TREBALL DE FI DE GRAU

Grau en Matemàtiques

Títol: Anàlisi de tràfic de dades orientat a la generació de mostres sintètiques

Autor: Laura García-Fogeda Roca

Directors: Luis Velasco Esteban i Marc Ruiz Ramírez

Departament: Departament d'Arquitectura de Computadors

Convocatòria: 3 de Maig de 2021

UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Facultat de Matemàtiques i Estadística

Facultat de Matemàtiques i Estadística

Treball de fi de grau

Anàlisi de tràfic de dades orientat a la generació de mostres sintètiques

Laura García-Fogeda Roca

Supervisat per
Luis Velasco Esteban
Marc Ruiz Ramírez

3 de Maig de 2021

Abstract

La irrupció de noves aplicacions i serveis està convertint la planificació i la gestió de xarxes de comunicacions d'operador en una tasca cada cop més complexa i difícil de dur a terme mitjançant tècniques tradicionals basades en procediments gestionats manualment. L'ús d'intel·ligència artificial es presenta com un dels principals habilitadors per dotar d'autonomia a aquestes infraestructures bàsiques de comunicacions. En general, les eines basades en intel·ligència artificial necessiten dades útils per poder operar amb precisió, per exemple, per entrenar models de predicció de tràfic que permetin anticipar o detectar situacions específiques que requereixin accions concretes. En aquest treball es presentarà un procediment de caracterització de tràfic de dades orientat a l'extracció de components periòdiques i residuals mitjançant algoritmes basats en la transformada de Fourier i l'anàlisi autoregressiu de sèries temporals. Aquesta caracterització permet, entre d'altres, reduir la dimensionalitat d'una base de dades de gran volum sense perdre una quantitat d'informació significativa. La metodologia principal es validarà tant a partir de conjunts de dades sintètics com tràfic realista. A més, es presentarà un cas d'ús il·lustratiu, on es mostrarà el potencial del procediment de caracterització a l'hora de detectar períodes de tràfic atípic en un entorn d'anàlisi dinàmic.

Paraules clau: Tràfic de dades, caracterització i modelització, transformada de Fourier, anàlisi de series temporals.

En primera instància m'agradaria agrair als meus supervisors Marc Ruíz Ramírez i Luís Velasco Esteban l'oportunitat de fer aquest treball amb ells. Gràcies per haver-me deixat formar part del vostre grup de recerca durant aquests mesos i haver-me permès fer un petit tast en aquest món. Gràcies per la vostra dedicació, el vostre temps, el vostre suport i tot el que he pogut aprendre gràcies a vosaltres.

Vull agrair també el suport incondicional de la meva família. Com sempre, m'han ajudat a fer que tot esforç sigui molt més amè i portador. Finalment, no puc oblidar-me dels meus amics i el seu suport moral durant tots aquests mesos. Sense tots vosaltres, no hauria estat possible la realització d'aquest treball.

Índex

Capítol 1

Introducció

En aquesta primera secció aprofitarem per explicar quina és la motivació que ens ha impulsat a fer aquest treball, així com els principals objectius que es volen assolir i una petita explicació sobre l'organització de la memòria.

Motivació del treball

El tràfic de dades és la quantitat d'informació que flueix en la xarxa a causa de la participació dels usuaris en un determinat servei. Mesurar-lo i modelar-lo pot ser molt útil a l'hora de crear estratègies, optimitzar i invertir recursos de manera encertada.

En els últims anys, estan fent-se més comuns i predominants serveis com Netflix o Twitch que tenen característiques diferents a les que hi havia fins ara [1]. Algunes d'aquestes característiques són per exemple l'increment de l'amplada de banda que consumeixen o els seus alts requisits de qualitat de servei, com ara una latència molt baixa. Pot haver-hi milers d'usuaris en una xarxa d'àmbit metropolità consumint Netflix a 4 o 8k i la xarxa ha de ser capaç de suportar-ho. Això està provocant canvis substancials en les estratègies de planificació i operació de les xarxes que han de connectar els proveïdors de serveis amb els usuaris [2].

Per tal d'afrontar aquesta planificació que ha passat a ser molt més complexa a causa de les necessitats dinàmiques que es presenten, poden aplicar-se tècniques d'intel·ligència artificial enfocades a la gestió de les xarxes. Un dels objectius d'aquestes eines i recursos d'intel·ligència artificial és el de dotar de més autonomia a les xarxes [3], fet que inclou la caracterització i predicció de com serà el tràfic que s'haurà de suportar per tal d'anticipar-se a accions com l'augment de la capacitat de la xarxa o la reconfiguració de connexions, de tal manera que puguin assolir-se els requisits que es demanden.

A fi de poder crear aquests models –que poden basar-se per exemple, en mètodes de regressió de qualsevol mena–, és necessari recopilar dades que permetin entrenar-los com és degut. El fet de treballar i gestionar aquestes dades és un tema complex, ja que és difícil que un operador tingui una col·lecció de dades suficientment completa per a entrenar bons models. Imaginem el cas en el qual un operador vol introduir un servei nou en la xarxa. Com és evident, l'operador en qüestió no té cap manera de tenir dades del servei que vol introduir. Què pot fer aleshores? Una de les opcions que pot plantejar-se és, en cas de tenir-hi accés, integrar dades d'una altra xarxa que estigui proveint el mateix servei. Una altra opció és la del tractament de dades d'altres serveis que hagi recollit en la seva xarxa per tal de reaprofitar-les per construir un model pel nou servei que vol incorporar. El més probable que passi en aquest segon escenari, és que el volum de dades que hagi recollit sigui molt gran i molt difícil de tractar. Les dades poden no estar ben balancejades, havent-hi patrons que es repeteixin en excés, mentre que d'altres, en canvi, només apareguin de manera esporàdica, fent així difícil o impossible l'entrenament de bons models.

El millor plantejament davant d'aquesta necessitat, és el de la generació de conjunts de dades que siguin suficientment representatives i permetin entrenar el model com és degut. Per tal d'incorporar les característiques dels nous serveis i emular com serà el tràfic que s'haurà de suportar, és imprescindible conèixer el comportament de les dades de les quals es parteix com a referència.

L'objectiu d'aquest treball és presentar una metodologia basada en la combinació de les sèries de Fourier i els mètodes d'anàlisi de sèries temporals, de manera que es pugui caracteritzar el tràfic en components periòdiques que després utilitzarem per a la generació de tràfic sintètic. Per tal de posar a prova la metodologia que desenvoluparem, es durà a terme l'anàlisi d'un tràfic de dades de gran volum amb l'objectiu de reduir-ne la dimensionalitat i caracteritzar-lo en un espai més petit que contingui la màxima informació possible. Aquesta metodologia serà aplicable de manera dinàmica, de manera que podrem anar analitzant el tràfic a mesura que l'anem rebent per tal de caracteritzar-lo i detectar tant la presència de dies atípics com l'aparició de noves components periòdiques que incorporarem en la caracterització del tràfic.

La manipulació de les dades, creació de models i obtenció de resultats es realitzarà a través del llenguatge de programació *Python*, destacant essencialment les llibreries *numpy* i *pandas*.

Objectius principals

Aquest treball està enfocat a la construcció d'un procediment que ens permeti caracteritzar dades de tràfic agregat, que es caracteritzen per ser sèries temporals univariants (tassa de bits per unitat de temps). Els punts en els quals treballarem per arribar a desenvolupar aquest model són:

- Detectar i extraure les components periòdiques que caracteritzen un tràfic de dades.
- Analitzar els residuals que no s'expliquen a partir de les components periòdiques, a fi de trobar altres components deterministes com components d'autocorrelació residuals.
- Reduir la dimensionalitat d'un tràfic de dades de gran volum en un espai més petit de tal manera que es perdi la mínima informació possible.
- Construir un predictor de tràfic sintètic a partir de les components detectades esmentades previament, de tal manera que es puguin generar dades que segueixin uns determinats patrons.
- Crear un detector de dies atípics que donats dos tràfics de dades, determini si la distribució que segueixen aquests dos tràfics és igual o no.
- Integrar les metodologies desenvolupades del tal manera que es pugui analitzar un tràfic de manera dinàmica, detectant l'aparició de dies atípics i actualitzant el generador de dades amb les noves components periòdiques que vagin apareixent.

Organització de la memòria

En referència a com hem organitzat la memòria, en el Capítol 2 presentarem tots els fonaments matemàtics que hem utilitzat per desenvolupar els models. En el Capítol 3, explicarem amb detall el disseny dels algoritmes que hem desenvolupat i els acompanyarem de part del codi implementat. Al Capítol 4 presentarem els resultats que fan referència a cadascun dels models i algoritmes del procediment complet. Finalment, en el Capítol 5 exposarem una síntesi de les conclusions extretes, així com una petita valoració personal i una proposta de feina futura.

Capítol 2

Background matemàtic

L'objectiu d'aquest capítol és introduir de manera rigorosa tots els conceptes matemàtics que hem utilitzat al llarg del treball. Començarem amb una breu introducció de teoria de la senyal, definint totes les idees que utilitzarem a posteriori. Continuarem parlant de l'anàlisi de Fourier, començant amb una contextualització històrica i explicant el pas de les sèries de Fourier a la transformada, fent menció també a l'algoritme de la Transformada Ràpida de Fourier. Inclourem una breu introducció dels principals conceptes estadístics que hem utilitzat, i acabarem la secció explicant què és un model ARIMA i de quina formulació matemàtica es parteix per construir-lo.

2.1 Teoria de la senyal

El primer que farem serà exposar tots els conceptes i enunciats relacionats amb teoria de la senyal.

Per entendre què modelen i com són les dades amb les quals hem treballat, començarem introduint què és una sèrie temporal i què és el tràfic de dades. Una **sèrie temporal** és una col·lecció d'observacions d'una variable recollides seqüencialment en el temps de manera equiespaiada. El **tràfic de dades** el definim com la quantitat de dades enviades, representades en quantitat de bits per unitat de temps, p. ex. Gigabits per segon (Gb/s) que s'envia i es rep a conseqüència de l'activitat dels usuaris en un determinat servei.

Aquests dos conceptes són fonamentals, ja que tota la modelització que hem fet va enfocada al tractament de sèries temporals que mesuren el tràfic de dades d'un determinat servei.

Una altra de les característiques del nostre tràfic, és que en essència, està format per la superposició de diverses funcions periòdiques.

Definició (Funció periòdica)**.** Diem que una funció $f : \mathbb{R} \to \mathbb{R}$ és **periòdica** amb període $T \in \mathbb{R}$, $T > 0$ si $f(x+T) = f(x)$, on T és el menor nombre real positiu que compleix aquesta propietat.

En el nostre cas, parlarem de **components periòdiques** quan fem referència a cadascuna de les funcions periòdiques que conformen el tràfic amb el qual treballem. Cadascuna d'aquestes components ve determinada per l'equació

$$y(t) = Asin(\frac{2\pi}{T}t + \varphi_0)$$

on A és l'amplitud del tràfic, T és el període de cada component i φ_0 és el desplaçament de fase, que correspon a l'angle associat al valor del tràfic en l'instant inicial $t = 0$. Tenint en compte que l'expressió $\omega = 2\pi/T$ relaciona la freqüència angular amb el període, podem simplificar l'equació anterior de la següent manera

$$y(t) = Asin(\omega t + \varphi_0)$$

A banda d'aquestes components també cal considerar un soroll addicional, i en alguns casos, una **component contínua**, que no és res més que una funció constant que té freqüència zero i que caldrà tenir en compte a l'hora de treballar amb tràfics que no estiguin centrats en zero. Quan parlem de **soroll**, ens referirem al conjunt d'informació que no és d'interès, degrada i distorsiona el tràfic original i en dificulta el seu estudi i tractament.

A l'hora de parlar de processament de senyals, cal tenir clares les diferències entre una senyal analògica i una de digital. Una **senyal analògica** és aquella en la que la informació de la senyal és contínua, mentre que una **senyal digital** pren valors discrets al llarg del temps. El tràfic amb el qual hem treballat és continu, motiu pel qual no pot ser enregistrat i processat en un ordinador ni pot fer-se'n cap tractament. És per això que cal considerar una sèrie de mostres –valors de la senyal en un instant de temps– en forma de sèrie temporal, de manera que aquesta mostra sigui prou representativa per a no perdre informació i al mateix temps, ens permeti emmagatzemar una quantitat de dades que no sigui massa gran. La **freqüència de mostreig** és el nombre de mostres que es prenen per unitat de temps. La seva unitat de mesura són els Hertz (Hz), que equivalen a s^{-1} i mesuren el nombre de repeticions d'un determinat fenomen físic durant un segon. Per acabar, introduirem el concepte de **digitalització**, que fa referència al procés de convertir el tràfic analògic en un tràfic digital a través del mostreig.

Una bona tria de la freqüència de mostreig ens permet controlar el procés d'emmagatzematge de les dades per tal de ser capaços de no perdre informació i per altra banda, de no recollir més dades que les que són necessàries i alentir-ne el posterior tractament.

En la següent imatge (Figura 2.1) veiem uns quants exemples de senyals analògiques amb diferents punts que corresponen a la tria de diferents freqüències de mostreig. En alguns casos es veu clarament com la tria d'aquests punts no és gens representativa a l'hora de reproduir correctament la senyal.

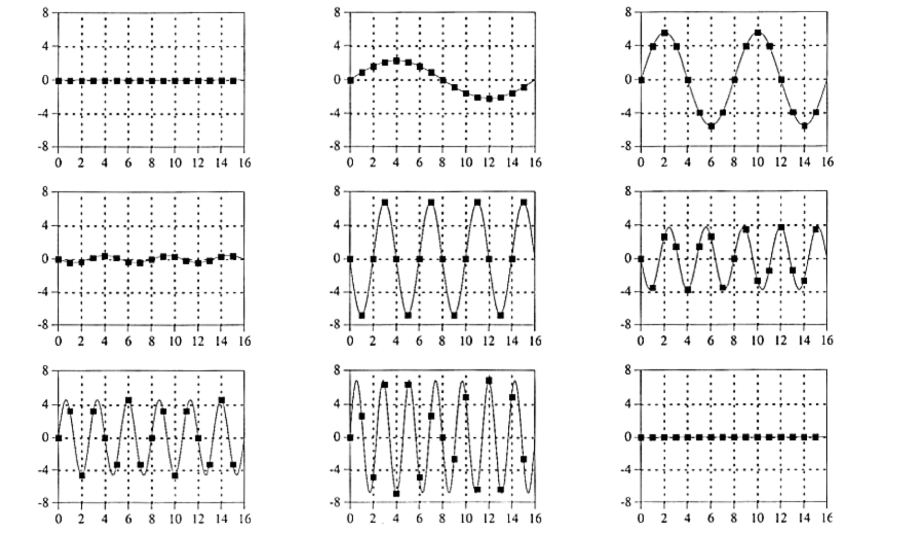

Figura 2.1: Tria de diferents freqüències de mostreig en el procés de digitalització d'una senyal [5].

Un cop introduït el concepte de digitalització i de freqüència de mostreig i seguint el fil d'un dels principals objectius del treball que és el de la generació de dades a partir del tractament d'un determinat tràfic, és imprescindible tenir present el següent enunciat:

Teorema (Teorema de Nyquist)**.** *La reconstrucció exacta d'una senyal contínua periòdica a partir de les seves mostres, és matemàticament possible sempre que la freqüència de mostreig*

sigui com a mínim el doble de la freqüència més alta que es vol reconstruir.

Un exemple que il·lustra el que exposa aquest teorema és que per recollir dades d'un fenomen periòdic que succeeix amb una periodicitat d'un segon (és a dir, amb freqüència d'1 Hz), cal que el mostreig de les dades es realitzi com a màxim, cada mig segon, és a dir, cal que la freqüència de mostreig mínima de les dades sigui de 2 Hz.

2.2 Anàlisi de Fourier

La modelització de les components periòdiques del tràfic amb el qual hem treballat l'hem dut a terme a través d'algoritmes ràpids de la transformada de Fourier. En aquesta secció treballarem des de la base tots els fonaments matemàtics necessaris per entendre en què estan basats aquests algoritmes.

Per la realització d'aquesta secció s'han consultat [10], [12], els quals proveeixen coneixements més extensos de l'àmbit que poden ser d'interès pel lector.

Context històric

Les sèries trigonomètriques van sorgir en la matemàtica del segle XVIII, en relació amb l'estudi de les petites oscil·lacions de medis elàstics.

Tot va començar el 1715, quan Brook Taylor[1], va proposar en la seva obra *Methodus incrementorum directa et inversa*, el problema de la corda vibrant. Aquest problema tracta de determinar el moviment d'una corda elàstica, així com el seu temps de vibració, si aquesta és tensada mitjançant l'aplicació de certa força externa i després es deixa lliure.

El primer matemàtic que va elaborar un model apropiat pel problema, va ser D'Alembert[2], qui el 1747 va demostrar (fent referència a petites oscil·lacions), que la funció u que és solució del problema ha de satisfer les següents equacions en derivades parcials:

$$\begin{cases} \dfrac{\partial^2 u(x,t)}{\partial^2 t} = \dfrac{\partial^2 u(x,t)}{\partial^2 x}, & 0 < x < \pi, t > 0 \\ u(x,0) = f(x), & 0 \leq x \leq \pi \\ \dfrac{\partial u(x,0)}{\partial t} = 0, & 0 \leq x \leq \pi \\ u(0,t) = u(\pi,t) = 0, & t \geq 0 \end{cases} \tag{2.1}$$

La primera condició en (2.1), és coneguda com **equació d'ones**. La segona fa referència a la posició inicial de la corda, la tercera significa que la velocitat inicial de la corda és zero i l'última expresa que els extrems de la corda es mantenen fixats.

D'Alembert també va demostrar que la solució de (2.1) és causada per:

$$u(x,t) = \frac{1}{2}[\tilde{f}(x+t) + \tilde{f}(x-t)] \tag{2.2}$$

on $\tilde{f}$ és una extensió apropiada de la funció f. No entrarem en més detall per no perdre el fil de l'objectiu del nostre estudi.

Leonhard Euler[3], també va demostrar la fórmula (2.2), diferint D'Alembert de les corbes que podien considerar-se com a condicions inicials. D'aquí va sorgir una de les primeres manifestacions sobre els problemes de la definició de la noció de funció.

[1] Brook Taylor (1685 - 1731) va ser un matemàtic britànic

[2] Jean Le Rond D'Alembert (1717 – 1783) va ser un matemàtic i filòsof francès i un dels màxims exponents de la Il·lustració.

[3] Leonhard Euler (1707 – 1783) va ser un matemàtic suís, considerat un dels més brillants de la història.

Per altra banda, el 1753, Daniel Bernoulli[4], va proposar una alternativa completament diferent per l'obtenció de la solució de (2.1). Aquesta proposta es fonamentava en la idea que la solució u podia obtenir-se com la superposició d'ones senzilles, concretament de la forma:

$$u_n = \sin(nx)\cos(nt), \quad \forall n \in \mathbb{N} \tag{2.3}$$

on per cada temps t fixat, la funció (2.3) és un múltiple de la funció $\sin(nx)$ que s'anul·la en exactament $n-1$ punts de l'interval $(0,\pi)$.

És possible que Bernoulli fes ús dels seus coneixements musicals per tal d'arribar a la idea anterior, que exposaven que matemàticament la solució de (2.1) ha de poder expressar-se de la forma:

$$u(x,t) = \sum_{n=1}^{\infty} a_n \sin(nx)\cos(nt) \tag{2.4}$$

on cal triar els coeficients a_n de tal manera que satisfacin l'equació (2.1).

Les idees de Bernoulli van rebre dures crítiques per part de D'Alembert i Euler, i no va ser fins 54 anys després quan Joseph Fourier[5] va reprendre-les per tractar l'estudi de la teoria de la calor en cossos sòlids.

Fourier va considerar una vareta fina de certa longitud π, els extrems de la qual es mantenen a 0^{o} centígrads i que té la superfície lateral aïllada. Si la distribució inicial de temperatura de la vareta ve donada per $f(x)$, la qüestió que va plantejar-se va ser quina seria la temperatura de qualsevol punt de la vareta x en un instant de temps t.

Fourier va demostrar que si $u(x,t)$ representa la temperatura de la vareta en la secció x i en el temps t, aleshores la funció u ha de satisfer:

$$\begin{cases} \dfrac{\partial^2 u(x,t)}{\partial^2 x} = \dfrac{\partial u(x,t)}{\partial t}, & 0 < x < \pi, 0 < t < T \\ u(0,t) = u(\pi,t) = 0, & 0 \le t \le T \\ u(x,0) = f(x), & 0 \le x \le \pi \end{cases} \tag{2.5}$$

la primera condició és **l'equació de la calor**, la segona representa la condició que els extrems es mantenen a 0^{o} i la tercera fa referència a la temperatura inicial de la vareta.

Partint de les idees de Bernoulli, Fourier va preguntar-se quina havia de ser la tria dels coeficients a_n per tal que l'única solució de (2.5) fos de la forma:

$$u(x,t) = \sum_{n=1}^{\infty} a_n e^{-n^2 t} \sin(xt) \tag{2.6}$$

En el seu estudi, va concloure que els coeficients que complissin aquests requisits havien de ser de la forma:

$$a_n = \frac{2}{\pi}\int_0^{\pi} f(\xi)\sin(n\xi)\,d\xi, \quad \forall n \in \mathbb{N} \tag{2.7}$$

En reconeixement a la tenacitat pionera de Fourier, aquest tipus de sèries van ser anomenades **sèries de Fourier**.

[4]Daniel Bernoulli (1700 – 1782), matemàtic, estadístic, físic i metge suís destacat per les seves aportacions tant en el camp de les matemàtiques pures com aplicades.

[5]Jean Baptiste Joseph Fourier (1768 – 1830), conegut com més endavant veurem, pels seus treballs sobre la descomposició de sèries periòdiques en sèries trigonomètriques.

Sèries de Fourier

Per tal de descriure els problemes bàsics de la teoria de sèries de Fourier, és necessari introduir el concepte d'espai L_2, producte escalar i algunes idees sobre funcions ortogonals.

Definició. Sigui f una funció suficientment regular, anomenem $L_2(a,b)$ al conjunt de les $f:[a,b]\to\mathbb{R}$ pels quals la integral $\int_a^b f^2(x)dx$ existeix i es defineix com

$$L_2(a,b)=\{f:[a,b]\to\mathbb{R} \quad \int_a^b f^2(x)dx<\infty\}$$

La condició anterior de suficient regularitat correspon al fet que la funció sigui mesurable. Els detalls d'aquesta definició s'escapen dels objectius d'aquest treball i en cas de ser d'interès del lector poden consultar-se a [12].

Definició. Siguin $f,g\in L_2(a,b)$. Definim el seu **producte escalar** com el nombre real $\langle f,g\rangle$ que ve donat per:

$$\langle f,g\rangle=\int_a^b fg$$

Donat un espai vectorial dotat d'un producte escalar $\langle\cdot,\cdot\rangle$, sempre és possible definir una norma mitjançant l'aplicació

$$\|f,f\|=\langle f,f\rangle^{\frac{1}{2}}$$

Definició. Un espai vectorial dotat d'una norma és un **espai normat**.

Definició. Donat un espai normat, un **sistema ortonormal** $\mathcal{S}$ de l'espai és un conjunt de vectors tals que tot element de l'espai pot escriure's com a combinació lineal dels vectors de la base. Per tal que la base sigui ortonormal, els seus elements han de complir que:

1. $\|v\|=1,\forall v\in\mathcal{S}$
2. $\langle v,w\rangle=0,\forall v,w\in\mathcal{S}$ tals que $v\neq w$

Definició. Anomenem **espai de Hilbert** a un espai vectorial normat i complet (és a dir, un espai vectorial dotat d'una norma i on tota successió de Cauchy és convergent).

Tenim doncs, que $L_2(a,b)$ és un espai normat amb les operacions usuals de suma de funcions i producte d'un escalar per una funció. És un espai de dimensió infinita, pel que podem trobar subconjunts de $L_2(a,b)$ que siguin linealment independents, que continguin infinits elements, i que formin una base ortonormal del nostre espai. A més a més, també és un espai complet. Així doncs, $L_2(a,b)$ és un espai de Hilbert, i en particular ho és $L_2(-\pi,\pi)$. Utilitzant el producte escalar que hem definit anteriorment, es té que el següent conjunt d'elements és una base ortonormal de $L_2(-\pi,\pi)$

$$\varphi_0=\frac{1}{\sqrt{2\pi}},\quad \varphi_{2n-1}=\frac{\cos(nx)}{\sqrt{\pi}},\quad \varphi_{2n}=\frac{\sin(nx)}{\sqrt{\pi}},\quad n=1,2,... \tag{2.8}$$

Definició. Donats $\mathcal{S}=\varphi_0,\varphi_1,\varphi_2,...$ un sistema ortonormal en $L_2(-\pi,\pi)$ i una funció $f\in L_2(-\pi,\pi)$, definim la **sèrie de Fourier de** f respecte $\mathcal{S}$ com

$$SF\mathcal{S}(f)(x)=\sum_{n=0}^{\infty}c_n\varphi_n(x) \tag{2.9}$$

on els coeficients de la sèrie són

$$c_n = \langle f, \varphi_n \rangle = \int_{-\pi}^{\pi} f(x)\varphi_n(x)dx$$

Si substituïm les φ_i de l'expressió (2.9) pels coeficients c_i trobats, tenim doncs, que la sèrie de Fourier trigonomètrica a $L_2(-\pi, \pi)$ és

$$SF\mathcal{T}(f)(x) = c_0\frac{1}{\sqrt{2\pi}} + \sum_{n=1}^{\infty}\left(c_{2n-1}\frac{\cos(nx)}{\sqrt{\pi}} + c_{2n}\frac{\sin(nx)}{\sqrt{\pi}}\right)$$

amb

$$\begin{aligned} c_0 &= \int_{-\pi}^{\pi} f(x)\frac{1}{\sqrt{2\pi}}dx \\ c_{2n-1} &= \int_{-\pi}^{\pi} f(x)\frac{\cos(nx)}{\sqrt{\pi}}dx \\ c_{2n} &= \int_{-\pi}^{\pi} f(x)\frac{\sin(nx)}{\sqrt{\pi}}dx \end{aligned}$$

Per treballar de manera més còmoda, s'acostumen a redefinir els coeficients anteriors de la següent manera:

$$a_0 = 2\frac{c_0}{\sqrt{2\pi}}, \quad a_n = \frac{c_{2n-1}}{\sqrt{\pi}}, \quad b_n = \frac{c_{2n}}{\sqrt{\pi}}$$

i s'obté així la fórmula:

$$f(x) = \frac{a_0}{2} + \sum_{n=1}^{\infty}(a_n\cos(nx) + b_n\sin(nx)) \tag{2.10}$$

amb

$$\begin{aligned} a_0 &= \frac{1}{\pi}\int_{-\pi}^{\pi} f(x)dx \\ a_n &= \frac{1}{\pi}\int_{-\pi}^{\pi} f(x)\cos(nx)dx, \forall n \in \mathbb{N}, n \geq 1 \\ b_n &= \frac{1}{\pi}\int_{-\pi}^{\pi} f(x)\sin(nx)dx, \forall n \in \mathbb{N} \end{aligned}$$

Per últim, manipularem (2.10) per obtenir una expressió matemàticament equivalent, però que ens serà útil com a punt de partida de la següent secció.

Suposem que $f : [-\pi, \pi] \to \mathbb{R}$, és una funció periòdica de període 2π. Sabem que podem expressar f tal com es mostra en l'equació (2.10), però també podem expressar-la com a sèrie de Fourier en forma complexa fent ús de les següents identitats trigonomètriques

$$\begin{aligned} \cos(nx) &= \frac{1}{2}(e^{inx} + e^{-inx}) \\ \sin(nx) &= \frac{1}{2i}(e^{inx} - e^{-inx}) \end{aligned}$$

Partint de (2.10) podem reescriure l'expresió de la sèrie de Fourier en forma complexa:

$$f(x) = \frac{a_0}{2} + \sum_{n=1}^{\infty} (a_n \cos(nx) + b_n \sin(nx)) = \frac{a_0}{2} + \sum_{n=1}^{\infty} \left(a_n \frac{e^{inx} + e^{-inx}}{2} + b_n \frac{e^{inx} - e^{-inx}}{2i} \right)$$
$$= \frac{a_0}{2} + \sum_{n=1}^{\infty} \frac{a_n - ib_n}{2} e^{inx} + \sum_{n=1}^{\infty} \frac{a_n + ib_n}{2} e^{-inx} = \sum_{n=-\infty}^{\infty} c_n e^{inx} \tag{2.11}$$

on hem fet servir la següent notació

$$c_0 = \frac{a_0}{2}, \quad c_n = \frac{a_n - ib_n}{2}, \quad c_{-n} = \frac{a_n + ib_n}{2}$$

Transformada de Fourier

Intuïtivament, la transformada de Fourier ens permet descompondre qualsevol senyal contínua i periòdica en funcions sinusoidals. Si calculem la transformada de Fourier d'una funció que és suma de diverses components sinusoidals amb diferents freqüències, amplituds i desplaçaments de fase, obtindrem una funció complexa, el valor absolut de la qual presentarà un pic per cada funció periòdica que conforma la funció original. L'amplitud de cada component vindrà donada per l'ordenada de cada pic, mentre que el seu valor d'abscissa correspondrà a la seva freqüència. Per obtenir el desplaçament de fase, haurem de fixar-nos en l'argument complex de cadascun dels pics.

Il·lustrem amb un exemple (Figura 2.2) el que acabem d'explicar. Si partim de dues senyals determinades cadascuna d'elles per una ona sinusoidal sense desplaçament de fase i amb una freqüència de 4 i 7Hz respectivament, podem definir una tercera senyal formada per la superposició d'aquestes primeres dues. Com podem veure en la següent imatge, si calculem la transformada de Fourier d'aquesta tercera senyal, el que obtindrem és una funció que presenta dos pics d'amplitud 1 en les freqüències de 4 i 7Hz.

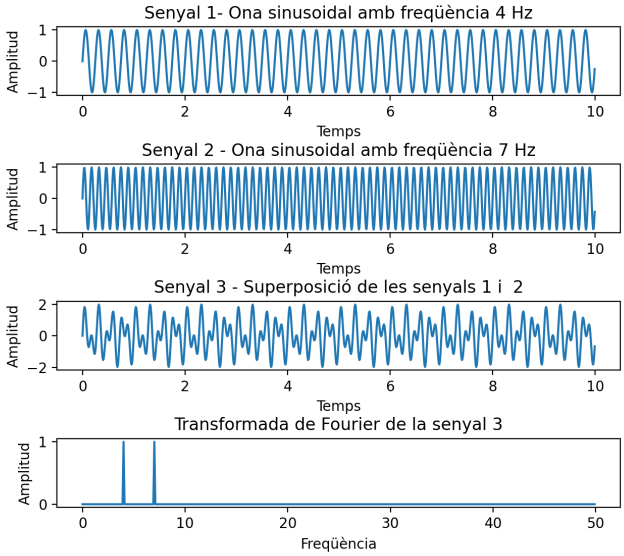

Figura 2.2: Exemple d'una funció i la seva transformada de Fourier.

En aquesta subsecció, prendrem com a referència l'estudi que hem fet fins ara de les sèries de Fourier per entendre la formulació matemàtica que s'amaga darrere d'aquest fenomen.

Formalment, podem deduir l'expressió de la transformada de Fourier a partir de la sèrie, fent tendir a infinit el període T. Fent un canvi de variable en l'equació (2.11) i expressant-la en funció de t i la seva freqüència ω_0 s'obté

$$f(t) = \sum_{n=-\infty}^{\infty} c_n e^{in\omega_0 t}, \tag{2.12}$$

on

$$c_n = \frac{1}{T}\int_{-T/2}^{T/2} f(x)e^{-in\omega_0 t}dt \tag{2.13}$$

Tenint en compte que $\omega_0 = 2\pi/T$ i substituïnt en (2.12) i (2.13), es té que:

$$\begin{aligned} f(t) &= \sum_{n=-\infty}^{\infty}\left[\frac{1}{T}\int_{-T/2}^{T/2} f(x)e^{-in\omega_0 x}dx\right]e^{in\omega_0 t} \\ &= \sum_{n=-\infty}^{\infty}\left[\frac{1}{2\pi}\int_{-T/2}^{T/2} f(x)e^{-in\omega_0 x}dx\right]\omega_0 e^{in\omega_0 t} \end{aligned}$$

Ara bé, com $\omega_0 = 2\pi/T$, si $T \to \infty$ aleshores ω_0 tendeix a zero. Sigui $\omega_0 = \triangle\omega$, en el límit $T \to \infty$ es té que $\triangle\omega \to d\omega$, i el sumatori anterior es converteix en la integral sobre ω, pel que l'expressió anterior de la funció f està determinada per

$$\begin{aligned} f(t) &= \int_{-\infty}^{\infty}\left[\frac{1}{2\pi}\int_{\infty}^{\infty} f(x)e^{-in\omega x}dx\right]e^{in\omega t}d\omega \\ &= \frac{1}{2\pi}\int_{-\infty}^{\infty}\left[\int_{\infty}^{\infty} f(x)e^{-in\omega x}dx\right]e^{in\omega t}d\omega \end{aligned} \tag{2.14}$$

Si definim

$$F(\omega) = \int_{-\infty}^{\infty} f(t)e^{-i\omega t}dt \tag{2.15}$$

(2.14) es converteix en

$$f(t) = \frac{1}{2\pi}\int_{-\infty}^{\infty} F(\omega)e^{i\omega t}d\omega \tag{2.16}$$

A la forma (2.15) se l'anomena **transformada de Fourier d'una funció** f i a la forma (2.16) **transformada inversa de Fourier**.

Hem de tenir en compte però, que en el cas particular del processament de la nostra senyal, com que haurem hagut de digitalitzar-la, tindrem que la funció que volem estudiar pren valors discrets, pel que caldrà adaptar les fórmules de (2.15) i (2.16) per poder calcular transformades de Fourier de funcions discretes.

Sigui f la funció de l'expressió (2.15), considerem una seqüència de valors complexo $f(t_k) = x_k$, $k = 0, ..., N-1$. Podem expressar l'equació de la **transformada discreta de Fourier** com

$$X_k = \sum_{n=0}^{N-1} x_n e^{-2\pi ikn/N}, \qquad k = 0, ..., N-1 \tag{2.17}$$

De manera anàloga es té que la **transformada inversa discreta de Fourier** és

$$x_n = \frac{1}{N} \sum_{k=0}^{N-1} X_k e^{2\pi i k n/N}, \qquad n = 0, ..., N-1$$

Implementació d'algoritmes

Dedicarem aquesta secció a explicar la importància de l'algoritme de la tansformada ràpida de Fourier –coneguda àmpliament com a FFT, pel seu nom en anglès *Fast Fourier Transform*–, sense entrar en el detall de la seva implementació.

D'entrada, és important entendre que la rellevància de la Transformada de Fourier ve donada pel seu ampli ventall d'aplicacions, que van des del tractament digital de senyals fins a la resolució d'equacions en derivades parcials o algoritmes de multiplicació ràpida d'enters molt grans.

Per tal que tot el potencial que s'amaga darrere d'aquesta eina sigui realment útil i aplicable, cal trobar una manera eficient d'implementar els algoritmes necessaris per dur a terme els corresponents càlculs. L'avaluació directa de la fórmula (2.17) requereix de l'ordre de $O(N^2)$ operacions aritmètiques, pel que fa que sigui poc usable per a N molt grans a causa de la seva lentitud. Arran d'aquest fet, es van buscar alternatives per esquivar aquest problema, i va ser així com cap a l'any 1965 James William Cooley i John Wilder Tukey van popularitzar l'algoritme de la **Transformada ràpida de Fourier**, que consisteix en un replantejament del mètode en el qual es descomposa la transformada en altres transformades més simples i que permet reduir el nombre d'operacions a $O(N \log(N))$. Per fer-nos a la idea que suposa aquest estalvi, si considerem $N \approx 1000$ el nombre d'operacions que podem evitar mitjançant l'ús de la FFT respecte a l'avaluació directa de la transformada és de l'ordre del 99%.

2.3 Eines estadístiques

Hem de tenir present, que en el procés d'anàlisi i tractament de dades, l'estadística hi juga un paper fonamental. En la construcció dels nostres models, hem utilitzat diversos conceptes i idees que pertanyen a aquesta branca.

En aquesta secció, partirem de la base que el lector té certes nocions d'estadística i obviarem la definició d'alguns conceptes fonamentals. En cas de voler-hi aprofundir, es poden consultar els detalls a [14].

Començarem definint què és la correlació entre dues variables. Després introduirem el concepte de covariància, idea de la qual partirem per definir el nostre estadístic de mesura de la correlació: el coeficient de correlació de Pearson.

Definició (Correlació)**.** Anomenem **correlació** al mètode estadístic que estudia la dependència lineal entre dues variables. Siguin X i Y dues variables, diem que estan correlacionades entre elles si al disminuir els valors d'X també disminueixen els de Y i viceversa.

En referència a la covariància i al coeficient de correlació de Pearson, aquests poden definir-se tant per a variables aleatòries com per a mostres d'aquestes variables. A nosaltres, però, només ens interessa definir-les per a mostres i deixarem de banda les definicions que fan referència purament a variables aleatòries.

Definició (Covariància mostral)**.** Anomenem **covariància mostral** al paràmetre que indica el grau de variació conjunta de dues variables aleatòries. Siguin X i Y aquestes variables, definim aquest paràmetre com

$$Cov(X, Y) = \frac{\sum_{i=1}^{n} (x_i - \bar{x})(y_i - \bar{y})}{n-1}$$

on n és la mida de les mostres, x_i, y_i el valor de les observacions i-èssimes i $\bar{x}, \bar{y}$ la mitjana de cadascuna d'elles.

Definició (Coeficient de correlació de Pearson mostral)**.** Anomenem **coeficient de correlació de Pearson mostral** (r_{xy}) a l'estadístic que serveix per mesurar la correlació entre dues mostres x i y i que correspon a la covariància estandaritzada. Es calcula segons la fórmula

$$r_{xy} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}}$$

on n és la mida de les mostres, x_i, y_i el valor de les observacions i-èssimes i $\bar{x}, \bar{y}$ la mitjana de cadascuna d'elles.

Aquest estadístic sempre pren valors en l'interval $[-1, 1]$. Valors negatius d'aquest coeficient indiquen dependència lineal negativa. Si el valor és 0 vol dir que no hi ha dependència lineal entre les variables, mentre que valors positius de l'estadístic estan associats a correlacions positives. Com més proper a 1 sigui en valor absolut, més correlacionades estan les mostres entre si.

En la següent figura (Figura 2.3) es mostren diferents gràfiques on hi ha representades parells de variables i el seu coeficient de correlació de Pearson associat.

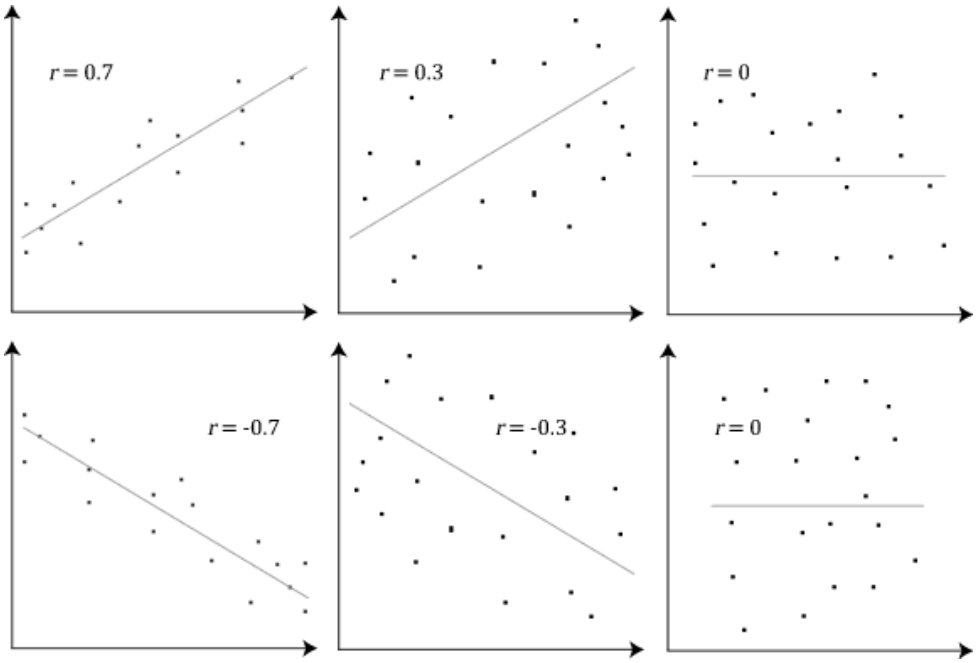

Figura 2.3: Coeficients de correlació de Pearson associats a diferents parelles de mostres [6].

De cara a estudiar les característiques que compleix una determinada població, introduirem els conceptes necessaris per entendre què és un test d'hipòtesis i què és el p-valor.

Definició (Hipòtesi nul·la)**.** Anomenem **hipòtesis nul·la** (H_0) a la hipòtesis que volem contrastar. Correspon a la hipòtesis que plantegem inicialment i és la que mantindrem a no ser que les dades demostrin la seva falsedat.

Definició (Hipòtesi alternativa)**.** Anomenem **hipòtesis alternativa** (H_1) a la negació de la hipòtesis nul·la.

Definició (Nivell de significació)**.** Anomenem **nivell de significació** α, a la probabilitat de rebutjar H_0 quan aquesta és certa[6].

$$P(rebutjar\ H_0 \mid H_0\ certa) = \alpha$$

Definició (Test d'hipòtesis)**.** Un **test d'hipòtesis** és una regla que, donat un cert valor de significació α, determina per a quins valors d'una mostra acceptarem o rebutjarem H_0.

Definició (p-valor)**.** El **p-valor** és el nivell de significació més petit pel qual la mostra particular obtinguda obligaria a rebutjar H_0. És a dir:

$$\text{p-valor} = P(rebutjar\ H_0\ per\ x_1, ..., x_n \mid H_0\ certa)$$

Això vol dir, que quan el p-valor del nostre test prengui valors menors que el nivell de significació que hem definit, haurem de rebutjar H_0, mentre que per valors majors del p-valor continuarem mantenint-la.

Acabarem la secció amb un exemple particular d'un test d'hipòtesis que mesura la discrepància entre una distribució de probabilitat teòrica i una d'observada, estudi que s'anomena bondat d'ajust. Tal com veurem més endavant, aquest test ens ha servit per ser capaços de detectar l'aparició de dies atípics en el tràfic.

Definició (Test chi-quadrat)**.** El **test chi-quadrat** és un test d'hipòtesis de bondat d'ajust que s'utilitza per comprovar si una mostra d'unes dades segueix una determinada distribució.

Una de les característiques més importants del test chi-quadrat és que pot aplicar-se a qualsevol distribució de probabilitat univariant per la qual pugui calcular-se la seva funció de distribució acumulada. Per tal d'aplicar aquest test, cal que les dades estiguin classificades en classes, és a dir, expressades en forma d'histograma o taula de freqüències.

La prova chi-quadrat es defineix segons les següents hipòtesis:

$$\begin{cases} H_0 : \text{Les dades segueixen una distribució especificada.} \\ H_1 : \text{Les dades no segueixen la distribució especificada.} \end{cases}$$

Per dur a terme el càlcul d'aquest test, cal que les dades estiguin classificades en k classes. Definim l'estadístic de la prova com:

$$\chi^2 = \frac{\sum_{i=1}^{k}(O_i - E_i)^2}{Ei}$$

on O_i és la freqüència observada de la classe i i E_i és la freqüència esperada de la classe i. La freqüència esperada E_i es calcula segons

$$E_i = n(F(Y_u) - F(Y_l))$$

on F és la funció distribució acumulada que estem contrastant, Y_u i Y_l són respectivament el límit superior i inferior de la classe i-èssima i n és la mida de la mostra.

[6]El fet de rebutjar H_0 quan aquesta és certa s'anomena error tipus I.

2.4 ARIMA

En el procés de modelització i desagregació del tràfic, hem utilitzar un model ARIMA per tal d'estudiar els residuals resultants d'extraure al tràfic les components periòdiques que s'han trobat. En aquesta secció, explicarem amb detall en què consisteixen aquest tipus de models.

Els models ARIMA són un subconjunt dels models de regressió lineal que tenen com a objectiu utilitzar les observacions passades dels valors d'una determinada variable per tal d'analitzar-la i poder predir-la per a temps futurs. El nom d'ARIMA fa referència a **Autoregressive Integrated Moving Average**. A continuació detallarem cadascuna de les parts que conformen aquest tipus de model per ser capaços d'entendre'l i veure com està construït.

AR: Autoregressive

El concepte d'autoregressió fa referència al fet que la idea que hi ha darrere d'aquesta part, és descriure la variable objectiu a partir dels valors que ha pres en els temps anteriors.

Sigui X la variable que nosaltres volem descriure, X_0 el valor que pren en l'instant actual, $X_{lag(i)}$ el valor que ha pres fa i instants de temps, el que es vol és expressar X de la següent manera:

$$X_0 = \beta_0 + \sum_{i=1}^{i=n} \beta_i X_{lag(i)}$$

El que estem exposant amb aquesta equació —coneguda de manera comuna per $AR(n)$ *model*— , no és res més que el valor de X_0 és una combinació lineal dels valors que ha pres en els n instants anteriors, on n és un paràmetre que nosaltres escollim i les β_i són els coeficients de regressió lineal que trobem en entrenar el model.

Si anomenem $X_{forward(i)}$ al valor que prendrà X un cop hagin transcorregut i instants de temps, podem calcular $X_{forward(1)}$ reformulant l'equació anterior

$$X_{forward(1)} = \beta_0 + \beta_1 X_0 + \beta_2 X_{lag(1)} + ... + \beta_n X_{lag(n-1)} \tag{2.18}$$

I: Integrated

La integració indica que aplicarem un pas diferencial a les dades. És a dir, en lloc d'aplicar la regressió lineal tal com hem vist en l'equació (2.18), el que farem serà restar a cada observació l'observació que ha succeït en l'instant de temps anterior

$$X_{forward(1)} - X_0 = \beta_0 + \beta_1(X_0 - X_{lag(1)}) + ... + \beta_n(X_{lag(n-1)} - X_{lag(n)})$$

El que continuem dient amb aquesta equació no és res més que els valors que prendrà la variable X són una combinació lineal dels valors que ha pres en el passat. El motiu pel qual volem aplicar una diferenciació és perquè generalment, les diferències són molt més estacionàries (varien molt menys al llarg del temps) que els valors de $X_{lag(i)}$. Quan modelitzem sèries temporals, és interessant que la variància mitjana de les nostres dades sigui estacionària, ja que aquest fet significarà que les principals propietats estadístiques del nostre model no dependran de quan s'hagi pres la mostra. Així doncs, els modes basats en dades estacionàries són generalment més robustos.

MA: Moving Average

Per tractar aquesta part, començarem recapitulant i recordant que l'equació completa d'un model bàsic de regressió lineal és

$$X = \beta_0 + \beta_1 t + \epsilon$$

on l'ϵ fa referència al fet que el resultat que obtinguem de la regressió $X = \beta_0 + \beta_1 t$ és només una aproximació de la variable X. En la següent imatge (Figura 2.4) es reflecteix clarament el fet que estem exposant. Els punts negres són els valors objectius de la variable que volem predir, mentre que la línia blava és el nostre model de regressió lineal. Així doncs, els ϵ representen la diferència entre el valor exacte que pren la variable X i l'aproximació que estem predint amb el nostre model.

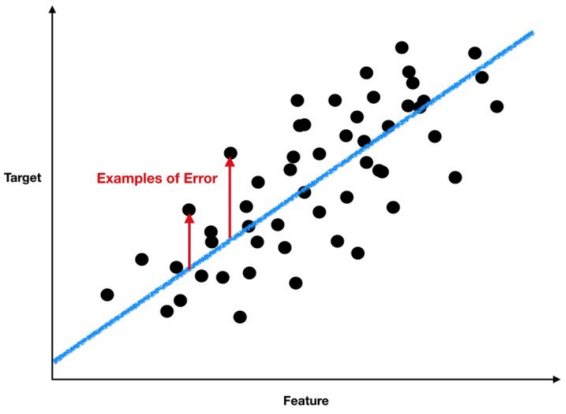

Figura 2.4: Model de regressió lineal

El problema que trobem en la definció que hem fet del paràmetre ϵ, és que per tal de poder trobar-lo, primer de tot necessitem un model. A diferència dels valors que pren la nostra variable objectiu X, tenim que ϵ és un paràmetre que no és directament observable. Per tant, a l'hora d'estimar els paràmetres del nostre model ARIMA no podrem remetre'ns a mètodes com el de mínims quadrats i necessitarem un mètode iteratiu com el de l'aproximació per màxima versemblança [13], que ens permetrà estimar simultàniament els paràmetres β i els residuals ϵ.

Per tant, el que fa un model MA és predir X utilitzant els errors que ha comès del model en temps passats. De manera similar al model AR, cal indicar quants errors passats volem que tingui en compte.

Així doncs, un model de mitjana mòbil és resumeix en la següent expressió

$$X = \beta_0 + \sum_{i=1}^{i=n} \beta_i \epsilon_{lag(i)}$$

Capítol 3

Algoritmes principals

En aquesta secció inclourem una explicació detallada de quin és l'objectiu dels principals algoritmes que s'han desenvolupat al llarg del treball i com s'han implementat.

Una premissa de la que s'ha partit al realitzar l'estudi i necessària per poder aplicar els models que hem desenvolupat, és que el tràfic de dades amb el que es treballa no ha de tenir tendència creixent ni decreixent. Noti's que, en cas de partir d'unes dades amb tendència, aquesta s'hauria de tractar i eliminar prèviament a partir de tècniques estadístiques clàssiques com la diferenciació [17].

Començarem explicant com s'ha generat el tràfic i la posterior anàlisi que s'ha fet per tal de desagregar-lo. A continuació, exposarem com s'ha implementat el principal cas il·lustratiu amb el qual s'ha treballat: el mòdul de detecció de dies atípics en un entorn dinàmic.

3.1 Generació del tràfic

Malgrat que l'objectiu del treball sigui construir models que permetin treballar amb tràfic real proveït per una teleoperadora, el que s'ha fet per arribar a desenvolupar aquests models ha estat partir d'un tràfic senzill que hem generat nosaltres mateixos. La generació de tràfic sintètic s'ha fet mitjançant la superposició de diverses components periòdiques, un soroll suplementari, i en alguns casos, l'addició de valors anòmals.

El nostre generador de tràfic està definit de tal manera que els paràmetres de generació de dades estan informats en forma de diccionari que conté dues claus: *base* i *anomalies*. Alhora, cadascuna d'aquestes claus conté diverses subclaus. Els paràmetres que s'especifiquen en la clau *base* fan referència a les característiques generals del tràfic, mentre que els d'*anomalies* només cal informar-los en cas de voler afegir valors anòmals al tràfic que generem.

La clau *anomalies* conté les subclaus *t0_params*, *dur_params*, *mg_params* i *n_anomalies*. Les subclaus *t0_params*, *dur_params* i *mg_params* estan definides a partir d'un vector de dos nombres reals (el segon d'ells ha de ser positiu) que corresponen a la mitjana i a la desviació estàndard d'una funció de probabilitat gaussiana que modela el temps entre anomalies expressat en minuts, la seva duració expressada en minuts, i la seva magnitud. La subclau *n_anomalies* determina el nombre d'anomalies que generem. Addicionalment, poden afegir-se dues claus opcionals definides cadascuna d'elles per un escalar: *start* i *end*. En el cas d'estar informades, la clau *start* més la funció distribució determinada per *t0_params* determinaran l'instant en el qual succeeix la primera anomalia, i el paràmetre *end* correspondrà al temps final de les anomalies. En el cas que aquest paràmetre estigui informat, primarà sobre el paràmetre *n_anomalies*, deixant de generar anomalies en cas de ser necessari per no sobrepassar *end*.

A continuació, presentem la funció *anomaly_generator*, la qual retorna tres llistes *t0*, *t1* i *mg* que corresponen al temps inicial, temps final i magnitud de cada anomalia. Utilitzarem aquests

valors per afegir-los a tràfic que després generarem.

```
# Funcio de generacio d'anomalies
def anomaly_generator(n_anomalies,t0_params,dur_params,mg_params,start=None,end=None):
    t = start if start is not None else 0 # En cas d'estar definit, prenem start com temps inicial. Altrament comencem en 0

    t0 = [] # Llista on guardarem els temps inicials de cada anomalia
    t1 = [] # Llista on guardarem els temps finals de cada anomalia
    mg = [] # Llista on guardarem la magnitud de cada anomalia

    for i in range(n_anomalies):
        # Per cada anomalia, calculem el seu instant d'inici, final i la seva magnitud seguint una normal definida pels parametres informats
        t0_val = t + np.random.normal(loc=t0_params[0], scale=t0_params[1])
        dur_val = np.random.normal(loc=dur_params[0], scale=dur_params[1])
        t1_val = end if end is not None and t0_val + dur_val > end else t0_val + dur_val
        mg_val = np.random.normal(loc=mg_params[0], scale=mg_params[1])

        # Afegim a les corresponents llistes els temps inicial, final i magnitud de cada anomalia
        t0.append(t0_val)
        t1.append(t1_val)
        mg.append(mg_val)

        t = t1_val # Actualitzem t
        # En cas d'estar definit el parametre end i sobrepassar-lo, ens aturem
        if end is not None and t >= end: break

    return t0, t1, mg
```

Per altra banda, dins la clau *base* trobem les subclaus *cycles*, *shifts*, *per*, *amp* i *sd*. La subclau *cycles* fa referència al nombre de dades que generem i es defineix a partir d'un enter que correspon al nombre de cops que es repeteix la component de periodicitat màxima. Imaginem que volem generar un tràfic com a superposició de m components periòdiques. Cadascuna de les subclaus *shifts*, *per* i *amp* estarà definida a partir d'un vector d'escalars de m components. Aquestes subclaus corresponen, respectivament, al desplaçament de fase, a la periodicitat en minuts i a l'amplitud de cadascuna de les components. En última instància, la subclau *sd* fa referència al domini de generació de soroll. Aquest soroll és afegit a la suma de les components periòdiques i està determinat per una funció de probabilitat uniforme que pren valors en l'interval $(-\sigma, \sigma)$.

El format en el qual cal introduir tots aquests paràmetres és el següent:

```
params = [
    {'base': {'cycles': k, 'shifts': [φ1, φ2, ..., φm],
    'per': [p1, p2, ..., pm], 'amp': [A1, A2, ..., Am], 'sd': σ}},
    'anomalies': {'t0_params': [μ1, σ1], 'dur_params': [μ2, σ2],
    'mg_params': [μ3, σ3], 'n_anomalies': n}]
```

Acabarem presentant la funció de generació de dades *main_data*. Aquesta funció rep com a inputs els paràmetres *datafilename*, que en cas d'estar informat correspon al nom de l'arxiu csv d'on es llegiran les dades, el paràmetre *verbose* que per cada cop que cridem la funció permet controlar si volem mostrar o no un plot per pantalla, i el paràmetre *params* que s'informa dels paràmetres de generació de dades. La funció retorna una llista *amplitude_all*, que correspon a una sèrie temporal amb el tràfic generat.

```python
# Funcio de generacio de dades
def main_data(datafilename, verbose, params=None):
    # En cas d'estar informat datafilename llegim el csv
    if datafilename is not None:
        amplitude_all = pd.read_csv(datafilename)
        amplitude_all = amplitude_all.values.tolist()
        amplitude_all = [item for sublist in amplitude_all for item in sublist]
    # En cas de no estar informat, generem les dades
    else:
        amplitude_all = [] # Llista on guardarem el trafic que generem
        # Per cada component definida:
        for comp in params:
            periods = comp['base']['cycles'] # Longitud dades -> periodicitat max * cycles
            shifts = comp['base']['shifts'] # Desplacament de fase
            per = comp['base']['per'] # Periodicitat
            amp = comp['base']['amp'] # Amplitud
            sd = comp['base']['sd'] # Soroll
            ppp = max(per) # Periodicitat maxima

            # Inicialitazem a 0 amplitude_1
            n = int(np.ceil(periods * ppp))
            amplitude_1 = np.zeros(n)

            # Generem amplitude_1
            for j in range(len(shifts)):
                a2 = amp[j] * np.array([np.sin((2 * np.pi * (i + shifts[j]) / float(per[j]))) for i in range(n)])
                a2 = np.add(a2, np.random.uniform(low=-sd, high=sd, size=n))
                amplitude_1 = np.add(amplitude_1, a2)

            # Parametres de generacio d'anomalies
            if 'anomalies' in comp:
                t0_params = comp['anomalies']['t0_params'] # Promig i sd de la normal que modela el temps entre anomalies
                dur_params = comp['anomalies']['dur_params'] # Promig i sd de la normal que modela la duracio de les anomalies
                mg_params = comp['anomalies']['mg_params'] # Promig i sd de la normal que modela la magnitud de les anomalies
                n_anomalies = comp['anomalies']['n_anomalies'] # Nombre d'anomalies

                # Generem les anomalies
                [t0, t1, mg] = anomaly_generator(n_anomalies, t0_params, dur_params, mg_params)
                t0 = [int(x) for x in t0]  # Convertim t0 vector d'enters
                t1 = [int(x) for x in t1]  # Convertim t1 a un vector d'enters

                # Afegim les anomalies a amplitude_1
                for j in range(len(t0)):
                    aux = np.zeros(n)
                    aux[t0[j]:t1[j]] = mg[j]
                    amplitude_1 = np.add(amplitude_1, aux)

            amplitude_all = amplitude_all + list(amplitude_1)

    if verbose:
        # Plot de les dades
        plotter.plot(amplitude_all)
        plotter.show()

    return amplitude_all
```

3.2 Anàlisi i modelització del tràfic

En aquesta secció explicarem com hem trobat les components periòdiques del tràfic, com les hem fet servir per generar noves dades i l'anàlisi que hem fet dels residuals que no s'expliquen un cop s'han tret al tràfic les components periòdiques trobades.

Detecció de les components periòdiques

Una de les premisses de les que hem partit en implementar la detecció de components periòdiques, és que la freqüència de mostreig del tràfic analitzat ha de ser de 1/60 Hz.

Tots els senyals que no estan centrades en zero, tenen associades una component contínua que té freqüència zero. Per tal de poder treballar de manera còmoda i no tenir problemes de divisibilitat, hem començat centrant el tràfic en zero per tal d'evitar la detecció d'aquesta component. Ens hem guardat el valor mitjà del tràfic, per tal de poder després dur a terme la corresponent reconstrucció.

Seguidament, hem realitzat la FFT. L'amplitud i freqüència A_i i f_i de cada component periòdica, corresponen respectivament a l'abscissa i ordenada de cadascun dels pics que apareixen en la part real de la transformada. Siguin φ_i els respectius desplaçaments de fase, aquests vénen determinats per $\varphi_i = Im(A_i)/Re(A_i) + \frac{\pi}{2}$.

Amb l'objectiu de filtrar algunes components associades al soroll, el que hem fet ha estat obviar tots els pics amb amplitud inferior a 1/200 del valor màxim de la transformada –xifra que correspon a l'amplitud màxima de les components trobades–. La cerca d'aquest llindar s'ha fet a base de prova i error. Malgrat que d'entrada vam plantejar-nos fixar-lo en 1/50, vam adonar-nos que era necessari rebaixar-lo per tal de ser capaços d'explicar tràfics més complexos que estan determinats per la composició de moltes components que tenen amplituds petites. La funció de detecció de components s'anomena *component_detection*, i rep com a inputs *amplitude_all* que correspon al tràfic que volem analitzar i el paràmetre *verbose* que ens permet controlar la generació de plots cada cop que cridem a la funció. La funció retorna la transformada filtrada *filtered*, la mitjana de les dades *data_mean*, i la periodicitat *per_clusters*, amplitud *clusters_amplitude* i desplaçament de fase *phase_shift* de cada component trobada.

```
# Funcio de deteccio de components periodiques
def component_detection(amplitude_all, verbose):
    data_mean = np.mean(amplitude_all) # Mitjana de les dades
    amplitude_all = amplitude_all - data_mean #Substraiem la mitjana al trafic
    original

    samplingFrequency_1 = 1 / float(60)  # Freq. de mostreig
    fourierTransform_1 = np.fft.fft(amplitude_all) * 2 / len(amplitude_all)  #
    Fem la transformada amb les dades normalitzades segons la seva longitud
    fourierTransform_1 = fourierTransform_1[range(int(len(amplitude_all) / 2))]
     # Excloiem la frequencia de mostratge

    # Definim l'espai de frequencies
    tpCount_1 = len(amplitude_all)
    values_1 = np.arange(int(tpCount_1 / float(2)))
    timePeriod_1 = tpCount_1 / samplingFrequency_1
    frequencies_1 = values_1 / timePeriod_1

    if verbose:
    # Plot de la transformada de Fourier
        plotter.plot(frequencies_1, abs(fourierTransform_1))
        plotter.title('FFT')
        plotter.xlabel('frequencies')
        plotter.ylabel('amplitude of the components')
```

```
        plotter.show()

    # FILTRATGE DE COMPONENTS RELLEVANTS
    # Totes les frequencies que estan per sota d'un cert llindar les assignem a 0
    f = (abs(fourierTransform_1)).max() / 200
    pos = np.where(abs(fourierTransform_1) > f) # Posicions de les frequencies que guardem
    filtered = len(amplitude_all) * np.array(fourierTransform_1)  # Abans ho hem normalitzat pel nombre de dades. Ho desfem
    filtered[np.where(abs(fourierTransform_1) <= f)] = 0 # Assignem a 0 les components no rellevants

    # Caracateritzacio de les components
    per_clusters = np.multiply(np.power(frequencies_1[pos], -1), samplingFrequency_1)  # Periodicitats components
    clusters_amplitude = abs(filtered[pos] / len(amplitude_all)) # Amplitud components
    phase_shift = np.arctan2(np.imag(filtered[pos]), np.real(filtered[pos])) + np.pi/2 # Desplacament de fase components


    return filtered, data_mean, per_clusters, clusters_amplitude, phase_shift
```

Modelització de les components periòdiques

De cara a reconstruir el tràfic original amb l'algoritme de la IFFT proveït per la llibreria *numpy*, ens hem trobat amb la limitació de què aquest no era vàlid per tràfics no centrats en zero i que tampoc ens servia per generar tràfics de major durada que el tràfic original.

El que hem fet doncs, ha estat crear un generador de tràfic que superposi les components periòdiques determinades per A_i, f_i i φ_i que s'han trobat mitjançant la FFT i les desplaci el corresponent valor mitjà *data_mean*. D'aquesta manera hem creat un model que ens permet crear tràfics de la durada que nosaltres desitgem, i que com més endavant veurem, ens serà molt útil per la detecció de comportaments atípics.

La funció generadora de tràfic s'anomena *traffic_generator* i rep com a paràmetres les dades *original_data* que volen analitzar-se, un enter i que informa del nombre de dies d'*original_data* dels quals s'extrauran les components (en cas de ser 0, considera totes les dades), el paràmetre n que fa referència al nombre de dies de dades que volem generar a partir de les components que trobem, i el paràmetre *verbose*, que permet controlar si volem mostrar plots per pantalla en l'execució del programa. La funció retorna el tràfic sintètic *synthetic_traffic* que s'ha generat, i per cada component periòdica que s'ha utilitzat en la generació d'aquest tràfic, la seva periodicitat *per_clusters*, amplitud *clusters_amplitude* i desplaçament de fase *phase_shift*.

```
# Funcio generadora de trafic sintetic
def traffic_generator(original_data, i, n, verbose):
    # Si i == 0 realitzem la deteccio de components de totes les dades originals
    if i == 0:
        initial_data = original_data
    # Si i != 0 analitzem les components de tants dies com haguem informat
    else:
        initial_data = original_data[:i * 1440]

    # Obtenim les components que caracteritzen la nostra senyal a traves de la transformada
    [filtered, data_mean, per_clusters, clusters_amplitude, phase_shift] = fourier_transform.component_detection(initial_data, verbose)

```

```
    # GENERAICIO DE DADES
    # Inicialitzem el trafic sintetic a la mitjana del trafic original
    synthetic_traffic = [data_mean for j in range(n * 1440)]

    # Transformem el desplacament de fase per poder reconstruir el model
    for j in range(len(phase_shift)):
        phase_shift[j] = (per_clusters[j] * phase_shift[j]) / (2 * np.pi)

    # Per cada component detectada l'afegim al trafic sintetic
    for j in range(len(phase_shift)):
        a2 = clusters_amplitude[j] * np.array([np.sin((2 * np.pi * (k+
    phase_shift[j])) / float(per_clusters[j])) for k in range(n * 1440)])

        synthetic_traffic = np.add(synthetic_traffic, a2)


    #Fem el plot del trafic generat i del trafic original
    if verbose:
        line_ori_data, = plotter.plot(original_data, label='Original data')
        line_syn_traf, = plotter.plot(synthetic_traffic, label='Synthetic
    traffic')
        plotter.legend([line_ori_data, line_syn_traf], ['Original data', '
    Synthetic traffic'])
        plotter.title('Synthetic traffic')
        plotter.show()

    return [synthetic_traffic, per_clusters, clusters_amplitude, phase_shift]
```

Modelització de les components residuals

La nostra idea inicial en referència a aquesta secció era analitzar els residuals no explicats a partir de la detecció de components periòdiques de tal manera que, en cas que hi hagués correlació entre ells, construir un model ARIMA que ens permetés reconstruir-los. D'aquesta manera, hauríem inclòs una predicció dels residuals en el generador de tràfic que hem creat. Ens hem trobat amb l'inconvenient, però, de què per falta de temps, no hem estat capaços d'implementar un algoritme que determini amb precisió quins han de ser els paràmetres p i q que ha d'utilitzar el model ARIMA en cada cas, i per això hem acabat no incloent els residuals en el generador. Malgrat aquest inconvenient, aprofitarem per exposar els avanços que hem fet en aquest punt.

Si anomenem *residuals* als residuals que hem obtingut i partim de la premissa que segueixen un model ARIMA de paràmetres p, q i $d = 0$, ja que com les dades amb les quals treballem són estacionàries els residuals també ho seran, la funció ARIMA de la llibreria *statsmodels.tsa.arima.model* ens permet executar de manera senzilla del càlcul del model.

```
# Calcul del model ARIMA
model = ARIMA(residuals, order=(p, d, q))
model_fit = model.fit()
print(model_fit.summary())
```

La variable *model_fit* retorna tots els paràmetres que fan referència al model. Seguidament, hem realitzat una tria dels coeficients AR i MA amb p-valor menor a una significança fixada en 0.05, de tal manera que puguem determinar quins ens són útils de cara a la generació de dades que segueixin aquest patró.

Per tal de dur a terme aquesta tria dels coeficients, hem utilitzat la funció *residuals_correlation*, que rep com a input els paràmetres p, q i d i el model *model_fit* que hem construït, i retorna els coeficients significatius *phi* i *theta* que s'han seleccionat i que fan referència a les parts AR i MA respectivament, així com un dataframe *sigma2* que retorna informació dels residuals que no explica el model.

```
# Funcio que selecciona els parametres significatius pel model ARIMA
def residuals_correlation(p, d, q, model_fit):
    # Vectors on es guardaran els coeficients significatius
    phi = [] # Hi guardarem els coeficients AR
    theta = [] # Hi guardarem els coeficients MA

    # pvalors dels residuals del model
    arima_pvalues = model_fit.pvalues
    sigma2 = {}  # Estudia la variancia del residuals del model ARIMA. Si el
    pvalor del sigma2 es significatiu voldra dir que la distribucio dels
    residuals segueix una normal
    # Mirem si hi ha algun pvalor que ens indiqui correlacio entre els residus
    for i in range(len(arima_pvalues)):
        if i in range(0, p + q):
            if arima_pvalues[i+1] < significance:
                res_correlation = True
                if i < p:
                    phi.append(model_fit.arparams[i])
                else:
                    theta.append(model_fit.maparams[i - p])
        else:
            sigma2['pvalue'] = arima_pvalues[i]

    sigma2["mean"] = np.mean(residuals)
    sigma2["std"] = np.std(residuals)
    sigma2 = pd.DataFrame(sigma2)

    return [phi, theta, sigma2]
```

Un cop trobats aquests coeficients, el paquet *arima_process* de la llibreria *statsmodels.tsa*, permet la generació de dades que segueixin aquest determinat model.

3.3 Casos d'ús: Model dinàmic de detecció de dies atípics

Partint de la premissa que la recol·lecció real de tràfic no és estàtica, ja que diàriament es rep tràfic nou que pot incloure la incorporació de noves components, el que hem fet ha estat construir un predictor de tràfic que detecta els dies atípics i incorpora les noves components periòdiques que van apareixent. D'aquesta manera tenim un model dinàmic que es va actualitzant alhora que van apareixent comportaments no previstos en el tràfic.

Siguin t i n dos nombres enters tals que $t < n$, imaginem que partim d'un tràfic de dades de t dies. Tal com hem explicat en la secció anterior, la funció *traffic_generator* analitza aquests t dies de tràfic i a partir de la detecció de components periòdiques és capaç de generar n dies de tràfic sintètic.

Ara, però, el que farem serà, des del dia $t+1$ fins al dia n anar comparant el tràfic que es rep amb el tràfic sintètic que s'ha generat, amb l'objectiu de detectar si per un dia concret, el tràfic predit és significativament diferent del tràfic rebut. Aquesta comparació es farà a través del test chi-quadrat lleugerament modificat. Recordem que l'estadístic que es defineix en el test chi-quadrat és $\chi^2 = \frac{\sum_{i=1}^{k}(O_i - E_i)^2}{Ei}$. Existeix la problemàtica que quan els valors esperats són molt propers a zero, aquest estadístic pren valors molt grans i els resultats que s'obtenen en el test no són realistes. Hem determinat un llindar que hem fixat en $1e{-}3$ i per dur a terme el test, només hem considerat els valors esperats E_i tals que $|E_i| > 1e{-}3$ i els seus corresponents valors observats O_i. L'estadístic obtingut en el test segueix una distribució chi-quadrat amb $n - t$ graus de llibertat.

La funció que realitza aquest test s'anomena *chi2_test*. Els paràmetres que rep són els dos tràfics que volen comprar-se *observed* i *expected* i els paràmetres t i n que abans hem esmentat.

Després de realitzar el test chi-quadrat la funció retorna el valor de l'estadístic *chi2_value* que s'ha obtingut pel test i el seu corresponent pvalor *pvalue*.

```
# Test chi-quadrat modificat
def chi2_test(observed, expected, t, n):
    expected_aux = []
    observed_aux = []
    # Descartem tots els valors que son molt propers a zero
    for k in range(len(expected)):
        if abs(expected[k]) > 1e-3h:
            expected_aux.append(expected[k])
            observed_aux.append(observed[k])
    observed = observed_aux
    expected = expected_aux
    # Calculem l'estadistic del test
    chi2_value = [x - y for x, y in zip(observed, expected)]
    chi2_value = [i * i for i in chi2_value]
    chi2_value = np.divide(chi2_value, [abs(number) for number in expected])
    chi2_value = np.sum(chi2_value)
    # Calculem el pvalor associat a l'estadistic trobat
    pvalue = 1 - stats.chi2.cdf(chi2_value, n - t)

    return [chi2_value, pvalue]
```

Suposem que per al dia k-èssim, on k és un enter tal que $t < k < n$, el model detecta que els tràfics són significativament diferents. Aleshores, es torna a fer una anàlisi del tràfic rebut fins al dia k per tal de detectar-ne de nou totes les components periòdiques, i tornar a generar així, un tràfic sintètic de longitud n amb el que seguir comparant el tràfic que es va recollint.

La funció que analitza i genera tràfic de manera dinàmica s'anomena *dynamic_traffic_generator* i rep com a inputs el tràfic que es vol generar *original data* i els paràmetres t i n i *verbose* amb els que hem treballat fins ara.

```
# Generador dinamic de trafic
def dynamic_traffic_generator(original_data, t, n, verbose):
    # Generem un trafic sintetic de n dies de durada a partir de la deteccio de
    les component dels t primers dies
    [synthetic_traffic, per_clusters, clusters_amplitude] = traffic_generator(
    original_data, t, n, True)
    # Des del dia t + 1 fins al dia n comparem el trafic generat amb el trafic
    original
    for i in range(t+1, n):
        observed = synthetic_traffic[1440*i:1440*(i+1)]
        expected = amplitude_1[1440*i:1440*(i+1)]
        # Realitzem el test chi quadrat
        chi2_value, pvalue = chi2_test(observed, expected, t, n)
        # Si el pvalor del test informa d'un dia atipic, actualitzem el trafic
    sintetic amb les noves components trobades
        if pvalue < 0.05:
            [synthetic_traffic, per_clusters, clusters_amplitude] =
    traffic_generator(original_data, i, n, True)
```

Capítol 4

Resultats

La idea d'aquest capítol és presentar els resultats que s'han obtingut aplicant les metodologies exposades per tal de validar-les i alhora buscar quines són les seves limitacions. El que farem serà dividir els resultats en 3 blocs: en el primer d'ells, presentarem els resultats relacionats amb la detecció de les diverses components esmentades en els capítols anteriors i la corresponent reconstrucció que hem fet del tràfic. Reservarem el segon i el tercer bloc per presentar els casos d'ús; la detecció de dies atípics i el model dinàmic de predicció de tràfic.

4.1 Detecció de components periòdiques

Per tal de validar la metodologia i determinar les limitacions que presenta, el que farem serà observar com es comporta en model en diferents escenaris. Començarem treballant amb dades sinusoidals obtingudes mitjançant el generador presentat en la secció 3.1. Per tal de veure fins a quin punt és robust el model, per una banda, partirem d'unes periodicitats fixades i farem variar les amplituds de les components mentre afegim soroll a les dades. Per altra banda, utilitzant també dades obtingudes mitjançant el generador de 3.1, fixarem les amplituds de les components i farem variar les periodicitats. Finalment, sense perdre de vista que l'objectiu del treball és desenvolupar una tècnica que ens permeti treballar amb tràfic real, acabarem validant aquesta secció amb un exemple on treballem amb un exemple de tràfic realista que barreja diferents serveis com són vídeo a demanda i joc en línia, entre d'altres [7].

Detecció de components periòdiques de dos tràfics en funció de l'amplitud de les components

Considerem d'entrada dos tràfics 1 i 2 generats cadascun d'ells pels següents paràmetres:

```
# TRAFIC 1
data_params = [{
'base': {'cycles': 5, 'shifts': [250, 0, 50, 400],
'per': [1440, 2880, 7200, 14400], 'amp': [1, 2, 2, 1], 'sd': 0.0}}]
```

```
# TRAFIC 2
data_params = [{
'base': {'cycles': 5, 'shifts': [250, 0, 50, 400],
'per': [1440, 2880, 7200, 14400], 'amp': [0.25, 0.5, 0.5, 0.25], 'sd': 0.25}}]
```

L'única diferència entre els dos tràfics és l'amplitud de les seves components i el fet que en el tràfic 1 no hi ha soroll afegit, mentre que en el tràfic 2 s'ha agregat un soroll que pren valors en

el mateix interval que dues de les components que el conformen. El que farem per estudiar com es comporta el model respecte a aquest fenomen, serà comparar els tràfics 1 i 2 (Figura 4.1), les seves respectives FFT (Figura 4.2) i la reconstrucció que s'obté dels tràfics inicials a partir de les components periòdiques que es detecten (Figura 4.3).

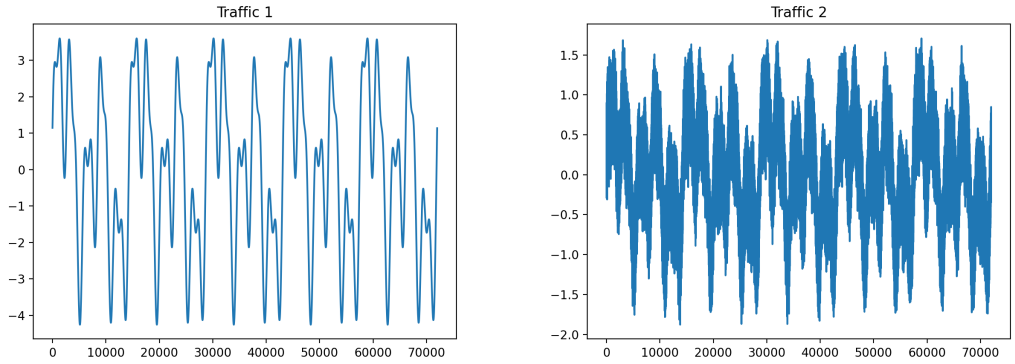

Figura 4.1: Tràfics de dades 1 i 2.

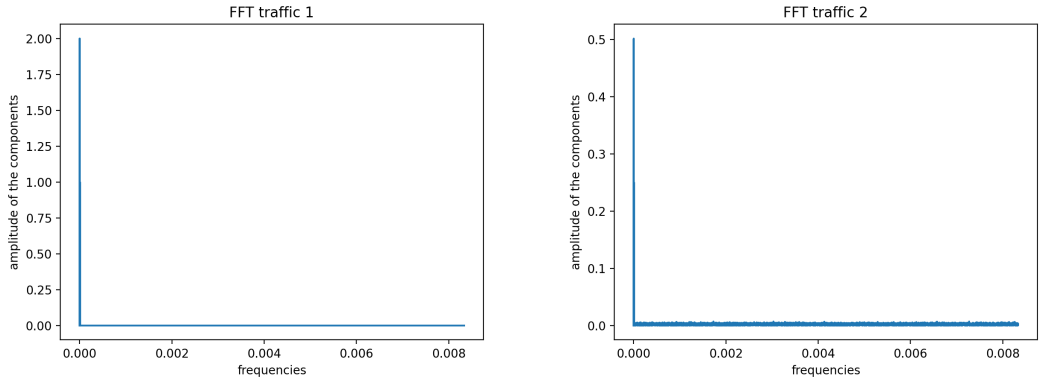

Figura 4.2: FFT dels tràfics 1 i 2.

Malgrat que a priori les dues transformades siguin similars (a excepció de l'amplitud de les components que les conformen), fixem-nos en el seu eix de les abscisses. Si observem amb detall, podem apreciar com l'eix de la FFT del tràfic 2 sembla més gruixut que el del tràfic 1. Aquest fet és conseqüència de què, pel segon tràfic, es detecten moltes components fictícies de petites amplituds a fi de poder explicar tot el soroll de fons que té incorporat.

Les components periòdiques que detecta el model en cadascun dels tràfics són les següents:

```
 # OUTPUT TRAFIC 1
'Nombre components:' 4
'Clusters periodicity:' [14400.  7200.  2880.  1440.]
'Clusters amplitude:' [1. 2. 2. 1.]
'Phase shift:' [ 1.74532925e-01  4.36332313e-02 -3.10862447e-15  1.09083078e+00]
'Component detection accuracy:' 100.0 %
```

```
 # OUTPUT TRAFIC 2
'Nombre components:' 9235
'Clusters periodicity:' [7.20000000e+04 1.80000000e+04 1.44000000e+04 ...
   2.00077808e+00 2.00061130e+00 2.00022225e+00]
'Clusters amplitude:' [0.00286889 0.00321831 0.24936616 ... 0.00504433
   0.00332563 0.00250397]
'Phase shift:' [ 0.62558876  3.86196979  0.17579841 ...  2.26179137  3.08794783
   -0.9764811]
'Component detection accuracy:' 95.7769 %
```

En el cas del tràfic 1 en el que no hi ha soroll afegit, el model detecta perfectament les components periòdiques que caracteritzen les dades i és capaç de reconstruir amb un 100% de precisió el tràfic inicial. Per altra banda, com abans hem mencionat, en l'anàlisi del tràfic 2, apareixen moltes components fictícies. Observem que en aquest cas, malgrat que la correlació respecte al tràfic inicial ha disminuït, el model és capaç d'explicar més d'un 95% del tràfic inicial a través de les components trobades.

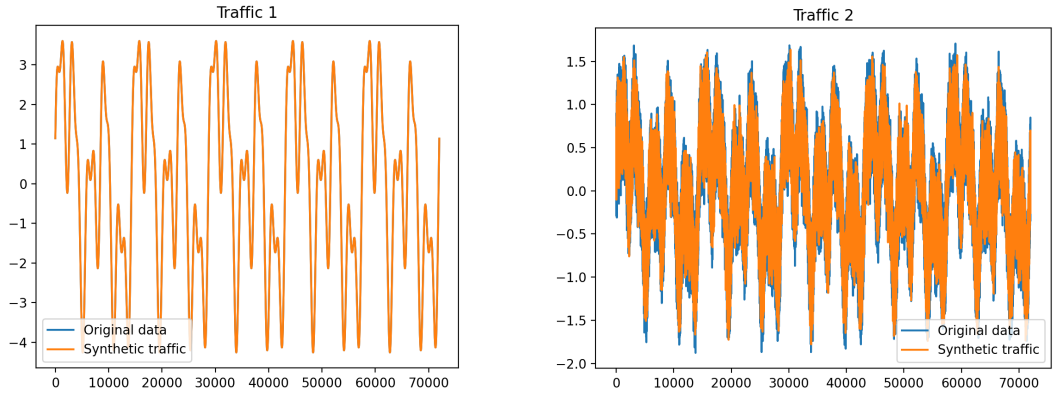

Figura 4.3: Reconstrucció dels tràfics 1 i 2.

Detecció de components periòdiques de dos tràfics en funció de la freqüència de les components

El que farem ara serà generar dos tràfics amb el mateix nombre de components, mateixes amplituds i mateixos desplaçaments de fase però amb periodicitats significativament diferents. Analitzarem com es comporta el model en aquests dos casos.

```
 # TRAFIC 3
data_params = [{
'base': {'cycles': 10, 'shifts': [0, 0, 0, 0, 0, 0, 0, 0],
'per': [60, 120, 180, 240, 360, 480, 720, 1440],
'amp': [1, 2, 2, 4, 1, 8, 2, 1], 'sd': 0.1}}]
```

```
 # TRAFIC 4
data_params = [{
'base':  {'cycles': 10, 'shifts': [0, 0, 0, 0, 0, 0, 0, 0],
'per': [600, 1200, 1800, 2400, 3600, 4800, 7200, 14400],
'amp': [1, 2, 2, 4, 1, 8, 2, 1], 'sd': 0.1}}]
```

A continuació presentem una comparació dels tràfics 3 i 4 (Figura 4.4) i també presentarem una comparació de les seves FFT (Figura 4.5).

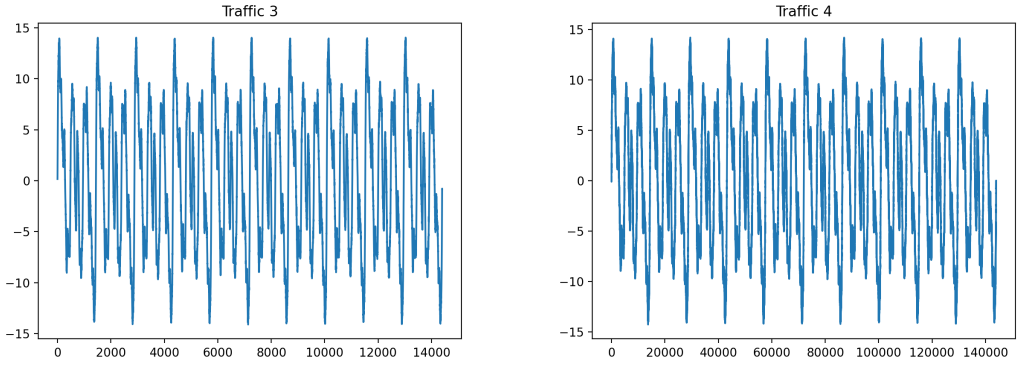

Figura 4.4: Tràfics de dades 3 i 4.

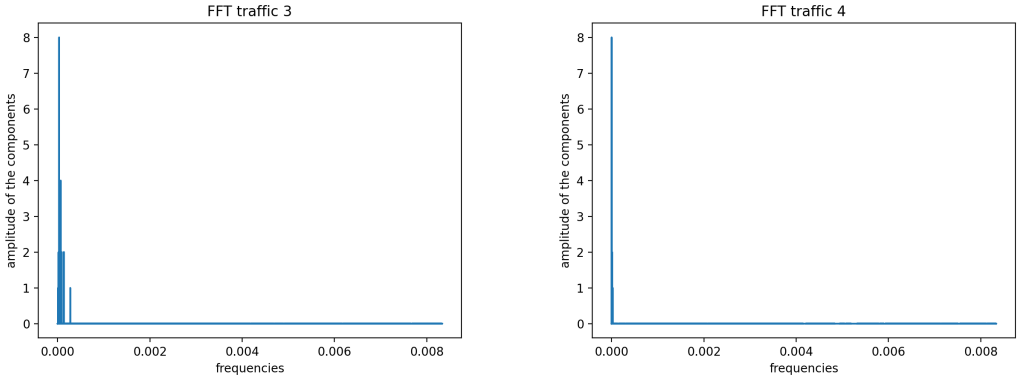

Figura 4.5: FFT dels tràfics 3 i 4.

Aprofitarem aquest exemple ampliar les transformades i veure com varia la freqüència de les components trobades en cada cas (Figura 4.6). Prendrem com a referència la component que té amplitud 8 en cadascun dels tràfics i en calcularem les freqüències associades. En el cas del tràfic 3, aquesta component té associada una periodicitat de 480 minuts, a la qual li correspon una freqüència de $\frac{1}{480*60} \simeq 3,47\text{e}{-5}$ Hz. En referència a la mateixa component del tràfic 4, té una periodicitat de 4800 minuts, i per això li correspon una freqüència de $\frac{1}{4800*60} \simeq 3,47\text{e}{-6}$ Hz. Si observem l'ordenada que li correspon al pic d'amplitud 8 de la FFT de cadascun dels tràfics representats en la Figura 4.6, podem observar com en els dos casos, aquesta quadra perfectament amb les freqüències que hem calculat. Aquest comportament és anàleg per a la resta de components que caracteritzen el tràfic.

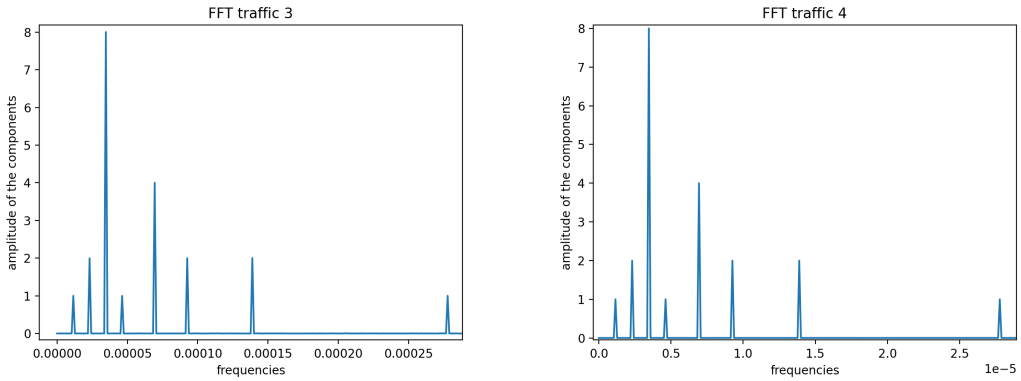

Figura 4.6: FFT ampliada dels tràfics 3 i 4.

```
 # OUTPUT TRAFIC 3
'Nombre components:' 8
'Clusters:' [1440.  720.  480.  360.  240.  180.  120.   60.]
'Clusters amplitude:' [1.00289517 2.00029063 8.0010207  0.999149   4.00065776
   2.00108368
 1.99957566 1.00183671]
'Phase shift:' [-0.00052873 -0.00029548  0.00021612 -0.00042891 -0.00037108
   0.00108326
  0.00062547  0.00146185]
'Component detection accuracy:' 99.9718 %
```

```
 # OUTPUT TRAFIC 4
'Nombre components:' 8
'Clusters periodicity:' [14400.  7200.  4800.  3600.  2400.  1800.  1200.
   600.]
'Clusters amplitude:' [1.00062396 2.00075758 7.99987207 1.00130725 4.00029144
   1.99990464
 2.00001532 0.99990187]
'Phase shift:' [-2.58795443e-04 -2.78572644e-04  9.66962179e-05 -3.06864678e-04
  1.65642540e-04 -2.73605028e-04 -8.52718963e-05 -5.59590257e-04]
'Component detection accuracy:' 99.9719 %
```

Observem que en els dos casos, malgrat haver inclòs un soroll en la generació de les dades, el model detecta perfectament les components periòdiques que conformen el tràfic, sense presentar cap diferència aparent respecte a la variació de les periodicitats.

Finalment, reconstruirem els tràfics 3 i 4 mitjançant la detecció de components periòdiques que s'han trobat en la FFT (Figura 4.7).

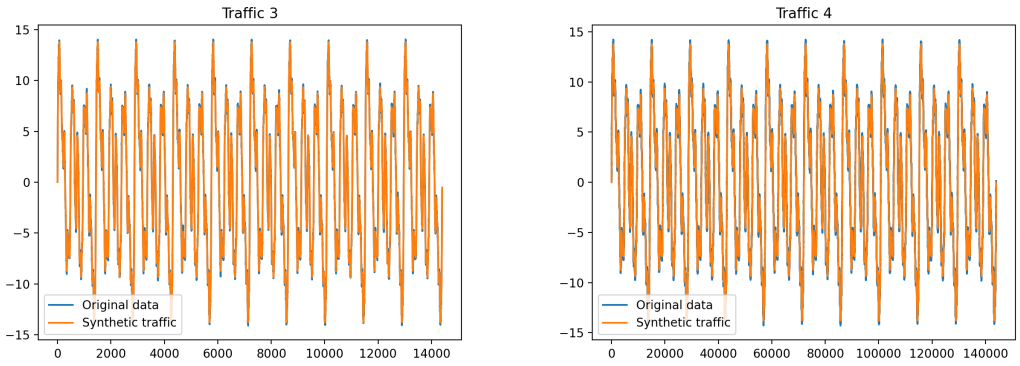

Figura 4.7: Reconstrucció dels tràfics 3 i 4.

Detecció de components periòdiques d'un tràfic de dades real

Considerem ara un tràfic de dades real que anomenarem tràfic 5 en el que es recull un exemple de tràfic real basat en [7], on es barregen serveis com vídeo a demanda, joc en línia, i sincronització de servidors en centres de processament de dades distribuïts. En aquest cas no sabem les funcions sinusoidals que el determinen, i no inclourem l'arxiu csv d'on s'han extret les dades a causa de la seva gran dimensionalitat. De la mateixa manera que en els casos anteriors, el que farem serà aplicar l'algoritme de la FFT per trobar les components periòdiques que el descomponen i utilitzar-les per reconstruir-lo.

Començarem representant aquest tràfic de dades i calculant-ne la FFT (Figura 4.8), i de la mateixa manera que en la resta de seccions, inclourem la reconstrucció que hem fet del tràfic original a partir de les components trobades mitjançant la FFT (Figura 4.9).

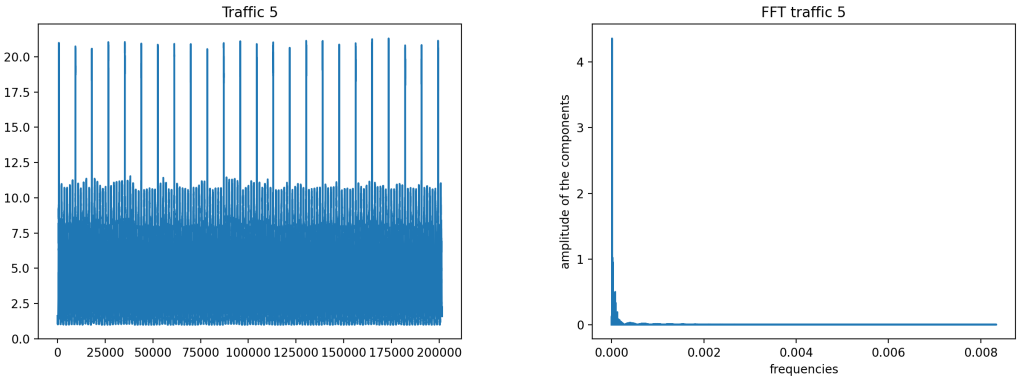

Figura 4.8: Tràfic real de dades i la seva FFT.

```
 # OUTPUT TRAFIC 5
'Nombre components:' 285
'Clusters periodicity:' [2.01599000e+05 1.00799500e+05 6.71996667e+04 ...
   3.74024119e+01 3.69228938e+01 3.64555154e+01]
'Clusters amplitude:' [0.02611362 0.02576082 0.03598097 ... 0.02486799
   0.02401817 0.02343899]
'Phase shift:' [ 0.54095847  1.08033407 -0.91270207 ... 4.64534522  2.97974763
   1.38034163]
'Component detection accuracy:' 99.4568 %
```

Notem que com en aquest cas, en el que el tràfic que es vol explicar és més complex, són necessàries moltes més components periòdiques per a caracteritzar-lo.

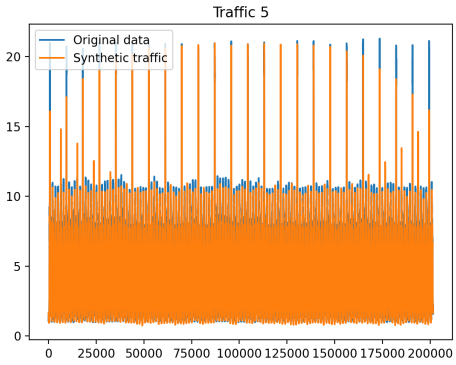

Figura 4.9: Reconstrucció del tràfic 5.

Malgrat que la correlació entre el tràfic sintètic predit i les dades originals sigui major al 99%, s'observa que els pics de major amplitud que es troben a l'inici i al final dels dos tràfics no es corresponen amb exactitud amb el tràfic original.

4.2 Detecció de dies atípics

Tot i que més endavant veurem una aplicació integrada de la detecció de dies atípics quan exposem els resultats que fan referència al predictor dinàmic de tràfic, començarem presentant de manera aïllada algun resultat que fa referència a aquesta secció.

Comparació de dos tràfics amb una parella de dies de dades permutats

Considerem un tràfic de dades que anomenarem tràfic 7 i que està generat a partir dels següents paràmetres:

```
# TRAFIC 7
data_params = [{
    'base': {'cycles': 5, 'shifts': [0, 0, 0], 'per': [1440, 2880, 14400],
    'amp': [1, 4, 2], 'sd': 0.0}
}]
```

Considerem ara un tràfic 8, generat pels mateixos paràmetres però en els quals s'ha intercanviat una parella de dies de dades; les dades del dia 13 s'han permutat amb les del dia 35. En la següent imatge (Figura 4.10) veiem una representació d'aquests dos tràfics.

Utilitzant el detector de dies atípics que compara els dos tràfics dia a dia mitjançant el test chi-quadrat modificat, es troba que efectivament els dos únics de dies pels quals existeixen discrepàncies entre els tràfics són en els dies 13 i 35 tal com era d'esperar. A continuació, presentem el detall dels dos dies de tràfic pels quals el generador ha detectat discrepàncies (Figura 4.11).

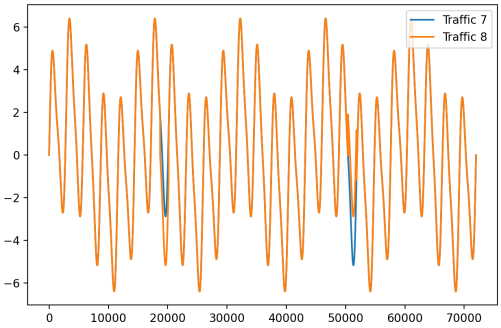

Figura 4.10: Tràfics de dades 7 i 8.

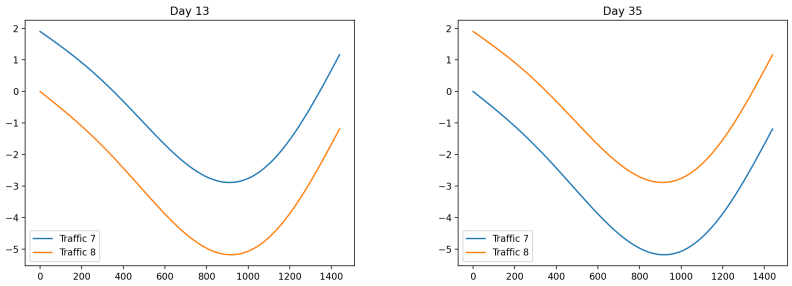

Figura 4.11: Tràfic dels dies que s'han detectat com atípics.

El estadístics que s'obtenen en el test per aquests dies són:

```
'DAY:'  13
'pvalue:'  0.0
'chi2_value:'  16304.762505418315

'DAY:'  35
'pvalue:'  0.0
'chi2_value:'  6171.135783840864
```

El p-valor obtingut en el test per aquests dos dies ens fa rebutjar la hipòtesi nul·la que els dos tràfics segueixen la mateixa distribució i ens permet classificar correctament els dies 13 i 35 com dies atípics.

Comparació de dos tràfics a partir de la generació de tràfic mitjançant la detecció de componenets periòdiques

Continuarem testejant la metodologia a través d'un altre exemple. Prenem ara 10 dies de dades del tràfic 7 i en calculem la FFT. Les components periòdiques que trobem són les següents:

```
'Nombre components:'  3
'Clusters periodicity:'  [14400.  2880.  1440.]
'Clusters amplitude:'  [2. 4. 1.]
'Phase shift:'  [-2.22044605e-16 -6.66133815e-16 -1.33226763e-15]
```

Un cop determinades les components periòdiques, les utilitzem per generar 50 dies de dades. Anomenarem tràfic 9 a aquest nou tràfic que hem generat a partir de les components que hem trobat.

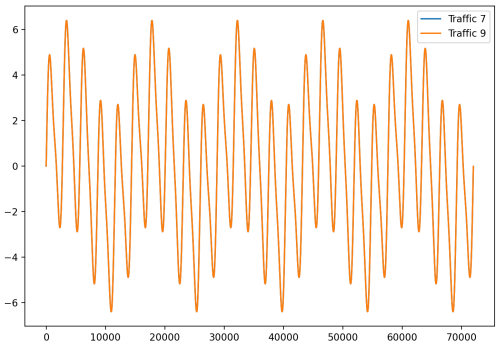

Figura 4.12: Reconstrucció del tràfic 7.

Si utilitzem el detector de dies atípics i comparem els dos tràfics dia a dia (com que hem utilitzat 10 dies del tràfic 7 per generar el tràfic 9, els compararem des del dia 11 fins al dia 50), no classifica cap dia com atípic tal com era d'esperar. En la següent imatge (Figura 4.12) observem com la predicció dels 50 dies de tràfic a partir de les components periòdiques extretes a partir dels 10 primers dies concorda perfectament amb el tràfic original.

4.3 Model dinàmic de predicció de tràfic

En la secció 4.1 hem vist que el detector de components periòdiques funciona tal com s'espera amb tràfics que tenen soroll afegit i que ens permet reconstruir amb precisió els tràfics originals. Així doncs, per motius de simplicitat, en aquesta secció reaprofitarem el tràfic 7 (Figura 4.10) per veure com es comporta el predictor dinàmic de tràfic. Recordem que la durada del tràfic 7 és de 50 dies, i està format per tres components periòdiques de 1, 2 i 10 dies amb diferents amplituds cadascuna d'elles i sense soroll afegit ni desplaçament de fase en cap component.

El que es farà a continuació, és prendre 1 dia de dades, detectar-ne les components periòdiques i utilitzar-les per predir 50 dies de dades. Des del dia 2 fins al dia 50, s'anirà comparant el tràfic 7 amb el tràfic que s'ha predit i quan per un dia concret, el model detecti que la parella de dies de tràfic que s'està comparant és significativament diferent, es tornaran a extraure les components periòdiques del tràfic original fins al dia en qüestió per actualitzar la predicció dels 50 dies de tràfic sintètic mitjançant les noves components trobades.

En la següent imatge (Figura 4.13) observarem com s'actualitza el predictor dinàmic de tràfic respecte a la detecció de dies atípics per cada dia de dades que analitza, i compararem el tràfic predit en cada cas amb el tràfic original. En la imatge hem volgut incloure el tràfic predit a partir de l'extracció d'un dia de dades, malgrat que per aquest cas particular, no s'hagi dut a terme el test chi-quadrat, ja que s'utilitza com a cas base per inicialitzar el model.

Observem que el model es comporta tal com s'espera, detectant com a dies atípics els dies entre el 2n i el 10è, ja que en aquest període de temps no ha recollit suficient informació per explicar el comportament del tràfic. Des del dia 11 fins al dia 50, és capaç de predir correctament el comportament del tràfic que analitza i no torna a detectar valors atípics en cap cas. En l'última imatge (Figura 4.14) veiem com per l'últim dia de dades que el model detecta com atípic (dia 10), la predicció del tràfic coincideix a la perfecció amb el tràfic original analitzat.

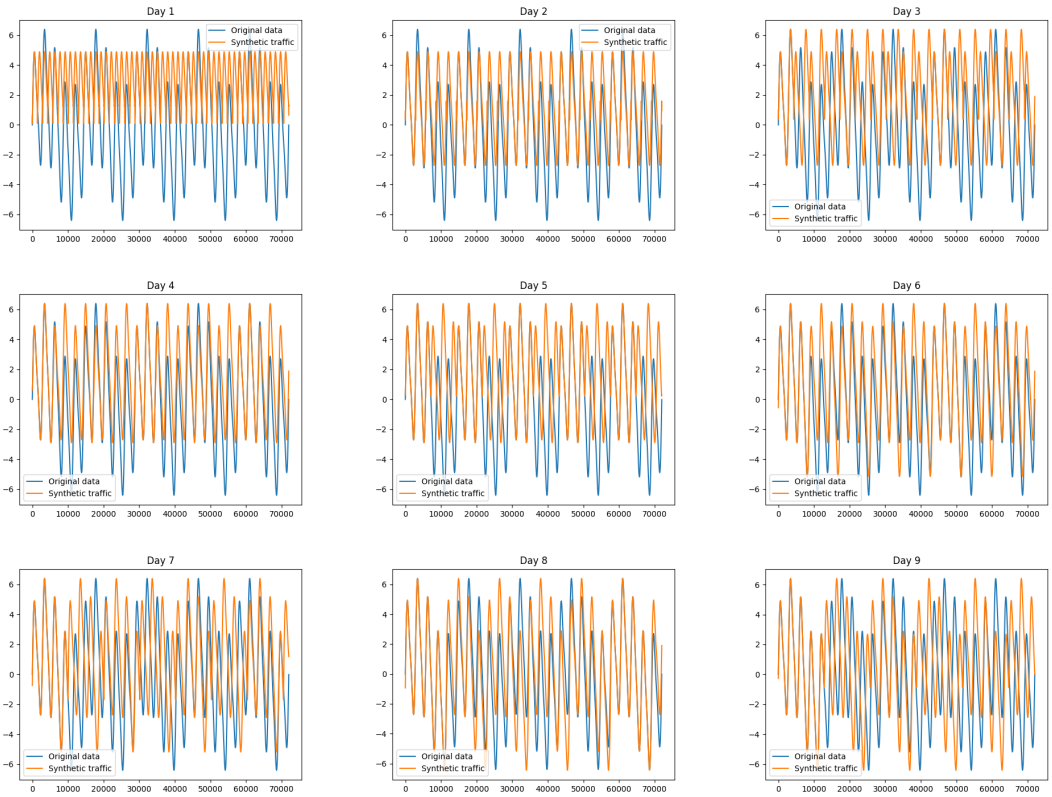

Figura 4.13: Comparació del tràfic predit amb el tràfic original.

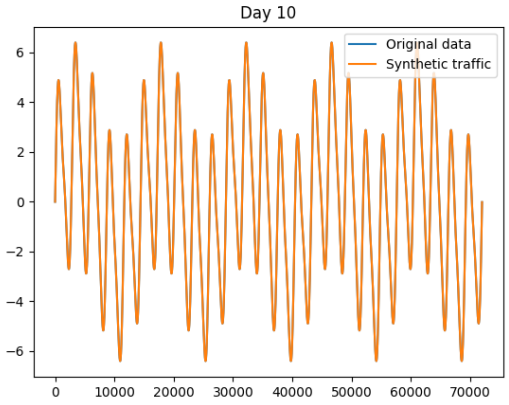

Figura 4.14: Reconstrucció del tràfic 7.

Hem estat capaços de construir doncs, un model que caracteritza les components periòdiques d'un tràfic de dades en diversos escenaris, sense veure's influenciat per l'amplitud ni la freqüència de les components periòdiques que conformen el tràfic, i hem pogut validar el model analitzant un exemple de tràfic de dades real. Per altra banda, també hem implementat un detector de dies de tràfic atípics que no només és capaç de detectar quan apareixen comportaments desconeguts en les dades, sinó que també quan aquests comportaments no succeeixen en el moment previst, com hem vist en la comparació de dos tràfics amb dos dies permutats. Finalment, hem integrat aquestes dues metodologies per construir un predictor dinàmic de tràfic mitjançant la detecció

de components periòdiques per cada dia de dades pel qual es detecten comportaments atípics, i hem comprovat que aquest predictor funciona a la perfecció quan es tracta d'analitzar tràfics senzills.

Capítol 5

Conclusions

En aquest projecte hem construït un model que és capaç de desagregar un tràfic de dades en les diferents components periòdiques que el caracteritzen mitjançant la transformada de Fourier. Hem comprovat com el model funciona bé en diferents escenaris, començant per l'anàlisi de tràfics senzills fins al processament d'un tràfic de dades real. Aquesta caracterització del tràfic en les components periòdiques que el conformen ens ha permès reduir la dimensionalitat d'un tràfic de gran volum en un espai molt més petit de tal manera que la pèrdua d'informació ha estat menor a l'1%. Ens ha faltat refinar la cerca dels paràmetres p i q del model ARIMA, per tal de poder analitzar els residuals no explicats a partir de les components periòdiques que s'han trobat.

Per altra banda, també hem pogut construir un generador de tràfic sintètic a partir de l'extracció de les components trobades. A més a més, hem integrat el predictor de tràfic amb un detector de dies atípics, de tal manera que el tràfic s'analitza de manera dinàmica i es va actualitzant el predictor respecte els comportaments desconeguts que van apareixent.

Finalment, mencionar que aquest TFG s'emmarca dins de les línies de recerca en Intel·ligència Artificial dutes a terme pel Grup de Comunicacions Òptiques (GCO) de la Universitat Politècnica de Catalunya (UPC). El GCO forma part com a membre fundador del Centre de Comunicacions Avançades de Banda Ampla (CCABA), centre de referència en sistemes de comunicacions de l'àmbit tecnològic 5G i posterior.

Valoració personal

Un dels punts que m'agradaria destacar, és que durant el transcurs d'aquest projecte he après a programar amb *Python*, llenguatge de programació amb el que no havia treballat fins al moment i que té un gran potencial de cara a treballar amb l'anàlisi i processament de dades. També he pogut familiaritzar-me amb un ampli ventall de conceptes del camp de teoria de la senyal i l'anàlisi de sèries temporals que desconeixia i també he après amb profunditat totes les idees que fan referència a la transformada de Fourier i a les seves aplicacions. Per altra banda, com que a l'inici del projecte no teníem clar quines eines ens serien útils de cara a la implementació dels models, vaig treballar amb exemples senzills de xarxes neuronals, i per això també he adquirit coneixements superficials respecte a aquest camp, malgrat que més tard vam veure que no seria necessari aplicar-los en el desenvolupament del nostre model. Una altra matèria transversal que he hagut de treballar i en la que he millorat força, és en la planificació i execució d'un projecte que ha durat diversos mesos, de tal manera que he hagut d'aprendre a organitzar-me i a ser constant en la recerca.

En referència a les eines que m'ha proporcionat el grau que m'han ajudat a desenvolupar aquesta tasca, destacar sobretot la capacitat de ser autodidacta, que m'ha permès afrontar pel

meu compte la recerca i aprenentatge de nous conceptes, així com el rodatge que havia adquirit treballant amb diferents llenguatges de programació, que m'ha fet molt més fàcil l'aprenentatge de *Python*. Destacar també totes les nocions d'estadística que havia assolit i he pogut aplicar i consolidar, a més de tot el que fa referència al camp de l'anàlisi funcional i les sèries de Fourier, sobre les quals estan fonamentades totes les idees que s'amaguen darrere de la transformada.

Finalment mencionar que el punt que se m'ha fet més difícil ha estat el de la planificació de la feina, ja que d'entrada no teníem clars els passos a seguir pel desenvolupament del model i ha estat necessària una recerca paral·lela a la construcció del model en base als resultats que hem anat obtenint. Sense el suport i els coneixements dels meus tutors, aquesta feina hauria estat molt més difícil o impossible.

Feina futura

Un cop finalitzat l'estudi i en base als resultats obtinguts, una proposta de feina futura que podria realitzar-se és la següent:

- Refinar la cerca dels paràmetres p i q del model ARIMA en funció de les dades que s'estan estudiant.
- Millorar el model de generació de tràfic sintètic utilitzant el model ARIMA per incloure-hi el perfil residuals que no estan contemplats en les components periòdiques.
- Ampliar els models per tal de que siguin capaços d'analitzar tràfics en els quals les dades tinguin una tendència de creixement o decreixement.
- Seguir desenvolupant el detector de dies atípics de tal manera que per cada irregularitat que es detecti es dugui a terme un anàlisi exhaustiu que permeti determinar amb claredat si es tracta d'una anomalia o d'una nova component que cal incorporar en la generació del tràfic.

Bibliografia

[1] J. Martin, Y. Fu, N. Wourms & T. Shaw (1 de gener de 2013) *Characterizing Netflix bandwidth consumption.* https://www.researchgate.net/publication/261056157_Characterizing_Netflix_bandwidth_consumption

[2] METRO High bandwidth, 5G Application-aware optical network, with edge storage, compUte and low Latency (METRO-HAUL), H2020-ICT-2016-2, 2017-2020.

[3] D. Rafique & L. Velasco, *Machine Learning for Optical Network Automation: Overview, Architecture and Applications*, (Invited Tutorial) IEEE/OSA Journal of Optical Communications and Networking (JOCN), vol. 10, pp. D126-D143, 2018.

[4] F. Morales, Ll. Gifre, F. Paolucci, M. Ruiz, F. Cugini, P. Castoldi & L. Velasco, *Dynamic Core VNT Adaptability based on Predictive Metro-Flow Traffic Models*, IEEE/OSA Journal of Optical Communications and Networking (JOCN), vol. 9, pp. 1202-1211, 2017.

[5] Cultura científica - Unidad didáctica FFT. http://www.culturacientifica.org/textosudc/unidad_didactica_fft.pdf

[6] By Laerd Statistics - https://statistics.laerd.com/statistical-guides/pearson-correlation-coefficient-statistical-guide.php, CC BY-SA 4.0, https://commons.wikimedia.org/w/index.php?curid=84255641

[7] M. Ruiz, F. Coltraro & L. Velasco, *CURSA-SQ: A Methodology for Service-Centric Traffic Flow Analysis*, IEEE/OSA Journal of Optical Communications and Networking (JOCN), vol. 10, pp. 773-784, 2018.

[8] C. Márquez, M. Gramaglia, M. Fiore, A. Banchs & Z. Smoreda, *Identifying Common Periodicities in Mobile Services Demands with Spectral Analysis*

[9] López Ortega, M. (Juny, 2016) *Una perspectiva histórica de los métodos de Fourier.* Treball de final de Grau Matemàtiques - Universitat de Granada

[10] Sandro Salsa. *Partial Differential Equations in Action: From Modelling to Theory.* Springer, 2015.

[11] Trinidad, F. A. (Desembre, 2017) *Transformada de Fourier y su aplicación en procesamiento digital de imágenes.* Treball de final de Grau Matemàtiques - Universitat de Puebla

[12] C. Batlle & E. Fossas (2002). *Apunts d'anàlisi real.*

[13] T. Yiu (7 de març de 2020). *Understanding Maximum Likelihood Estimation (MLE).* Towards data science. https://towardsdatascience.com/understanding-maximum-likelihood-estimation-mle-7e184d3444bd

[14] P. Delicado, G. Gómez & J. Graffelman (9 de Febrer de 2015) *Estadística – Grau en Matemàtiques Tercer Curs*. Universitat Politècnica de Catalunya

[15] N. Ye & Q. Chen (2001). *An anomaly detection technique based on a chi-square statistic for detecting intrusions into information systems.* Quality and Reliability Engineering International. 17. 105 - 112. 10.1002/qre.392.

[16] R. Rahul, K. Zubair & Khan M. H. (3 de març de 2012). *Network Anomalies Detection Using Statistical Technique : A Chi- Square approach.* Department of CSE, Invertis University Bsareilly India and Department of CSE, IET, UPTU Lucknow, India

[17] Shumway, Robert H., and David S. Stoffer. Time Series Analysis and Its Applications: With R Examples. New York: Springer, 2006.]