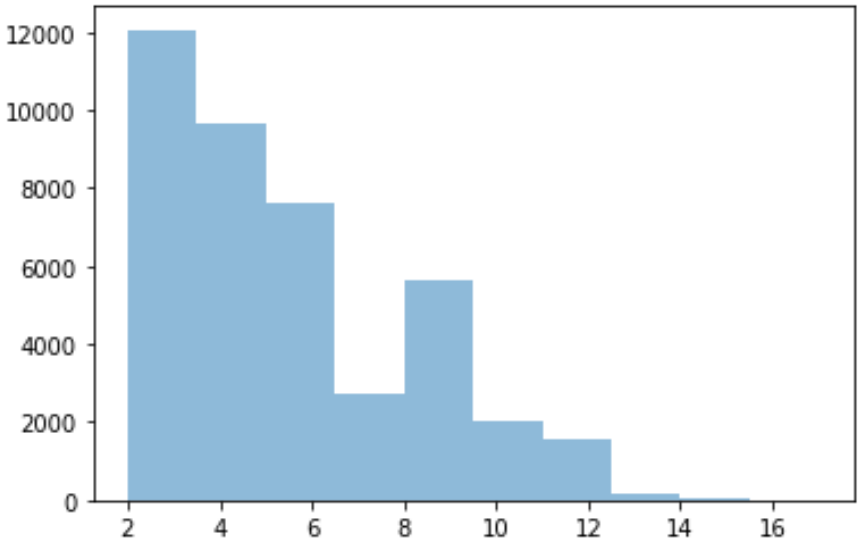


Figura 4.23: Distribución del grado máximo de los grafos horizontales de visibilidad

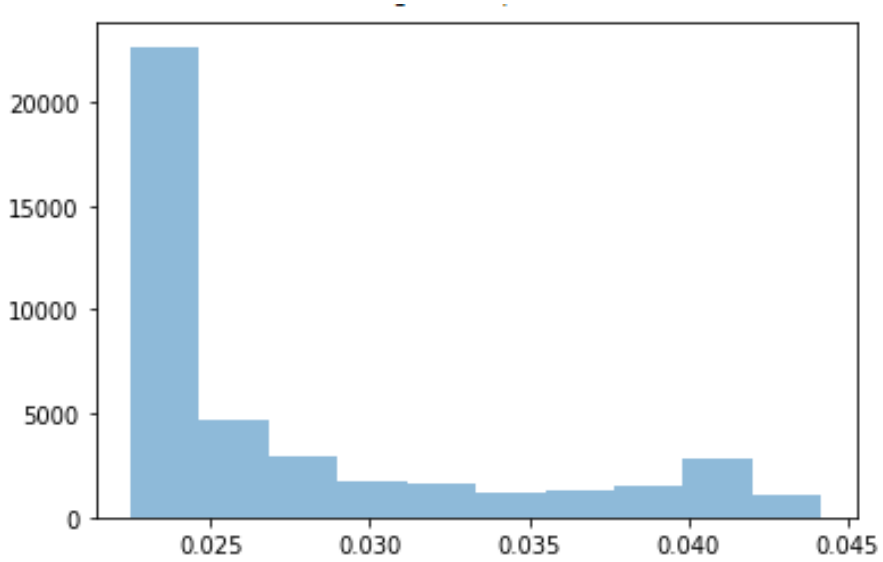


Figura 4.24: Distribución de la densidad de los grafos horizontales de visibilidad

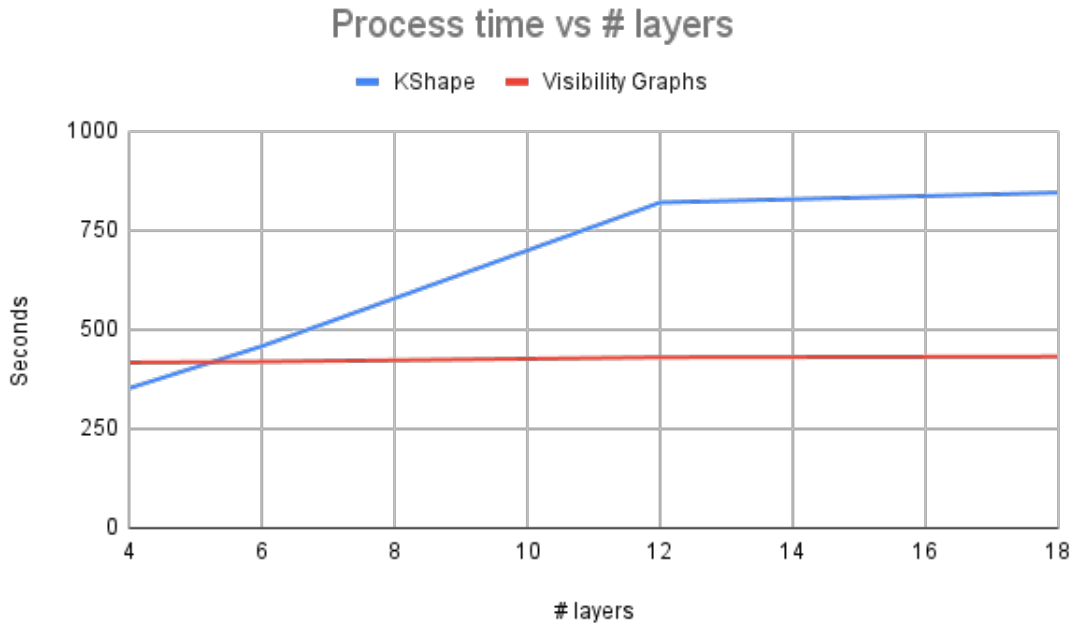


Figura 4.25: Tiempo de proceso vs número de capas

4.7. Discusión

Al igual que las investigaciones anteriores, en este trabajo nos centramos en las ventajas del uso de redes multiplexadas de comportamiento temporal para traducir la complejidad de la vida real en una red compleja. En esta investigación somos capaces de reducir los viajes en taxi en 6 meses en una red multiplexada en la que se incluye toda la información temporal sobre la relación entre las zonas de taxi. En esta investigación podemos reducir 44 millones de viajes en taxi en un pequeño conjunto de datos que describe cada zona de taxis. Además, la creación de las capas en función del comportamiento temporal nos proporciona un enfoque nuevo e ingenuo para crear redes.

El resultado final de la investigación valida la tesis inicial y proporciona una previsión a seis meses para el sector inmobiliario en la ciudad de Nueva York, reduciendo el modelo algorítmico y aumentando la capacidad de agrupar la información mediante una red multiplexada de comportamiento temporal. Así, hemos reducido 44 millones de viajes en taxi a 12 atributos por zona de taxi. Este nuevo tipo de agrupación de la información puede reducir el tiempo del proceso en futuros casos en los que el esfuerzo computacional no nos permita analizar los datos.

La nueva metodología descrita ofrece dos características distintivas:

- Reducción del número de eventos de viajes en taxi a características de zonas de taxi. Reducimos 44 millones de viajes en taxi a 12 características por zona de taxi.
- Reducción de las capacidades computacionales para fusionar toda la información. Utilizando una red de multiplexación del comportamiento temporal, podemos optimizar las capacidades computacionales para fusionar estos 44 millones de eventos.

La validación de la hipótesis inicial nos anima a buscar nuevos campos de investigación en los que la agrupación temporal de datos sobre relaciones entre entidades pueda ser diferencial.

A partir de este trabajo, hay varios estudios nuevos en los que se centrará en el futuro:

- Cambiar nuestras técnicas de agrupación de series temporales no supervisadas basadas en k-shape por nuevas técnicas supervisadas para detectar nuevos comportamientos temporales. Este enfoque nos permitirá comprender mejor la descripción de cada cluster.
- Cambiar las técnicas de señalización por otra técnica para reducir el tiempo de proceso. El algoritmo k-shape aumenta el tiempo de procesamiento a medida que se incluyen más series temporales de forma exponencial. Sería interesante

buscar nuevas formas de reducir el tiempo de procesamiento en entornos muy exigentes.

- Utilizar otras técnicas relacionadas con el uso de series temporales para obtener más información sobre el comportamiento de las relaciones e interacciones entre nodos.
- Utilizar algoritmos de clasificación de nodos dentro de la red multiplexada como alternativa mejorada para obtener nuevas características de la red, en lugar de tomar características y utilizarlas en un algoritmo Random Forest.

Y por último, pero no menos importante, dado que la cantidad de datos recogidos en las ciudades hoy en día es enorme, un reto significativo en el que nos gustaría utilizar el tipo de red multiplex de comportamiento temporal definido en este trabajo sería en el contexto de un análisis profundo del comportamiento de las ciudades inteligentes. Esta cantidad de información puede darnos un nuevo enfoque para utilizarla en diversos retos diferentes no considerados hasta ahora. La granularidad real de los datos nos da la oportunidad de evaluar el comportamiento temporal de los activos dentro de la ciudad.

Como en anteriores capítulos, este primer caso de uso, nos abre múltiples capacidades para realizar nuevas investigaciones y desarrollos en las diferentes capacidades realizadas.

En este capítulo hemos considerado la representación de las series temporales como un grafo, y hemos estudiado sus características que representan las características de temporalidad de las relaciones entre los nodos.

Con todo ello, hemos abierto la puerta a nuevas capacidades de representación de las series temporales con grafos, transformadas de fourier, u otra posible representación que nos permita clasificarlas de forma óptima y rápida.

Este capítulo, nos ofrece la posibilidad de poder analizar nuevas formas de realización de las clasificaciones temporales de las series con metodologías diferentes que se enfoquen en la rapidez de la solución.

5. DISCUSIÓN GENERAL

Los sistemas de adquisición de información en los últimos años nos han permitido crear simulaciones de la realidad incluyendo cada vez un número mayor de datos. Toda esta nueva información permite analizar interacciones en la realidad que anteriormente no podíamos medir. Sin embargo, esta situación crea un nuevo reto para que dicha información sea manejada obteniendo un valor computacionalmente asequible. Para este fin, la representación de los datos a través de grafos donde se muestre la evolución de las interacciones entre cada una de las entidades que representan la realidad puede ser una solución. Para ello, en los últimos años la utilización de grafos ha permitido agrupar las conexiones entre grandes cantidades de datos de una forma más precisa aumentando la información sobre las relaciones entre las entidades de la realidad.

Sin embargo, estas capacidades de representación de la realidad se enfrentan a la condición de proporcionar atributos temporales de los grafos. Hasta la fecha se han definido muchos grafos temporales que representaban la evolución de nodos y aristas a lo largo del tiempo. Sin embargo, no simplificaban el número de interacciones que se realizaban entre cada uno de los nodos, creando aristas para cada acción entre nodos.

Esta tesis se centra en poder disminuir todas las interacciones entre los nodos a lo largo del tiempo, en nuevos atributos temporales dentro del grafo. De esta forma podemos reducir millones de interacciones entre los nodos que crearían diferentes aristas en cada interacción, en un grafo multiplex que agrega toda esta información y crea diferentes capas de relación entre los nodos.

Se propone el análisis del comportamiento de los grafos a lo largo del tiempo en un período determinado y con una frecuencia de muestreo definida. Este nuevo análisis permite estudiar la evolución del grafo de muchas formas en función del plazo de tiempo que se quiera estudiar. No es lo mismo el estudio de tiempos cortos como días u horas que el estudio de rangos de tiempos más largos como anuales. Esto nos permite la creación de características del grafo en diferentes frecuencias aumentando los atributos temporales que se pueden reflejar en estos nuevos grafos multiplex con características temporales.

Como ejemplo, se presentan dos estudios con principios similares que son la detección de intrusiones en grandes compañías que realiza un estudio horario de la evolución de los comportamientos de los equipos existentes que se realiza de forma

horaria y otro estudio que estudia los movimientos de pasajeros de taxis entre las diferentes zonas de Nueva York que se realiza de forma diaria.

En el primero de los casos, se utiliza como evolución temporal las comunicaciones que existen entre los diferentes equipos. Al comparar el número de bytes que se envían a lo largo del tiempo, podemos agrupar las comunicaciones entre los diferentes equipos en función de la forma de la serie temporal a lo largo del tiempo. El objetivo será que en el mismo grupo residan las comunicaciones que son similares en el tiempo, es decir, todas las navegaciones a internet en el mismo cluster, las interacciones de los administradores de las bases de datos en otro, ... El objetivo es poder conocer cuántas interacciones tiene cada equipo en cada una de las capas temporales. Esta información nos permite detectar equipos con un patrón de conexiones fuera de la normalidad.

En el segundo de los casos, utilizamos la evolución temporal de los viajeros de taxis en Nueva York. Nuestra premisa consiste en que los patrones entre los barrios van a reflejar el tipo de flujos que existe entre ambos, como pueden ser, financieras, lúdicas, sanitarias,... A cada uno de estos patrones, se le asignará un cluster diferente en el análisis temporal, y por tanto, estará en una capa diferente del grafo multiplex. Con esta exploración, disponemos de información sobre el tipo de relaciones que tiene un barrio con el resto. Como ejemplo, podemos apreciar que las zonas financieras reciben tráfico durante la semana, al contrario que zonas lúdicas que reciben el pico de tráfico en los fines de semana. Toda esta investigación nos permite predecir la evolución del precio de los bienes raíces sin conocer datos estáticos ni demográficos de las zonas.

Ambos casos, avalan nuestra primera premisa en la que el grafo era capaz de recopilar información temporal de una nueva forma y por tanto, obtener unos atributos o características complejas que aporten un valor diferencial sobre la evolución de las relaciones entre las entidades.

El tercer capitulo, se centra en posibles evoluciones de la primera definición del grafo multiplex con atributos temporales para reducir los tiempos de computación de la creación del grafo multiplex y mejorar la precisión de los atributos temporales.

5.1. Trabajo futuro

Esta tesis presenta una nueva línea de investigación en el que la combinación de técnicas basadas en grafos y en series temporales aportando nuevos resultados para la resolución de retos altamente complejos. Por ello, puede ser punto de partida de nuevos estudios basados en análisis de diferentes técnicas en ambos ámbitos: series temporales y grafos.

Las líneas de investigación futuras que pueden iniciarse desde este estudio residen en dos bases principales:

- Reducción de tiempos de procesamiento y creación de características temporales más complejas: en esta línea nos podemos centrar en la detección de nuevas técnicas o combinación de ellas que permita la reducción de capacidades de cómputos o tiempos de ejecución con los algoritmos propuestos anteriormente. Dentro de este punto podemos centrarnos en la creación de un framework de programación o la búsqueda de novedosas técnicas más eficientes.
- Utilización de otras capacidades de las series temporales para obtener atributos dentro del grafo multiplex. Dentro de las series temporales podemos centrarnos en otras capacidades analíticas como pueden ser la predicción o la detección de anomalías como nuevo campo de estudio.

De estas dos líneas de trabajo se proponen estos nuevos aspectos a desarrollar:

- Estudio de las anomalías en las series temporales para la reducción de número de aristas a aquellas que dispongan de anomalías en el rango temporal estudiado. De forma similar a la utilización de métodos de clasificación de las series temporales, se puede analizar la utilización de detección de anomalías en las series temporales para obtener atributos dentro del grafo multiplex.
- Otra de las capacidades de la representación de las interacciones de los nodos a través de series temporales es la utilización de las capacidades de predicción de las series temporales que permitan analizar las futuras interacciones del grafo en el futuro, pudiendo aportar las relaciones entre los nodos en el futuro.
- Creación de un framework de creación de características temporales de los grafos: la creación de una librería o capacidades analíticas automáticas nos permitiría la generación de cada vez más atributos de forma automática que podrían ampliar las capacidades de estudio de la tesis actual con más características a estudiar en las series temporales que conforman las aristas. En este

sentido se podrían aumentar los estudios temporales de las series temporales analizadas. En la actualidad nos hemos centrado en la segmentación de las series temporales, pero también pueden utilizarse otras técnicas de series temporales como son la detección de anomalías y la predicción de futuros valores de las series para aportar más características al grafo multiplex que agrupe toda esta información temporal descrita.

6. CONCLUSIONES GENERALES

Para finalizar esta memoria se presentan unas conclusiones generales finales sobre los principales resultados que se han encontrado durante el período de investigación en el que se ha realizado esta tesis doctoral:

- La utilización de dos tecnologías que hasta ahora parecían aisladas y no complementarias, pueden ser utilizadas en conjunto para poder aportar un valor diferencial. Durante los capítulos de esta tesis se presentan varios casos de uso en los que la utilización de grafos multiplex con características temporales aporta un valor para la agregación de un elevado conjunto de eventos en una red compleja desde la que se pueden obtener características complejas para la descripción de cada uno de los nodos. En cada uno de ellos se toma como punto de partida un conjunto de millones de eventos en los que se describen las relaciones entre las diferentes entidades de las que se pretende analizar su evolución, y lo sustraen a un conjunto muy reducido de atributos de cada una de estas entidades que agrupan los atributos temporales que describen la evolución de cada uno.
- La selección del número de capas en el grafo multiplex con características temporales es vital para la adquisición de atributos temporales dentro de dicho grafo. El incremento del número de capas en las que se segmenta el grafo multiplex permite la inclusión de más información temporal dentro del grafo.
- La utilización de segmentaciones no supervisadas basadas en algoritmia de señales permite poder dividir en cada una de las capas según diferentes formas de segmentar. Al existir diversas formas de segmentación por señales como son por modelo, por forma y por características, es posible aportar diferentes preferencias a la segmentación de las señales para representar los casos más óptimos en cada uno de los casos.
- En el caso de la utilización de una clusterización no supervisada basada en la forma de las series temporales, se puede crear un grafo multiplex con características temporales que permite una descripción básica del comportamiento de cada una de las capas. En alguno de los casos de uso, se ha podido describir el comportamiento al visualizar el patrón de cada uno de los clusters. Por poner un ejemplo, gracias a la clusterización a través de forma de los trayectos de los taxis de Nueva York se ha podido inferir cuál es la capa de los viajes financieros y cuál es la de los viajes lúdicos. Los primeros se concentraban dentro

de los días laborables de la semana y los segundos se centraban en los fines de semana. Esta agregación aportaba información inexistente hasta la fecha en cualquier grafo temporal.

- La reducción de dimensionalidad de los datos temporales, también viene asociada por una limitación del proceso de cómputo para la consecución de los objetivos. El poder agrupar toda esta información en un entorno de redes complejas permite la reducción de la complejidad de la solución permitiendo que los procesos computacionales se contengan. Los incrementos computacionales, debido al incremento en millones de eventos, no son apreciables ya que el número de activos se mantiene constante.

BIBLIOGRAFÍA

[1] DataReportal. "Mobile Data Evolution." (2021), [En línea]. Disponible en: https://datareportal.com/reports/digital-2021-global-overview-report (Acceso: 05-04-2022).

[2] blocksandfiles. "Internet Storage." (2021), [En línea]. Disponible en: https://blocksandfiles.com/2020/05/14/idc-disk-drives-will-store-over-half-world-data-in-2024/ (Acceso: 05-04-2022).

[3] K. Erciyes, "Distributed graph algorithms for computer networks," 2013.

[4] S. K. Pal y S. S. Sarma, "Computer Network Topology Design in Limelight of Pascal Graph Property," *arXiv preprint arXiv:1003.5432*, 2010.

[5] F. Riaz y K. M. Ali, "Applications of graph theory in computer science," en *2011 Third International Conference on Computational Intelligence, Communication Systems and Networks*, IEEE, 2011, pp. 142-145.

[6] S. J. Mason, "Feedback theory-some properties of signal flow graphs," *Proceedings of the IRE*, vol. 41, n.º 9, pp. 1144-1156, 1953.

[7] J. Aspnes et al., "A theory of network localization," *IEEE Transactions on Mobile Computing*, vol. 5, n.º 12, pp. 1663-1678, 2006.

[8] P. Holme y J. Saramäki, "Temporal networks," *Physics reports*, vol. 519, n.º 3, pp. 97-125, 2012.

[9] N. Masuda y R. Lambiotte, *A guide to temporal networks*. World Scientific, 2016.

[10] A. Li, S. P. Cornelius, Y.-Y. Liu, L. Wang y A.-L. Barabási, "The fundamental advantages of temporal networks," *Science*, vol. 358, n.º 6366, pp. 1042-1046, 2017.

[11] P. Rozenshtein y A. Gionis, "Mining temporal networks," en *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, 2019, pp. 3225-3226.

[12] P. J. Brockwell y R. A. Davis, *Time series: theory and methods*. Springer Science & Business Media, 2009.

[13] M. Khatami y F. Akbarzadeh, "Algorithms for Segmenting Time Series," *Global Analysis and Discrete Mathematics*, vol. 3, n.º 1, pp. 65-73, 2018.

[14] X. Wang, A. Mueen, H. Ding, G. Trajcevski, P. Scheuermann y E. Keogh, "Experimental comparison of representation methods and distance measures for time series data," *Data Mining and Knowledge Discovery*, vol. 26, n.º 2, pp. 275-309, 2013.

[15] V. Bettaiah y H. S. Ranganath, "An analysis of time series representation methods: data mining applications perspective," en *Proceedings of the 2014 ACM Southeast Regional Conference*, 2014, pp. 1-6.

[16] M. G. Baydogan y G. Runger, "Time series representation and similarity based on local autopatterns," *Data Mining and Knowledge Discovery*, vol. 30, n.º 2, pp. 476-509, 2016.

[17] M. Sifuzzaman, M. R. Islam y M. Ali, "Application of wavelet transform and its advantages compared to Fourier transform," 2009.

[18] S. Daw, C. Finney y E. Tracy, "A Review of Symbolic Analysis of Experimental Data," *Review of Scientific Instruments*, vol. 74, pp. 915-930, feb. de 2003. DOI: 10.1063/1.1531823.

[19] R. L. Davidchack, Y.-C. Lai, E. M. Bollt y M. Dhamala, "Estimating generating partitions of chaotic systems by unstable periodic orbits," *Phys. Rev. E*, vol. 61, pp. 1353-1356, 2 feb. de 2000. DOI: 10.1103/PhysRevE.61.1353. [En línea]. Disponible en: https://link.aps.org/doi/10.1103/PhysRevE.61.1353.

[20] L. Lacasa, B. Luque, F. Ballesteros, J. Luque y J. C. Nuno, "From time series to complex networks: The visibility graph," *Proceedings of the National Academy of Sciences*, vol. 105, n.º 13, pp. 4972-4975, 2008.

[21] B. Zhao, H. Lu, S. Chen, J. Liu y D. Wu, "Convolutional neural networks for time series classification," *Journal of Systems Engineering and Electronics*, vol. 28, n.º 1, pp. 162-169, 2017.

[22] M. Hüsken y P. Stagge, "Recurrent neural networks for time series classification," *Neurocomputing*, vol. 50, pp. 223-235, 2003.

[23] H. Ismail Fawaz, G. Forestier, J. Weber, L. Idoumghar y P.-A. Muller, "Deep learning for time series classification: a review," *Data mining and knowledge discovery*, vol. 33, n.º 4, pp. 917-963, 2019.

[24] Z. Cui, W. Chen e Y. Chen, "Multi-scale convolutional neural networks for time series classification," *arXiv preprint arXiv:1603.06995*, 2016.

[25] M. Christ, N. Braun, J. Neuffer y A. W. Kempa-Liehr, "Time Series FeatuRe Extraction on basis of Scalable Hypothesis tests (tsfresh – A Python package)," *Neurocomputing*, vol. 307, pp. 72-77, 2018. DOI: https://doi.org/10.1016/j.neucom.2018.03.067. [En línea]. Disponible en: https://www.sciencedirect.com/science/article/pii/S0925231218304843.

[26] M. Christ, A. W. Kempa-Liehr y M. Feindt, *Distributed and parallel time series feature extraction for industrial big data applications*, 2017. arXiv: 1610.07717 [cs.LG].

[27] A. W. Kempa-Liehr, J. Oram, A. Wong, M. Finch y T. Besier, "Feature Engineering Workflow for Activity Recognition from Synchronized Inertial Measurement Units," en *Pattern Recognition*, M. Cree, F. Huang, J. Yuan y W. Q. Yan, eds., Singapore: Springer Singapore, 2020, pp. 223-231.

[28] D. Knuth. "tsfresh: Time Series Feature extraction based on scalable hypothesis tests." (), [En línea]. Disponible en: https://github.com/blue-yonder/tsfresh. (accessed: 01.09.2021).

[29] F. Pedregosa et al., "Scikit-learn: Machine Learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825-2830, 2011.

[30] C. Faloutsos, M. Ranganathan e Y. Manolopoulos, "Fast Subsequence Matching in Time-Series Databases," *ACM SIGMOD Record*, vol. 23, jun. de 2000. DOI: 10.1145/191839.191925.

[31] I. Popivanov y R. Miller, "Similarity search over time-series data using wavelets," en *Proceedings 18th International Conference on Data Engineering*, 2002, pp. 212-221. DOI: 10.1109/ICDE.2002.994711.

[32] F. Korn, H. V. Jagadish y C. Faloutsos, "Efficiently supporting ad hoc queries in large datasets of time sequences," *Acm Sigmod Record*, vol. 26, n.º 2, pp. 289-300, 1997.

[33] A. Bagnall y G. Janacek, "A run length transformation for discriminating between auto regressive time series," *Journal of classification*, vol. 31, n.º 2, pp. 154-178, 2014.

[34] M. Corduas y D. Piccolo, "Time series clustering and classification by the autoregressive metric," *Computational statistics & data analysis*, vol. 52, n.º 4, pp. 1860-1872, 2008.

[35] P. Smyth, "Clustering sequences with hidden Markov models," *Advances in neural information processing systems*, vol. 9, 1996.

[36] T. Oates, L. Firoiu y P. R. Cohen, "Clustering time series with hidden markov models and dynamic time warping," en *Proceedings of the IJCAI-99 workshop on neural, symbolic and reinforcement learning methods for sequence learning*, Citeseer, 1999, pp. 17-21.

[37] S. Ghassempour, F. Girosi y A. Maeder, "Clustering multivariate time series using hidden Markov models," *International journal of environmental research and public health*, vol. 11, n.º 3, pp. 2741-2763, 2014.

[38] A. Abanda, U. Mori y J. A. Lozano, "A review on distance based time series classification," *Data Mining and Knowledge Discovery*, vol. 33, n.º 2, pp. 378-412, 2019.

[39] Z. Xing, J. Pei y E. Keogh, "A Brief Survey on Sequence Classification," *SIGKDD Explor. Newsl.*, vol. 12, n.º 1, pp. 40-48, nov. de 2010. DOI: 10.1145/1882471.1882478. [En línea]. Disponible en: https://doi.org/10.1145/1882471.1882478.

[40] X. Xi, E. Keogh, C. Shelton, L. Wei y C. A. Ratanamahatana, "Fast time series classification using numerosity reduction," en *Proceedings of the 23rd international conference on Machine learning*, 2006, pp. 1033-1040.

[41] H. Ding, G. Trajcevski, P. Scheuermann, X. Wang y E. Keogh, "Querying and mining of time series data: experimental comparison of representations and distance measures," *Proceedings of the VLDB Endowment*, vol. 1, n.º 2, pp. 1542-1552, 2008.

[42] J. Lines, L. M. Davis, J. Hills y A. Bagnall, "A shapelet transform for time series classification," en *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2012, pp. 289-297.

[43] C. Faloutsos, M. Ranganathan e Y. Manolopoulos, "Fast subsequence matching in time-series databases," *ACM Sigmod Record*, vol. 23, n.º 2, pp. 419-429, 1994.

[44] E. Keogh y C. Ratanamahatana, "Exact indexing of dynamic time warping. Knowl," *Inf. Syst*, vol. 7, n.º 3, 2005.

[45] M. Müller, "Dynamic time warping," *Information retrieval for music and motion*, pp. 69-84, 2007.

[46] P. Senin, "Dynamic time warping algorithm review," *Information and Computer Science Department University of Hawaii at Manoa Honolulu, USA*, vol. 855, n.º 1-23, p. 40, 2008.

[47] A. Mueen y E. Keogh, "Extracting optimal performance from dynamic time warping," en *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016, pp. 2129-2130.

[48] J. Paparrizos y L. Gravano, "k-shape: Efficient and accurate clustering of time series," en *Proceedings of the 2015 ACM SIGMOD international conference on management of data*, 2015, pp. 1855-1870.

[49] J. Yang et al., "k-Shape clustering algorithm for building energy usage patterns analysis and forecasting model accuracy improvement," *Energy and Buildings*, vol. 146, pp. 27-37, 2017.

[50] L. Yang y Z. Zhang, "A Deep Attention Convolutional Recurrent Network Assisted by K-shape Clustering and Enhanced Memory for Short Term Wind Speed Predictions," *IEEE Transactions on Sustainable Energy*, 2021.

[51] M. Kivelä, A. Arenas, M. Barthelemy, J. P. Gleeson, Y. Moreno y M. A. Porter, "Multilayer networks," *Journal of Complex Networks*, vol. 2, n.º 3, pp. 203-271, jul. de 2014. DOI: 10.1093/comnet/cnu016. eprint: https://academic.oup.com/comnet/article-pdf/2/3/203/9130906/cnu016.pdf. [En línea]. Disponible en: https://doi.org/10.1093/comnet/cnu016.

[52] M. Newman, *Networks*. Oxford university press, 2018.

[53] S. Wasserman, K. Faust et al., "Social network analysis: Methods and applications," 1994.

[54] S. Boccaletti, V. Latora, Y. Moreno, M. Chavez y D.-U. Hwang, "Complex networks: Structure and dynamics," *Physics reports*, vol. 424, n.º 4-5, pp. 175-308, 2006.

[55] S. Fortunato, "Community detection in graphs," *Physics reports*, vol. 486, n.º 3-5, pp. 75-174, 2010.

[56] A. Lancichinetti, M. Kivelä, J. Saramäki y S. Fortunato, "Characterizing the community structure of complex networks," *PloS one*, vol. 5, n.º 8, e11976, 2010.

[57] M. A. Porter, J.-P. Onnela, P. J. Mucha et al., "Communities in networks," *Notices of the AMS*, vol. 56, n.º 9, pp. 1082-1097, 2009.

[58] G. Chartrand, L. Lesniak y P. Zhang, *Graphs & digraphs*. Chapman & Hall London, 1996, vol. 22.

[59] J. Bang-Jensen y G. Z. Gutin, *Digraphs: theory, algorithms and applications*. Springer Science & Business Media, 2008.

[60] J.-C. Bermond y C. Thomassen, "Cycles in digraphs–a survey," *Journal of Graph Theory*, vol. 5, n.º 1, pp. 1-43, 1981.

[61] M. E. Newman, "Analysis of weighted networks," *Physical review E*, vol. 70, n.º 5, p. 056 131, 2004.

[62] A. Barrat, M. Barthelemy, R. Pastor-Satorras y A. Vespignani, "The architecture of complex weighted networks," *Proceedings of the national academy of sciences*, vol. 101, n.º 11, pp. 3747-3752, 2004.

[63] R. L. Breiger, "The duality of persons and groups," *Social forces*, vol. 53, n.º 2, pp. 181-190, 1974.

[64] A. Hagberg, P. Swart y D. S Chult, "Exploring network structure, dynamics, and function using NetworkX," Los Alamos National Lab.(LANL), Los Alamos, NM (United States), inf. téc., 2008.

[65] P. Holme y J. Saramäki, "Temporal networks," *Physics reports*, vol. 519, n.º 3, pp. 97-125, 2012.

[66] J. G. De Gooijer y R. J. Hyndman, "25 years of time series forecasting," *International journal of forecasting*, vol. 22, n.º 3, pp. 443-473, 2006.

[67] B. Lim y S. Zohren, "Time-series forecasting with deep learning: a survey," *Philosophical Transactions of the Royal Society A*, vol. 379, n.º 2194, p. 20 200 209, 2021.

[68] A. A. Cook, G. Mısırlı y Z. Fan, "Anomaly detection for IoT time-series data: A survey," *IEEE Internet of Things Journal*, vol. 7, n.º 7, pp. 6481-6494, 2019.

[69] V. Chandola, A. Banerjee y V. Kumar, "Anomaly detection: A survey," *ACM computing surveys (CSUR)*, vol. 41, n.º 3, pp. 1-58, 2009.

[70] V. Chandola, A. Banerjee y V. Kumar, "Anomaly detection for discrete sequences: A survey," *IEEE transactions on knowledge and data engineering*, vol. 24, n.º 5, pp. 823-839, 2010.

[71] P. John y G. Luis, "k-shape: Efficient and accurate clustering of time-seris," en *Proc. SIGMOD*, 2015.

[72] B. Luque, L. Lacasa, F. Ballesteros y J. Luque, "Horizontal visibility graphs: Exact results for random time series," *Physical Review E*, vol. 80, n.º 4, p. 046 103, 2009.

[73] E. Tibau, A.-A. Ludl, S. Rüdiger, J. G. Orlandi y J. Soriano, "Neuronal spatial arrangement shapes effective connectivity traits of in vitro cortical networks," *IEEE Transactions on Network Science and Engineering*, vol. 7, n.º 1, pp. 435-448, 2018.

[74] S. Boccaletti et al., "The structure and dynamics of multilayer networks," *Physics Reports*, vol. 544, n.º 1, pp. 1-122, nov. de 2014. DOI: 10.1016/j.physrep.2014.07.001. [En línea]. Disponible en: http://dx.doi.org/10.1016/j.physrep.2014.07.001.

[75] M. Kivelä, A. Arenas, M. Barthelemy, J. P. Gleeson, Y. Moreno y M. A. Porter, "Multilayer networks," *Journal of complex networks*, vol. 2, n.º 3, pp. 203-271, 2014.

[76] R. Criado, J. Flores, A. Garcia del Amo, M. Romance, E. Barrena y J. A. Mesa, "Line graphs for a multiplex network," *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 26, n.º 6, p. 065 309, 2016.

[77] S. Dorogovtsev, *Complex networks*. Oxford University Press, Oxford, UK, 2010.

[78] S. H. Strogatz, "Exploring complex networks," *nature*, vol. 410, n.º 6825, pp. 268-276, 2001.

[79] L. d. F. Costa et al., "Analyzing and modeling real-world phenomena with complex networks: a survey of applications," *Advances in Physics*, vol. 60, n.º 3, pp. 329-412, 2011.

[80] V. Chapela, R. Criado, S. Moral y M. Romance, *Intentional risk management through complex networks analysis*. Springer, 2015.

[81] R. Criado, S. Moral, Á. Pérez y M. Romance, "On the edges' PageRank and line graphs," *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 28, n.º 7, p. 075 503, 2018.

[82] E. Estrada, M. Fox, D. J. Higham y G.-L. Oppo, *Network science: complexity in nature and technology*. Springer Science & Business Media, 2010.

[83] V. Latora, V. Nicosia y G. Russo, *Complex networks: principles, methods and applications*. Cambridge University Press, 2017.

[84] S. Moral, V. Chapela, R. Criado, Á. Pérez y M. Romance, "Efficient algorithms for estimating loss of information in a complex network: Applications to intentional risk analysis," *Networks & Heterogeneous Media*, vol. 10, n.º 1, p. 195, 2015.

[85] M. Brede, *Networks—An Introduction. Mark EJ Newman.(2010, Oxford University Press.)* 65,38, 35,96(*hardcover*), 772*pages.IS BN* − 978 − 0 − 19 − 920665 − 0., 2012.

[86] M. Zanin, M. Romance, S. Moral y R. Criado, "Credit card fraud detection through parenclitic network analysis," *Complexity*, vol. 2018, 2018.

[87] M. Zanin, D. Papo, M. Romance, R. Criado y S. Moral, "The topology of card transaction money flows," *Physica A: Statistical Mechanics and its Applications*, vol. 462, pp. 134-140, 2016.

[88] A. Partida, R. Criado y M. Romance, "Identity and Access Management Resilience against Intentional Risk for Blockchain-Based IOT Platforms," *Electronics*, vol. 10, n.º 4, p. 378, 2021.

[89] A. Partida, R. Criado y M. Romance, "Visibility Graph Analysis of IOTA and IoTeX Price Series: An Intentional Risk-Based Strategy to Use 5G for IoT," *Electronics*, vol. 10, n.º 18, p. 2282, 2021.

[90] A. Criado-Alonso, E. Battaner-Moro, D. Aleja, M. Romance y R. Criado, "Using complex networks to identify patterns in specialty mathematical language: a new approach," *Social Network Analysis and Mining*, vol. 10, n.º 1, pp. 1-10, 2020.

[91] A. A. Aburomman y M. B. I. Reaz, "Review of IDS development methods in machine learning," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 6, n.º 5, pp. 2432-2436, 2016.

[92] C.-F. Tsai, Y.-F. Hsu, C.-Y. Lin y W.-Y. Lin, "Intrusion detection by machine learning: A review," *expert systems with applications*, vol. 36, n.º 10, pp. 11 994-12 000, 2009.

[93] N. T. Van, "A Combination of Temporal Sequence Learning and Data Description for Anomalybased NIDS," *International Journal of Network Security & Its Applications (IJNSA) Vol*, vol. 11, 2019.

[94] F. Wang, S. Yang, C. Wang y Q. Li, "A Novel Intrusion Detection System for Malware Based on Time-Series Meta-learning," en *International Conference on Machine Learning for Cyber Security*, Springer, 2020, pp. 50-64.

[95] S. D. Anton, L. Ahrens, D. Fraunholz y H. D. Schotten, "Time is of the essence: Machine learning-based intrusion detection in industrial time series data," en *2018 IEEE International Conference on Data Mining Workshops (ICDMW)*, IEEE, 2018, pp. 1-6.

[96] S. Staniford-Chen et al., "GrIDS-a graph based intrusion detection system for large networks," en *Proceedings of the 19th national information systems security conference*, Citeseer, vol. 1, 1996, pp. 361-370.

[97] L. Akoglu, H. Tong y D. Koutra, "Graph based anomaly detection and description: a survey," *Data mining and knowledge discovery*, vol. 29, n.º 3, pp. 626-688, 2015.

[98] S. I. Pérez, S. Moral-Rubio y R. Criado, "A new approach to combine multiplex networks and time series attributes: Building intrusion detection systems (IDS) in cybersecurity," *Chaos, Solitons & Fractals*, vol. 150, p. 111 143, 2021.

[99] N. Moustafa y J. Slay, "UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)," en *2015 military communications and information systems conference (MilCIS)*, IEEE, 2015, pp. 1-6.

[100] B. A. Tama y K.-H. Rhee, "Attack classification analysis of IoT network via deep learning approach," *Res. Briefs Inf. Commun. Technol. Evol.(ReBICTE)*, vol. 3, pp. 1-9, 2017.

[101] A. Sonule, M. Kalla, A. Jain y D. Chouhan, "UNSWNB15 Dataset and Machine Learning Based Intrusion Detection Systems," *International Journal of Engineering and Advanced Technology (IJEAT)*, vol. 9, n.º 3, pp. 2638-2648, 2020.

[102] S. I. Pérez, S. Moral-Rubio y R. Criado, "A new approach to combine multiplex networks and time series attributes: Building intrusion detection systems (IDS) in cybersecurity," *Chaos, Solitons & Fractals*, vol. 150, p. 111 143, 2021.

[103] S. I. Pérez, S. Moral-Rubio y R. Criado, "Combining multiplex networks and time series: A new way to optimize real estate forecasting in New York using cab rides," *Physica A: Statistical Mechanics and its Applications*, vol. 609, p. 128 306, 2023.

[104] F. Ballesteros, F. Luque, L. Lacasa, B. Luque y J. Nuno, "From time series to complex networks: the visibility graphs," *Proc Natl Acad Sci USA*, vol. 105, p. 4972, 2008.

[105] R. Criado, J. Flores, A. Garcıa del Amo, J. Gómez-Gardenes y M. Romance, "A mathematical model for networks with structures in the mesoscale," *International Journal of Computer Mathematics*, vol. 89, n.º 3, pp. 291-309, 2012.

[106] S. Martinčić-Ipšić, D. Margan y A. Meštrović, "Multilayer network of language: A unified framework for structural analysis of linguistic subsystems," *Physica A: Statistical Mechanics and its Applications*, vol. 457, pp. 117-128, 2016.

[107] J. D. Kelleher, *Fundamentals of machine learning for predictive data analytics : algorithms, worked examples, and case studies / John D. Kelleher, Brian Mac Namee, Aoife D'Arcy.* eng. Cambridge, Massachusetts: The MIT Press, 2015 - 2015.

[108] J. D. Kelleher y B. Tierney, *Data science*. MIT Press, 2018.

[109] P. Lorenzo, F. Stefano, A. Ferreira y P. Carolina, *Artificial Intelligence and Cybersecurity: Technology, Governance and Policy Challenges*, 2021.

[110] P. Holme y J. Saramäki, "Temporal networks," *Physics Reports*, vol. 519, n.º 3, pp. 97-125, oct. de 2012. DOI: 10.1016/j.physrep.2012.03.001. [En línea]. Disponible en: http://dx.doi.org/10.1016/j.physrep.2012.03.001.

[111] R. Albert y A.-L. Barabási, "Statistical mechanics of complex networks," *Rev. Mod. Phys.*, vol. 74, pp. 47-97, 1 ene. de 2002. DOI: 10.1103/RevModPhys.74.47. [En línea]. Disponible en: https://link.aps.org/doi/10.1103/RevModPhys.74.47.

[112] S. Boccaletti, V. Latora, Y. Moreno, M. Chavez y D. U. Hwang, "Complex Networks: Structure and Dynamics," *Physics Reports*, vol. 424, pp. 175-308, 2006.

[113] S. Wasserman, K. Faust et al., "Social network analysis: Methods and applications," 1994.

[114] L. Solá, M. Romance, R. Criado, J. Flores, A. Garcıa del Amo y S. Boccaletti, "Eigenvector centrality of nodes in multiplex networks," *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 23, n.º 3, p. 033 131, 2013.

[115] F. Pedroche, M. Romance y R. Criado, "A biplex approach to PageRank centrality: From classic to multiplex networks," *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 26, n.º 6, p. 065 301, 2016.

[116] M. Romance, L. Solá, J. Flores, E. Garcıa, A. G. Del Amo y R. Criado, "A Perron–Frobenius theory for block matrices associated to a multiplex network," *Chaos, Solitons & Fractals*, vol. 72, pp. 77-89, 2015.

[117] P. Vandewalle, J. Kovacevic y M. Vetterli, "Reproducible research in signal processing," *IEEE Signal Processing Magazine*, vol. 26, n.º 3, pp. 37-47, 2009.

[118] D. R. Fuhrmann, J. P. Browning y M. Rangaswamy, "Signaling strategies for the hybrid MIMO phased-array radar," *IEEE Journal of Selected Topics in Signal Processing*, vol. 4, n.º 1, pp. 66-78, 2010.

[119] G. Ali y K. Zaman, "Do house prices influence stock prices? Empirical investigation from the panel of selected European Union countries," *Economic Research-Ekonomska Istraživanja*, vol. 30, n.º 1, pp. 1840-1849, 2017. DOI: 10.1080/1331677X.2017.1392882. eprint: https://doi.org/10.1080/1331677X.2017.1392882. [En línea]. Disponible en: https://doi.org/10.1080/1331677X.2017.1392882.

[120] A. Yuksel, "The relationship between stock and real estate prices in Turkey: Evidence around the global financial crisis," *Central Bank Review*, vol. 16, abr. de 2016. DOI: 10.1016/j.cbrev.2016.03.006.

[121] H. Rahman, G. Ali, S. Khan y E. Aidoo, "Are Asian stock and house prices integrated or segmented?" *International Journal of Electronic Finance*, vol. 10, p. 2020, oct. de 2020. DOI: 10.1504/IJEF.2020.110297.

[122] J. K. * y J. W. V. D. End, "Do stock prices affect house prices? Evidence for the Netherlands," *Applied Economics Letters*, vol. 11, n.º 12, pp. 741-744, 2004. DOI: 10.1080/1350485042000254863. eprint: https://doi.org/10.1080/1350485042000254863. [En línea]. Disponible en: https://doi.org/10.1080/1350485042000254863.

[123] S. Bourassa y V. S. Peng, "Hedonic Prices and House Numbers: The Influence of Feng Shui," *International Real Estate Review*, vol. 2, n.º 1, pp. 79-93, 1999. [En línea]. Disponible en: https://EconPapers.repec.org/RePEc:ire:issued:v:02:n:01:1999:p:79-93.

[124] M. J. Bailey, R. F. Muth y H. O. Nourse, "A regression method for real estate price index construction," *Journal of the American Statistical Association*, vol. 58, n.º 304, pp. 933-942, 1963.

[125] J. M. N. Tabales, J. M. Caridad, F. J. R. Carmona et al., "Artificial neural networks for predicting real estate price," *Revista de Métodos Cuantitativos para la Economıa y la Empresa*, vol. 15, pp. 29-44, 2013.

[126] *Zillow*.

[127] *NYTaxi. URL:https://www1.nyc.gov/site/tlc/about/tlc-trip-record-data.page*.

[128] S. Aghabozorgi, A. S. Shirkhorshidi y T. Y. Wah, "Time-series clustering–a decade review," *Information Systems*, vol. 53, pp. 16-38, 2015.

[129] A. Cutler, D. R. Cutler y J. R. Stevens, "Random forests," en *Ensemble machine learning*, Springer, 2012, pp. 157-175.

[130] L. Breiman, "Random forests," *Machine learning*, vol. 45, n.º 1, pp. 5-32, 2001.

[131] T. Hastie, R. Tibshirani y J. Friedman, "Random forests," en *The elements of statistical learning*, Springer, 2009, pp. 587-604.