UNIVERSITAT POLITÈCNICA DE VALÈNCIA

DSIC DEPARTAMENT DE SISTEMES INFORMÀTICS I COMPUTACIÓ

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Dpto. de Sistemas Informáticos y Computación

Previsión sobre series temporales: el caso Logifruit

Trabajo Fin de Máster

Máster Universitario en Inteligencia Artificial, Reconocimiento de Formas e Imagen Digital

AUTOR/A: García Cucó, Arnau

Tutor/a: Botti Navarro, Vicente Juan

Cotutor/a: Palanca Cámara, Javier

CURSO ACADÉMICO: 2023/2024

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

DEPARTAMENT DE SISTEMES INFORMÀTICS I COMPUTACIÓ

Màster universitari en intel·ligència artificial, reconeixement de forma i imatge digital

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

“Previsió sobre sèries temporals: el cas Logifruit.”

TFM

Autor/a:
Arnau Garcia i Cucó

Tutor/a:
Vicente Botti Navarro
Javier Palanca i Cámara

VALÈNCIA, 2024

Abstract

Time series prediction is an area in machine learning relevant in many fields, since weather forecasts to sales pronostics. This project, encompassed inside the master's degree thesis of MUIARFID from the Universitat Politècnica de València, aims to apply some forecast methodologies to the Logifruit's logistics. The main goal is to develop a model that predicts the amount of boxes needed to be cleaned within a week, using the RMSE and SMAPE error metrics. To achieve such goal, it has been applied numerous approaches, since the more classic ones such as ARIMA or exponential smoothing, to more complex models of deep neural networks. The results prove that the statistical models perform better in this case than the deep learning ones, most likely due to the lack of data across time. It is also emphasized the pretrained models' efficiency and their benefits when applied to problems without much data.

Keywords Time series; machine learning; deep learning; ARIMA; exponential smoothing; RNN; GRU; LSTM; CNN; Transformer; pretrained models

Resumen

La predicción de series temporales es un área del aprendizaje automático relevante en muchos campos, desde pronósticos climáticos hasta previsión de ventas. Este proyecto, englobado dentro del trabajo de final de máster del MUIARFID de la Universitat Politècnica de València, pretende aplicar algunas metodologías de predicción de series temporales a la logística de la empresa Logifruit. El objetivo es desarrollar un modelo que prediga la cantidad de cajas a limpiar con una semana de antelación usando las métricas de error RMSE y SMAPE. Para conseguirlo, se han usado numerosas aproximaciones, desde las más clásicas como el ARIMA o el suavizado exponencial, hasta modelos más complejos de redes neuronales profundas. Los resultados demuestran que los modelos estadísticos funcionan mejor en este caso que los de redes profundas, muy probablemente debido a la falta de datos a lo largo del tiempo. También se recalca la eficiencia de los modelos preentrenados y los beneficios en problemas sin demasiados datos.

Palabras clave Series temporales; aprendizaje automático; aprendizaje profundo; ARIMA; suavizado exponencial; RNN; GRU; LSTM; CNN; Transformer; modelos preentrenados

Resum

La predicció de sèries temporals és una àrea de l'aprenentatge automàtic rellevant en molts camps, des de pronòstics climàtics fins a previsió de vendes. Aquest projecte, englobat dins del treball de final de màster del MUIARFID de la Universitat Politècnica de València, pretén aplicar algunes metodologies de predicció de sèries temporals a la logística de l'empresa Logifruit. L'objectiu és desenvolupar un model que prediga la quantitat de caixes a netejar amb una setmana d'antelació utilitzant les mètriques d'error RMSE i SMAPE. Per a aconseguir-ho s'han emprat nombroses aproximacions, des de les més clàssiques com

l'ARIMA o el suavitzat exponencial, fins a models més complexos de xarxes neuronals profundes. Els resultats demostren que els models estadístics funcionen millor en aquest cas que els de xarxes profundes, molt probablement a causa de la manca de dades més prolongadament amb el temps. També es recalca l'eficiència dels models preentrenats i els beneficis en problemes sense gaires dades.

Paraules clau Sèries temporals; aprenentatge automàtic; aprenentatge profund; ARIMA; suavitzat exponencial; RNN; GRU; LSTM; CNN; Transformer; models preentrenats

Contents

1. Introducció **9**
- 1.1. Motivació i objectius . . . 9
- 1.2. Estructura de la memòria . . . 10

2. Revisió bibliogràfica **11**
- 2.1. Models clàssics . . . 11
  - 2.1.1. Models de suavitzat . . . 12
  - 2.1.2. Models ARIMA . . . 12
- 2.2. Xarxes neuronals . . . 14
  - 2.2.1. Xarxes neuronals recurrents . . . 15
  - 2.2.2. Mecanismes d'atenció . . . 19
  - 2.2.3. Transformer . . . 20
- 2.3. Altres models competitius . . . 27

3. Anàlisi del problema **29**
- 3.1. Anàlisi de qualitat de dades . . . 29
- 3.2. Anàlisi univariant de dades . . . 31
- 3.3. Anàlisi multivariant . . . 33

4. Metodologia **37**
- 4.1. Models estadístics . . . 38
  - 4.1.1. Models de suavitzat exponencial . . . 38
  - 4.1.2. Models ARIMA . . . 39
  - 4.1.3. Models ARIMAX . . . 42
- 4.2. Models de xarxes recurrents . . . 43
- 4.3. Models basats en el Transformer . . . 45
- 4.4. Models de xarxes convolucionals . . . 47
- 4.5. Models preentrenats . . . 49
- 4.6. Models Light-GBM . . . 49

5. Resultats **51**
- 5.1. Models estadístics . . . 51
  - 5.1.1. Models de suavitzat exponencial . . . 51
  - 5.1.2. Models ARIMA . . . 52
  - 5.1.3. Models ARIMAX . . . 53
- 5.2. Models de xarxes recurrents . . . 56
- 5.3. Models basats en el Transformer . . . 60
- 5.4. Models de xarxes convolucionals . . . 65
- 5.5. Models preentrenats . . . 68

5.6. Models Light-GBM 69
5.7. Comparació de models 70

6. Conclusions **73**

A. Objectius de desenvolupament sostenible **83**

Agraïments

Agraïments a ValGrAI pel seu finançament en el Màster Universitari en Intel · ligència Artificial, Reconeixement de Formes i Imatge Digital, a la meua família pel seu suport incondicional i als meus tutors del TFM pels seus consells i ajuda inestimable.

1. Introducció

Després de la pandèmia de la COVID-19, el camp de la intel · ligència artificial ha tingut un auge important [1], degut, molt probablement a la democratització de la IA amb projectes com el ChatGPT [2]. Tanmateix, dins de la IA no només existeixen els models generatius, sinó que hi ha moltes altres ferramentes, entre les quals destaquen les prediccions a futur.

El pronòstic de sèries temporals és una tasca crítica en nombroses àrees i dominis. Des de predir l'energia que es consumirà a una xarxa elèctrica la setmana vinent, fins a pronosticar quines seran les vendes d'una empresa el mes següent, en nombroses aplicacions es requereix un bon modelat de les sèries temporals.

Aquest és un procés complex que inclou nombroses variables, com els patrons que es repeteixen al llarg del temps, variables exògenes que puguen tindre efectes sobre els nostres models i valors predits. Existeixen nombrosos models que s'han utilitzat i s'utilitzen per a resoldre aquests problemes. Per un costat, es presenten els models estadístics, amb els suavitzats i l'ARIMA com a principals exponents. D'altra banda, l'altre grup de models més emprat ara per ara, són les xarxes neuronals recurrents i, més recentment, els anomenats Transformers.

1.1. Motivació i objectius

L'empresa Logifruit és una empresa proveidora de serveis logístics especialitzada en la gestió d'envasos reutilitzables [3]. Busquen satisfer les necessitats d'envasat i transport dels seus clients mitjançant solucions logístiques que asseguren la màxima qualitat en el servei. Entre els seus clients destaca Mercadona, una de les principals empreses de supermercats físics i en línia d'Espanya.

Aquesta empresa des de fa un cert temps té certa relació amb la Universitat Politècnica de València (UPV d'ara endavant) i aquesta associació ha desembocat en que es requerisquen els serveis de la UPV per a multitud de tasques. Al 2023 es va plantejar un projecte en el marc d'una beca de col · laboració en la qual es buscava predir a una setmana vista la quantitat de caixes que havien de netejar cada dia per tal de poder reutilitzar-les.

Així doncs, l'objectiu d'aquest treball final de màster serà trobar en un període de sis mesos el model que millor s'ajuste a la sèrie temporal present utilitzant com a mesures d'error les conegudes mètriques RMSE y SMAPE.

Aquest objectiu es pot descompondre en d'altres més menuts:

- Realitzar un anàl · lisi descriptiu de les dades.
- Realitzar una revisió bibliogràfica dels models més rellevants actualment.
- Desenvolupar un o més models univariants de suavitzat sobre la sèrie.
- Desenvolupar un o més models SARIMA i SARIMAX que integren les dades disponibles.

- Desenvolupar un o més models basats en xarxes neuronals recurrents que integren les dades disponibles.
- Desenvolupar un o més models convolucionals que integren les dades disponibles.
- Desenvolupar un o més models Transformers que integren les dades disponibles.

1.2. Estructura de la memòria

Aquesta memòria va a dividir-se en les parts que s'especifiquen a continuació. A la primera part, ja explicada, s'introduirà el treball i s'explicaran els objectius i la motivació de desenvolupar aquest treball.

Al segon capítol parlarem sobre què és realment una sèrie temporal i quins són els mecanismes i models més comuns que s'utilitzen per a estudiar aquest tipus de dades.

A la tercera part desenvoluparem el context del nostre problema. Explicarem quines són les dades que tenim, d'on s'han extret i quines són les característiques i patrons que es poden observar amb un simple anàl · lisi descriptiu.

Al quart capítol explicarem formalment quina és la metodologia emprada a l'experimentació. És a dir, parlarem sobre les mètriques que hem utilitzat per a validar els models i com s'han desenvolupat els experiments que els estudien. També desenvoluparem teòricament els mateixos models i quines eren les hipòtesis inicials sobre els resultats que extrauriem.

A la cinquena part presentarem els resultats obtinguts i explicarem quin o quins són els millors models per a predir la quantitat de caixes que s'haurien de netejar.

A l'última secció desenvoluparem les conclusions del projecte, interpretant els resultats i parlarem sobre quins treballs es podrien dur a terme en el futur.

2. Revisió bibliogràfica

Parlem ara del context que envolta la predicció de sèries temporals a la literatura. Però abans de res, hem d'explicar què és una sèrie temporal. Una sèrie temporal en la seua forma més simple és una seqüència ordenada de dades d'una variable observada al llarg del temps en intervals regulars. Es tracta, doncs, d'un procés estocàstic, és a dir, d'un seguit de variables aleatories les característiques de les quals fluctuen amb el temps. Aquest procés podrà ser estacionari en cas que les funcions de mitjanes i de variàncies d'aquestes característiques es mantinguen constants al llarg del temps o no estacionari en cas que aquesta condició no es complisca.

Freqüentment s'assumeix que les sèries temporals consten de cinc components fonamentals: el nivell, la tendència, la estacionalitat, els cicles i la component aleatòria [4]. El nivell i la tendència parlen sobre el comportament general de la sèrie. El nivell és el valor mitjà de la sèrie, mentre que la tendència és la variació del nivell al llarg de la sèrie. És a dir, és el comportament de la sèrie a llarg del temps. D'altra banda, tant la estacionalitat i els cicles parlen sobre els comportaments repetitius de la sèrie. La seua diferència és que la estacionalitat són els comportaments a curt terme i els cicles són els comportaments a llarg terme. Finalment, la component aleatòria és aquella que explica els moviments a l'atzar de la sèrie al voltant de la tendència.

Tanmateix, s'ha comprovat en diversos estudis, com [5], que els mètodes que s'enfoquen en prediccions tenint només en compte la mateixa sèrie, fallen quan se succeeixen certs events que canvien la sèrie, com ara poden ser festivitats o vacances. Per això és necessari tindre en compte variables externes a la pròpia sèrie, que anomenem exògenes.

2.1. Models clàssics

Si assumim que el temps linial, podem enfocar la predicció de tres maneres. Els models projectius [6] assumeixen que l'entorn es manté constant i tenen com a objectiu obtindre previsions utilitzant l'extrapolació de com la sèrie es comportava al passat. És a dir, utilitzen dades passades per a predir el futur. D'altra banda, els models prospectius es basen en que l'entorn va a variar constantment i buscaran identificar els patrons de canvi a l'entorn per a definir com es comportarà la sèrie en un futur llunyà. Finalment, els models integrals utilitzaran les dades passades per a el · laborar pronòstics del comportament de la sèrie i utilitzaran els patrons de canvi de l'entorn per a corregir aquestes prediccions. En el nostre cas desenvoluparem un model projectiu, ja que ens interessa estudiar com es comportarà la sèrie només dins de set dies.

Els mètodes de predicció projectius són molt variats però generalment es poden agrupar en tres tipus. Els mètodes qualitatius empren judicis de valor d'experts per a predir el comportament futur de la sèrie. D'altra banda, els mètodes estadístics utilitzen dades històriques de la sèrie per a predir el seu comportament. Finalment els models causals relacionen les variables

entre elles per a obtindre una predicció sense tindre en compte el factor temporal. En aquest apartat ens enfocarem en els mètodes estadístics, que han demostrat fins i tot en recents investigacions i competicions, com la M5 [5], que continuen tenint un paper rellevant a l'hora de realitzar prediccions precises de sèries temporals. Nosaltres en concret ens concentrarem en anal · litzar els models de suavitzat exponencial i els models basats en ARIMA.

2.1.1. Models de suavitzat

Els models de suavitzat foren els primers models el · laborats per a sèries temporals i estaven basats en els models de mitjanes mòbils [7]. És a dir, utilitzen els P punts més pròxims de la sèrie per a obtindre un punt mitjà.

$$AM(2K+1)_t = \frac{\sum_{k=1}^{2K+1} x_{t-K-1+k}}{K}$$

on x_t és el valor de la sèrie en l'instant de temps t i $P = 2K + 1$

Existeixen diversos tipus de models de suavitzat, evolucionats com a la suma ponderada de les mitjanes mòbils [7]. Els models de suavitzat exponencial simple [8] utilitzen les previsions anteriors i els seus errors per a corregir el nou pronòstic. D'altra banda, els models dobles [8] consideren la tendència i permet establir prediccions a llarg terme. Finalment, el model de suavitzat exponencial triple també anomenat Holt-Winters [9] considera tant la tendència com la estacionalitat, amb la qual cosa es poden establir millors pronòstics.

$$F_{t+m} = (S_t + m \cdot b_t) \cdot E_{t+m-s}$$
$$S_t = \alpha \frac{x_t}{E_{t-s}} + (1-\alpha)(S_{t-1} + b_{t-1})$$
$$b_t = \beta(S_t - S_{t-1}) + (1-\beta)b_{t-1}$$
$$E_t = \gamma \frac{x_t}{S_t} + (1-\gamma)E_{t-s}$$

on S_t representa el nivell, m indica quants intervals a futur volem tindre en compte, b_t indica la tendència de la sèrie a l'instant t i E_t indica la estacionalitat del model a l'instant t amb cicle estacional s. Es considera que α, β, γ són hiperparàmetres a optimitzar compresos entre 0 i 1.

2.1.2. Models ARIMA

Tal i com hem explicat a l'inici del capítol 2, les sèries temporals són processos estocàstics molt sovintment autocorrelacionats. És a dir, els valors futurs estan força correlacionats amb els valors passats. Els models ARIMA [10], [7] (Auto-regressive integrated moving averages models) busquen utilitzar aquesta informació per millorar els models de suavitzat. Així doncs, els models ARIMA compten principalment amb dues parts: un model autorregressiu i un model de mitjanes mòbils.

Els models autorregressius busquen fer servir la autocorrelació per a fer un pronòstic de la serie a l'instant t (X_t). D'aquesta manera, un model autorregressiu d'ordre p s'expressaria com a la suma entre un sumatori dels p valors previs de la sèrie ponderats segons uns hiperparàmetres ϕ i la component aleatoria (a) en l'instant t de la predicció.

$$\hat{X}_t = \sum_{i=1}^{p}(\phi_i X_{t-i}) + a_t$$

Si considerem l'operador retard $BX_t = X_{t-1}$, podem expressar el polinomi autorregressiu d'ordre p de manera compacta.

$$\phi_p(B) = 1 - \sum_{i=1}^{p}(\phi_i B^i)$$

D'aquesta manera, podem notar que l'expressió $\phi_p(B)X_t = a_t$ és equivalent a un model autorregressiu d'ordre p.

La segona part del model, la part de mitjanes mòbils, busca predir la sèrie en base a la seua component aleatòria. D'aquesta manera, un model de mitjanes mòbils d'ordre q s'expressa en funció d'uns hiperparàmetres θ i una constant c a estimar.

$$\hat{X}_t = c + a_t - \sum_{j=1}^{q}(\theta_j a_{t-j})$$

$$\theta_q(B) = 1 - \sum_{j=1}^{q}(\theta_j B^j)$$

$$X_t = \theta_q(B)a_t$$

Així doncs, un model ARMA, que compren el model autorregressiu d'ordre p i el model de mitjanes mòbils d'ordre q s'expressaria com a un sumatori entre ambdós models.

$$\hat{X}_t = c + \sum_{i=1}^{p}(\phi_i X_{t-i}) + a_t - \sum_{j=1}^{q}(\theta_j a_{t-j})$$

$$\phi_p(B)X_t = c + \theta_q(B)a_t$$

Ara bé, aquest model té una sèrie de requeriments, com són la estacionarietat i aquestes condicions no sempre es compleixen en les sèries reals. Per solucionar aquest inconvenient, l'ARIMA proposa un decalament en la sèrie. És a dir, en compte de predir la sèrie temporal es busca predir la variació de la sèrie. Per a això s'inclou un operador de diferència regular (∇) d'ordre d, definit segons l'operador retard, de manera que el model ARIMA$_{p\times d\times q}$ es defineix respecte a aquesta variació i no respecte a la sèrie original.

$$\nabla^d = (1 - B)^d$$

$$\phi_p(B)\nabla^d X_t = c + \theta_q(B)a_t$$

Aquest model es pot millorar incloent la component estacional definint un segon model ARIMA sobre l'original, però tenint en compte un decalat igual a la longitud del període estacional (S). És a dir, si es repeteix un patró cada setmana, caldrà tindre en compte aquesta longitud. Així doncs, un model SARIMA$_{(p\times d\times q)\times(P\times D\times Q)_S}$ es defineix sobre els polinomis de la part regular de l'ARIMA ($\phi_p(B)$, $\theta_q(B)$) i sobre els de l'ARIMA estacional

($\Phi_P(B^S)$, $\Theta_Q(B^S)$).

$$\phi_p(B)\Phi_P(B^S)\nabla^d\nabla_S^D X_t = c + \theta_q(B)\Theta_Q(B^S)a_t$$

$$\Phi_P(B^S) = 1 - \sum_{i=1}^{P}(\Phi_i B^{iS})$$

$$\Theta_q(B^S) = 1 - \sum_{j=1}^{Q}(\Theta_j B^{jS})$$

Aquest model encara es pot millorar més si incloem informació sobre variables exògenes, és a dir, altres sèries temporals que poden tindre algun tipus de relació amb la que volem predir. Per a això s'inclou la variable com a un sumand i amb l'hiperparàmetre β que regule el pes de la variable exògena Z.

$$\phi_p(B)\Phi_P(B^S)\nabla^d\nabla_S^D X_t = c + \beta Z_t + \theta_q(B)\Theta_Q(B^S)a_t$$

2.2. Xarxes neuronals

Les xarxes neuronals artificials (ANN) són un tipus de model capaç de discernir molt bé patrons que, inspirats per sistemes biològics, són capaços d'aprendre i generalitzar a partir de l'experiència.

A [11] es presenten algunes característiques d'aquests models que els fan molt atractius per a modelar sèries temporals. En primer lloc, són models dirigits per les dades, de tal manera que amb molt poques assumpcions a priori, són capaces d'obtenir bons resultats. Açò els fa molt bons per a problemes on hi ha poques assumpcions teòriques, però amb moltes observacions. L'únic problema que pot aparéixer és que el model s'adeqüe massa bé al conjunt de dades i no generalitze bé, perquè aquesta és una de les altres grans característiques de les ANN: la seua capacitat per generalitzar malgrat tindre soroll. Com que el pronòstic en sèries temporals es basa en la predicció basant-se en observacions passades, les ANN semblen ser molt adequades per a modelar aquest tipus de problema.

D'altra banda, qualsevol model de sèrie temporal assumeix que hi ha una relació en les dades al llarg de la sèrie. És a dir, s'assumeix que una funció pot ajustar-se a la sèrie. Com que està demostrat per nombrosos autors que les ANN són aproximadors universals de qualsevol funció [12][13] sembla prou clar que aquest tipus de model podrien encaixar per predir els valors d'una sèrie temporal. Les aproximacions més tradicionals que hem explicat abans assumeixen que aquesta funció subjacent serà de caràcter lineal, però tal com s'explica a [14], sovint al món real els sistemes no són lineals. En canvi, els models basats en xarxes neuronals, són capaços de modelar funcions no lineals gràcies a les seues funcions d'activació.

Tanmateix, tradicionalment les ANN han funcionat pitjor que els models estadístics ([15], [16]). A [17] s'hipotetitza que açò pot ser perquè les sèries temporals individuals necessiten moltes dades per modelar-se de manera complexa, ja que la informació que es pot extraure d'una sèrie individual curta és limitada ([18] [11]). Una altra possible raó és que les característiques de les sèries temporals canvien amb el temps (heterocedasticitat), de tal manera que els patrons més distants de la sèrie tenen molta menys rellevància que els patrons més

propers. Per això és necessari que les dades tinguen una longitud adequada i que estiguen generades per un sistema relativament estable. A més, necessitem moltes dades que ara, al segle XXI, tenim. Finalment, també cal considerar que la precisió del model depén molt de com l'arquitectura de xarxa s'adequa al problema que es vol resoldre.

De tota manera, des que als anys seixanta [19] es van començar a emprar ANN per a predir sèries temporals s'ha avançat molt, especialment des que l'any 1986 es va presentar l'algorisme BackProp [20]. A la competició M4 de predicció de sèries temporals [21] es va aconseguir demostrar que els models basats en el perceptró multicapa són capaços de millorar els resultats dels models més tradicionals, amb el cas guanyador [22]. Ara com ara, les arquitectures basades en xarxes neuronals que més triomfen en aquest tipus de problemes són les basades en recurrències i les basades en mecanismes d'atenció.

2.2.1. Xarxes neuronals recurrents

Les xarxes neuronals recurrents (RNN d'ara endavant) són un tipus de xarxa neuronal que sovint s'utilitza per a modelar seqüències i si tenim en compte que les sèries temporals es poden considerar seqüències, no és forassenyat considerar-les per fer pronòstics sobre aquest tipus de dades.

La primera recurrent va ser desenvolupada a [23] als anys huitanta per Williams i Zisper, buscant millorar les ANN bàsiques mitjançant bucles a les neurones que permeten capturar l'ordre i dependències temporals de les dades. Aquesta primera xarxa era de caràcter dens i prompte molts autors es van adonar que les connexions podien ser parcials, com a la Elman [24] o a la Jordan [25].

El principal avantatge d'aquest tipus d'arquitectura és la seua capacitat per capturar dependències temporals a curt termini de les dades gràcies al seu aprenentatge adaptatiu i el seu mapejat dinàmic. Tanmateix, també du una sèrie d'inconvenients.

Per començar, un dels problemes que presenten és la caiguda a mínims locals. Per corregir aquest problema, a [26] es proposa, seguint la línia discursiva de les xarxes wavelet adaptatives i les wavelets de graella fixa, emprar una wavelet com a funció d'activació local en lloc d'una funció d'activació global. Així doncs, a l'AFCRWN (Adaptive Fully Connected Recurrent Wavelet Network) intenten combinar aquest tipus de xarxa amb les recurrents, de manera que es mantenen les propietats de les RNN (aprenentatge adaptatiu, mapejat dinàmic i memòria a curt termini amb les bones prediccions de les Wavelet networks. Per a fer-ho, ells prediuen els pesos i la translation i la dilation de les wavelets amb un nombre de neurones fix, de tal manera que el model té molta més flexibilitat que els anteriors.

Un segon problema de les xarxes recurrents és la poca capacitat de predir a llarg terme. A [27] intenten resoldre aquest problema dilatant les connexions recurrents, és a dir, en lloc de connectar-les de manera contigua, ells proposen connectar-les a una major distància. D'aquesta manera, aconsegueixen, alleugerir el problema de l'esvaïment del gradient i estenen el rang de les dependències temporals de manera similar a les connexions omeses (skip connections). A més, es redueix el temps computacional i el nombre d'hiperparàmetres de manera exponencial, de manera que es poden apilar més capes i formar xarxes més profundes.

Tal com hem dit, el problema de l'esvaïment per gradient és un dels més greus que presenten les xarxes recurrents. Per solucionar-ho hi ha dues aproximacions bàsiques segons [28]. Per un costat, es pot dissenyar un millor algorisme que un simple descens per gradient estocàstic. Per exemple es pot emprar l'anomenat gradient clipping o mètodes de segon ordre que siguen

menys susceptibles al problema. L'altra opció és alterar l'estructura de les neurones amb arquitectures noves o funcions d'activació més complexes.

Seguint aquesta línia, a [29] es va introduir la LSTM, una unitat recurrent que és capaç de modelar a partir d'una entrada ($\vec{x}_t \in \mathbb{R}^d$) les dependències a llarg termini gràcies a l'ús de tres portes que empren la funció sigmoide (σ): la d'entrada o *input* ($\vec{i}_t \in (0,1)^h$), la d'oblidament o *forget* ($\vec{f}_t \in (0,1)^h$) i la d'eixida o *output* ($\vec{o}_t \in (0,1)^h$).

$$\vec{f}_t = \sigma(\bar{W}_f \vec{x}_t + \bar{U}_f \vec{h}_{t-1} + \vec{b}_f)$$
$$\vec{i}_t = \sigma(\bar{W}_i \vec{x}_t + \bar{U}_i \vec{h}_{t-1} + \vec{b}_i)$$
$$\vec{o}_t = \sigma(\bar{W}_o \vec{x}_t + \bar{U}_o \vec{h}_{t-1} + \vec{b}_o)$$

Per a fer-ho, defineixen un vector d'activació ($\vec{c}_t \in \mathbb{R}^h$) aplicada a cada entrada a la unitat LSTM i un vector ocult ($\vec{h}_t \in (-1,1)^h$) d'eixida per a cada cèl·lula utilitzant un seguit de pesos ($\bar{W} \in \mathbb{R}^{h \times d}, \bar{U} \in \mathbb{R}^{h \times h}, \vec{b} \in \mathbb{R}^h$) i operacions amb les portes.

$$\tilde{c} = tanh(\bar{W}_c \vec{x}_t + \bar{U}_c \vec{h}_{t-1} + \vec{b}_c)$$
$$\vec{c}_t = \vec{f}_t \odot \vec{c}_{t-1} + \vec{i}_t \odot \tilde{c}_t$$
$$\vec{h}_t = \vec{o}_t \odot tanh(\vec{c}_t)$$

Aquest sistema de portes contribueix a filtrar informació d'entrada irrellevant i a obtindre una precisió més alta que les aproximacions clàssiques de RNN a costa de perdre precisió en dependències a curt termini.

D'altra banda, la GRU, publicada a [30], és una arquitectura que busca reunir el millor de les RNN clàssiques amb el millor de les LSTM, creant un mecanisme capaç de predir dependències a curt i llarg termini. Per a fer-ho, usa una estructura similar a la LSTM però basada en només dues portes que empren la funció sigmoide (σ): la d'actualització o *update* ($\vec{z}_t \in (0,1)^e$) i la d'inicialització o *reset* ($\vec{r}_t \in (0,1)^e$).

$$\vec{z}_t = \sigma(\bar{W}_z \vec{x}_t + \bar{U}_z \vec{h}_{t-1} + \vec{b}_z)$$
$$\vec{f}_t = \sigma(\bar{W}_z \vec{x}_t + \bar{U}_z \vec{h}_{t-1} + \vec{b}_z)$$

D'aquesta manera, defineixen també un vector ocult d'eixida ($h_t \in (-1,1)^e$) a partir d'un seguit de pesos ($\bar{W} \in \mathbb{R}^{e \times d}, \bar{U} \in \mathbb{R}^{e \times e}, \vec{b} \in \mathbb{R}^e$) i operacions amb les portes.

$$\tilde{h} = tanh(\bar{W}_h \vec{x}_t + \bar{U}_h(\vec{r}_t \odot \vec{h}_{t-1}) + \vec{b}_c)$$
$$\vec{h}_t = (1 - \vec{z}_t) \odot \vec{h}_{t-1} + \vec{z}_t \odot \tilde{h}_t$$

Així, mentre que la LSTM controla la quantitat de memòria vista mitjançant la porta d'entrada, amb la porta d'eixida la quantitat de memòria nova afegida i amb la porta d'oblidament l'eixida de la neurona, la GRU no té cap control per a la quantitat de memòria vista i no diferencia entre eixida i oblidament.

Tanmateix, ambdues unitats són de caràcter additiu i basades en portes, cosa que facilita que recorden l'existència d'una característica significativa anterior i creen dreceres que permeten que el gradient es retropropague millor. D'aquesta manera, a [28] es demostra que

aquest tipus d'unitats neuronals acaben sent més eficients que les unitats tradicionals de xarxes recurrents. D'entre la GRU i la LSTM, nombroses investigacions com les de [31] han intentat discernir quina d'elles funciona millor i sembla que és molt complicat identificar quin és l'escenari més adient per a cadascuna de les dues arquitectures.

Una altra aproximació per solucionar aquest inconvenient de les xarxes recurrents és l'encreuament amb altres models. Així les xarxes NARX, publicades a [32], buscaven combinar els models autoregressius no lineals amb variables exògenes amb les xarxes recurrents, però segons [31] les unitats amb portes continuaven funcionant millor. Un altre exemple són les xarxes recurrents convolucionals, com la ConvLSTM, publicada a [33], que integra dades espaciotemporals dins la mateixa neurona LSTM emprant convolucions, superant així els resultats de les LSTM bàsiques però augmentant el seu cost computacional.

Una unitat recurrent es pot agrupar en diferents arquitectures per a formar una xarxa recurrent. Molts models es basen en els models de capes d'unitats recurrents apilades, com el publicat a [34], que busca realitzar prediccions sobre múltiples sèries temporals alhora. Per a aconseguir-ho, ells proposen separar la sèrie en característiques utilitzant la metodologia de [35] i després agrupar les sèries en clústers segons les seues característiques, de manera que resumeixen i descriuen millor la sèrie temporal. Açò, segons [36] fa que el model sia més interpretable i resilient davant valors anòmals. Així, ells suggereixen entrenar un model LSTM distint per a cada clúster, estabilitzar la variància de la sèrie i després corregir l'estacionalitat mitjançant retards a la sèrie i la tendència fent servir una tècnica de normalització per finestres.

Un altre tipus d'arquitectura, plantejada a [37] és la LSTM en graella. Aquesta arquitectura aplica unitats LSTM a través de totes les dimensions, incloent-hi la profunditat de la xarxa. Per a fer-ho, cada bloc rep N vectors ocults $(\vec{h_1}, \vec{h_2}, ..., \vec{h_N})$ que concatena en una matriu de vectors ocults $(\bar{H})$ i N vectors memòria $(\vec{m_1}, \vec{m_2}, ..., \vec{m_N})$. Així, cada unitat del bloc buscarà aplicar la LSTM sobre la matriu de vectors ocults i el vector de memòria pertinent.

A [38] s'afirma que aquest tipus d'arquitectura pot ser adequada per a prediccions en paral · lel de múltiples eixides que poden ser independents o dependents linealment o no-linealment, ja que, com s'explica a [39] connectar les diferents dimensions apilant diverses capes dona més solidesa al model.

Altres models estan basats en jerarquia, és a dir, divideixen el problema general en subproblemes i els organitzen de manera piramidal. Així, són capaços de reduir el problema i condensar la informació. N'és un exemple [40], en el que s'empra un model jeràrquic per a predir dependències a llarg termini considerant un mecanisme LSTM jeràrquic aplicat sobre una convolucional de cada instant d'un vídeo. A [22], campions de la competició M4 [21], també utilitzen un mètode jeràrquic, que creua el suavitzat exponencial amb les LSTM de manera que el suavitzat modela de millor manera les dependències lineals mentre que la LSTM modela les característiques que no tenen aquesta linealitat. Així doncs, la LSTM rep una sèrie desestacionalitzada, normalitzada i de valors reduïts. L'arquitectura de la xarxa consta de diferents blocs LSTM apilats l'un damunt de l'altre de manera que a mesura que avancem a la xarxa, les connexions es van dilatant, de manera que cada unitat LSTM agrupa més informació. Aquestes unitats, a més, contenen addicions de connexions residuals similars a la ResNet [41] per ajudar a retropropagar el gradient. Finalment, a l'última capa hi ha un adaptador (una unitat lineal) que configura la mida de la capa d'eixida a l'horitzó de previsió adequat.

$$\forall i: \qquad (\vec{h_i}', \vec{m_i}') = LSTM(\bar{H}, \vec{m}_i, \bar{W}_i)$$

Però sens dubte l'arquitectura que més triomfa és la seqüèncial, introduïda a [42]. Aquestes arquitectures consten, principalment de dues parts: una dedicada a codificar la entrada a la que comunment anomenem *encoder* o codificador i una altra a generar l'eixida, anomenada *decoder* o descodificador. El codificador busca representar cada entrada de la sèrie $x_j, 1 \leq j \leq T$ com un estat ocult ($\vec{s_j} \in \mathbb{R}^n$) seguint una funció no lineal F que dependrà de la unitat recurrent escollida.

$$\vec{s_j} = F(x_j, \vec{h}_{j-1})$$

De vegades, podrem trobar que l'estat ocult $\vec{s}$ està format per dos vectors ocults concatenats, prenent com a entrada l'estat ocult del davant i del darrere ($\vec{h}_j = f(x_j, \vec{h}_{j-1})$ i $\vec{h}'_j = f(x_j, \vec{h}'_{j+1})$).

$$\vec{s_j} = [\vec{h}_j; \vec{h}'_j] \in \mathbb{R}^{2n}$$

D'altra banda, el descodificador, similar al codificador, associa una eixida ($\vec{y_i}, 1 \leq i \leq T'$) al vector de'estats ocults ($\vec{s_i} \in \mathbb{R}^n$) a partir d'uns pesos.

$$\vec{y_i} = \bar{W}_{out}\vec{s_i} + \vec{b}_{out}$$

Per exemple, a [30] es proposa una arquitectura que, mitjançant una LSTM, llig cada valor de la seqüència en cadena, modificant així el seu estat ocult. Aquest estat en l'última lectura serà un resum de la seqüència d'entrada i a partir d'aquest el descodificador, una altra LSTM, generarà una probabilitat del següent símbol.

Alguns estudis demostren que aquest tipus d'arquitectures podrien superar les tradicionals, com per exemple [43], que demostrava com el model seqüència a seqüència basat en GRU superava els altres models en diferents conjunts de dades. A més, a [44] es demostra que aquest tipus d'arquitectures, però ara basades en LSTM, superen els models tradicionals basats en aquesta mateixa unitat recurrent, puix són capaces de capturar adequadament tant les dependències a curt termini com les dependències a llarg termini, mentre que les arquitectures de LSTM ordinàries no són capaces de detectar tan bé les dependències a curt terme.

Així, a DeepAr publicat a [45], també usen aquesta arquitectura, tot i que ells empren la mateixa estructura i els mateixos pesos per al codificador que per al descodificador i a la pràctica aquests normalment difereixen. Un altre exemple és [46], on es busca modelar una successió d'esdeveniments mitjançant una arquitectura jeràrquica de codificador i descodificador. Així, es presenten els subesdeveniments com a components dels esdeveniments. L'arquitectura que plantegen té tres capes LSTM. La primera és un codificador que pren els valors inicials i els passa una LSTM per codificar-los en un embedding, obtenint així els subesdeveniments. La segona capa, és un altre codificador LSTM que codifica els conjunts d'*embeddings* dels subesdeveniments per a formar un superembedding que mapetge els esdeveniments. Finalment, tenim un descodificador LSTM que pren els superembeddings i els classifica segons esdeveniments.

Com que a les sèries temporals és molt factible que es perda l'estacionarietat, és possible

que les dades a predir continguen informació no present al conjunt d'entrenament i a l'inrevés, provocant que hi haja sobreentrenament i subajustament. Per a resoldre-ho, alguns autors han començat a emprar autoencoders per a extraure les característiques intrínseques de les sèries temporals durant l'entrenament i calculant la diferència entre aquestes característiques i les assolides al conjunt de validació. Aquesta extracció de característiques pot utilitzar-se també per expandir i contraure la xarxa, tal com fan a [39] o a [47], de manera que s'optimitza el flux d'informació de la neurona i porta a una millora de l'actualització del seu mecanisme d'actualització tant per a dependències a curt terme com a dependències a llarg terme.

A [48], s'incorpora un autoencoder d'LSTM per aconseguir extraure aquestes característiques que permeten detectar situacions anòmales. Així, demostren que per a aquesta tasca en concret, els autoencoders serien una millor solució que una arquitectura LSTM clàssica

2.2.2. Mecanismes d'atenció

Com que les sèries temporals posseeixen components estacionals, es pot argumentar que la informació contextual és necessària per a realitzar una bona predicció i, per tant, un model amb atenció podria ser molt eficaç en aquest tipus de tasca. El treball de Suilin a [49] per a la competició *Wikipedia Web Traffic Forecast* de Kaggle [50] reforça aquesta idea, emprant un model seqüència a seqüència d'unitats GRU amb finestres d'atenció per modelar l'estacionalitat. D'aquesta manera i combinant amb la memòria pròpia de les unitats GRU, Suilin aconseguia predir adequadament la sèrie.

A [51] anaven una passa més enllà i utilitzaven l'arquitectura seqüencial amb memòria per incloure context. Cal que ens detinguem ací per explicar que una arquitectura seqüència amb memòria segueix la mateixa arquitectura que una seqüencial ordnària però inclou un vector de context ($\vec{c}_i$) que correspon a l'eixida del model i es defineix a partir dels vectors ocults a partir d'una funció q que resumeix la seqüència d'entrada. D'aquesta manera, el vector a partir del qual es genera l'eixida al descodificador ($\vec{s} \in \mathbb{R}^n$) està definit per una funció G a partir d'ell mateix, del vector d'eixides ($\vec{y}_{i-1} \in \mathbb{R}^{T'}$) i del vector de context $\vec{c}_i$.

$$\vec{c} = q([\vec{h}_1, ..., \vec{h}_j, ..., \vec{h}_T])$$
$$\vec{s}_i = G(\vec{y}_{i-1}, \vec{s}_{i-1}, \vec{c}_i)$$

Típicament, aquesta funció pot ser senzillament l'últim estat ocult $\vec{h}_T$, que resumeix tot el conjunt d'estats ocults al ser recurrent, però a [51] proposen un model d'atenció per generar aquests vectors Ells defineixen els vectors de context o d'atenció, com els anomenen ells, com la suma ponderada dels estats ocults de l'encoder. La ponderació dependrà dels pesos d'atenció (α_{ij}) que marquenla importància de l'entrada j a l'hora de predir l'eixida i. Per a definir-los s'utilitza una softmax ($\mathcal{S}$) sobre un vector d'enmascarament $\vec{e}_{ij}$ calculat per retropropagació a partir d'uns pesos ($\bar{W}_a, \bar{U}_a, \vec{v}_a$).

$$\vec{c}_i = \sum_{j=1}^{T} \alpha_{ij} \vec{h}_j$$
$$\alpha_{ij} = \mathcal{S}(\vec{e}_{ij})$$
$$\vec{e}_{ij} = \vec{v}_a^T tanh(\bar{W}_a \vec{s}_{i-1} + \bar{U}_a \vec{h}_j)$$

Cinar et al. a [52] consideraven, a més, que la sèrie contenia pseudo-períodes, de tal manera que l'objectiu del seu model és capturar tots els pseudo-períodes amb un vector de tants valors reals com existisquen a la història de la sèrie ($\vec{\pi} \in \mathbb{R}^{T' \times T}$).

Aquest vector, codifica la importància de cada sub-esdeveniment j sobre la sèrie en el moment i. D'aquesta manera ells plantegen utilitzar el vector $\vec{\pi}$ per modificar els pesos amb un altre vector binari de T dimensions ($\vec{\Delta}^{(ij)} \in (0,1)^T$) de tal manera que val 1 en la dimensió $(i+T-j)$ i 0 en les altres. Aquest vector s'encarrega de seleccionar la coordenada de $\vec{\pi}$ que corrspon a la posició entre l'entrada j i l'eixida i.

$$\vec{e_{ij}} = \begin{cases} \vec{v_a^T} tanh(\bar{W}_a \vec{s}_{i-1} + < \vec{\pi}, \vec{\Delta}^{(ij)} > \bar{U}_a \vec{h}_j) & \text{si } (i+T-j) \leq T \\ 0 & \text{d'altra manera} \end{cases}$$

D'aquesta manera, l'eixida completa dependria d'aquest nou vector de sub-esdeveniments $\vec{\pi}$.

$$\vec{c}_i = \sum_{j=1}^{T} \alpha_{ij} \vec{h}_j$$

$$\alpha_{ij} = \mathcal{S}(\vec{e}_{ij})$$

$$\vec{e_{ij}} = \vec{v_a^T} tanh(\bar{W}_a \vec{s}_{i-1} + < \vec{\pi}, \vec{\Delta}^{(ij)} > \bar{U}_a \vec{h}_j)$$

Tanmateix, al mateix any (2017), es va publicar a [53] l'arquitectura Transformer, que va desbancar tota la resta de models d'atenció.

2.2.3. Transformer

La innovació del Transformer en l'aprenentatge profund [53] ha atret l'interés de moltes àrees tals com el reconeixement automàtic de la parla [54], la visió per computador [55] o el domini que ens interessa: la predicció de sèries temporals [56]. Aquest tipus de models es consideren aproximadors universals de les funcions seqüència-seqüència [57] i són capaços de mantindre les relacions temporals a llarg terme.

La forma bàsica que podem veure a la figura 2.1 d'aquests models segueix la mateixa arquitectura seqüencial de la qual parlàvem als models recurrents i als primers models d'atenció. Les seues unitats codificador i descodificador són pràcticament idèntiques (amb la diferència de l'emmascarament) i ambdues compten amb codificadors posicionals i mecanismes d'atenció.

Pretractament de les dades

Tal com hem explicat abans, les sèries temporals solen ser multivariants, on les diferents característiques estan relacionades entre elles. Lim et al. a [58] expliquen que utilitzar totes les característiques presents al model és ineficaç, ja que afegim una gran quantitat d'informació redundant. Ells proposen al mateix article una selecció de variables usant xarxes neuronals feed-forward amb oblidament, de manera que cada variable tindrà un pes que li donarà major o menor rellevància al model. És a dir, plantegen una selecció de variables segons la seua utilitat a la predicció.

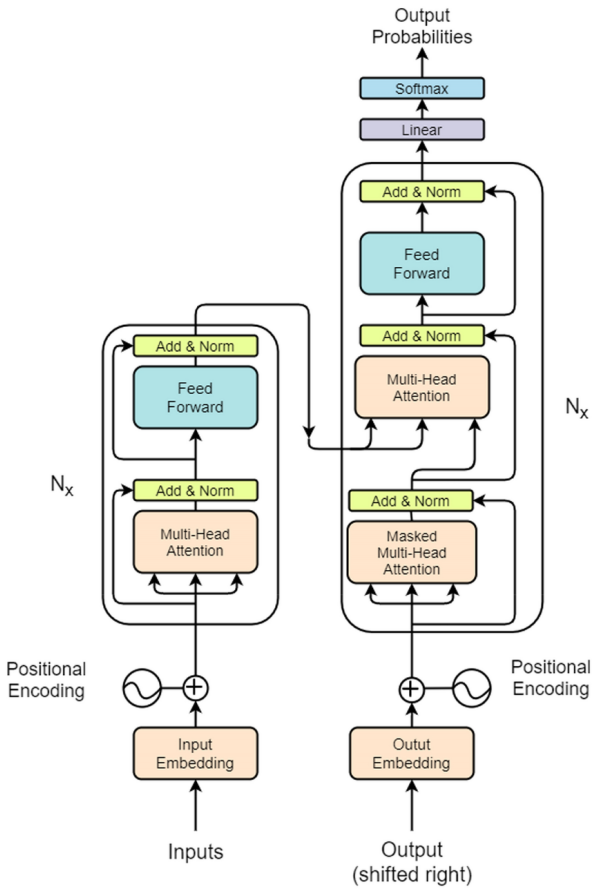


Figure 2.1.: arquitectura del Transformer descrita a [53]

.

Tanmateix, com hem parlat a l'inici del capítol 2, tractem molt sovint amb sistemes de sèries temporals multivariants. Nie et al. a [59] entenen que una sèrie temporal no és més que un senyal amb múltiples canals, on els tokens introduïts al Transformer poden provindre d'un únic canal o de diversos i expliquen que s'han de tractar de manera independent per a obtindre uns millors resultats. D'altra banda, Zhang i Yan a [60] conceben les sèries multivariants de la mateixa manera, però exploten les interdependències entre variables per a obtindre els seus resultats. Tanmateix, aquesta discussió continua oberta, puix ambdues variants obtenen grans resultats.

Aquestes dades s'han de mapejar mitjançant *embeddings*. Zerveas et al. a [61] proposen utilitzar xarxes denses feed-forward per tal d'aprendre aquests vectors, tot i que també podrien emprar-se altres tipus de mecanismes. Zhang i Yan ([60]) entenen, tanmateix, que dins les sèries temporals s'ha de distingir entre característiques temporals i posicionals, de manera que tant el temps com els valors en aquest mateix instant d'altres sèries afectaran a la nostra previsió. Per tant, ells proposen separar l'*embedding* en dues dimensions: una que tinga en compte les característiques temporals, mentre que l'altra pare atenció a les característiques espacials. Així doncs, l'*embedding* resultant serà $\vec{x} \in \mathbb{R}^{L \times d \times 2}$ on L és la longitud de la seqüència, és a dir, la quantitat de característiques temporals que tenim en compte i d és el

nombre de variables amb les quals comptem.

Per a tindre en compte aquestes característiques espacials, A l'article original [53], donada la frase de J paraules x_1^J, es defineix l'*encoding* posicional de la posició j (on $1 \leq j \leq J$) és $p_j \in \mathbb{R}^{D_p}$.

$$p_{j,2k} = sin(\frac{j}{10000^{2k/d_{model}}})$$

$$p_{j,2k+1} = cos(\frac{j}{10000^{2k/d_{model}}})$$

En aquesta equació k és la dimensió de l'*embedding*. És a dir, cada dimensió de la codificació és de caràcter sinusoidal i cosinusoidal. Altres tipus de Transformer com el Whisper [62], dedicat al reconeixement de la parla, proposen emprar només la codificació sinusoidal, però a les sèries temporals la major part d'*encodings* posicionals s'aprenen mitjançant feed-forwards, com és el cas dels articles [61] i [58]. A aquest últim, Lim et al. afirmen també que incloure com a agregat de l'*embedding* i l'*encoding* posicional algun tipus d'*encoding* estàtic (com podria ser la codificació genètica d'un pacient en sèries temporals d'evolucions d'afeccions) podria millorar significativament la predicció.

Tanmateix, en aquestes codificacions i *embeddings* no estem parant atenció a la possibilitat que la sèrie siga no estacionària. Els Transformers assumeixen que la distribució de la seqüència d'entrada és invariable amb el temps, de manera que usar una sèrie no estacionària dificulta la capacitat de previsió d'aquest tipus de models.

Molts autors opten per estacionaritzar la sèrie abans de processar-la dins del Transformer. Típicament, a les xarxes neuronals s'opta per normalitzar les dades, és a dir, convertir el rang dels valors entre zero i u, o estandarditzar-los, convertint-los en valors entre menys u i u. Aquest mètode, usat per alguns autors com Zhou et al a [63], estacionaritza la sèrie evitant problemes a l'interior del Transformer. Altres autors, com Wu et al. a [64], prefereixen emprar altres tècniques d'estandardització com el decalat que s'empra a l'ARIMA.

El problema d'aplicar una estandardització és que s'elimina informació intrínseca de la sèrie, provocant que el mecanisme d'atenció generalitze massa i no siga capaç de detectar alguns esdeveniments anòmals a la sèrie. És a dir, en estacionalitzar podríem convertir dues observacions distintes en la mateixa representació, per la qual cosa el mecanisme d'atenció podria confondre-les.

Per a solucionar aquest problema, tot i que Vaswani et al. a [53] proposaven una normalització per capa, alguns autors han decidit alterar aquesta normalització. Així, Zerveas et al. proposen a [61] normalitzar per remeses (*batch normalization*), de manera que mitiguen l'efecte dels *outliers*. Altres investigadors, com Chen et al. a [65], prefereixen aplicar una normalització utilitzant mitjanes mòbils amb l'objectiu de descompondre la variància i introduir aquesta informació al Transformer.

D'altra banda, Liu et al. proposen a [66] introduir la informació dins del mecanisme d'atenció, de manera que tot i que s'estacionaritze i es desestacionaritze abans i després del modelatge respectivament, no es perda la informació intrínseca de la sèrie. Aquesta idea d'introduir la informació dins del mecanisme d'atenció ha estat utilitzada per altres autors, com Chen et al., que al seu QuatFormer publicat a [65] imputen la freqüència dins la comparació entre claus i consultes. Sabent que la funció kernel de la *softmax* del mecanisme d'atenció no és més que una multiplicació de vectors, ells proposen que una multiplicació

de quaternions podria ser equivalent. D'aquesta manera, defineixen unes funcions (ψ, ϕ) per convertir vectors a quaternions utilitzant una freqüència i una fase que s'aprenen mitjançant quatre convolucionals aplicades a les matrius Q i K originals (dos per cada matriu: una convolucional per a la freqüència i una altra per a la fase). Aquests quaternions es roten després utilitzant la freqüència de la sèrie i es multipliquen, de tal manera que el resultat inclou més informació que si només incloem la sèrie bàsica.

$$LTRAtt(Q, K, V) = \mathcal{S}(\frac{\psi(Q) \times \phi(K)}{\sqrt{D_K}})V$$

En canvi, altres autors, opten per una altra aproximació: canviar de domini. Zhou et al. proposen a [67] el FedFormer, que canvia el domini al de les freqüències utilitzant el que anomenen bloc de freqüència ampliada (FEB). D'aquesta manera primerament projecten linealment les dades d'entrada $X \in \mathbb{R}^{L \times d}$ (on L és la longitud de la seqüència i d és el nombre de variables disponibles) en una matriu U utilitzant una matriu de pesos aprenentable $W \in \mathbb{R}^{d \times d}$.

$$U = XW$$

Aquesta projecció és aleshores transformada al domini de la freqüència utilitzant la transformada de Fourier o una transformada discreta de Wavelet ($T(\vec{u_i}) \in \mathbb{C}^{L \times d}$). Després proposen multiplicar aquests valors per un kernel de paràmetres $R \in \mathbb{C}^{d \times d \times L}$, de manera que s'afegeix una major complexitat.

$$U \odot R = \sum_{i}^{d} U_{L,i} \cdot R_{i,o,L}$$

on i és la dimensió del canal d'entrada i o és la dimensió del canal d'eixida

Aquests autors també afirmen que mantindre tota la freqüència no serà útil perquè introduirem una gran quantitat de soroll. Per a resoldre-ho, ells consideren seleccionar aleatòriament una quantitat M de fileres de la matriu U, reduint així la dimensió de $\tilde{U} \in \mathbb{C}^{M \times d}$ i la de

$$R \in \mathbb{C}^{d \times d \times M}$$

. Després, a l'eixida del FEB, s'hi afig *padding* i la transformada inversa per tornar al domini original.

$$U = XW$$

$$\tilde{U}_i = Select(F(\vec{u_i}))$$

$$\tilde{U} \odot R = \sum_{i}^{d} \tilde{U}_{M,i} \cdot R_{i,o,M}$$

$$FEB(x_i) = F^{-1}(Padding(\tilde{U}_i \odot R))$$

Mecanisme d'atenció

La part fonamental del Transformer és el mecanisme d'atenció, que permet al model recordar i concentrar-se en certes parts (passades i/o futures) del context. Per a aconseguir-ho, el mecanisme d'atenció (*self-attention*) transforma linealment les entrades (X) en tres matrius diferents: les consultes ($Q = XW_Q$), les claus ($K = XW_K$) i els valors ($V = XW_V$) on W_Q, W_K i $W_V \in \mathbb{R}^{L \times D_K}$. És a dir són matrius aprenentables amb tantes files com llarga és la seqüència d'*embeddings* (L) i tantes columnes com dimensions volem que tinguen les matrius (D_K). Una vegada hem obtingut les tres matrius, es mapegen les consultes sobre les claus, de manera que el resultat a la fila i de la matriu de consultes Q és una suma ponderada (segons una softmax) de tots els valors de V atenent al conjunt de claus K.

$$Att(Q, K, V)_i = \mathcal{S}(\frac{q_i K^T}{\sqrt{D_K}})V$$

En el cas dels mecanismes d'atenció amb H caps (*multihead-attention*), s'utilitzen H conjunts diferents de mecanismes d'atenció en lloc de només emprar-ne un.

$$MultiHeadAtt(Q, K, V) = Concat(Att(XW_1^Q, XW_1^K, XW_1^V), ..., Att(XW_H^Q, XW_H^K, XW_H^V))W^O$$

D'aquesta manera, el *decoder* d'un Transformer conté un mecanisme d'atenció amb diversos caps que compara les consultes de l'eixida de l'*encoder* amb les claus i valors de l'eixida del primer mecanisme d'atenció del *decoder* (*cross-attention*).

El desavantatge d'aquest procés és que el cost temporal d'aquestes operacions depén quadràticament de la dimensió de Q, de K i de V, és a dir, de la longitud de la seqüència d'*embeddings* L. Així doncs, mantindre sèries temporals massa llargues resulta en un problema de còmput.

A la literatura s'ha abordat aquest problema de distintes formes. Cristea et al. van proposar a [68] separar la sèrie d'entrada en segments de grandària S. D'aquesta manera, ells proposen usar finestres corredisses que posteriorment s'utilitzarien com a *embeddings*. Nie et al. a [59] van un pas més enllà i afirmen que, com un simple valor en el temps no té cap significat semàntic (com podria tindre'l una paraula dins d'una frase), extreure tota la informació en forma de *patch* és essencial per analitzar les seues connexions.

Tanmateix, el problema d'aquests models radica en el fet que en tindre menys dades, també es té menys informació sobre patrons que apareixen més distesament en el temps, amb la qual cosa, aquest tipus de metodologia no acaba de ser útil. A més, per a fer prediccions a més llarg termini, es requereix utilitzar mètodes autoregressius, és a dir, mecanismes que utilitzen les prediccions per a generar noves previsions.

Per solucionar aquest inconvenient, Wu et al. proposaren a [69] un entrenament adversarial que millorara la generació minimitzant l'error acumulat. A aquest article utilitzen una xarxa densa feed-forward amb una funció d'activació *Leaked ReLU* per tal de discernir si els valors futurs generats són reals o no. Amb açò i retropropagant l'error de manera similar a les GANs, s'aconsegueix millorar el funcionament del model en prediccions a llarg terme.

Altres mecanismes opten per corregir el cost computacional del model canviant directament el mecanisme d'atenció. L'AutoFormer, dissenyat per Wu et al. a [64], pren l'ARIMA com a inspiració i assumeix que la matriu Q serà molt similar a la sèrie decalada τ instants de

temps, de manera que en lloc de buscar similituds basades en la divergència entre les matrius QK^T tal com proposaven Vaswani et al. a [53], busca similituds emprant el coeficient de correlació de Pearson. D'aquesta manera, es té en compte l'autocorrelació i s'aconsegueix reduir el cost a $O(LlogL)$.

$$AutoFormerAtt(Q,K,V) = \mathcal{S}(\frac{cov(Q,K)}{\sigma_Q \sigma_K})V$$

Molts altres models s'aprofiten de la capacitat dels Transformers per tindre connexions escasses (*sparsity*). Li et al. a [70] proposen el LogTrans, que connecta aleatòriament $log(L)+1$ claus amb cada consulta, a diferència del model de Vaswani et al., que connecta totes les eixides anteriors. D'aquesta manera, en lloc de multiplicar QK^T, multiplica QK'^T on $K' \in \mathbb{R}^{log(L)+1 \times D_K}$ és una matriu escassa. Així es redueix el cost computacional de $O(L^2)$ a $O(LlogL)$.

El model Informer de Zhou et al. publicat a [63] segueix la mateixa teoria, però en lloc de seleccionar les connexions aleatòriament, mesura amb la divergència Kullback-Leibler l'eficàcia de les connexions i només n'utilitza les C millors. Tanmateix, mesurar l'eficàcia de totes les consultes torna a tindre un cost quadràtic, per la qual cosa no es mesura la divergència entre tots els parells possibles, sinó entre cada consulta i el vector mitjana de les claus, reduint el cost de les divergències a lineal amb la seqüència. Així, si considerem que Q' és una matriu escassa tal que $Q' \in \mathbb{R}^{log(L)+1 \times D_K}$, el cost es reduirà novament a $O(LlogL)$.

El model Fedformer presentat per Zhou et al. a [67] també segueix una aproximació similar. Com hem vist abans, aquests autors decidiren canviar el domini cap al de les freqüències emprant o bé la transformada ràpida de Fourier o bé una transformada discreta wavelet, ja que consideraven que així incloïen més informació que la definida per la seqüència de la sèrie temporal. Tanmateix, també sostenen que mantindre tots els components de la nova matriu de dades, és a dir, tots els valors de la transformada de Fourier o la wavelet, no és útil, puix molts dels canvis a altes freqüències es deuen a entrades amb soroll. Tot i això, tampoc consideren que s'haja de mantindre solament valors de baixa freqüència, puix els canvis de tendència s'observen habitualment amb freqüències altes. Per tant, ells plantegen emprar matrius escasses, en les quals es trien aleatòriament M característiques de la matriu. És a dir, descomponen la sèrie utilitzant les consultes, claus i valors, però transformen aquestes matrius en matrius disperses de freqüències.

$$FEAtt - f(Q,K,V) = \mathcal{S}(\frac{\tilde{Q}\tilde{K}^T}{\sqrt{D_K}})\tilde{V}$$
$$\tilde{Q_i} = Select(F(\vec{q_i})$$
$$\tilde{K_i} = Select(F(\vec{k_i})$$
$$\tilde{V_i} = Select(F(\vec{v_i})$$

En aquest sistema F denota la transformada ràpida de Fourier o la transformada discreta de Wavelet i la selecció de les files i és la mateixa tant per a les consultes com per a les claus i els valors. Com que $\tilde{Q}, \tilde{K}, \tilde{V} \in \mathbb{C}^{M \times D}$, assumim que el cost passa a ser $O(M^2)$ i si M és un valor molt inferior a L, Zhou et al. afirmen que podem denotar el cost computacional com a $O(L)$.

Una altra manera de reduir el cost és mitjançant vectors directors o *routers*. El QuatFormer, dissenyat per Chen et al. a [65], segueix aquesta idea, plantejada per Zhang i Yan a [60]. Aquest mètode consisteix a descompondre una sèrie llarga en una més reduïda $M' \in \mathbb{R}^c$ que condense la informació de la sèrie original. D'aquesta manera el mecanisme d'atenció creuada passa a descompondre's en dos: un mecanisme que infereix M' a partir dels valors d'entrada i dels valors d'una matriu M que s'aprén utilitzant descens per gradient amb momentum, i un altre mecanisme que infereix el resultat H a partir de la sèrie intermèdia i els valors contra els quals els volem comparar Y.

$$M' = Att(X, M) \rightarrow M \in \mathbb{R}^c$$
$$H = Att(M', Y) \rightarrow H \in \mathbb{R}^L$$

D'aquesta manera, el cost computacional final era de $O(2cL)$, que, d'altra banda, es pot considerar $O(L)$.

Una altra de les idees plantejades per solucionar aquest problema és atacar la funció *softmax* per introduir-hi l'escassetat (*sparsity*). Al model AST, explicat a [71], Wu et al. proposen l'ús una funció que, a diferència de la *softmax* puga assignar valors nuls. Per a açò, ells plantegen la funció $\alpha - entmax(\vec{z})$, basada en els multiplicadors de Lagrange (τ) on $\vec{z}$ és el conjunt de dades sobre el que s'aplica la funció i α és un hiperparàmetre.

$$\alpha - entmax(\vec{z}) = ReLU((\alpha - 1)\vec{z} + \tau\vec{1})^{\frac{1}{\alpha - 1}}$$
$$ASTAtt(Q, K, V) = (\alpha - entmax(\frac{QK^T}{\sqrt{D_K}}))V$$

Si aquest hiperparàmetre és u, el model serà una *softmax*, però si és dos, s'assumeix que el Transformer ha assolit la cota màxima d'escassetat. Per tant, Peters et al. a [72] consideren que $\alpha = 1.5$ és un valor prudent, arribant a reduir el cost computacional a $O(LlogL)$.

Finalment, un altre dels mètodes més freqüents a l'hora de resoldre el problema de cost computacional als Transformers aplicats a sèries temporals és el canvi d'arquitectures. Tenen un èxit especial els models jeràrquics que, com el Pyraformer que Liu et al. van proposar a [69] o el Triformer que Cristea et al. van desenvolupar a [68], van aconseguir assolir un cost lineal $O(L)$. La idea darrere d'aquesta tècnica és que es poden agregar les característiques temporals a un nivell superior. Per exemple, podem recollir la informació hora per hora, però, tot i això, segurament encara tindrem algun tipus d'estacionalitat diària, setmanal o fins i tot mensual i anual. Per tant, aquest tipus de models redueixen l'arquitectura a una forma piramidal, de manera que els mecanismes d'atenció no busquen tots els elements previs, sinó solament en els seus nodes fills en el cas d'ambdós models, i en els seus nodes adjacents, en el cas del Pyraformer. El Triformer, per solucionar el problema de retindre menys informació, utilitza una recurrència que té en compte les observacions passades.

Finalment, cal que parlem de l'SSDNet, desenvolupat per Lin et al. a [73]. Aquest model pren una aproximació diferent de les altres i busca combinar el Transformer amb un altre model: els mètodes d'espais d'estats (*Space State Models*). Aquests models tenen un gran recorregut en previsió de sèries temporals i compten un avantatge davant dels Transformers i és que són interpretables gràcies a la seua capacitat per recuperar tendència i estacionalitat. Lin et al. proposen emprar el Transformer com una ajuda als SSM de manera que el Trans-

former obtinga les característiques més importants del model i les utilitze per a predir els pesos que l'SSM hauria de tindre. Aquest model obri la porta a la combinació de l'arquitectura del Transformer amb altres mètodes a l'hora de realitzar previsions de sèries temporals.

2.3. Altres models competitius

En aquest punt, ja hem considerat una gran quantitat de models que dins la literatura tenen un paper significatiu. Tot i això, per concloure aquest capítol cal que considerem un parell d'aproximacions que no acaben d'estar incloses en els apartats anteriors.

En primer lloc, és necessari que parlem dels **Light-GBM**. Aquesta tècnica, publicada a [74] no és més que un algorisme que busca millorar la eficiència dels *Gradient Boosting Decision Trees* (GBDT d'ara endavant). Aquest tipus de models, els GBDT, són molt utilitzats perquè són capaços d'obtindre resultats sòlids sobre una gran varietat de tasques, però tenen com a inconvenient la seua poca eficiència computacional. A [74] proposen accelerar aquests processos mitjançant la reducció de mostres i de característiques de manera eficient.

A la competició M4 [21], es va comprovar l'eficiència d'aquest tipus de models quan el segon guanyador [75] va ser un model que precisament emprava aquest tipus de models. A la competició M5 del 2022 aquest tipus de models va tindre un paper encara més important, sent que molts participants, inclòs el guanyador, utilitzaven aquest algorisme. Queda demostrat així la seua capacitat de processar de manera efectiva nombroses sèries correlacionades.

Finalment, cal destacar la importància dels models híbrids, que combinen diferents aproximacions clàssiques amb aproximacions d'ANN i obtenen grans resultats, com el treball desenvolupat per Smyl a [22] o l'AutoFormer de [76].

3. Anàlisi del problema

Una vegada ja hem parlat del marc teòric de les sèries temporals, és hora d'abordar directament el nostre problema.

Tal com hem explicat al capítol 1, Logifruit és una empresa d'envasament i de transport i especialitzada en la gestió d'envasos reutilitzables. Un dels seus clients és Mercadona a qui ofereixen les seues caixes que després l'empresa de supermercats els retorna. La distribució de les caixes, visible a l'esquema de la figura 3.1, passa de Logifruit als productors de Mercadona. Aquests, omplin les caixes i les envien als supermercats, que, una vegada venen el producte, les retornen brutes a Logifruit. Aquests netegen les caixes i tornen a enviar-les als productors, aconseguint així establir un bucle.

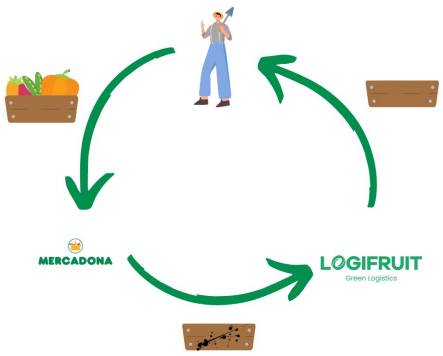


Figure 3.1.: esquema logístic de Logifruit.

.

Aquest cicle es repeteix constantment per a múltiples plantes en diferents regions i per a tots els distints tipus de caixa amb els que Logifruit compta.

Tanmateix, per assolir una distribució eficient de les caixes, cal saber la quantitat de caixes a netejar. Per fer-ho, Logifruit ha desenvolupat una sèrie de models sense gaire èxit, fins que va recórrer a la UPV per elaborar aquesta previsió. D'aquesta manera el nostre problema consisteix a desenvolupar un model de sèries temporals que prediga la quantitat de caixes a netejar amb una setmana d'antelació.

3.1. Anàlisi de qualitat de dades

El nostre objectiu era estudiar les dades recopilades a l'estació d'Abrera, una població propera a Barcelona, Catalunya. Teníem dades de dos tipus de caixes: dades de palets complets i dades de mitjos palets. Com hem explicat abans, la nostra intenció era obtindre una predicció sobre la quantitat de caixes que Logifruit havia de netejar per a poder enviar-les als productors, però a més, comptàvem amb la quantitat de caixes desplaçades pels clients cap als centres

de Mercadona. Finalment, l'últim conjunt de dades de què disposàvem eren els festius a la localitat d'Abrera.

Abans de posar-nos a treballar amb el conjunt de dades de què disposàvem, era convenient dur a terme una anàlisi de la qualitat de les dades per poder identificar possibles problemes. El primer que vam fer va ser una anàlisi de completesa, per observar valors nuls al conjunt de dades. Tal com podem veure a la figura 3.2, no trobàvem cap valor faltant, de manera que podíem continuar operant.

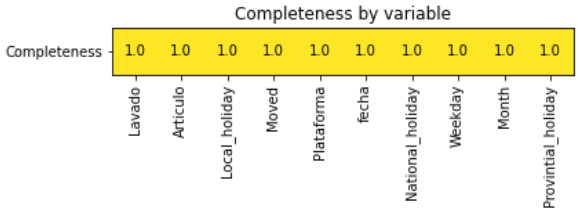


Figure 3.2.: gràfica de completitud per a les dades després de combinar els repositoris.

.

Tanmateix, les dades de quantitat de caixes que rebia Mercadona tenien certes incongruències tal com es pot comprovar a la figura 3.3, ja que existien valors negatius, i això no tenia cap sentit. Parlant amb els tècnics que ens havien oferit les dades, vam descobrir que molt probablement aquestes dades foren causades per algun error perquè s'apunten de manera manual i ens van instar a no considerar-les. Per tant, vam decidir imputar-les com a la mitjana. A més també ens van informar que quan existira una coincidència entre valors negatius i positius, es deuria també a una correcció sobre la quantitat rebuda el dia anterior. Per exemple, hi havia un cas en què s'havia anotat un valor de trenta mil un dia i l'endemà un valor negatiu exacte per a corregir l'error anterior. Ambdós valors devien ser establerts com a zero i així ho vam fer.

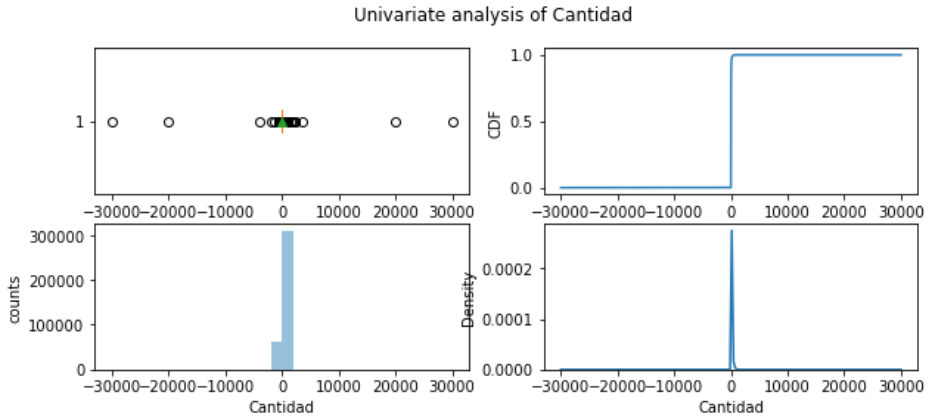


Figure 3.3.: anàl · lisi exploratori de la quantitat de caixes desplaçades a magatzems de Mercadona abans d'agregar-les per dia.

.

Per tant, les variables finals amb què comptàvem eren la quantitat de caixes que es netejava

diàriament, la quantitat de caixes que es transportaven des de, els festius locals, provincials i nacionals, el tipus d'article i el dia en què s'anotava l'observació.

3.2. Anàlisi univariant de dades

Si observem la quantitat de caixes netejades a la figura 3.4, podem veure que té una distribució una mica estranya. En primer lloc, podem comprovar que hi ha una gran quantitat de valors iguals a zero. Açò és, tal com veurem a la secció 3.3, a causa de la influència de festius. Tanmateix, també comprovem que hi ha dues modes, causades com veurem també a la secció 3.3 per una mescla de poblacions entre els dos tipus de caixes.

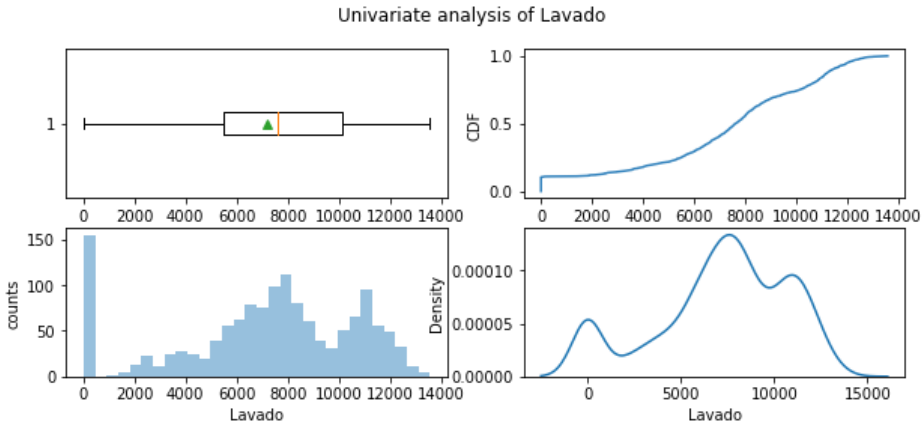


Figure 3.4.: anàl · lisi exploratori de la quantitat de caixes netejades.

.

En el cas de la quantitat de caixes desplaçades cap a Mercadona, també semblava haver-hi una mescla de poblacions, però molt més lleu, tal com podem veure a la figura 3.5. En aquest cas la distribució sí que és una mica més similar a una normal, tot i que trobem que la mitja (sis mil huitanta-quatre) estava una mica desplaçada respecte a la mitjana (sis mil dos-cents quaranta-cinc).

Respecte a la quantitat d'observacions de palets i de mitjos palets, cal dir que teníem set-centes dues observacions de mitjos palets i set-centes tres observacions de palets sencers, valors molt parells. Comptàvem també amb un 3.63% d'observacions en dies festius entre locals, provincials i nacionals i la distribució entre els dies de la setmana era molt parella, tal com podem comprovar en la figura 3.6

Seguidament, vam disgregar el conjunt de dades per a separar entre la sèrie dels palets complets i la sèrie dels mitjos palets, ja que vam pensar que aquesta mescla de poblacions que veiem a les figures 3.4 i 3.5 podria ser causada pel tipus de caixa a netejar.

A aquest conjunt separat de dades li vam aplicar el test *Augmented Dickey-Fuller* [77] per tal de comprovar si les quatre sèries (la de quantitat netejada i la de quantitat traspassada per a cada tipus de palet) eren estacionàries. Els resultats que vam obtindre eren prou concloents i en el cas de mitjos palets, tant per a la quantitat de pallets retornada com per a la quantitat netejada, les sèries eren estacionàries, mentre que per a la quantitat de palets

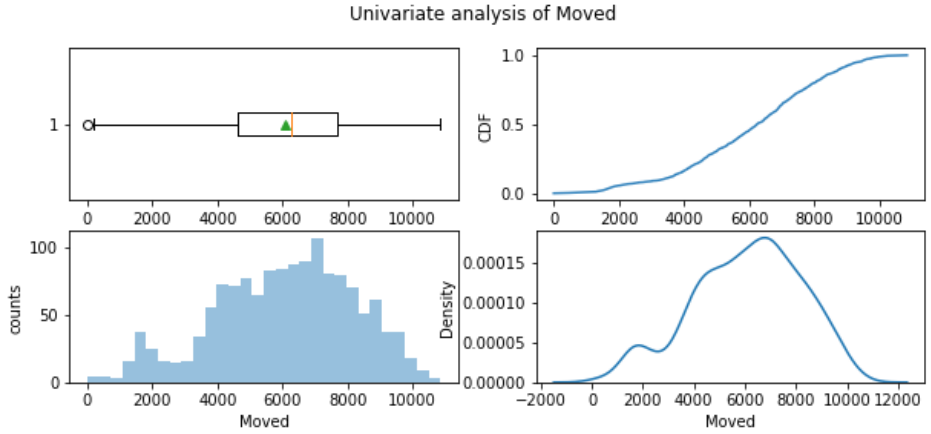


Figure 3.5.: anàl · lisi exploratori de la quantitat de caixes retornades a Mercadona després d'agregar-les per dia.

.

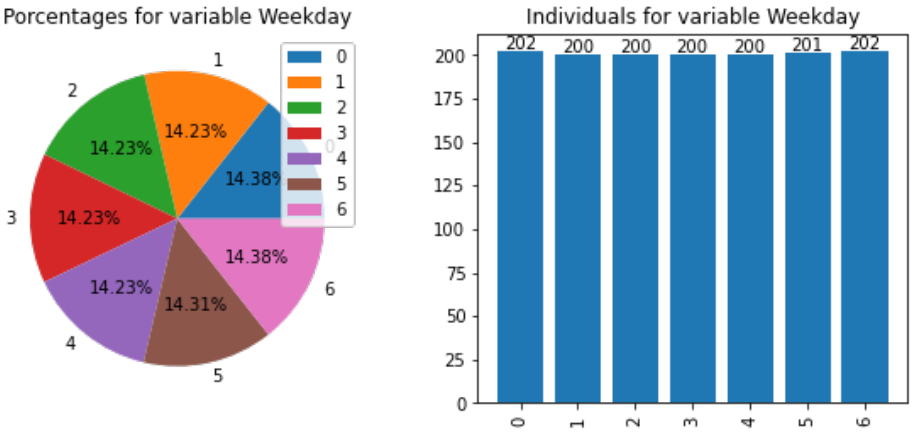


Figure 3.6.: anàl · lisi exploratori de la quantitat d'observacions registrades per a cada dia de la setmana.

.

sencers, ni la quantitat retornada ni la netejada eren estacionàries. Així doncs, havíem de tractar de manera distinta les sèries de palets sencers i les sèries de mitjos palets.

D'altra banda, calia veure la possible estacionalitat que podria tindre la sèrie. Per a fer-ho, podríem haver fet una descomposició STL, com s'explica a [4], però aquest mètode no ens permetia discernir clarament els patrons estacionals de la sèrie, tal com podem observar a la figura 3.7. Per tant, vam decidir aplicar una matriu de correlació [78] amb retards, de manera que els valors actuals de la sèrie s'avaluaven contra els valors anteriors.

Aquesta matriu, disponible a la figura 3.8, marcava clarament que l'autocorrelació en valors més propers era dèbil, però que estava molt influenciada pels valors una setmana abans. D'aquesta manera, podem argumentar que la sèrie presenta una profunda estacionalitat setmanal que haurem de tractar d'alguna manera. Aquesta matriu es desenvolupava de manera

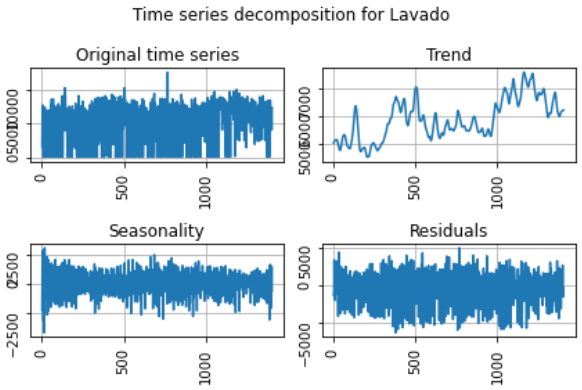


Figure 3.7.: descomposició STL per a la quantitat de mitjos pallets llavada.

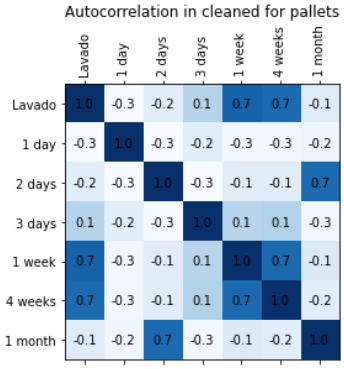


Figure 3.8.: matriu d'autocorrelació per a la quantitat de mitjos pallets netejats.

similar tant per a la quantitat de palets sencers netejats, com per a la quantitat de mitjos palets i palets sencers retornats.

3.3. Anàlisi multivariant

Tanmateix, encara mancàvem d'informació per establir un bon model, puix no sabíem exactament quina era la relació entre les distintes variables. El primer que vam fer en aquesta anàlisi multivariant era examinar si el tipus de caixa afectava la quantitat a netejar. Per a avaluar-ho vam pensar a utilitzar una anàlisi de variància [79] per tal de comprovar si hi havia una diferència significativa entre les mitjanes de les poblacions. Aquest test assumeix normalitat en les dades i homoscedasticitat i, per tant, calia comprovar aquests supòsits sobre les dues poblacions.

Tal com veiem a la figura 3.9, sembla que ambdues poblacions no són normals i tenen molt probablement variàncies distintes. Els tests de Shapiro-Wilk [80] i de Levene [81] confirmen aquest fet que podem comprovar a simple vista. Aquest problema amb les poblacions és molt probable que siga degut als valors nuls que podem observar a la figura i que són causats per dies festius, com veurem una mica més endavant.

Davant la impossibilitat d'aplicar una ANOVA, vam decidir emprar un test de Kruskal-

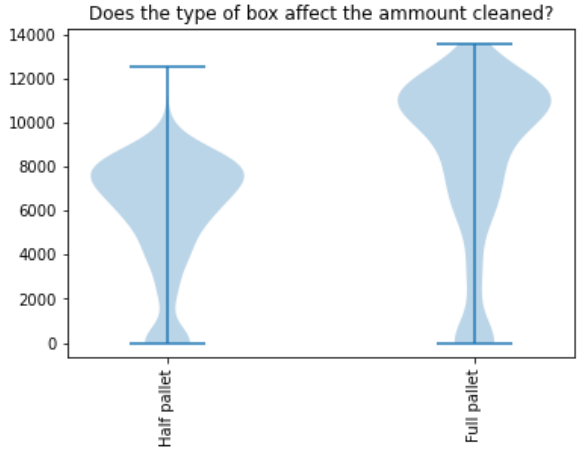


Figure 3.9.: distribució de la quantitat netejada segons el tipus de caixa.

Wallis [82] per analitzar si hi havia una diferència significativa, però en aquest cas sobre les medianes. Aquest test no necessita supòsits com l'ANOVA i és molt més robust. En aplicar-lo, vam comprovar que, efectivament, existeix una diferència entre la quantitat d'ambdós valors, com podem veure a la figura 3.10.

Aquest mateix problema i amb les mateixes solucions es repeteix per a la quantitat de caixes retornades, per la qual cosa podem assumir que hi ha, en general, un major moviment de palets sencers a la plataforma d'Abrera. Açò implica també, que els errors en la predicció de palets sencers seran de major envergadura que els de mitjos palets, per la qual cosa haurem d'estar més atents als problemes que açò pot comportar.

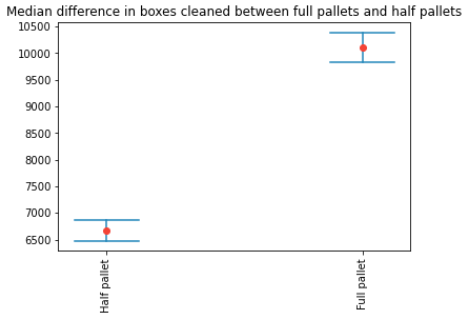


Figure 3.10.: diferència de medianes entre quantitat de palets sencers netejats i quantitat de mitjos palets netejats.

Ara calia comprovar si les variables de quantitat de caixes que els proveïdors enviaven a Mercadona estava correlacionada amb la quantitat de caixes que es netejaven diàriament a la plataforma de Logifruit d'Abrera. Tal com comprovem a la figura 3.11, es tracta de dues variables molt correlacionades on l'una afecta l'altra, perquè, com hem vist a l'inici d'aquest capítol, es tracta d'un sistema circular. També descobrirem que les caixes que es netegen durant el dia estan molt correlacionades amb les caixes rebudes per Mercadona el dia anterior. Tanmateix, nosaltres busquem llençar prediccions a una setmana vista i, en

conseqüència, no comptarem amb les dades del dia anterior en moltes ocasions. Imputar-les duria a una cascada d'errors i, per tant, decidirem no utilitzar-les. De tota manera, la quantitat de caixes netejades durant un dia també està molt correlacionada amb el total de caixes que rep Mercadona huit dies abans, per la qual cosa serà molt interessant fer servir aquesta informació per a elaborar la previsió.

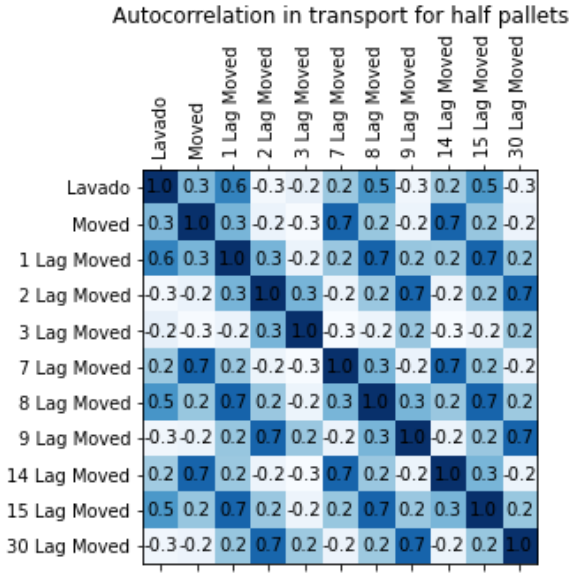


Figure 3.11.: matriu de correlació entre la quantitat desplaçada i la quantitat llavada per a palets sencers.

Un altre aspecte interessant a estudiar és l'impacte del dia de la setmana sobre la quantitat de caixes netejada, ja que hom podria pensar que els dissabtes i diumenges la plataforma d'Abrera podria no funcionar. Per avaluar aquest impacte hem tornat a elaborar un test de Kruskal-Wallis, car es tractava de poblacions amb distribucions anormals i heterocedàstiques i els resultats foren que, efectivament, hi havia una diferència entre les mitjanes. A la figura 3.12 s'aprecia que aquesta diferència és sobretot acusada els diumenges, amb valors propers a zero. És a dir, podem deduir que els diumenges la plataforma d'Abrera pren festius.

Ara caldria comprovar si els festius també tenen un efecte sobre la quantitat de caixes netejada. Per a fer-ho tornem a aplicar un test de Kruskal-Wallis, puix les assumpcions de l'ANOVA no es compleixen i obtenim que existeix una clara diferència entre medianes. A la figura 3.13 observem que la quantitat netejada es redueix dràsticament en festius provincials i en festius nacionals, però que no es redueix a zero. Durant els festius locals, en canvi, no sembla que hi haja cap mena de canvi de producció respecte als festius locals. Per a considerar si hi havia un canvi de producció entre els festius provincials i els nacionals, vam elaborar un altre test de Kruskal-Wallis, però en aquest cas, no vam trobar diferències significatives entre les medianes.

Finalment, l'última anàlisi multivariant que vam elaborar va ser una anàlisi per mesos. En aquest cas, les poblacions de cada mes tampoc no es podien ajustar amb una distribució normal i les seues variàncies semblaven distintes. Els resultats que vam assolit van ser que, de nou, existia una diferència entre medianes, tal com comprovem a la figura 3.14. A aquesta

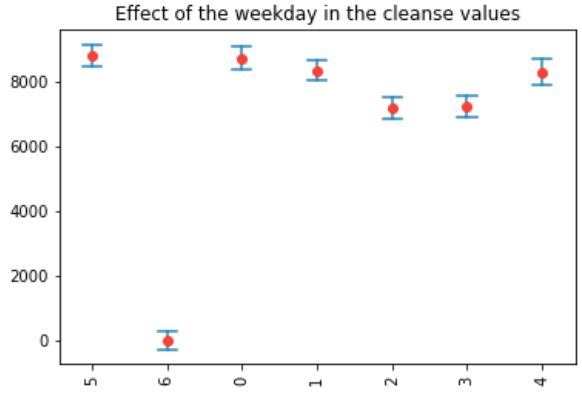


Figure 3.12.: diferència de medianes segons el dia de la setmana.

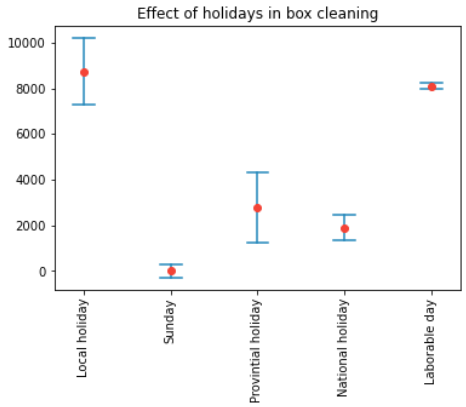


Figure 3.13.: diferència de medianes segons els festius.

mateixa figura podem discernir que hi ha una certa tendència que els mesos d'estiu el nombre de caixes netejades augmente i que a l'hivern baixe. Açò podria estar relacionat amb algun tipus d'estacionalitat, tot i que l'autocorrelació sembla desmentir-ho.

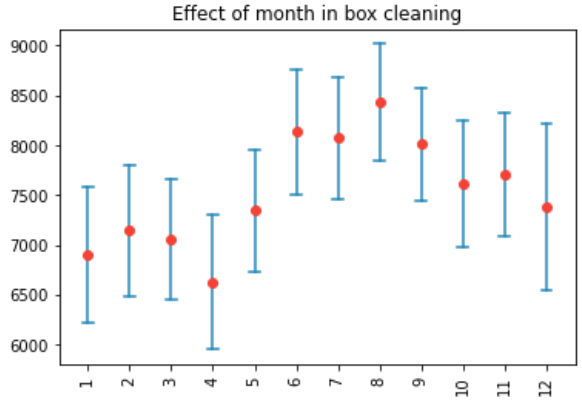


Figure 3.14.: diferència de medianes segons el mes.

4. Metodologia

Una vegada havíem explorat les dades, era moment d'elaborar els models predictius per a la sèrie. Però abans, calia establir una metodologia adequada per a poder obtindre bons resultats.

El primer que havíem de decidir era quin conjunt utilitzaríem com a entrenament. Tal com hem explicat al capítol 3, vam separar el conjunt de dades en dos: una sèrie temporal que inclouria només les dades referents a palets sencers i una altra sèrie que només tindria en compte la informació dels mitjos palets. Per tant, tots els models s'havien d'aplicar tant per a un conjunt de dades com per a l'altre.

Ara bé, per a cada una d'aquestes sèries, calia diferenciar entre dos subconjunts més: un que aprofitaríem per a entrenar els models i l'altre que empraríem per a validar-los. A [5] es destaca el valor d'aplicar una validació creuada quan s'apliquen models d'aprenentatge automàtic, com ara les xarxes neuronals. Tanmateix, els models més clàssics de tipus estadístic com els models de suavitzat exponencial o els models ARIMA, fan ús de tota la seqüència de dades fins al moment en el qual s'efectua la predicció per a elaborar la previsió. Per tant, no té gaire sentit segmentar les dades i agafar un dels conjunts de dades del principi o del mig. A més, en els models de xarxes profundes, açò també implica un augment considerable del temps de còmput, per la qual cosa vam decidir que nosaltres faríem servir només un subconjunt d'entrenament i un subconjunt de validació.

Així doncs, vam decidir que els subconjunts d'entrenament comptarien amb un 80% dels valors dels conjunts de cada palet, mentre que els subconjunts de validació serien només el 20% de les dades de la mostra. A més, el caràcter lineal de la sèrie, vigent en l'autocorrelació, ens impedeix mesclar les dades, per la qual cosa, aquest conjunt d'entrenament no va ser reordenat. Així doncs, el primer 80% de les dades va ser usat per a entrenar els models i el restant 20%, el vam utilitzar per validar-los.

Per decidir quines mètriques aprofitaríem per a avaluar els nostres models, cal que tinguem en compte que el nostre sistema conté una gran quantitat de zeros. Açò vol dir que no podrem recórrer a mètriques com el MAPE [83], ja que aquest valor que depén del nombre de mostres (n) i dels valors prevists ($\tilde{y}$) conté una divisió amb els valors reals (y).

$$MAPE = \frac{100}{n} \cdot \sum_{i}^{n} |\frac{y - \tilde{y}}{y}|$$

Nosaltres, vam decidir que empraríem l'MSE [84] per a mesura de pèrdua als models de xarxes i que ens serviríem de la seua arrel per comparar els models entre ells. Tanmateix, érem molt conscients que fer ús de la pèrdua o un dels seus derivats per avaluar el model no és gaire just per als mètodes estadístics que no s'entrenen amb aquesta funció i a més, pot dur-nos a biaixos si caiem en alguns dels punts dèbils de la mètrica. Per aquesta raó vam decidir fer servir una avaluació addicional.

Existeix una tècnica anomenada *rolling window* [85] que consisteix a elaborar una finestra

d'informació que desplacem a través del temps i incloure aquesta informació. En desplaçar-la a través del temps, podem aplicar una mètrica per avaluar els resultats. Açò dona més resiliència al model per establir paràmetres que es mantinguen constants al llarg del temps. Encara més, a [5] i a [86] es destaca la importància de combinar models per obtindre una millor predicció. Podem emprar aquesta finestra lliscant per a descobrir quin és el millor valor per a cada instant de predicció i combinar els distints models que oferisquen els millors resultats per a cada instant de predicció. Per exemple, posem que volem fer una predicció per a les pròximes vint-i-quatre hores i amb la finestra lliscant descobrim que per a les primeres dotze funciona millor una ARIMA i que per a les darreres dotze funciona millor una xarxa recurrent. Podem combinar ambdós models de tal manera que obtinguem un primer model que prediga les primeres dotze hores amb l'ARIMA i les darreres dotze amb la recurrent.

Com a mètrica per a avaluar la *rolling window* vam decidir utilitzar dues mètriques. La primera, el SMAPE [87], una mètrica similar al MAPE, però que pot tindre valors reals zero. D'altra banda, i per tindre més informació sobre els resultats, també vam decidir aplicar la finestra lliscant amb el MAE [88], una mesura clàssica de desviació de prediccions.

Els models que hem usat són molts i molt diversos. Així doncs, per explicar-les totes, la resta d'aquest capítol es dividirà en cinc seccions. En la primera, parlarem dels models basats en mètodes estadístics com el suavitzat exponencial o l'ARIMA. En la segona, explicarem els mètodes de xarxes recurrents que hem emprat. En el tercer apartat, desenvoluparem els mètodes de xarxes convolucionals adaptats al nostre problema. En el quart, parlarem de la implementació de models basats en el Transformer aplicats a les nostres dades. Finalment, en l'últim segment explicarem una mica quins models preentrenats hem fet servir, reentrenant-los per a adaptar-se al nostre cas.

4.1. Models estadístics

Començarem parlant dels models estadístics que hem utilitzat per a realitzar les previsions. Aquest apartat es dividirà en tres subseccions: una sobre models de suavitzat exponencial, una altra sobre models ARIMA i una altra sobre models ARIMAX.

4.1.1. Models de suavitzat exponencial

Els models de suavitzat exponencials poden ser de tres tipus, tal com hem vist a la secció 2.1.1. En el nostre cas, hem observat que tenim, com a mínim, una estacionalitat setmanal, de forma que serà completament necessari utilitzar un model exponencial triple, altrament dit model de Holt-Winters.

Tanmateix, aquest model no permet valors zero, ja que inclou una divisió entre l'estacionalitat i una divisió entre la tendència, que impedeixen al model prendre valors nuls. Per això vam decidir que faríem un preprocessat afegint deu unitats (el valor mínim observat excluint els zeros) a tots els registres, de manera que obteníem valors superiors a zero en tot cas.

Els models Holt-Winters poden tindre diverses formes segons si decidim afegir la tendència i l'estacionalitat de forma additiva o multiplicativa. Nosaltres vam decidir provar totes les combinacions possibles. És a dir, per a cada tipus de caixa (palets sencers i mitjos palets) vam aplicar un model on la tendència i l'estacionalitat eren additives, un altre en el que la tendència era additiva i l'estacionalitat multiplicativa, un tercer en el qual la tendència

era multiplicativa i l'estacionalitat era additiva i, finalment, un darrer model amb les dues multiplicatives.

4.1.2. Models ARIMA

Seguidament, vam decidir aplicar un model ARIMA seguint la metodologia Box-Jenkins [10]. En total vam acabar desenvolupant un total de deu models: sis per a palets sencers i quatre per a mitjos palets.

Palets sencers

Abans de començar a aplicar un model ARIMA, era convenient observar els gràfics de les funcions d'autocorrelació de la figura 4.1. En aquest cas, podem veure que hi ha clarament una estacionalitat setmanal (cada set observacions), però no es revela a priori cap estacionalitat mensual (cada trenta-una observacions) com podríem haver deduït a la secció 3.3. Per tant, caldrà utilitzar un model SARIMA, que tinga en compte aquesta component estacional. A més, podem comprovar que el gràfic d'autocorrelacions parcials presenta una major concentració de valors propers a u, per la qual cosa podem deduir que és molt probable que necessitem una part de mitjanes mòbils per corregir aquesta autocorrelació. Finalment, cal recordar que aquestes dades no són estacionàries, per la qual cosa és pràcticament segur que necessitarem una part integrada.

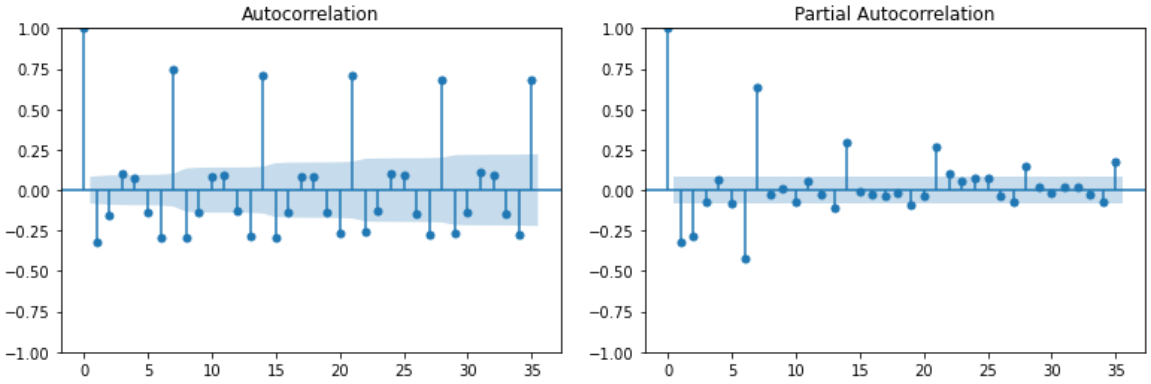
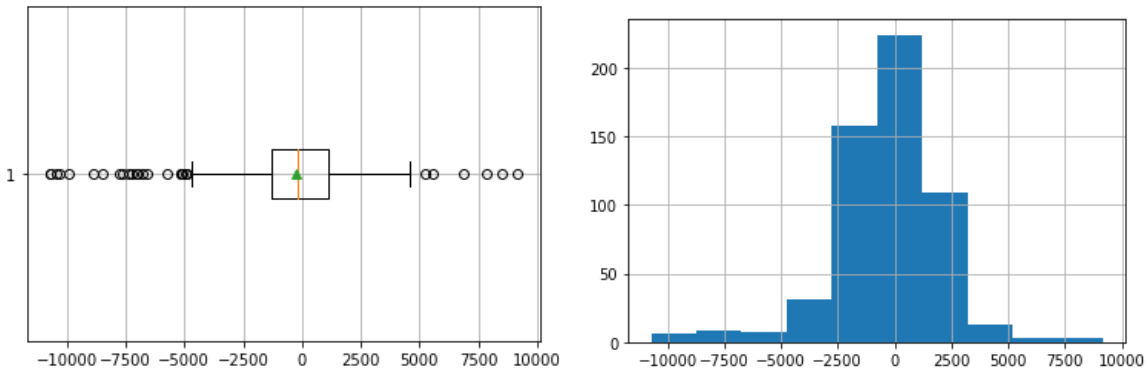


Figure 4.1.: funcions d'autocorrelació per a les dades de palets complets.

El primer model que vam desenvolupar, era un model SARMA $(0,0,2)\ (0,0,3)_7$. Continuant amb la metodologia de Box i Jenkins explicada a [10], ara era necessari comprovar que els residus de la sèrie eren un soroll blanc. És a dir, els residus del model havien de ser normals, havien de tindre una mitjana nul · la, havien de ser estacionaris i no havien d'estar autocorrelats. Per tant, vam aplicar el test augmentat de Dickey-Fuller i vam comprovar que no hi havia cap estacionalitat i vam observar els intervals de confiança per a la mitjana i vam obtindre que aquesta es trobava en interval entre -369.45 i 169.57, comprenent el valor nul. Tanmateix, quan vam aplicar els tests de Shapiro [80] i el de Ljung-Box [89][90], vam assolir mal resultats. Per una banda, vam confirmar que la distribució no era normal, puix la curtosi definia la distribució com a leptocúrtica, és a dir, la distribució era més corbada i

acusada que en una normal, el que es tradueix en molts valors similars i alguns valors anòmals. D'altra banda, els tests de Ljung-Box, ens informaven que a partir del període set hi havia una autocorrelació palpable entre els residus. Tot açò impedia que poguérem assumir que aquests residus eren soroll blanc i, per tant, vam haver de descartar aquest model.

Així doncs, i després d'experimentar una mica més amb les dades, desenvolupàrem un segon model SARIMA $(0,1,2)\ (0,1,3)_7$. Segons el test d'autocorrelació de Ljung-Box aquest model ja no presentava problemes d'autocorrelació de cap mena i l'ADF ens indicava que tampoc no hi havia estacionalitat. Tanmateix, quan vam aplicar els tests de Shapiro i vam establir els intervals de confiança, vam observar que la mitjana era una mica inferior a zero i que la distribució no era exactament normal. En examinar les gràfiques de la figura 4.2, comprovàrem que existia una cua a l'esquerra de la mitjana i que es devia, sobretot, a valors anòmalament baixos, provocats segurament per errors en festius o diumenges, car no estàvem comptant amb aquestes dades. Per tant, vam decidir assumir que la distribució dels residus sí que era la d'un soroll blanc.



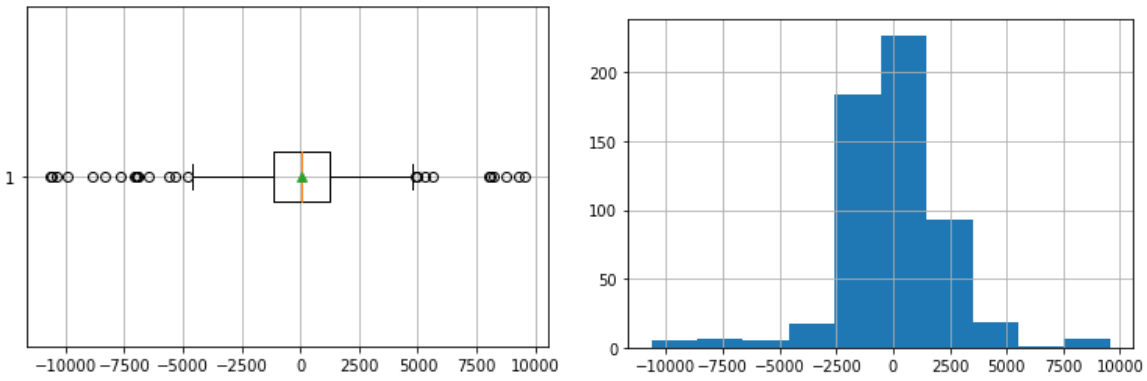
(a) Gràfic Box&Whiskers dels residus.

(b) Histograma dels residus.

Figure 4.2.: gràfics d'anàlisi dels residus del primer model SARIMA per a palets sencers.

El segon model ARIMA que vam aplicar va ser un model SARIMA $(1,1,1)\ (1,0,2)_7$, de manera que obteníem gràfics ACF i PACF plans. En aquest cas, els tests de Ljung-Box van demostrar que, de nou els residus no estaven autocorrelats. D'altra banda, el test ADF demostrava que eren de caràcter estacionari i l'interval de confiança per a la mitjana incloïa el valor zero. Tot i això, en examinar amb el test Shapiro la distribució dels residus, comprovàrem que no eren normals. Aquesta vegada, observant les gràfiques de la figura 4.3, comprovàrem que es tractava d'una distribució leptocúrtica, amb molts valors en les cues en ambdós costats. Novament, vam pensar que aquests valors podrien ser causats per l'efecte d'alguna de les variables exògenes amb les quals comptàvem. Per tant, vam decidir obviar aquest error i assumir que la distribució d'aquests residus era similar a la del soroll blanc.

Els dos següents models per als palets sencers foren molt similars: un SARIMA $(2,1,1)\ (1,0,3)_7$ i un SARIMA $(2,1,2)\ (1,0,3)_7$. En ambdós casos es complien les condicions de mitjana nul · la, estacionarietat i la no autocorrelació. Tanmateix, també en els dos casos teníem distribucions leptocúrtiques, com en el segon model ARIMA. Davant d'açò vam tornar a assumir que serien errors causats per festivitats o per la quantitat de caixes rebudes per Mercadona i vam obviar aquest resultat per assumir que es tractava de soroll blanc.



(a) Gràfic Box&Whiskers dels residus.

(b) Histograma dels residus.

Figure 4.3.: gràfics d'anàlisi dels residus del segon model ARIMA per a palets sencers.

Finalment, l'últim model ARIMA que vam aplicar sobre aquestes dades va ser un model calculat automàticament seguint l'algorisme *Stepwise* dissenyat per Hyndman i Khandakar [91]. Aquest model resultant ens definia un model SARIMA $(5,1,0)$ $(1,0,2)_7$. Això no obstant, analitzant els residus, no podíem comptar amb ells, ja que tot i que els residus eren estacionaris i la seua mitjana podia assumir-se com a nul · la, tenien problemes d'autocorrelació des del primer interval i no s'adequaven a una distribució normal. Per aquesta raó vam acabar descartant també aquest model.

Mitjos palets

En el cas dels mitjos palets, com en l'apartat anterior, era convenient examinar abans d'aplicar cap model, quina era la situació respecte a les dades amb les quals comptàvem. Per això, vam observar els gràfics d'autocorrelació, presents a la figura 4.4. En aquest cas, podem veure com altra vegada tenim una estacional setmanal i no mensual, com podíem haver hipotetitzat a la secció 3.3. D'aquesta manera, serà necessari imputar aquesta component estacional en el model ARIMA que usem. A més, en aquest cas descobrim que la distribució d'autocorrelació en dades properes és molt diversa, per la qual cosa els models que hem d'utilitzar poden arribar a ser molt complexos mesclant autoregressió amb mitjanes mòbils. Finalment, cal que remarquem que aquest model no presentava estacionarietat, per la qual cosa no hauria de ser necessari aplicar una component integrada.

El primer model que vam aplicar després d'una llarga experimentació de set iteracions, fou un model SARMA $(2,1)$ $(4,4)_7$ sense part integrada. Els seus residus no presentaven autocorrelació ni estacionarietat i tenien mitjana nul · la, però el test de Shapiro assumia anormalitat en la distribució de les dades, causada, de nou, per una distribució leptocúrtica. Com que es tractava de la influència de valors anòmals, vam decidir assumir que era soroll blanc i obviar que no es tractara d'una distribució perfectament normal.

El segon i tercer model sí que comptaven amb una component integral, essent un SARIMA $(1,1,3)$ $(1,0,1)_7$ i un SARIMA $(1,1,3)$ $(1,1,1)_7$. Ambdós models tenien residus estacionaris, de mitja nul · la i sense autocorrelació, però tots dos també tenien problemes de normalitat segons el test Shapiro. Aquests problemes, altra vegada eren causats per la curtosi i molt

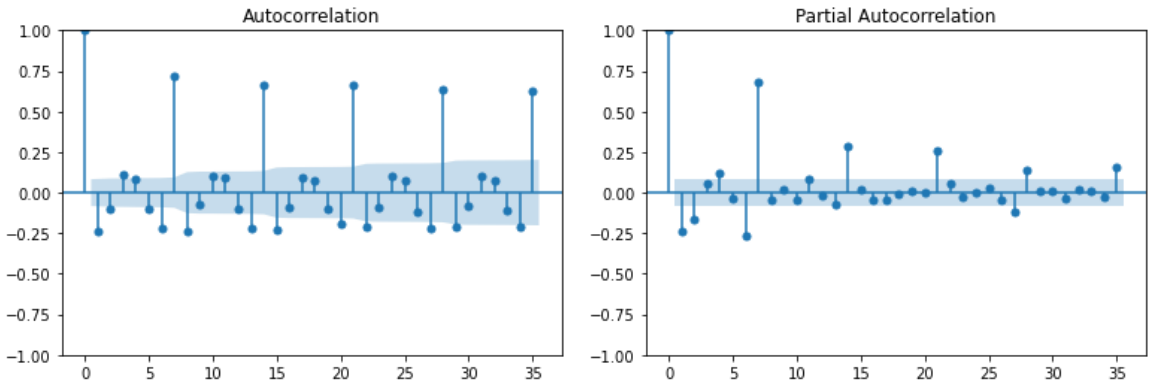
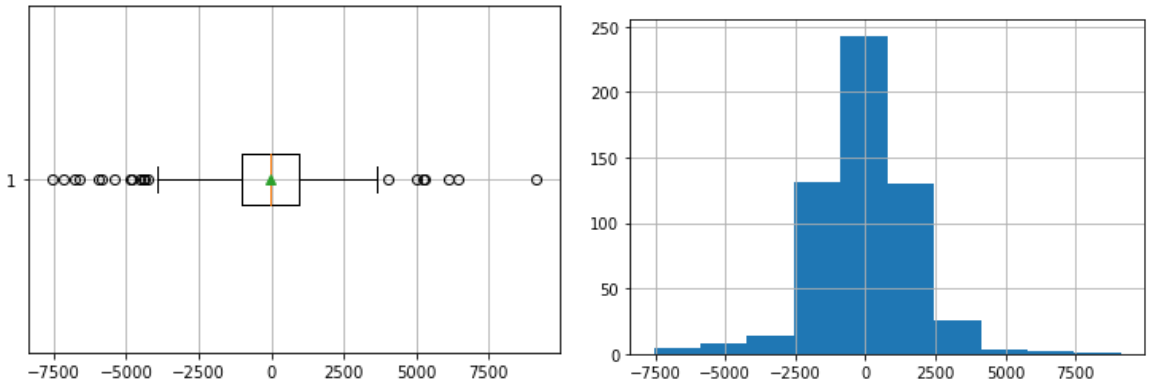


Figure 4.4.: funcions d'autocorrelació per a les dades de mitjos palets.

probablement eren deguts a l'efecte de variables exògenes, de tal manera que vam decidir obviar que no s'ajustara perfectament a una campana de Gauss i vam assumir que es tractava de soroll blanc.



(a) Gràfic Box&Whiskers dels residus.

(b) Histograma dels residus.

Figure 4.5.: gràfics d'anàlisi dels residus del segon model ARIMA per a mitjos palets.

Finalment, vam aplicar un model calculat automàticament segons l'algorisme *Stepwise*, obtenint un SARIMA (2,1,0) $(1, 0, 1)_7$. Aquest model, però, tenia greus problemes als residus, puix estaven molt autocorrelats en tots els seus intervals, tenien una mitjana inferior a zero i no seguien una distribució normal. Davant aquests inconvenients, no podíem considerar-los soroll blanc i, per tant, no podíem definir el model com a correcte.

4.1.3. Models ARIMAX

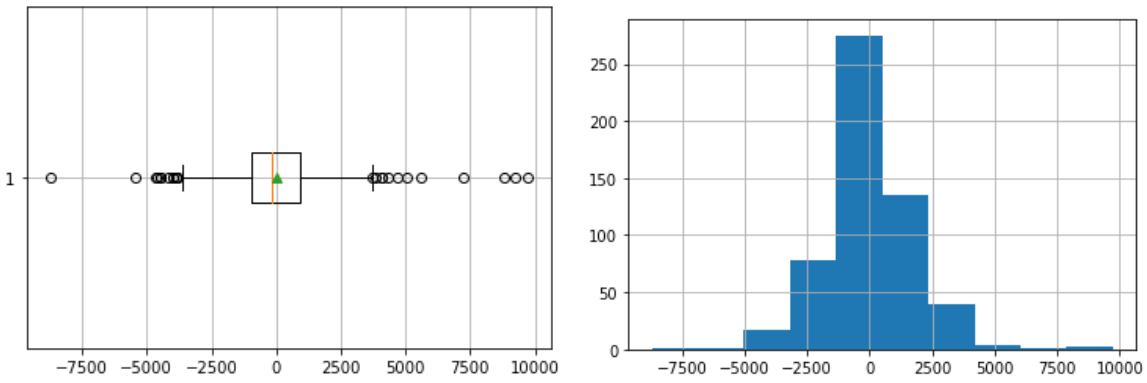
Finalment, el darrer dels models estadístics que comentarem serà l'evolució dels models ARIMA per integrar variables exògenes, és a dir, els models ARIMAX. Així doncs, necessitàvem saber a quins models afegiríem les dades exògenes per millorar les prediccions. Per descobrir-ho, cal que avancem una mica, fins a la secció 5.1, i seleccionem els tres models amb menor RMSE per a cada tipus de caixa. Aquests seran els models que emprarem com a

base abans d'afegir-los les variables exògenes.

D'altra banda, hem de descobrir quines són les variables que utilitzarem. Tal com hem vist a la secció 3.3, sembla que les variables de quantitat de caixes que rep Mercadona estan certament relacionades amb la quantitat de caixes a netejar per Logifruit. A la figura 3.11 observàvem que els valors del dia d'abans estan molt correlats, però també comentàvem que no podríem comptar amb aquestes dades i que l'única forma d'obtindre-les seria mitjançant una predicció que ens llençaria cap a una cascada d'errors. Per evitar-ho, no considerarem cap valor de la sèrie de retorns a Mercadona amb un marge inferior a set dies. Ara bé, analitzant altra vegada la matriu de correlació de la figura 3.11, comprovem que els valors d'enviaments cap a Mercadona amb més sentit per introduir al model ARIMA són els de huit dies abans, puix són els més correlats o, en tot cas, els de set dies.

D'altra banda, dins la secció 3.3 també véiem a la figura 3.13 que els festius tenien un efecte certament significatiu sobre els models de predicció, concretament als diumenges, als festius nacionals i als festius provincials. En principi l'estacionalitat de l'ARIMA ja devia tindre en compte l'efecte dels diumenges per a realitzar una millor predicció, de manera que nosaltres solament vam afegir una variable binària per a festius, que prenia valor 1 quan era o festiu nacional o festiu provincial i prenia valor 0 en qualsevol altre cas.

Amb això, esperàvem que els problemes dels residus de l'ARIMA descrits a la secció 4.1.2 se solucionaren i tinguérem, per fi, una distribució normal. Tanmateix, açò no es complia ni per als models de palets sencers ni per als models de mitjos palets i continuàvem tenint errors anòmalament alts o anòmalament baixos, com podem comprovar en la figura 4.6. Per tant, tot i que els residus s'adeqüen prou bé a un soroll blanc, encara tenim moments en què la predicció falla per molt.



(a) Gràfic Box&Whiskers dels residus.

(b) Histograma dels residus.

Figure 4.6.: gràfics d'anàlisi dels residus d'un model ARIMAX(1, 1, 1) $(1, 0, 3)_7$ per a palets sencers, incloent les festivitats i les quantitats rebudes per Mercadona set i huit dies abans.

4.2. Models de xarxes recurrents

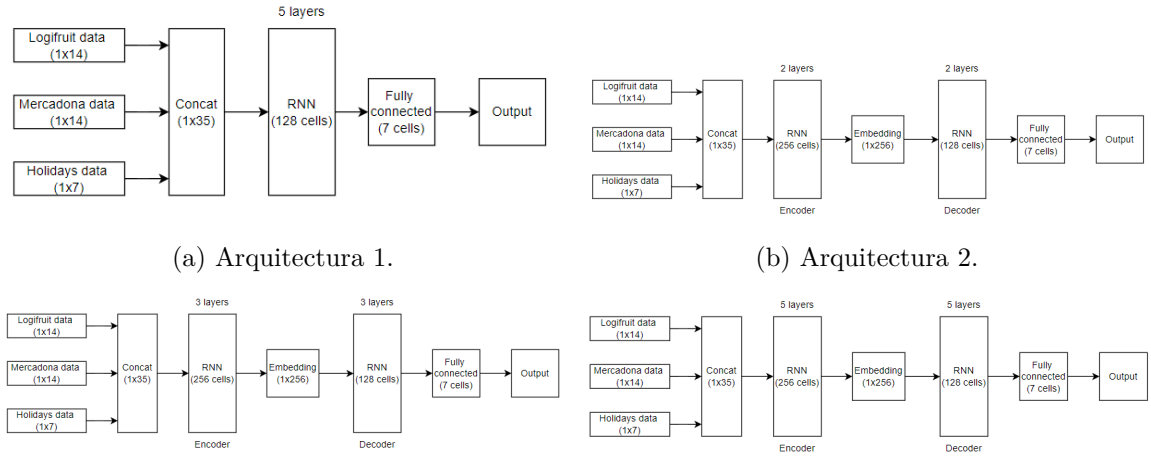
Passem ara a parlar dels models basats en l'algorisme perceptró, aquells que anomenem de xarxes neuronals. Tal com hem vist a la secció 2.2.1, existeixen diferents tipus de xarxes

basades en recurrències. Nosaltres vam a examinar els models unidireccionals (car no té sentit examinar les previsions tenint en compte dades futures) de les xarxes recurrents Elman, que segueixen l'arquitectura de neurona de [24]; les xarxes amb arquitectura neuronal GRU, definides a [29]; i les xarxes amb arquitectura cel · lular LSTM explicades a [30]. Esperàvem que les xarxes Elman foren les que pitjors resultats obtingueren, mentre que les GRU foren les xarxes amb millors prediccions.

En tots els experiments vam utilitzar una funció de pèrdua basada en l'MSE, tal com hem explicat a l'inici del capítol, ja que volíem millorar els resultats dels models estadístics penalitzant aquelles prediccions amb errors més extrems. A més per optimitzar l'ús del factor d'aprenentatge vam incloure un algoritme de refredament simulat en mesa [92] de manera que es variava aquest hiperparàmetre des de 10^{-2} fins a 10^{-5} amb una paciència de deu iteracions.

Per a l'entrenament (i la validació) vam usar remeses de setze observacions [93]. Aquestes remeses comptaven amb els valors de caixes netejades i caixes retornades a Mercadona de dues setmanes prèvies i els valors de festius de la setmana posterior. D'aquesta manera, es pretenia predir els valors de la setmana següent, coincidint amb els valors de festius. Aquests valors van ser tots normalitzats de tal manera que foren tots valors entre zero i u i s'esperava que el resultat fora també un valor en aquest rang.

Respecte a l'arquitectura de xarxes, en vam provar un total de quatre models. El primer d'ells, present a la figura 4.7a, era una xarxa senzilla, de caràcter seqüencial en la que se superposaven cinc capes de cent vint-i-huit unitats i una capa de regressió final densa de set unitats que s'encarregava d'establir els valors adequats.



(a) Arquitectura 1.

(b) Arquitectura 2.

(c) Arquitectura 3.

(d) Arquitectura 4.

Figure 4.7.: arquitectures de xarxes recurrents utilitzades.

Les altres tres arquitectures seguien el model seqüència a seqüència, amb un codificador (*encoder*) de dues-centes cinquanta-sis neurones i un descodificador (*decoder*) de cent vint-i-huit unitats, tal com podem veure a les figures 4.7b, 4.7c i 4.7d. La diferència entre aquest tipus d'arquitectures es trobava en la quantitat de capes que tenien l'*encoder* i el *decoder*, variant entre les dues capes del segon model fins a les cinc capes del quart. En aquest cas

esperàvem que els millors resultats foren els d'arquitectures *encoder-decoder*, reafirmant així els resultats trobats a la literatura.

Els experiments següents els vam desenvolupar tenint en compte les millors arquitectures per a cada tipus d'estructura recurrent. En primer lloc, vam provar d'alterar el factor d'abandonament (*drop out*). Una vegada aquest valor era òptim, vam provar d'alternar l'algorisme d'optimització. Vam provar amb Adam, RMSProp i descens del gradient estocàstic. Finalment, com que els resultats havien de ser entre zero i u, vam pensar que podia ser interessant comprovar si una funció d'activació darrere l'última capa densa podria ser interessant. Així, vam provar amb la Leaky ReLU i amb la Sigmoide sobre el millor model obtingut anteriorment.

4.3. Models basats en el Transformer

La metodologia seguida en els models basats en el Transformer era molt similar a la seguida en els models de xarxes recurrents. Altra vegada comptàvem amb l'MSE com a funció de pèrdua, l'algorisme Adam com a optimitzador, un algorisme de refredament simulat per optimitzar el factor d'aprenentatge (que aquesta vegada anava des de 10^{-1} fins a 10^{-6} amb una paciència de vint iteracions) i remeses de setze observacions per tal de millorar l'eficiència computacional i evitar caure en mínims locals. En aquest cas, però, no vam fer variacions ni en l'algorisme d'optimització i vam utilitzar l'Adam. D'altra banda, tampoc no vam modificar el *drop out*, sinó que vam analitzar només la variació de resultats en les dimensions del model i en el nombre de caps del mecanisme d'atenció múltiple.

Respecte a les arquitectures, cal distingir entre les cinc diferents que vam emprar. La primera va ser el model similar al Transformer original. Aquest model rebia les dades de caixes rebudes a Mercadona i netejades a Logifruit d'un mes previ a la setmana que volíem predir. Aquestes dades, convenientment normalitzades entre zero i u, eren rebudes per dues capes denses independents que s'encarregaven de codificar-les en *embeddings* del mateix format i a sobre dels quals s'aplicava una codificació posicional de caràcter sinusoidal. Després, seguint l'estructura de la figura 2.1, comptava amb un codificador que en el nostre cas capturava la informació de quantitat rebuda per Mercadona i un descodificador que creuava les dades de caixes netejades amb el resultat de l'*encoder*. Finalment, rere el *decoder* hi trobàvem una capa densa que ens feia el paper d'unitat regressora.

La segona arquitectura, visible a la figura 4.8, emprava les dades de festius, però, tanmateix, no comptava amb emmascarament. Açò significa que rebia les dades de festius únicament dels mesos previs. D'aquesta manera, es codificaven aquestes dades mitjançant dues xarxes denses independents i s'afegien a l'*embedding* de l'*encoder* i del *decoder* després d'haver aplicat la codificació posicional. Esperàvem que el model aprendria a introduir la informació de festius de la millor manera possible de tal manera que finalment obtinguérem unes millors previsions.

Tanmateix, com hem explicat, aquesta segona arquitectura no tenia en compte la informació de festius durant la setmana a avaluar, cosa que pot dur a errors molt grossos, ja que, com hem vist a la secció 3.3, hi ha una variació de producció molt important en els dies de vacances. Per tant, és molt rellevant tindre en compte aquesta informació. En la tercera arquitectura vam optar per introduir-la al final, prop de la capa de regressió. Així, com podem comprovar a la figura 4.9 la estructura prèvia era idèntica a la de la figura 4.8, però just abans de la capa densa final, hi afegíem les dades de festius de la setmana a predir

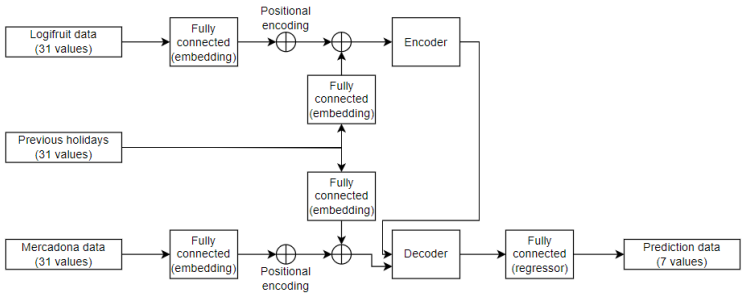


Figure 4.8.: segona arquitectura dels models basats en el Transformer.

codificades mitjançant una capa densa.

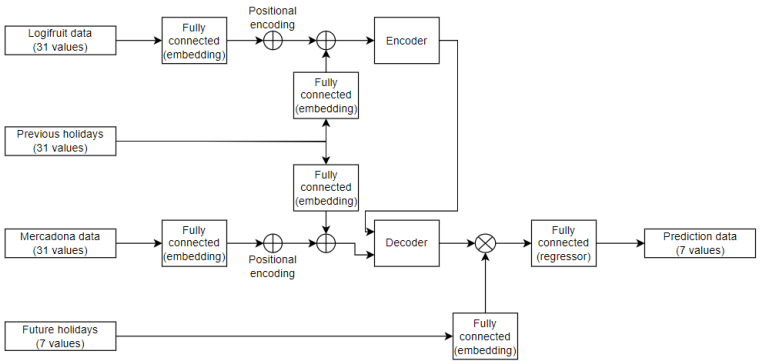


Figure 4.9.: tercera arquitectura dels models basats en el Transformer.

També podem incloure els festius previs mitjançant una codificació prèvia, que combine els festius del context històric i de la setmana que es vol estudiar de manera conjunta. Aquesta combinació serà de caràcter lineal, tal com es pot comprovar a la figura 4.10, de manera que es pot incloure com una altra codificació temporal just després de la codificació posicional.

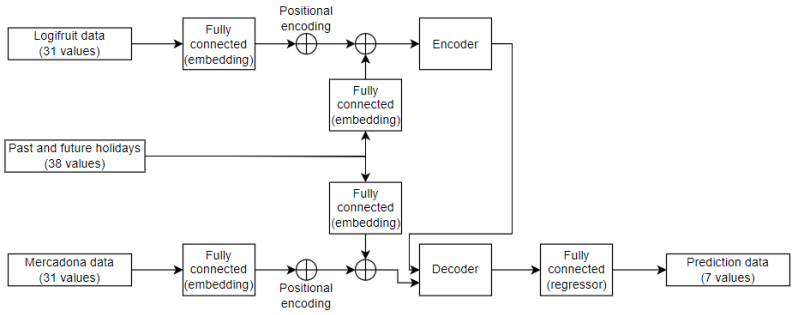


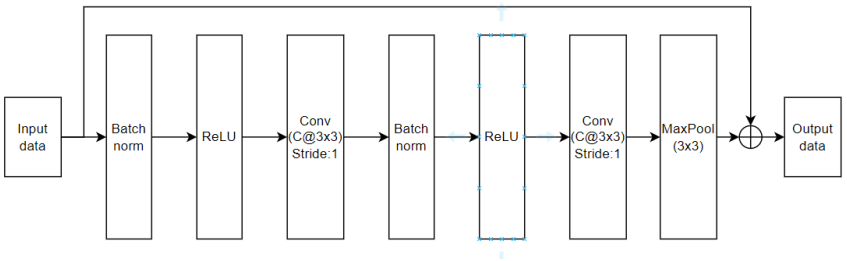
Figure 4.10.: quarta arquitectura dels models basats en el Transformer.

D'altra banda, tal com hem comprovat a la secció 2.2.3, existeixen alguns models d'estat

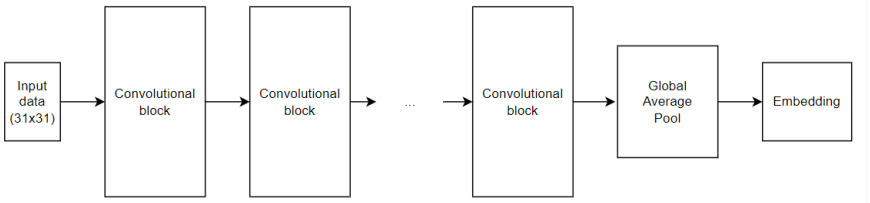
de l'art com el FedFormer de [67] o el QuatFormer de [65] introdueixen informació sobre la freqüència, millorant així els models anteriors. En el nostre cas, vam pensar que podia ser una bona manera de millorar encara més els resultats.

Per aquesta raó vam decidir ampliar la quantitat de dades utilitzant la transformada contínua de la wavelet de Morlet. Així, del vector original de dades $\mathbb{R}^{31}$ vam obtindre una matriu $\mathbb{R}^{31\times31}$ que desglossava la freqüència. Aquesta matriu s'havia de pretractar d'alguna manera per tal d'introduir-la al model Transformer. Nosaltres vam plantejar dues opcions. La primera, feia una descomposició de la imatge en fragments i els aplanava de manera similar a com treballen els VITs (*Visual Transformers*).

L'altra opció, que constituïa la cinquena arquitectura, seguint el model presentat a [94], consistia a emprar convolucions per a reduir la imatge a un vector. D'aquesta manera, tal com podem comprovar a la imatge 4.11, l'*embedding* de les dades s'extreia mitjançant una CNN de paràmetres variables. D'altra banda, a la sisena i última arquitectura dels models basats en Transformers, vam decidir aplicar-hi una convolució a l'interior del mecanisme d'atenció, tal com fan al mateix [94]. El problema d'aquestes aproximacions era que el cost computacional del model s'incrementava sobre manera.



(a) Arquitectura del bloc convolucional. C és el nombre de canals del bloc.



(b) Arquitectura de l'*embedding* convolucional.

Figure 4.11.: funcionament de l'*embedding* convolucional.

4.4. Models de xarxes convolucionals

Veient que els resultats després d'usar la transformada continua de la wavelet de Morlet eren prometedors, vam decidir prendre una nova aproximació emprant xarxes convolucionals. Per a això, concatenàvem les dades de Mercadona amb les dades de Logifruit, obtenint una pseudoimatge de dos canals i trenta-una característiques de trenta-una dimensions. Novament, vam emprar l'MSE com a funció de pèrdua, optimitzant-la mitjançant l'algorisme Adam i utilitzàvem refredament simulat en mesa per optimitzar els valors del factor d'aprenentatge.

La primera remesa d'experiments que vam conduir van ser convolucions sense cap mena d'agrupament (*pooling*) i amb una ReLU prèvia. Així, vam provar amb tres arquitectures diferents. La primera comptava amb set convolucions de quatre, sis huit, deu, dotze, catorze i setze filtres 3×3 amb *stride* unitari i sense *padding*. La segona tenia menys capes, només tres, ja que la gambada era de dos valors. Així doncs, teniem tres convolucions amb huit, setze i trenta-dos filtres 3×3 sense *padding*. Finalment, la darrera arquitectura d'aquest bloc constava d'una primera convolució de quatre filtres 3×3 amb *stride* u i *padding* u, una capa de huit filtres 3×3 de *stride* 1 i *padding* dos, una tercera capa de setze filtres 3×3 amb *stride* dos i sense *padding* i una darrera convolució amb trenta-dos filtres 2×2 amb *stride* 1. La idea d'aquesta última arquitectura era expandir la imatge amb el padding per a després contraure-la i obtindre informació més detallada.

A la segona remesa d'experiments, que comptava només amb dues arquitectures, li vam incloure agrupament màxim (*max pooling*) just després de cada convolució i vam millorar solament les dues primeres arquitectures descrites anteriorment. D'altra banda, el tercer bloc d'experiments incloïa una concatenació del vector obtingut després del *pooling* global amb el vector de properes festivitats. Novament, vam utilitzar com a base les dues arquitectures de la remesa anterior. En la quarta remesa d'experiments vam canviar les dues arquitectures anteriors per a en lloc de concatenar els dos vectors, sumar-los.

Per a la cinquena i sisena remesa d'experiments vam emprar l'arquitectura ResNet [95] amb connexions residuals. L'única diferència entre aquestes tres remeses era que a la sisena afegíem les dades de festius en un *embedding* abans d'aplicar-hi el classificador i a la sisena féiem el mateix però multiplicant-hi l'*embedding* dels festius. Així doncs, la primera arquitectura d'aquests dos grups d'experiments constava de quatre blocs convolucionals de quatre capes cadascun i seixanta-quatre, cent vint-i-huit, dos-cents cinquanta-sis i cinc-cents dotze filtres 3×3 cadascun amb un *stride* de la primera capa del bloc de dos píxels, on cada filtre té la seua corresponent normalització per remeses i funció d'activació ReLU. D'altra banda, la segona arquitectura canviava el *padding*, sent aquest de tres, quatre, sis i tres píxels per a cada bloc convolucional.

Les següents tres remeses eren exactament les mateixes, però en aquest cas aplicant l'arquitectura DenseNet [96]. Així doncs, la remesa huit es tractava d'una DenseNet on incloíem els festius sense *embedding*, mentre que la remesa nou codificava els festius en un *embedding* i la remesa deu no afegia sinó que multiplicava aquests festius tal com féiem en la tercera arquitectura dels models basats en el Transformer.

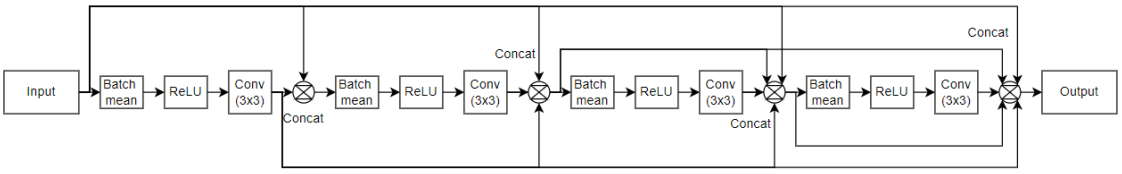


Figure 4.12.: arquitectura del DenseBlock.

Per a entendre aquesta xarxa, cal saber que consideràvem un bloc dens com a un seguit de convolucions i concatenacions, tal com es pot veure a la figura 4.12. Les arquitectures

d'aquests blocs eren sempre de tres blocs de quatre capes cada bloc. La primera de les dues arquitectures comptava amb convolucions de 64, 128 i altra vegada 64 filtres, mentre que la segona arquitectura era de 32, 64 i 32 filtres. A més, també disposàvem d'una capa de transició que constava d'una convolució de filtres 1×1 i un *max pooling*.

4.5. Models preentrenats

Un dels problemes més grans que tenia aquesta tasca era la quantitat de dades de què disposàvem. Solament comptàvem amb observacions d'aproximadament dos anys (set-cents tres dies, per ser més concrets) i, tal com hem explicat a l'inici de la secció 2.2, les xarxes neuronals artificials requereixen una gran quantitat de dades per a poder funcionar correctament. Davant aquesta problemàtica caben dues solucions: l'augmentació de les dades o l'ús de models preentrenats.

Dins de les tècniques d'augment de dades, podem incloure soroll Gaussià, però caiem en el risc de poder alterar l'estructura interna de la sèrie, provocant errors greus en les prediccions. Una altra opció seria capgirar les dades i col · locar les observacions inicials al final, assumint que la sèrie és simètrica. El problema és que, tal com hem observat a la secció 3, la sèrie de palets sencers no és estacionària i amb tendència variable, raó per la qual no podem emprar aquesta tècnica d'augmentació.

Una altra opció és expandir el conjunt de dades mitjançant el domini de les freqüències. A [97], Gao et al. proposen realitzar pertorbacions en amplitud i fase espectrals, per a augmentar el nombre d'observacions. Altres aproximacions, sobretot en el camp del reconeixement de la parla [98], proposen emprar mecanismes d'ampliació de dades utilitzant la freqüència Mel. En el nostre cas, aquest tipus d'ampliacions serien encara més eficaces aplicades sobre les convolucionals, que empren la transformada continua de Wavelet.

Finalment, també es poden aplicar tècniques més avançades com pertorbacions a l'espai d'*embedding* o entrenaments adversarials per a generar noves observacions.

En el nostre cas, però, hem preferit emprar models preentrenats. Per un costat hem emprat l'InFormer definit a [63] i l'AutoFormer de [76], dos models de l'estat de l'art entrenats per a predir la quantitat de turistes que visiten una població a escala mensual. En el nostre cas, vam realitzar les conversions pertinents i vam extraure un context de cent observacions amb un horitzó de predicció de set dies. Vam reentrenar tot el model utilitzant la *Negative-Log Likelihood* com a funció de pèrdua perquè era l'única funció acceptada pel *framework* i vam optimitzar aquesta funció mitjançant l'algorisme Adam amb refredament simulat per optimitzar el factor d'aprenentatge.

D'altra banda, també hem emprat models preentrenats que tinguen en compte les pseudoimatges obteses en la transformada continua de wavelet. D'aquesta manera, hem emprat dos Transformers com són el Google-ViT [99][100] i el SWIN [101]. A més, també ens hem enfocat en els models convolucionals, aplicant una ResNet-50 [102] i una DenseNet-201 [96].

4.6. Models Light-GBM

Finalment, tal com hem vist a la secció 2.3, els models Light-GBM, basats en l'optimització dels algorismes de *boosting* per a arbres, obtenen molt bons resultats en aquest tipus de tasca.

Per tant, vam pensar que seria una bona idea emprar-los per a realitzar les prediccions de la nostra sèrie temporal.

Tanmateix, calia discernir quines eren les dades que usaríem per a aplicar-hi a sobre el model LGBM. En primer lloc, podíem utilitzar la sèrie directament, però açò seria summament ineficient, puix hem fet un esforç per extraure les característiques més significatives en forma d'*embeddings*. Per tant, també té sentit utilitzar els vectors dels diferents models que disposem.

En el nostre cas, vam utilitzar tant la sèrie original, com el vector resultant de la capa prèvia a la de regressió en aquelles xarxes neuronals que tenien millors resultats. Esperàvem que aquest darrer vector fora capaç de reunir les característiques més significatives de la sèrie i que ajudaria a millorar els resultats del Light-GBM aplicat directament a la sèrie.

En el següent capítol s'observaran els resultats dels diferents experiments aplicats a cadascun d'aquests models, així com les comparacions entre les diverses tècniques que hem aplicat.

5. Resultats

Una vegada desenvolupats els procediments per a l'experimentació, és moment d'observar els resultats obtinguts. Aquest capítol es disgrega en sis seccions que comenten els diferents tipus de models separadament per a discernir quin n'és el millor mentre que una darrera secció s'encarrega d'obtenir quin és el millor model de tots els que hem obtingut.

Aquest capítol es divideix en set seccions: una per als models estadístics, una altra per als models basats en les xarxes recurrents, una tercera que adreça els resultats dels models basats en el Transformer, una quarta que s'enfoca en els models de xarxes convolucionals, una altra que se centra en l'experimentació amb models preentrenats, una penúltima que observa els resultats obtinguts amb els models Light-GBM i una darrera que compara els diferents models aplicats.

5.1. Models estadístics

En primer lloc, explicarem els resultats obtinguts mitjançant els models estadístics. Com hem exposat a la secció 4.1, hem aplicat una sèrie d'experiments amb models de suavitzat exponencial triple, amb models ARIMA i, finalment, amb models ARIMAX.

5.1.1. Models de suavitzat exponencial

Primerament, quan vam dissenyar els experiments dels models Holt & Winters, esperàvem que el que millor s'ajustara a les dades de palets sencers fora aquell que capturara l'estacionalitat de manera additiva mentre que la tendència fora de caràcter multiplicatiu. Aquesta deducció va ser extreta per descart, ja que després de desenvolupar la descomposició STL (de caràcter additiu) present a la figura 5.1, observàvem que encara persistia una tendència bastant marcada, mentre que l'estacionalitat pràcticament desapareixia.

D'altra banda, pensàvem que en el cas dels models de mitjos palets el millor model seria aquell que regulava tant la tendència com l'estacionalitat de forma additiva, ja que a la figura 3.7 comprovàvem que després d'aplicar una descomposició STL de caràcter additiu, tant la tendència com l'estacionalitat es diluïen, tot i que encara quedaven algunes restes causades, molt probablement, per valors anòmals.

Tendència	Estacionalitat	RMSE palets sencers	RMSE mitjos palets
Additiva	Additiva	2293.55	**1636.75**
Additiva	Multiplicativa	2871.29	1914.68
Multiplicativa	Additiva	**2305.66**	1642.84
Multiplicativa	Multiplicativa	5831.99	100202.71

Table 5.1.: resultats per als models Holt& Winters.

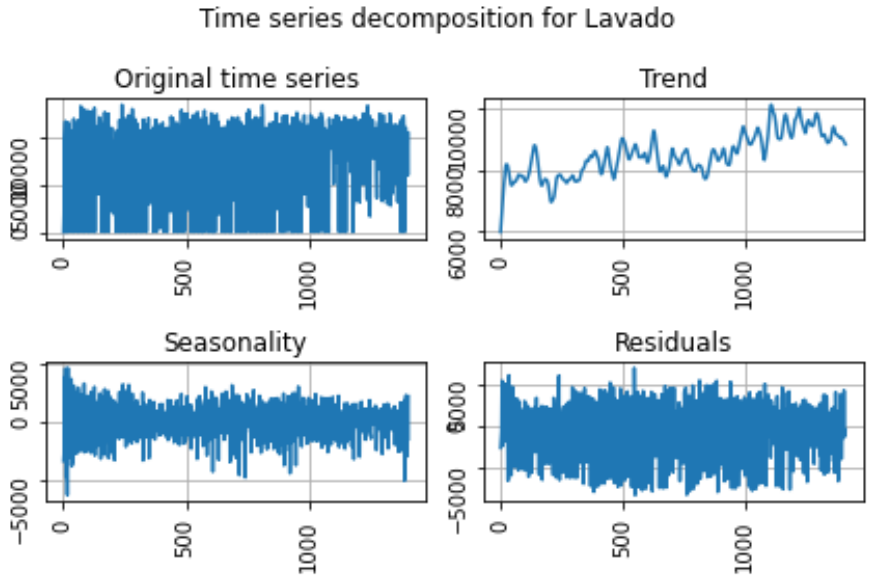


Figure 5.1.: descomposició STL per a la quantitat de palets sencers llavada.

A la taula 5.1 s'observa com els resultats avalen aquestes hipòtesis que plantejàrem. Cal remarcar els terribles resultats assolits amb els models multiplicatius, en ambdós casos amb errors més alts que amb la resta de models. D'altra banda, també hem de destacar que els errors en prediccions de la sèrie de mitjos palets són molt inferiors als resultats de la sèrie de palets sencers no només perquè la sèrie pren valors més baixos, com hem comprovat a la secció 3.3, sinó perquè es tracta d'una sèrie estacionària, molt més senzilla de modelar.

5.1.2. Models ARIMA

Observem ara, els resultats de la secció 4.1.2, on havíem definit quatre models ARIMA per a la sèrie dels palets sencers i tres models ARIMA per a la sèrie dels mitjos palets.

Model	RMSE
SARIMA (0, 1, 2) $(0, 1, 3)_7$	2347.54
SARIMA (1, 1, 1) $(1, 0, 2)_7$	2303.36
SARIMA (2, 1, 1) $(1, 0, 3)_7$	**2290.26**
SARIMA (2, 1, 2) $(1, 0, 3)_7$	2293.10

Table 5.2.: resultats per als models ARIMA aplicats a la sèrie temporal dels palets sencers.

La taula 5.2 mostra els resultats obtinguts per a les dades de palets sencers. Tal com podem veure, sembla que el millor model és el SARIMA (2, 1, 1) (1 0, $3)_7$ i aquest model deuria ser lleugerament millor que el millor model Holt & Winters, d'acord amb l'RMSE. En qualsevol cas, nosaltres ens quedarem amb els tres millors models (excloent el SARIMA (0, 1, 2) $(0, 1, 3)_7$) i aplicarem la tècnica de *rolling window* per obtindre quina seria la millor predicció en cada instant.

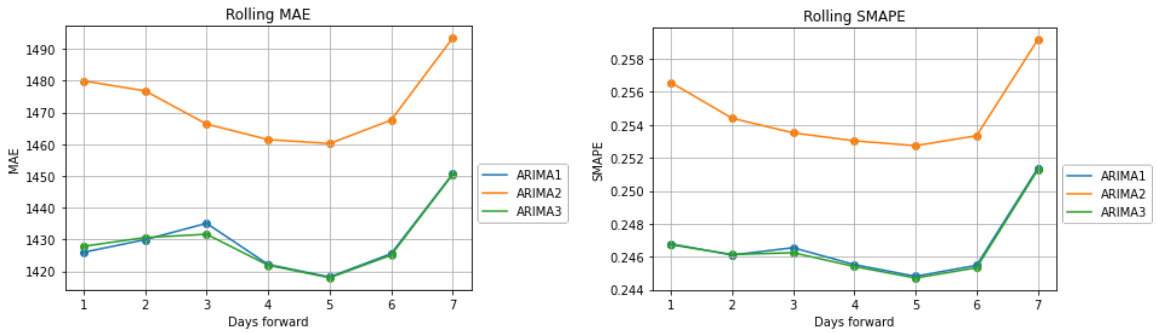


Figure 5.2.: avaluació d'errors per predicció en els models ARIMA aplicats a la sèrie de palets sencers.

A la figura 5.2 observem com l'ARIMA 2 (SARIMA (2, 1, 2), $(1, 0, 3)_7$) té un error una mica més gran que els altres dos models, que pràcticament no són diferenciables. Sembla que durant les primeres prediccions el model ARIMA 1 (SARIMA (2, 1, 1) $(1, 0, 3)_7$) funciona una mica millor, mentre que en la resta d'observacions seria el model ARIMA 3 (SARIMA (1, 1, 1), $(1, 0, 3)_7$). Per tant, si volem fer un *ensemble* de models ARIMA, caldria que les primeres dues prediccions es feren amb l'ARIMA 1, mentre que les darreres cinc observacions s'haurien de fer amb el model ARIMA 3.

D'altra banda, a la taula 5.3 hi trobem els resultats per als models ARIMA aplicats a la sèrie de mitjos palets. En aquest cas, cal que recordem que es tracta d'una sèrie estacionària, per la qual cosa no deuria ser necessari un decalat com en la sèrie de palets sencers. Els resultats semblen avalar aquesta hipòtesi, de manera que el millor model és el SARMA (2, 1) $(4, 4)_7$, que en aquest cas no supera al millor model Holt & Winters en RMSE.

Model	RMSE
SARMA (2, 1) $(4, 4)_7$	**1695.93**
SARIMA (1, 1, 3) $(1, 0, 1)_7$	1721.91
SARIMA (1, 1, 3) $(1, 1, 1)_7$	1705.87

Table 5.3.: resultats per als models ARIMA aplicats a la sèrie temporal dels mitjos palets.

Si novament organitzem aquests tres models per considerar les millors prediccions amb la tècnica de *rolling window*, podem obtindre una gràfica com la de la figura 5.3. Aquesta gràfica ens indica com el SARMA (2, 1) $(4, 4)_7$ presenta un error força més pronunciat que els altres dos models. Aquests dos semblen funcionar de manera semblant segons el MAE, però en el SMAPE s'observa que el model SARIMA (1, 1, 3) $(1, 0, 1)_7$ té menys error a la predicció, sent d'aquesta manera el millor dels tres.

5.1.3. Models ARIMAX

Finalment, tal com hem explicat a la secció 4.1.3, hem aplicat variables exògenes sobre els millors models ARIMA. Aquests valors han sigut triats segons l'efecte que pogueren tindre (més informació a la secció 3.3). Així doncs, hem observat la quantitat de caixes rebudes per Mercadona set i huit dies abans de la predicció i la presència de festius durant la predicció.

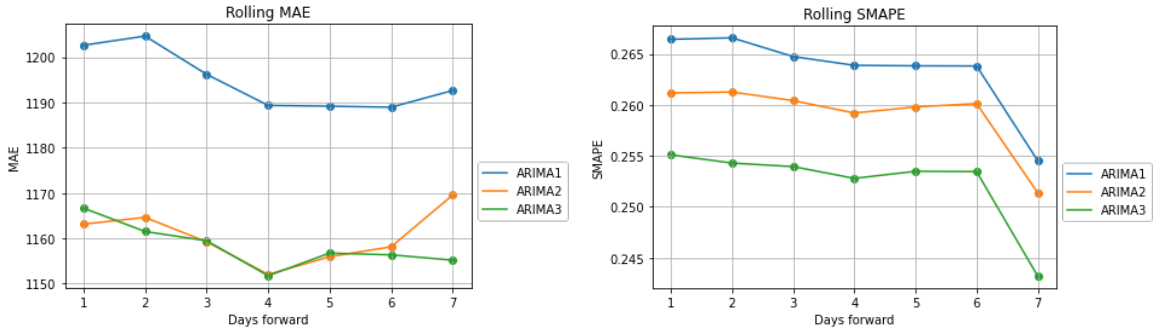


Figure 5.3.: avaluació d'errors per predicció en els models ARIMA aplicats a la sèrie de mitjos palets.

Model	Variables exògenes	RMSE
SARIMAX (2, 1, 1) $(1, 0, 3)_7$	Valors Mercadona 7 dies abans	2576.94
SARIMAX (2, 1, 2) $(1, 0, 3)_7$	Valors Mercadona 7 dies abans	2241.17
SARIMAX (1, 1, 1) $(1, 0, 2)_7$	Valors Mercadona 7 dies abans	2279.56
SARIMAX (2, 1, 1) $(1, 0, 3)_7$	Valors Mercadona 8 dies abans	2240.37
SARIMAX (2, 1, 2) $(1, 0, 3)_7$	Valors Mercadona 8 dies abans	2215.48
SARIMAX (1, 1, 1) $(1, 0, 2)_7$	Valors Mercadona 8 dies abans	2233.73
SARIMAX (2, 1, 1) $(1, 0, 3)_7$	Valors Mercadona 7 i 8 dies abans	2678.53
SARIMAX (2, 1, 2) $(1, 0, 3)_7$	Valors Mercadona 7 i 8 dies abans	2213.47
SARIMAX (1, 1, 1) $(1, 0, 2)_7$	Valors Mercadona 7 i 8 dies abans	2224.39
SARIMAX (2, 1, 1) $(1, 0, 3)_7$	Festius	1892.37
SARIMAX (2, 1, 2) $(1, 0, 3)_7$	Festius	1886.09
SARIMAX (1, 1, 1) $(1, 0, 2)_7$	Festius	1892.37
SARIMAX (2, 1, 1) $(1, 0, 3)_7$	Festius i valors Mercadona 7 dies abans	1889.06
SARIMAX (2, 1, 2) $(1, 0, 3)_7$	Festius i valors Mercadona 7 dies abans	1863.59
SARIMAX (1, 1, 1) $(1, 0, 2)_7$	Festius i valors Mercadona 7 dies abans	1890.15
SARIMAX (2, 1, 1) $(1, 0, 3)_7$	Festius i valors Mercadona 7 i 8 dies abans	1804.63
SARIMAX (2, 1, 2) $(1, 0, 3)_7$	Festius i valors Mercadona 7 i 8 dies abans	1917.22
SARIMAX (1, 1, 1) $(1, 0, 2)_7$	**Festius i valors Mercadona 7 i 8 dies abans**	**1803.77**

Table 5.4.: resultats per als models ARIMAX aplicats a la sèrie temporal dels palets sencers.

La taula 5.4 mostra els resultats de l'experimentació amb els models ARIMAX sobre la sèrie de palets sencers. Tal com podem comprovar, sembla que introduir les dades de caixes rebudes per Mercadona set dies abans de la predicció no ajuda gaire al model i, de fet, en el cas del SARIMAX (2, 1, 1) $(1, 0, 3_7$, empitjora els resultats. Tanmateix, introduir els mateixos valors, però de huit dies abans, té un efecte molt més beneficiós, aconseguint millorar tots els resultats dels ARIMA. Tot i això, la variable exògena que més millora els resultats és la de festius, disminuint l'error en gran quantitat. Finalment, el millor model segons l'RMSE s'ha obtingut d'aprofitar tant el nombre de caixes rebudes per Mercadona set i huit dies abans, com els festius.

Tal com hem fet amb els models ARIMA, podem avaluar els resultats mitjançant la tècnica

de *rolling window*. Així doncs, prenem els tres millors models d'acord amb la taula 5.4 i obtenim uns resultats com els de la figura 5.4. Aquesta gràfica ens demostra que el model que pitjor funciona dels tres tant en MAE com en SMAPE (i en RMSE també) és, sens dubte, el SARIMAX (2, 1, 1) $(1, 0, 3)_7$ prenent els festius i les entrades a Mercadona 7 dies abans com a variables exògenes. D'altra banda, els altres dos models sembla que funcionen de manera similar segons el SMAPE, tot i que el SARIMAX (1, 1, 1) $(1, 0, 2)_7$ amb totes les variables exògenes funciona millor tant en MAE com en RMSE que el model SARIMAX (2, 1, 1) $(1, 0, 3)_7$ amb totes les variables exògenes. Per tant, aquest serà el model ARIMAX òptim.

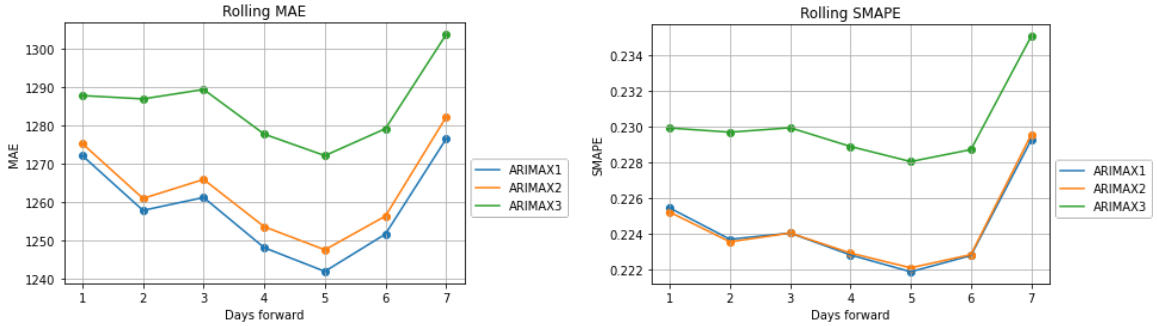


Figure 5.4.: avaluació d'errors per predicció en els models ARIMAX aplicats a la sèrie de palets sencers.

Finalment, la taula 5.5 desenvolupa els resultats obtinguts d'aplicar els models ARIMAX a la sèrie dels mitjos palets. La diferència més significativa respecte als models anteriors i salvant les distàncies, és que en aquest cas la imputació de les dades de Mercadona set dies abans sí que té un efecte beneficiós sobre els models, reduint-ne l'error.

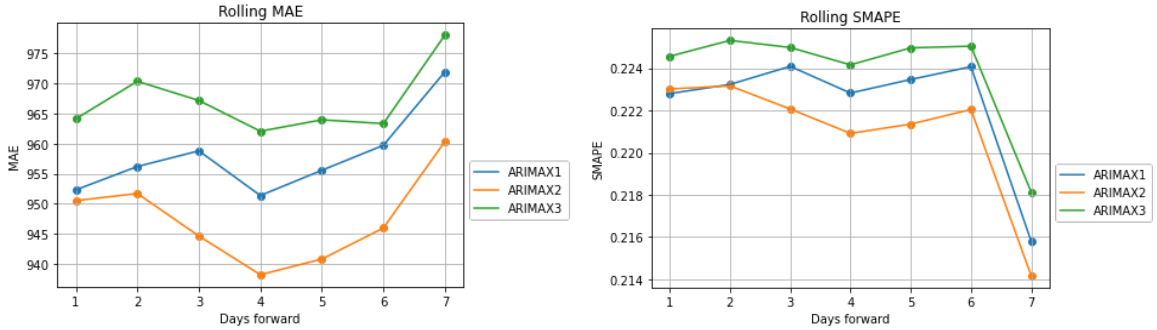


Figure 5.5.: avaluació d'errors per predicció en els models ARIMAX aplicats a la sèrie de mitjos palets.

A la figura 5.5 s'observen els resultats en aplicar la tècnica de *rolling window* sobre els tres millors models ARIMAX. Tal com podem veure, el tercer millor model segons l'RMSE (SARMAX (2, 1) $(4, 4)_7$ amb totes les variables exògenes) és el que major error registra tant en MAE com en SMAPE. D'altra banda, tot i que el model SARIMAX (1, 1, 3) $(1, 1, 1)_7$ amb totes les variables exògenes era el que millor funcionava, la veritat és que és superat tant

Model	Variables exògenes	RMSE
SARMAX (2, 1) $(4, 4)_7$	Valors Mercadona 7 dies abans	1643.85
SARIMAX (1, 1, 3) $(1, 1, 1)_7$	Valors Mercadona 7 dies abans	1685.87
SARIMAX (1, 1, 3) $(1, 0, 1)_7$	Valors Mercadona 7 dies abans	1637.32
SARMAX (2, 1) $(4, 4)_7$	Valors Mercadona 8 dies abans	1655.40
SARIMAX (1, 1, 3) $(1, 1, 1)_7$	Valors Mercadona 8 dies abans	1675.65
SARIMAX (1, 1, 3) $(1, 0, 1)_7$	Valors Mercadona 8 dies abans	1957.40
SARMAX (2, 1) $(4, 4)_7$	Valors Mercadona 7 i 8 dies abans	1645.58
SARIMAX (1, 1, 3) $(1, 1, 1)_7$	Valors Mercadona 7 i 8 dies abans	1681.74
SARIMAX (1, 1, 3) $(1, 0, 1)_7$	Valors Mercadona 7 i 8 dies abans	1641.36
SARMAX (2, 1) $(4, 4)_7$	Festius	1870.66
SARIMAX (1, 1, 3) $(1, 1, 1)_7$	Festius	1418.68
SARIMAX (1, 1, 3) $(1, 0, 1)_7$	Festius	1536.16
SARMAX (2, 1) $(4, 4)_7$	Festius i valors Mercadona 7 dies abans	1400.00
SARIMAX (1, 1, 3) $(1, 1, 1)_7$	Festius i valors Mercadona 7 dies abans	1387.56
SARIMAX (1, 1, 3) $(1, 0, 1)_7$	Festius i valors Mercadona 7 dies abans	1514.30
SARMAX (2, 1) $(4, 4)_7$	Festius i valors Mercadona 8 dies abans	1504.87
SARIMAX (1, 1, 3) $(1, 1, 1)_7$	Festius i valors Mercadona 8 dies abans	1411.80
SARIMAX (1, 1, 3) $(1, 0, 1)_7$	Festius i valors Mercadona 8 dies abans	1468.45
SARMAX (2, 1) $(4, 4)_7$	Festius i valors Mercadona 7 i 8 dies abans	1403.06
SARIMAX (1, 1, 3) $(1, 1, 1)_7$	**Festius i valors Mercadona 7 i 8 dies abans**	**1386.45**
SARIMAX (1, 1, 3) $(1, 0, 1)_7$	Festius i valors Mercadona 7 i 8 dies abans	1426.66

Table 5.5.: resultats per als models ARIMAX aplicats a la sèrie temporal dels mitjos palets.

en SMAPE com en MAE pel model SARIMAX(1, 1, 3) $(1, 1, 1)_7$ amb la quantitat registrada per Mercadona una setmana abans i les dades de festius com a variables exògenes. Per tant, aquest serà el model que escollirem per tal d'obtindre la millor predicció.

5.2. Models de xarxes recurrents

A continuació explicarem els resultats obtinguts durant l'experimentació amb xarxes recurrents. Tal com hem explicat a la secció 4.2, hem implementat les arquitectures Elman, GRU i LSTM tant amb *encoder* i *decoder*, com de manera simple.

Respecte a l'experimentació, vam començar observant les quatre diferents arquitectures que havíem implementat: una de caràcter simple i les altres de caràcter *encoder-decoder*. Vam prosseguir observant com el diferent nombre de capes afectava el funcionament del model. Seguidament, vam alterar el *drop out*, per comprovar si un valor més alt de connexions elidides ajudava el model a generalitzar o si, per contra, no arribava a entrenar adequadament. Finalment, vam provar amb diferents algorismes d'optimització, per comprovar si hi havia alguna diferència entre ells.

A la taula 5.6, trobem els resultats per a les xarxes de tipus Elman. Tal com podem veure, sembla que d'entre el model simple i el *encoder-decoder*, el primer funciona millor. Tanmateix, la diferència no és abismal, per la qual cosa continuarem estudiant els models

encoder-decoder. Així doncs, observem que un nombre de caps més elevat comporta a un augment poc significatiu de l'error i que un drop-out més alt sembla reduir una mica més aquesta diferència entre el valor predit i el real. Tanmateix, continuem amb valors molt alts on la diferència no és realment significativa. D'altra banda, en alterar l'algorisme optimitzador, comprovem que el descens per gradient estocàstic obté molts millors resultats, assolint un RMSE de 2441.80, quasi mil punts per davall del model més proper (3264.91).

Arquitectura	Capes	Drop out	Optimitzador	RMSE
Simple	5	0.2	Adam	3273.57
Encoder-decoder	3+3	0.2	Adam	3411.13
Encoder-decoder	5+5	0.2	Adam	3406.16
Encoder-decoder	2+2	0.2	Adam	3282.80
Encoder-decoder	5+5	0.5	Adam	3270.47
Encoder-decoder	5+5	0.8	Adam	3264.91
Encoder-decoder	5+5	0.1	Adam	3282.90
Encoder-decoder	5+5	0.3	Adam	3274.24
Encoder-decoder	5+5	0.2	RMSProp	3397.87
Encoder-decoder	**5+5**	**0.2**	**SGD**	**2441.80**

Table 5.6.: resultats per als models Elman aplicats a la sèrie temporal dels palets sencers.

D'altra banda, a la taula 5.7 s'observen els resultats de les xarxes Elman aplicades a la sèrie de mitjos palets. Aquesta taula demostra que el model simple sembla funcionar millor que la resta d'arquitectures. Tanmateix, com altra vegada aquesta diferència és força lleu, continuarem amb els models *encoder-decoder* amb cinc capes a l'*encoder* i cinc al *decoder*. Després d'efectuar una experimentació, comprovem que els valors de *drop out* alts semblen tindre millors resultats que els models de *drop out* baixos. D'entre aquests, el nostre millor model seria el de *drop out* 0.8. Finalment, quant a l'optimització dels paràmetres, l'Adam sembla ser l'algorisme que millor resultats produeix, per la qual cosa serà el que emprarem.

Arquitectura	Capes	Drop Out	Optimizador	RMSE
Simple	5	0.2	Adam	2426.47
Encoder-decoder	3+3	0.2	Adam	2434.96
Encoder-decoder	5+5	0.2	Adam	2461.36
Encoder-decoder	2+2	0.2	Adam	2604.81
Encoder-decoder	5+5	0.5	Adam	2438.61
Encoder-decoder	**5+5**	**0.8**	**Adam**	**2413.28**
Encoder-decoder	5+5	0.9	Adam	2532.36
Encoder-decoder	5+5	0.7	Adam	2432.35
Encoder-decoder	5+5	0.8	RMSProp	2437.01
Encoder-decoder	5+5	0.8	SGD	2481

Table 5.7.: resultats per als models Elman aplicats a la sèrie temporal dels mitjos palets.

Quant als models basats en GRU, comprovem a la taula 5.8 que amb una arquitectura simple ja assolim uns resultats millors que en el millor model dels Elman per als palets sencers (RMSE de 2429.65). Tanmateix, una vegada assolit, no aconseguírem millorar-lo de

cap de les maneres. Ni canviant l'arquitectura a *encoder-decoder*, ni alterant el nombre de capes, ni modificant el *drop out* ni fins i tot emprant un altre algorisme optimitzador. Així, el model triat finalment per representar les GRU en la sèrie de palets sencers seria aquest.

Arquitectura	Capes	Drop Out	Optimitzador	RMSE
Simple	**5**	**0.2**	**Adam**	**2429.65**
Encoder-decoder	3+3	0.2	Adam	3349.51
Encoder-decoder	5+5	0.2	Adam	3309.27
Encoder-decoder	2+2	0.2	Adam	3299.72
Simple	5	0.5	Adam	3330.36
Simple	5	0.8	Adam	3249.97
Simple	5	0.6	Adam	3356.01
Simple	5	0.4	Adam	3292.65
Simple	5	0.5	RMSProp	3351.14
Simple	5	0.2	SGD	3449.88

Table 5.8.: resultats per als models GRU aplicats a la sèrie temporal dels palets sencers.

Pel que fa a la sèrie dels mitjos palets, podem comprovar a la taula 5.9 quins són els resultats per als experiments desenvolupats. En primer lloc, observem que altra vegada una major profunditat tant en l'*encoder* com al *decoder* provoca que obtinguem uns resultats lleugerament millors. A més, també comprovem que en aquest cas un *drop out* més alt no millora els resultats sinó que pel contrari els empitjora, sent que els millors resultats que hem obtingut es registren amb un 20% de *drop out*. Per acabar, els algorismes d'optimització no milloren els resultats de l'Adam, tot i que cal que destaquem els resultats obtinguts amb el RMSProp, que són exactament els mateixos que els de l'Adam, per la qual cosa podem deduir que es deu tractar d'algun tipus de mínim (local segurament, puix altres models lineals com l'ARIMA obtenen millors resultats).

Arquitectura	Capes	Drop Out	Optimizador	RMSE
Simple	5	0.2	Adam	2384.69
Encoder-decoder	3+3	0.2	Adam	2432.29
Encoder-decoder	**5+5**	**0.2**	**Adam**	**2365.36**
Encoder-decoder	2+2	0.2	Adam	2487.94
Encoder-decoder	5+5	0.5	Adam	2380.93
Encoder-decoder	5+5	0.8	Adam	2397.24
Encoder-decoder	5+5	0.1	Adam	2420.77
Encoder-decoder	5+5	0.3	Adam	2500.34
Encoder-decoder	5+5	0.2	RMSProp	2365.36
Encoder-decoder	5+5	0.2	SGD	2602.55

Table 5.9.: resultats per als models GRU aplicats a la sèrie temporal dels mitjos palets.

Finalment, cal que parlem dels models basats en LSTM. La taula 5.10 reflexa els resultats assolits per aquests models. En aquest cas, observem que el model simple sembla funcionar millor que l'*encoder-decoder*. Tot i això, en disminuir el nombre de capes, aquest segon model assoleix uns resultats millors en RMSE que el millor model GRU i el millor model Elman.

Tanmateix, millorar aquest model sembla complicat, puix ni els canvis al *drop out* ni els canvis en algorisme optimitzador no han arribat a millorar el model.

Arquitectura	Capes	Drop Out	Optimitzador	RMSE
Simple	5	0.2	Adam	2586.54
Encoder-decoder	3+3	0.2	Adam	3268.61
Encoder-decoder	5+5	0.2	Adam	3260.88
Encoder-decoder	**2+2**	**0.2**	**Adam**	**2397.25**
Encoder-decoder	2+2	0.5	Adam	3271.70
Encoder-decoder	2+2	0.8	Adam	3254.04
Encoder-decoder	2+2	0.1	Adam	3262.10
Encoder-decoder	2+2	0.5	RMSProp	3334.32
Encoder-decoder	2+2	0.2	SGD	3428.71

Table 5.10.: resultats per als models LSTM aplicats a la sèrie temporal dels palets sencers.

Quant als models LSTM aplicats a la sèrie de mitjos palets, només cal que ens referim a la taula 5.11 per observar-ne els resultats. Com podem observar, sembla que el model *encoder-decoder* funciona millor que el model simple (tot i que molt lleugerament) i augmentar la profunditat té un efecte positiu en la reducció d'errors. D'altra banda, sembla que el millor valor per al *drop out* registrat seria amb 50%, puix que tant augmentar com disminuir aquest valor porta a uns pitjors resultats. Finalment, el millor algorisme d'optimització registrat és l'Adam, car tant el RMSProp com el SGD donen pitjors resultats.

Arquitectura	Capes	Drop out	Optimizador	RMSE
Simple	5	0.2	Adam	2382.87
Encoder-decoder	3+3	0.2	Adam	2385.82
Encoder-decoder	5+5	0.2	Adam	2363.13
Encoder-decoder	2+2	0.2	Adam	2387.46
Encoder-decoder	**5+5**	**0.5**	**Adam**	**2347.79**
Encoder-decoder	5+5	0.8	Adam	2416.08
Encoder-decoder	5+5	0.4	Adam	2360.95
Encoder-decoder	5+5	0.6	Adam	2348.60
Encoder-decoder	5+5	0.2	RMSProp	2455.15
Encoder-decoder	5+5	0.2	SGD	2588.98

Table 5.11.: resultats per als models LSTM aplicats a la sèrie temporal dels mitjos palets.

5.3. Models basats en el Transformer

Dins dels models basats en el Transformer, podem distingir aquells que no empren la freqüència i aquells que sí que l'utilitzen. En el cas dels primers, cal que expliquem que no vam obtindre gaire bons resultats, molt probablement per la falta de dades (recordem que només comptàvem amb dos anys d'observacions diàries).

Arquitectura	Profunditat	Caps d'atenció	Dimensió d'*embedding*	RMSE
Arquitectura 1	3	8	64	3264.56
Arquitectura 1	3	4	64	3285.91
Arquitectura 1	3	16	64	3301.25
Arquitectura 1	5	8	64	3299.33
Arquitectura 1	2	8	64	3273.41
Arquitectura 1	3	8	128	3293.41
Arquitectura 1	3	8	32	3390.12
Arquitectura 2	3	8	64	3299.68
Arquitectura 2	3	4	64	3377.04
Arquitectura 2	3	16	64	3369.63
Arquitectura 2	5	8	64	3303.13
Arquitectura 2	2	8	64	3306.72
Arquitectura 2	4	8	64	3268.40
Arquitectura 2	4	8	128	3279.97
Arquitectura 2	4	8	32	3365.80
Arquitectura 3	3	8	64	3271.59
Arquitectura 3	3	4	64	3127.77
Arquitectura 3	**3**	**16**	**64**	**3123.02**
Arquitectura 3	5	16	64	3276.57
Arquitectura 3	2	16	64	3284.22
Arquitectura 3	4	16	64	3270.80
Arquitectura 3	3	16	128	3271.84
Arquitectura 3	3	16	32	3198.17
Arquitectura 4	3	8	64	3395.30
Arquitectura 4	3	4	64	3300.96
Arquitectura 4	3	2	64	3375.59
Arquitectura 4	5	4	64	3258.51
Arquitectura 4	8	4	64	3379.26
Arquitectura 4	3	4	128	3249.93

Table 5.12.: resultats per als models basats en el Transformer aplicats a la sèrie temporal dels palets sencers.

La taula 5.12 mostra els resultats d'aquests models aplicats a la sèrie de palets sencers. Tal com podem veure a aquesta taula, afegir els festius previs com fem a la segona arquitectura (visible a la figura 4.8) no aporta gaire informació i, de fet, en molts casos els models funcionen pitjor en afegir aquestes dades. Tanmateix, quan afegim la informació dels festius futurs mitjançant un producte com fem a l'arquitectura 3 (disponible a la figura 4.9), els models

semblen presentar una menuda millora. De fet, el millor dels models basats en el Transformer sense tindre en compte la freqüència el registrem en aquest grup. D'altra banda, quan afegim els valors de tots els festius en format de combinació lineal com en l'arquitectura 4 (recordem la figura 4.10), els models semblen comportar-se pitjor.

En qualsevol cas, els valors d'RMSE obtinguts en els models basats en el Transformer sense tindre en compte la freqüència s'allunyen molt ja no dels millors models estadístics com l'ARIMAX, sinó fins i tot dels models basats en xarxes recurrents, el que situa aquests experiments com als que obtenen pitjors resultats.

Tanmateix, a la taula 5.13 trobem els resultats dels models basats en el ViT, que ja empren la freqüència per elaborar les prediccions. En aquest cas podem observar una clara millora en l'eficàcia del model per realitzar les prediccions, reduint l'error en quasi mil caixes respecte als models que no compten amb la freqüència. A més, la taula mostra com incrementar els caps d'atenció així com la profunditat del Transformer, ajuda a reduir l'error en aquesta sèrie. D'altra banda, la millor domensió d'*embedding* sembla ser seixanta-quatre, ja que a l'incrementar-la o en disminuir-la els resultats empitjoren. Finalment i referint-nos al *patch embedding*, el format de la secció de pseudoimatge seleccionada hauria de ser de 5×5 si volem minimitzar l'error.

Profunditat	Caps d'atenció	Tamany d'*embedding*	Tamany del *patch embedding*	RMSE
3	4	64	5×5	3236.84
3	8	64	5×5	3369.17
3	16	64	5×5	3266.93
5	16	64	5×5	3342.38
8	**16**	**64**	**5×5**	**2393.34**
8	16	32	5×5	3368.94
8	16	128	5×5	3257.12
8	16	64	3×3	3293.56
8	16	64	10×10	3362.55

Table 5.13.: resultats per als models basats en el Transformer per Visió aplicats a la sèrie temporal de palets sencers.

Els models basats en el Transformer Convolucional que comentàvem a la secció 4.3 encara milloren més els resultats del ViT. A la taula 5.14 comprovem que el millor model del Transformer Convolucional és lleugerament superior al millor dels models basats en el ViT i que, a més, els models registrats a la taula presenten uns valors més baixos en general que els obtinguts pel ViT. Per tant, podem arribar a inferir que les convolucions aporten uns millors resultats per a aquesta sèrie en particular. Si observem la taula més detingudament, podrem observar que l'experimentació ens porta a pensar que el nombre de convolucions aplicades a l'hora d'aconseguir l'*embedding* hauria de ser com més alt millor, puix els dos millors models (amb molta diferència) són dos dels tres que tenen més capes convolucionals. D'altra banda, el nombre òptim de caps d'atenció hauria de ser 4, car quan els augmentem acabem amb valors d'RMSE molt alts. Finalment, incrementar la profunditat acaba per tenir un efecte negatiu, tot i que no s'assoleixen errors tan greus com quan alterem el nombre de caps d'atenció.

Profunditat	Nombre de caps d'atenció	Dimensió d'*embedding*	Canals de convolució	RMSE
3	**4**	**64**	**4, 8 i 64**	**2389.18**
3	4	64	8, 16 i 64	3286.62
3	**4**	**64**	**16, 32 i 64**	**2399.49**
3	4	64	2, 4 i 64	3152.46
3	4	64	32 i 64	3249.31
3	4	64	16 i 64	3108.62
3	4	64	8 i 64	3247.69
3	4	64	64	3224.11
3	8	64	4, 8 i 64	3116.03
3	2	64	4, 8 i 64	3136.69
5	4	64	4, 8 i 64	2547.60

Table 5.14.: resultats per als models basats en el Transformer Convolucional aplicats a la sèrie temporal de palets sencers.

Si apliquem la tècnica de *rolling window* sobre els tres millors models dels estudiats tal com hem fet a les seccions 5.1.2 i 5.1.3 podem obtindre una gràfica com la que s'observa a la figura 5.6. En aquesta imatge observem com el millor model basat en el ViT es comporta de manera molt estable, seguint un error del 30% aproximadament. Tanmateix, en la segona observació, sí que comprovem que el segon millor model segons l'RMSE del Transformer Convolucional funciona una mica millor. D'altra banda, si mirem el gràfic del MAE, podem observar com els dos models basats en el Transformer Convolucional funcionen més o menys similar, tot i que el millor d'aquests dos models en termes de RMSE té un error mitjà absolut inferior. D'altra banda, el ViT sembla comportar-se prou pitjor, amb errors molt més alts que en els altres casos exceptuant la tercera observació, on comprovem que l'error és una mica més baix.

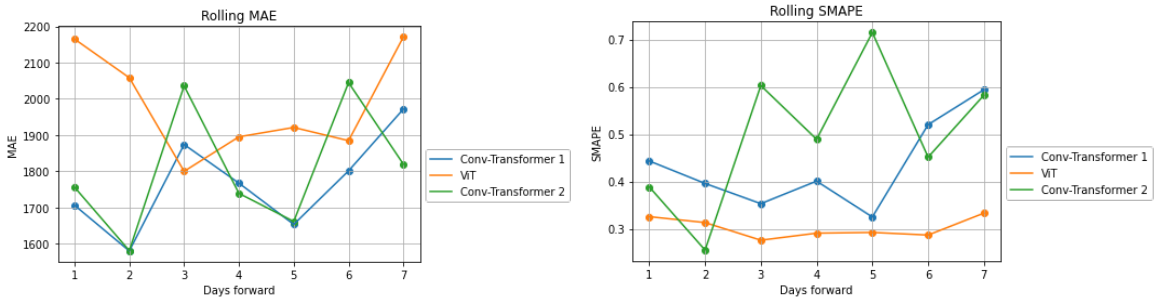


Figure 5.6.: avaluació d'errors per predicció en els models basats en el Transformer aplicats a la sèrie de palets sencers.

Per tant, a l'hora de desenvolupar el model *ensemble*, podem fer-ho de dues maneres: podem basar-nos en el millor model dels Transformers Convolucionals i emprar-lo exceptuant en la segona observació, en la que usarem el millor model ViT, o podem utilitzar el millor ViT en tot cas exceptuant la segona observació, en la que farem ús del segon millor model dels Transformers Convolucionals.

Endemés, si ens referim als models aplicats a les dades dels mitjos palets, la taula 5.15 mostra els resultats obtinguts per a l'experimentació amb els models que no tenen en compte la freqüència de la sèrie. Com podem comprovar, tal com passava en la sèrie de palets sencers, emprar la segona arquitectura porta a errors generalment més alts que utilitzar la primera. És a dir, introduir únicament els valors de festius previs confon el model i provoca que obtinga pitjors resultats. Tanmateix, si introduïm els festius dels dies que volem predir, com a la tercera o a la quarta arquitectura, podem arribar a millorar una mica els models de Transformers més tradicionals.

D'entre aquestes dues arquitectures, sembla que la tercera funciona força millor, obtenint uns resultats en RMSE que si bé no rivalitzen als dels models estadístics, sí que superen els valors dels millors models basats en les xarxes recurrents. Aquest millor model serà el que té més caps d'atenció i una dimensió d'*embedding* superior, però una profunditat d'únicament tres capes d'atenció.

Arquitectura	Profunditat	Caps d'atenció	Dimensió d'*embedding*	RMSE
Arquitectura 1	3	8	64	2489.68
Arquitectura 1	3	4	64	2557.51
Arquitectura 1	3	16	64	2519.90
Arquitectura 1	5	8	64	2513.48
Arquitectura 1	2	8	64	2436.76
Arquitectura 1	2	8	128	2517.80
Arquitectura 1	2	8	32	2535.23
Arquitectura 2	3	8	64	2522.50
Arquitectura 2	3	4	64	2643.50
Arquitectura 2	3	16	64	2473.34
Arquitectura 2	5	16	64	2390.82
Arquitectura 2	2	16	64	2375.94
Arquitectura 2	2	16	128	2516.23
Arquitectura 2	2	16	32	2517.43
Arquitectura 3	3	8	64	2411.34
Arquitectura 3	3	4	64	2551.84
Arquitectura 3	3	16	64	2299.91
Arquitectura 3	5	16	64	2390.63
Arquitectura 3	2	16	64	2491.50
Arquitectura 3	**3**	**16**	**128**	**2279.32**
Arquitectura 3	3	16	32	2499.19
Arquitectura 4	3	8	64	2511.04
Arquitectura 4	3	4	64	2362.49
Arquitectura 4	3	2	64	2538.46
Arquitectura 4	5	4	64	2461.13
Arquitectura 4	8	4	64	2516.13
Arquitectura 4	3	4	128	2490.09

Table 5.15.: resultats per als models basats en el Transformer aplicats a la sèrie temporal de mitjos palets.

Alternativament, podem considerar la freqüència i emprar algun altre tipus d'arquitectura. La taula 5.16 mostra els resultats obtinguts emprant l'arquitectura del ViT sobre les pseudoimatges obtingudes. Podem observar que sembla que mantenir un nombre reduït de caps d'atenció així com la profunditat, pot ser beneficiós perquè el model realitze adequadament les prediccions. També observem que els millors models són aquells que empren una dimensió d'*embedding* menor. Finalment, també comprovem que els models que empren un *patch* menut són els que millor funcionen. És a dir i com a resum, el millor model serà aquell més simple i que reculla més informació concentrada.

Profunditat	Caps d'atenció	Dimensió d'*embedding*	Grandària del patch	RMSE
3	4	64	5×5	2344.25
3	8	64	5×5	2500.09
3	16	64	5×5	2393.01
5	4	64	5×5	2460.94
8	4	64	5×5	2508.78
3	**4**	**32**	**5×5**	**1704.06**
3	4	128	5×5	2496.58
3	**4**	**32**	**3×3**	**1679.00**
3	4	32	10×10	2343.85

Table 5.16.: resultats per als models basats en el Transformer per Visió aplicats a la sèrie temporal de mitjos palets.

Finalment, a la graella 5.17 podem comprovar els resultats d'aplicar models basats en el Transformer Convolucional per resoldre el problema. En aquest cas, a diferència de com passava amb la sèrie palets sencers, sembla que els models basats en el Transformer Convolucional tenen pitjors resultats que no els Transformers basats en Visió. En qualsevol cas, aquests dos tipus de models tenen millors resultats que els Transformers que no tenen en compte la freqüència i que les xarxes recurrents, el que dona força a la nostra hipòtesi que aprofitar la freqüència podria dur a millorar els resultats.

Profunditat	Caps d'atenció	Dimensió d'*embedding*	Canals de convolució	RMSE
3	4	64	4, 8 i 64	2368.53
3	4	64	8, 16 i 64	2318.85
3	4	64	13, 32 i 64	2466.30
3	4	64	2, 4 i 64	1952.57
3	4	64	32 i 64	2364.84
3	4	64	16 i 64	2463.71
3	4	64	8 i 64	2537.23
3	4	64	64	2405.63
3	8	64	4, 8 i 64	2521.44
3	**2**	**64**	**2, 4 i 64**	**1810.27**
5	4	64	2, 4 i 64	2279.73

Table 5.17.: resultats per als models basats en el Transformer Convolucional aplicats a la sèrie temporal de mitjos palets.

A més a més, a la taula es comprova com altra vegada els millors models són aquells que tenen més capes, així com es comprova que augmentar la profunditat dels models comporta una pèrdua de precisió a les prediccions. D'altra banda, en aquest cas el millor model obtingut seria el que només empra dos caps d'atenció, de manera que podem concloure en el fet que el model que té l'atenció més senzilla i les convolucions més complexes és el que millor funciona.

Aquests experiments ens porten a pensar que si els millors models són aquells que empren mecanismes simples d'atenció i convolucions més complexes, potser podem estalviar-nos l'ús de l'atenció i emprar només xarxes convolucionals, tal com veurem a la següent secció.

Si apliquem la tècnica de la finestra corredissa (*rolling window*) sobre els tres millors models obtinguts, és a dir, els dos millors models basats en el Transformer per a Visió i el millor Transformer Convolucional, obtenim la figura 5.7.

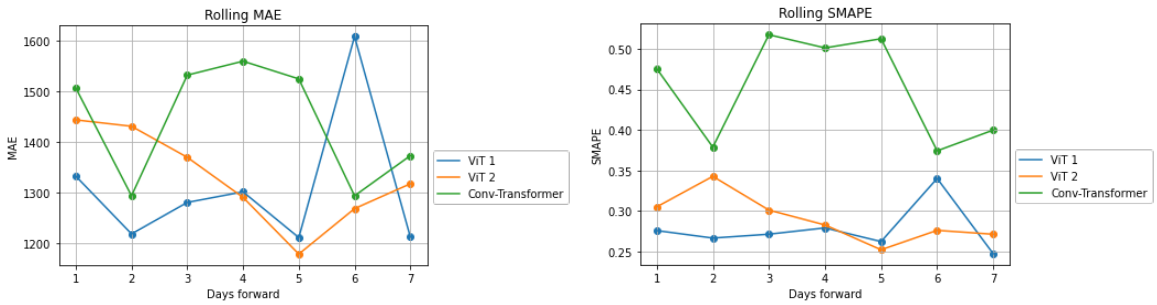


Figure 5.7.: avaluació d'errors per predicció en els models basats en el Transformer aplicats a la sèrie de mitjos palets.

Així, podem veure que per a les primeres tres observacions sense cap mena de dubte el model que realitza les prediccions més precises és el millor dels ViTs. Tanmateix, en la quarta predicció, el segon millor ViT assoleix unes cotes d'error similar, superant-lo en la cinquena i sisena predicció. Això no obstant, en el darrer pronòstic, el millor ViT torna a avançar-se com al model més precís. El Transformer Convolucional, d'altra banda, és el que pitjor funciona dels tres amb prou diferència.

Així doncs, si volem establir un model *ensemble* emprant aquests tres, usarem els dos millors ViTs. D'aquesta manera, el millor ViT s'encarregarà de predir els tres primers valors així com l'últim i el segon millor ViT s'encarregarà de predir els altres valors.

5.4. Models de xarxes convolucionals

Veient l'èxit que presentaven els models basats en el Transformer quan empraven la transformada contínua de wavelet, especialment el Transformer convolucional, vam decidir aplicar-hi sobre aquestes pseudoimatges un seguit de models convolucionals, tal com hem explicat a la secció 4.4.

La taula 5.18 mostra els resultats obtinguts en aquests experiments. Podem comprovar que el *MaxPool* sembla influir negativament sobre el model bàsic, empitjorant força els resultats obtinguts. A més a més, els festius tampoc no semblen aportar gaire informació en el cas de la sèrie de palets sencers, mentre que en el cas de la sèrie de mitjos palets sí que tenen un efecte notable.

Arquitectura	Festius	Profunditat	RMSE palets sencers	RMSE mitjos palets
CNN	-	7	2514.03	**1853.61**
CNN	-	3	**2327.33**	1998.57
CNN	-	4	**2272.24**	**1640.85**
CNN + maxpool	-	7	2904.96	2482.07
CNN + maxpool	-	3	2855.39	2015.12
CNN + maxpool	-	4	2700.60	1888.73
CNN + maxpool	Concatenats	3	2524.69	2025.65
CNN + maxpool	Concatenats	7	3235.18	2526.95
CNN + maxpool	Concatenats	4	2737.26	2045.44
CNN + maxpool	Sumats	3	2783.55	2109.76
CNN + maxpool	Sumats	7	3331.34	2468.23
CNN + maxpool	Sumats	4	2783.25	**1950.52**
ResNet-1	-	16	2768.23	2073.68
ResNet-2	-	16	2841.38	2348.68
ResNet-1	Sumats	16	2722.04	2082.84
ResNet-2	Sumats	16	2933.11	2223.54
ResNet-1	Multiplicats	16	$5.25 \cdot 10^{10}$	$1.18 \cdot 10^{11}$
ResNet-2	Multiplicats	16	**2546.94**	2104.17
DenseNet-1	-	12	3485.62	2504.84
DenseNet-2	-	12	3501.83	2709.10
DenseNet-1	Sumats	12	3301.14	2390.57
DenseNet-2	Sumats	12	3325.09	3135.94
DesneNet-1	Multiplicats	12	6359.37	4378.56
DenseNet-2	Multiplicats	12	3465.26	2818.70

Table 5.18.: resultats dels experiments amb models convolucionals.

D'altra banda, els models més complexos com la ResNet o la Densenet semblen funcionar una mica pitjor que la convolució simple, especialment en la segona arquitectura. La ResNet sembla funcionar una mica millor i la segona arquitectura plantejada és fins i tot capaç d'incorporar prou bé els valors de festius en el cas de la sèrie dels palets sencers. També es cert que els resultats obtinguts per a les dues sèries emprant la ResNet-1 amb els festius multiplicant, té els resultats més dolents de tota la sèrie d'experiments que hem desenvolupat, molt probablement perquè la formació de l'*embedding* no és prou complexa perquè puga aprendre l'efecte que aquesta variable té sobre la sèrie.

Així doncs, podem afirmar que els models més senzills tendeixen a funcionar millor en aquest problema que no els models complexos.

En aquest cas també podem aplicar la tècnica de *rolling window* per a observar quines serien les millors prediccions en cada instant de temps. En el cas de la sèrie de palets sencers, hem emprat les tres xarxes convolucionals més senzilles, puix eren com hem vist a la taula 5.18 les que donaven millors resultats. Així doncs, la figura 5.8 ens mostra com tant per al MAE com per a l'SMAPE la millor xarxa convolucional té valors baixos i estables mentre que la segona millor CNN té valors molt més inestables. És gràcies a aquesta inestabilitat

que aconsegueix superar el millor CNN en alguns casos, com en la segona, en la quarta o en l'última observació, on el millor CNN comet més errors. El tercer millor CNN, d'altra banda, és el que pitjor funciona, tenint quasi sempre valors de MAE més alts i sempre valors de SMAPE superiors als dels altres models.

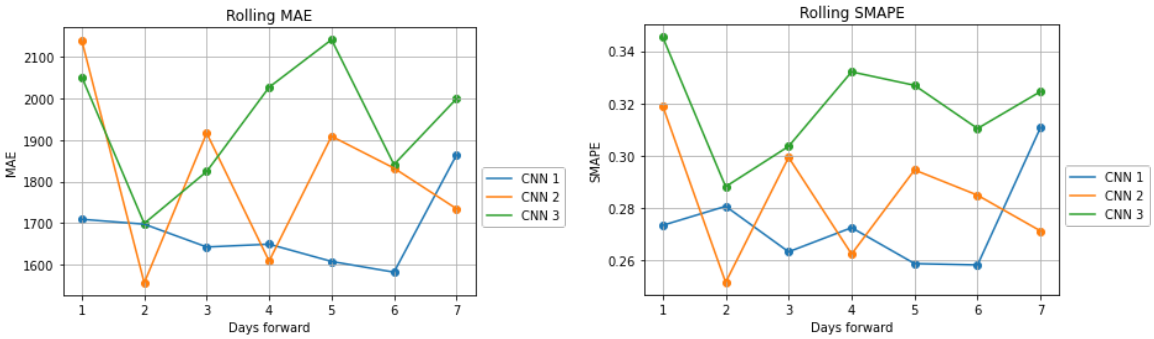


Figure 5.8.: avaluació d'errors per predicció en els models convolucionals aplicats a la sèrie de palets sencers.

Així, en cas de voler realitzar un *ensemble* emprant aquests models, caldria considerar totes les observacions de la millor CNN, exceptuant la segona, la quarta i l'última, prediccions que recaurien sobre la segona millor CNN.

Quant a la sèrie de mitjos palets, la figura 5.9 mostra els resultats dels tres millors models: dues CNN bàsiques i una CNN amb *MaxPool* i l'addició dels festius. D'aquesta manera podem comprovar que totes les prediccions durant el primer dia cometen més o menys el mateix error (la segona millor CNN comet una mica més) però des d'aquest punt endavant la millor CNN mostra la seua dominància amb valors més baixos i molt estables, mentre que la segona millor CNN assoleix un pic en errors amb la tercera predicció i després tendeix a baixar fins a la darrera predicció, on l'error torna a enlairar-se en MAE i s'estabilitza en SMAPE. La xarxa convolucional amb *MaxPool* i festius afegits és molt més inestable i si bé en alguns casos comet menys error que la segona CNN, en general té un MAE i un SMAPE més elevats.

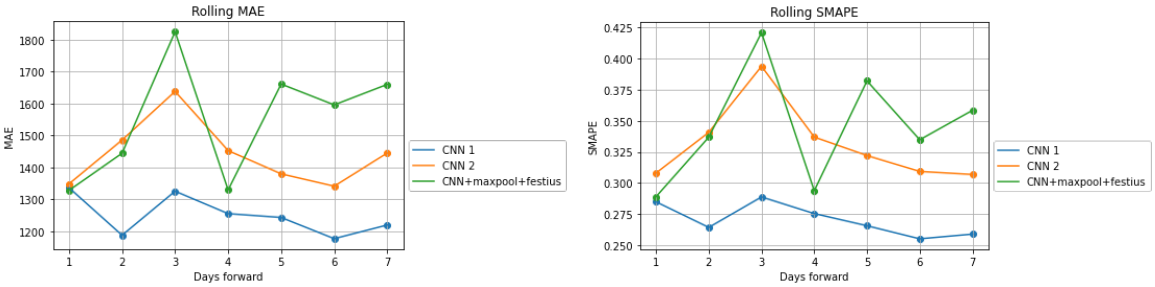


Figure 5.9.: avaluació d'errors per predicció en els models convolucionals aplicats a la sèrie de mitjos palets.

Per tant, aplicar un *ensemble* de models no té gaire sentit en aquest cas, ja que la primera CNN supera o iguala en tots els casos als altres models.

5.5. Models preentrenats

Tal com hem explicat anteriorment, un dels inconvenients més importants d'aquest problema era la carència de dades. Per aquesta raó vam utilitzar models preentrenats, que esperàvem que ens ajudaren a superar aquest obstacle. En total vam usar sis models: dos Transformers per a sèries temporals com són l'InFormer i l'AutoFormer; dos Transformers per a imatge, com són el ViT i el SWIN; i dues xarxes convolucionals com són la ResNet-50 i la DenseNet-201.

En el cas dels Transformers que s'apliquen directament sobre el model, ambdós estan preentrenats amb un *benchmark* reconegut com és l'Arxiu de Predicció de Sèries Temporals de Monash [103], concretament el conjunt de dades corresponent a quantitat de turistes mensuals. D'altra banda, els quatre models visuals estan entrenats amb el conjunt de dades d'ImageNet. Concretament, el Google ViT està entrenat amb les catorze milions d'imatges d'ImageNet-21k i reentrenat amb el milió d'imatges d'ImageNet 2012, mentre que el Microsoft SWIN, la ResNet i la DenseNet estan entrenats amb ImageNet 1k.

Arquitectura	RMSE palets sencers	RMSE mitjos palets
InFormer	2137.41	1607.45
AutoFormer	2730.08	2582.15
Google ViT	3857.07	3864.33
Microsoft SWIN	6539.62	7030.61
ResNet-50	2565.67	2378.78
DenseNet-201	2450.36	2384.38

Table 5.19.: resultats per als experiments amb models preentrenats.

A la taula 5.19 observem en ambdós casos els models InFormer i AutoFormer preentrenats funcionen significativament millor que els altres models que hem aplicat. De fet, el InFormer assoleix unes cotes de RMSE que en el cas dels mitjos palets són competitives amb els models estadístics univariants mentre que en el cas de palets sencers supera sense cap dubte tots els altres models de xarxes.

Els Transformers per a Visió no són tant prometedors. Podem observar que els resultats en el cas del Google ViT preentrenat són força pitjors als que havíem aconseguit entrenant des de zero, i açò encara empitjora més en el cas del SWIN. Aquesta diferència entre ambdós models segurament siga causada per la quantitat de dades amb les quals s'hi ha preentrenat els models. Com hem explicat, el Google ViT està entrenat amb moltes més imatges que el SWIN.

D'altra banda, el fet que aquests Transformers tinguen uns resultats tan dolents pot ser degut a diversos factors. En primer lloc, com hem comprovat a la secció 5.3, semblava que la complexitat del model incrementava la quantitat d'errors i el model preentrenat compta amb una gran quantitat de paràmetres. D'altra banda, també podria ser perquè ambdós models estan entrenats amb imatges que no tenen a veure amb les freqüències, però com explicarem a continuació açò és poc probable.

I és que les xarxes convolucionals, tot i estar entrenades amb les mateixes imatges, resulten en errors molt inferiors al cas dels Transformers per visió. Si comparem els resultats de la ResNet-50 amb els resultats de les ResNet entrenades des de zero, trobem que tot i no incloure dades de festius, obté resultats per a la sèrie de palets sencers molt similars a aquells que

si els inclouen, fet que ens porta a pensar que potser si incloguérem els festius obtindríem millors resultats. No és el cas quan parlem de la sèrie de mitjos palets, ja que els resultats obtinguts per la xarxa preentrenada són pitjors als resultats dels models entrenats des de zero.

En el cas de les DenseNets, utilitzar models preentrenats resulta en una millora dels resultats més que evidents. Altra vegada, tot i que no incloíem les dades de festius, els models preentrenats obtenien millors resultats que els models entrenats des de zero i aquesta vegada en ambdues sèries.

5.6. Models Light-GBM

Seguidament, explicarem una mica els resultats obtinguts quan emprem els mètodes Light-GBM. Aquesta tècnica, com ja hem explicat, està basada en els mètodes d'Arbres*Gradient Boosting*, però en lloc d'aplicar una aproximació de cerca en amplitud, realitza una cerca en profunditat de manera que obté pitjors resultats però molt més ràpidament.

Nosaltres proposem quatre models per a cada sèrie. El primer model serà aquell que utilitze la sèrie directament, sense aplicar-hi cap tècnica d'*embedding*. Esperàvem que aquesta fora la pitjor aproximació de les quatre emprades.

Les altres tres aproximacions consisteixen a utilitzar la penúltima capa dels tres millors models de xarxes obtinguts per a cada sèrie. Esperàvem que aquesta capa continguera tota la informació necessària perquè la darrera capa lineal fora capaç d'aplicar una adequada regressió. Així doncs, canviar aquesta capa lineal per un Light-GBM semblava adequat i és una pràctica que s'ha usat en algunes competicions com l'M4 [21] o l'M5 [5].

Embedding	RMSE
Serie temporal	**2244.83**
InFormer	2456.08
CNN	2412.67
Conv-Transformer	2313.22

Table 5.20.: resultats per als models Light-GBM aplicats a la sèrie temporal de palets sencers.

La taula 5.20 ens mostra els resultats d'aplicar aquest tipus de model a la sèrie temporal de palets sencers. Com podem veure, sembla que el millor Light-GBM és aquell que pren directament la sèrie temporal. Podríem pensar que açò es deu al fet que els models d'*embedding* no són capaços d'aconseguir una bona representació de la sèrie, però no és el cas, ja que com veiem per a l'InFormer, el model original amb una capa lineal com a regressora, funciona prou millor que no aquest Light-GBM. El mateix passa amb la xarxa convolucional i l'única excepció dels tres models és el Transformer convolucional, que sembla funcionar una mica millor amb un Light-GBM com a regressor. En el nostre cas, només utilitzarem el Light-GBM aplicat directament sobre la sèrie temporal per comparar-lo amb els altres models.

Els resultats aplicats a la sèrie de mitjos palets, visibles a la taula 5.21 són molt semblants. En aquest cas, el millor model és el que pren l'*embedding* directament del millor model convolucional, però no aconsegueix millorar-lo, sinó que el seu RMSE és prou major. Així mateix, els Light-GBM aplicats a partir dels *embeddings* de l'InFormer i el ViT tampoc no milloren els seus resultats respectius.

Embedding	RMSE
Serie temporal	1825.15
InFormer	1906.84
CNN	**1800.60**
ViT	1839.56

Table 5.21.: resultats per als models Light-GBM aplicats a la sèrie temporal de palets sencers.

Així doncs, després d'aquesta breu experimentació, podem afirmar que l'aplicació de Light-GBM sobre el nostre problema ha resultat en un empitjorament dels resultats originals, per la qual cosa no els tindrem en compte durant el següent apartat.

5.7. Comparació de models

Una vegada hem explorat tots els models presentats fins ara, arriba el moment de comparar-los entre ells. Per a això, en el cas de l'ARIMA, l'ARIMAX i el Transformer emprarem els *ensembles* definits a les seues respectives seccions, mentre que per al cas dels models Holt & Winters, Elman, GRU, LSTM i InFormer utilitzarem els millors models directament.

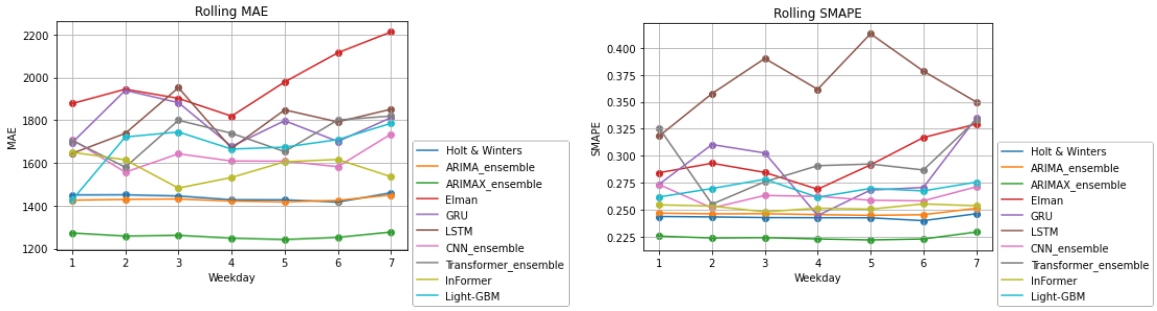


Figure 5.10.: avaluació d'errors en els models a la sèrie de palets sencers.

Com podem veure en la figura 5.10, els models estadístics són els models que funcionen millor, a pesar que l'InFormer comptara amb un RMSE menor. Açò és molt probablement causat perquè l'InFormer no comet errors molt grans (que penalitza l'RMSE), però sí que comet molts errors al llarg de totes les observacions. D'altra banda, cal que destaquem l'ARIMAX entre aquests models estadístics, ja que aconsegueix els menors errors amb molta diferència. Açò reafirma la nostra hipòtesi inicial que la quantitat de caixes rebudes per Mercacdona i els festius són clars indicadors per a establir una adequada predicció.

D'altra banda, dins dels models basats en xarxes, trobem que l'InFormer és un model competitiu si el comparem amb els models estadístics univariants en SMAPE. L'*ensemble* de CNN també és competitiu en algunes prediccions (sobretot la segona) en SMAPE, tot i que sí que és cert que té errors una mica majors, encara que estables. El cas de les xarxes GRU és semblant al de les xarxes convolucionals, tot i que la seua estabilitat és molt menor, cosa que implica que es cometen errors més greus durant les prediccions.

L'*ensemble* de Transformers, d'altra banda, té errors que ja no s'acosten tant als models estadístics univariants, tot i que en la segona predicció l'SMAPE s'acosta força als resultats

de l'*ensemble* de convolucionals. Finalment, les xarxes Elman i LSTM així com el model Light-GBM, tenen errors molt més grans i no són gaire competitius en SMAPE.

En qualsevol cas, també hem de remarcar que aquests models basats en xarxes, des de l'InFormer fins a les LSTM, així com el Light-GBM, es troben molt lluny dels models estadístics en MAE.

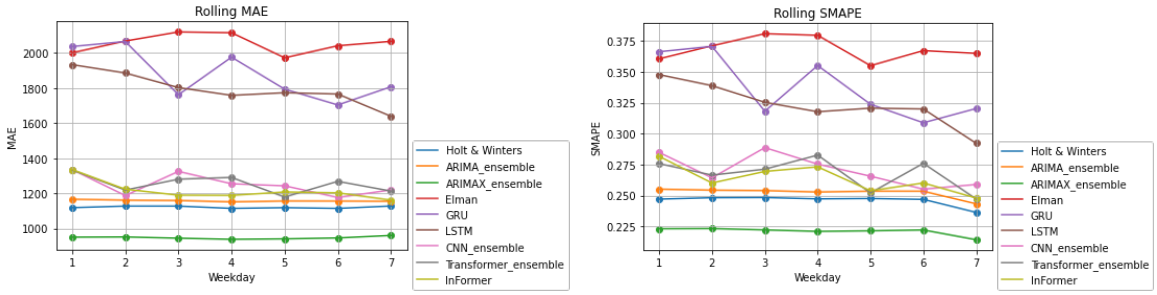


Figure 5.11.: avaluació d'errors en els models a la sèrie de mitjos palets.

Si observem els models aplicats a la sèrie de mitjos palets, el primer que podem veure a la figura 5.11 és que els models basats en xarxes recurrents (tant l'Elman com la GRU i la LSTM) són, clarament els que pitjor funcionen tant en MAE com en SMAPE, amb un error en aquest darrer d'aproximadament un 10% més que els altres models.

D'altra banda, en aquest cas els models estadístics són els que millor funcionen, sent el seu màxim exponent el model ARIMAX, que inclou dades exògenes. Així doncs, podem comprovar que incloure les variables de quantitat de caixes rebuda per Mercadona i de festius millora els resultats dels models.

Finalment, els models tant InFormer com l'*ensemble* de Transformers visuals com l'InFormer funcionen més o menys igual, destacant així els beneficis d'emprar la freqüència en realitzar les prediccions.

6. Conclusions

En aquest treball hem desenvolupat una sèrie d'experiments que provaven de resoldre un problema de regressió com és la predicció de sèries temporals.

Com hem vist, l'assumpte més delicat del nostre problema era l'escassa quantitat de dades que teníem. És per aquesta dificultat que els models estadístics semblen funcionar millor en general, tant per a la sèrie estacionària (mitjos palets) com per a la sèrie no estacionària.

A més, durant l'experimentació, tant en models estadístics com en models basats en xarxes hem observat que aquells que obtenen millors resultats són els que inclouen més informació, és a dir, els models que contenen variables exògenes. Per tant, és lògic suposar que les variables de quantitat rebuda per Mercadona i de festius són bons indicadors per a realitzar una predicció sobre les caixes netejades per Logifruit.

Quan parlem dels models basats en xarxes, la nostra experimentació ens porta a assumir que els models més senzills són aquells que funcionen millor. És a dir, els models que tenen menys paràmetres són els que ofereixen millors resultats. Segurament açò és provocat altra vegada per la carència de dades, ja que un major nombre de paràmetres a optimitzar requereix un entrenament més exhaustiu amb dades que no tenim.

D'altra banda, en aplicar les xarxes recurrents els resultats de les tres (Elman, LSTM i GRU) eren prou similars en ambdues sèries, sent més eficaces les GRU en el cas de la sèrie de palets sencers i les LSTM en el cas de la sèrie de mitjos palets. Tanmateix, en ambdós casos les RNN són els models que ofereixen uns pitjors resultats. Cal recalcar que tot i que les magnituds dels valors d'ambdues sèries són diferents, els errors no disten gaire, el que ens porta a pensar que aquestes xarxes funcionen una mica millor quan es tracta de modelar sèries no estacionàries.

L'experimentació en models Transformer ens ha oferit uns resultats prou d'acord amb les nostres primeres suposicions. Quan els aplicàvem sobre les sèries temporals directament, els models *from scratch* oferien uns resultats prou dolents, ja que la quantitat de dades era massa baixa perquè els models aprengueren els patrons intrínsecs de les sèries. Els models preentrenats, però, oferien uns resultats prou millors rivalitzant en alguns casos amb els models estadístics univariants, però sense arribar a superar els models ARIMAX.

A més a més, l'experimentació utilitzant la freqüència ens ha oferit uns resultats molt prometedors, tant en els Transformers per a Visió i els Transformers Convolucionals com per a les xarxes convolucionals, sent els models ANN *from scratch* que millor resultats obtenen. Així doncs, aquest projecte sembla remarcar els beneficis d'aplicar aquest tipus de models sobre sèries temporals amb poques dades.

Finalment, hom pot recalcar que els models Light-GBM aplicats sobre *embeddings* han obtingut resultats prou dolents, segurament a causa dels models d'*embedding*, que estaven basats en els models de xarxes, no obtenien la millor caracterització de la sèrie. D'altra banda, els Light-GBM aplicats directament sobre la sèrie obtenien uns millors resultats, especialment en el cas de la sèrie de palets sencers, on tenien uns errors similars als millors models ANN.

Tanmateix, encara quedaven molt lluny dels models estadístics.

Treball a futur

Aquesta investigació podria continuar de diverses maneres. En primer lloc, com que la sèrie temporal està autocorrelada, té sentit pensar que les pseudoimatges obtingudes mitjançant la transformada contínua de wavelet estaran relacionades entre elles. Per tant, no és forassenyat pensar que l'ús d'una xarxa Conv-LSTM, com la que es defineix a [33], puga donar uns resultats prou millors.

Una altra de les línies d'investigació que es podria seguir seria l'aplicació d'uns *embeddings* més adequats per a les sèries temporals. L'aplicació d'aquests *embeddings* no només ajudaria a millorar els resultats de tots els models ANN, sinó que també ens podria donar lloc a una reducció d'errors en els models Light-GBM.

També es podria buscar aplicar altres models que hem vist a la literatura, com el TriFormer o el QuatriFormer i, en cas de tindre una capacitat computacional major i més temps, es podria arribar a preentrenar tots els models amb els quals hem experimentat amb dades, per exemple, de l'Arxiu de Predicció de Sèries Temporals de Monash [103] per tal d'obtindre uns millors resultats.

Per acabar, recentment s'ha desenvolupat un tipus de model basat en l'algorisme Kolgomorov-Arnold [104], que podria en algun moment arribar a substituir l'algorisme perceptró com a base de les ANN. Es podria experimentar amb aquest algorisme per tal de comprovar quina seria la seua utilitat en la predicció de sèries temporals.

Bibliography

[1] J. Liu, X. Kong, F. Xia, X. Bai, L. Wang, Q. Qing, and I. Lee, "Artificial intelligence in the 21st century," *IEEE Access*, vol. 6, pp. 34403–34421, 2018.

[2] OpenAI, "Introducing chatgpt." `https://openai.com/blog/chatgpt`. Accessed: 2024-04-25.

[3] Logifruit, "Qué es Logifruit." `https://logifruit.es/quienes-somos/`. Accessed: 2024-04-25.

[4] J. C. García-Díaz, *Predicción en el dominio del tiempo. Análisis de series temporales para ingenieros*, ch. 1. Análisis descriptivo de series temporales, pp. 1–34. Universitat Politècnica de València, 2016.

[5] S. Makridakis, E. Spiliotis, and V. Assimakopoulos, "M5 accuracy competition: Results, findings, and conclusions," *International Journal of Forecasting*, vol. 38, no. 4, pp. 1346–1364, 2022. Special Issue: M5 competition.

[6] J. C. García-Díaz, *Predicción en el dominio del tiempo. Análisis de series temporales para ingenieros*, ch. 7. Predicción de series temporales, pp. 169–186. Universitat Politècnica de València, 2016.

[7] D. N. Gujarati and D. C. Porter, *Econometría*, ch. 22. Econometría de series de tiempo: pronósticos, pp. 773–800. Mc Graw Hill, 2010.

[8] J. C. García-Díaz, *Predicción en el dominio del tiempo. Análisis de series temporales para ingenieros*, ch. 2. Técnicas de suavizado de series temporales, pp. 169–186. Universitat Politècnica de València, 2016.

[9] J. C. García-Díaz, *Series temporales, análisis, predicción. Ejercicios prácticos*, ch. 2. Suavizado, pp. 27–35. Universitat Politècnica de València, 2011.

[10] J. C. García-Díaz, *Predicción en el dominio del tiempo. Análisis de series temporales para ingenieros*, ch. 3. Introducción a los modelos ARIMA, pp. 61–98. Universitat Politècnica de València, 2016.

[11] G. Zhang, E. Patuwo, B., and M. Y. Hu, "Forecasting with artificial neural networks: The state of the art," *International Journal of Forecasting*, vol. 14, pp. 35–62, March 1998.

[12] B. Irie and S. Miyake, "Capabilities of three-layered perceptrons," *IEEE 1988 International Conference on Neural Networks*, pp. 641–648 vol.1, 1988.

[13] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural Networks*, vol. 2, no. 5, pp. 359–366, 1989.

[14] C. W. J. Granger and T. Teräsvirta, *Modelling Nonlinear Economic Relationships*. Oxford University Press, 10 1993.

[15] S. Makridakis and M. Hibon, "The m3-competition: results, conclusions and implications," *International Journal of Forecasting*, vol. 16, pp. 451–476, 2000.

[16] J. Hyndman, Rob, "Rob j hyndman question answering time." `https://robjhyndman.com/hyndsight/qa-time/`. Accessed: 2024-04-23.

[17] H. Hewamalage, C. Bergmeir, and K. Bandara, "Recurrent neural networks for time series forecasting: Current status and future directions," *International Journal of Forecasting*, vol. 37, pp. 388–427, January–March 2021.

[18] W. Yan, "Toward automatic time-series forecasting using neural networks," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 23, no. 7, pp. 1028–1039, 2012.

[19] M. J. C. Hu and H. E. Root, "An adaptive data processing system for weather forecasting," *Journal of Applied Meteorology (1962-1982)*, vol. 3, no. 5, pp. 513–523, 1964.

[20] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, pp. 533–536, 1986.

[21] S. Makridakis, E. Spiliotis, and V. Assimakopoulos, "The m4 competition: 100,000 time series and 61 forecasting methods," *International Journal of Forecasting*, vol. 36, no. 1, pp. 54–74, 2020. M4 Competition.

[22] S. Smyl, "A hybrid method of exponential smoothing and recurrent neural networks for time series forecasting," *International Journal of Forecasting*, vol. 36, no. 1, pp. 75–85, 2020. M4 Competition.

[23] R. J. Williams and D. Zipser, "A learning algorithm for continually running fully recurrent neural networks," *Neural Comput.*, vol. 1, p. 270–280, jun 1989.

[24] J. L. Elman, "Finding structure in time," *Cognitive Science*, vol. 14, no. 2, pp. 179–211, 1990.

[25] M. I. Jordan, "Attractor dynamics and parallelism in a connectionist sequential machine," in *Artificial neural networks: concept learning*, 1990.

[26] Z. Motazedian and A. A. Safavi, "Nonlinear and time varying system identification using a novel adaptive fully connected recurrent wavelet network," *2019 27th Iranian Conference on Electrical Engineering (ICEE)*, pp. 1181–1187, 2019.

[27] S. Chang, Y. Zhang, W. Han, M. Yu, X. Guo, W. Tan, X. Cui, M. Witbrock, M. Hasegawa-Johnson, and T. S. Huang, "Dilated recurrent neural networks," 2017.

[28] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," 2014.

[29] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[30] K. Cho, B. van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using rnn encoder-decoder for statistical machine translation," 2014.

[31] F. M. Bianchi, E. Maiorino, M. C. Kampffmeyer, A. Rizzi, and R. Jenssen, *Recurrent Neural Networks for Short-Term Load Forecasting: An Overview and Comparative Analysis*. Springer International Publishing, 2017.

[32] T. Lin, B. Horne, P. Tino, and C. Giles, "Learning long-term dependencies in narx recurrent neural networks," *IEEE Transactions on Neural Networks*, vol. 7, no. 6, pp. 1329–1338, 1996.

[33] X. Shi, Z. Chen, H. Wang, D.-Y. Yeung, W. kin Wong, and W. chun Woo, "Convolutional lstm network: A machine learning approach for precipitation nowcasting," 2015.

[34] K. Bandara, C. Bergmeir, and S. Smyl, "Forecasting across time series databases using recurrent neural networks on groups of similar series: A clustering approach," *Expert Systems with Applications*, vol. 140, p. 112896, 2020.

[35] R. J. Hyndman, E. Wang, and N. Laptev, "Large-scale unusual time series detection," in *2015 IEEE International Conference on Data Mining Workshop (ICDMW)*, pp. 1616–1619, 2015.

[36] X. Wang, K. Smith, and R. Hyndman, "Characteristic-based clustering for time series data," *Data Mining and Knowledge Discovery*, vol. 13, pp. 335–364, May 2006.

[37] N. Kalchbrenner, I. Danihelka, and A. Graves, "Grid long short-term memory," 2016.

[38] B. Lindemann, T. Müller, H. Vietz, N. Jazdi, and M. Weyrich, "A survey on long short-term memory networks for time series prediction," *Procedia CIRP*, vol. 99, pp. 650–655, 2021. 14th CIRP Conference on Intelligent Computation in Manufacturing Engineering, 15-17 July 2020.

[39] A. Gensler, J. Henze, B. Sick, and N. Raabe, "Deep learning for solar power forecasting — an approach using autoencoder and lstm neural networks," in *2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pp. 002858–002865, 2016.

[40] R. Villegas, J. Yang, Y. Zou, S. Sohn, X. Lin, and H. Lee, "Learning to generate long-term future via hierarchical prediction," 2018.

[41] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," 2015.

[42] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," 2014.

[43] C. Peng, Y. Li, Y. Yu, Y. Zhou, and S. Du, "Multi-step-ahead host load prediction with gru based encoder-decoder in cloud computing," in *2018 10th International Conference on Knowledge and Smart Technology (KST)*, pp. 186–191, 2018.

[44] D. L. Marino, K. Amarasinghe, and M. Manic, "Building energy load forecasting using deep neural networks," in *IECON 2016 - 42nd Annual Conference of the IEEE Industrial Electronics Society*, IEEE, Oct. 2016.

[45] D. Salinas, V. Flunkert, and J. Gasthaus, "Deepar: Probabilistic forecasting with autoregressive recurrent networks," 2019.

[46] H. Xue, D. Q. Huynh, and M. Reynolds, "Ss-lstm: A hierarchical lstm model for pedestrian trajectory prediction," in *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 1186–1194, 2018.

[47] D. Hsu, "Multi-period time series modeling with sparsity via bayesian variational inference," 2018.

[48] N. P. Laptev, J. Yosinski, L. E. Li, and S. Smyl, "Time-series extreme event forecasting with neural networks at uber," in *Computer Science, Engineering*, 2017.

[49] A. Suilin, "Kaggle-web-traffic." `https://github.com/Arturus/kaggle-web-traffic`. Accessed: 2024-04-19.

[50] Google, "Web traffic time series forecasting." `https://www.kaggle.com/c/web-traffic-time-series-forecasting/`, 2017. Accessed: 2024-04-23.

[51] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," 2016.

[52] Y. G. Cinar, H. Mirisaee, P. Goswami, E. Gaussier, A. Ait-Bachir, and V. Strijov, "Position-based content attention for time series forecasting with sequence-to-sequence rnns," 2017.

[53] A. Vaswani, N. M. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Neural Information Processing Systems*, 2017.

[54] P. Karmakar, S. W. Teng, and G. Lu, "Thank you for attention: A survey on attention-based artificial neural networks for automatic speech recognition," *ArXiv*, vol. abs/2102.07259, 2021.

[55] S. Khan, M. Naseer, M. Hayat, S. W. Zamir, F. S. Khan, and M. Shah, "Transformers in vision: A survey," *ACM Computing Surveys*, vol. 54, p. 1–41, Jan. 2022.

[56] Q. Wen, T. Zhou, C. Zhang, W. Chen, Z. Ma, J. Yan, and L. Sun, "Transformers in time series: A survey," in *International Joint Conference on Artificial Intelligence*, 2022.

[57] C. Yun, S. Bhojanapalli, A. S. Rawat, S. J. Reddi, and S. Kumar, "Are transformers universal approximators of sequence-to-sequence functions?," 2020.

[58] B. Lim, S. . Arık, N. Loeff, and T. Pfister, "Temporal fusion transformers for interpretable multi-horizon time series forecasting," *International Journal of Forecasting*, vol. 37, no. 4, pp. 1748–1764, 2021.

[59] Y. Nie, N. H. Nguyen, P. Sinthong, and J. Kalagnanam, "A time series is worth 64 words: Long-term forecasting with transformers," *ArXiv*, vol. abs/2211.14730, 2022.

[60] Y. Zhang and J. Yan, "Crossformer: Transformer utilizing cross-dimension dependency for multivariate time series forecasting," in *International Conference on Learning Representations*, 2023.

[61] G. Zerveas, S. Jayaraman, D. Patel, A. Bhamidipaty, and C. Eickhoff, "A transformer-based framework for multivariate time series representation learning," 2020.

[62] A. Radford, J. W. Kim, T. Xu, G. Brockman, C. Mcleavey, and I. Sutskever, "Robust speech recognition via large-scale weak supervision," in *Proceedings of the 40th International Conference on Machine Learning* (A. Krause, E. Brunskill, K. Cho, B. Engelhardt, S. Sabato, and J. Scarlett, eds.), vol. 202 of *Proceedings of Machine Learning Research*, pp. 28492–28518, PMLR, 23–29 Jul 2023.

[63] H. Zhou, S. Zhang, J. Peng, S. Zhang, J. Li, H. Xiong, and W. Zhang, "Informer: Beyond efficient transformer for long sequence time-series forecasting," in *AAAI Conference on Artificial Intelligence*, 2020.

[64] H. Wu, J. Xu, J. Wang, and M. Long, "Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting," in *Advances in Neural Information Processing Systems* (M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, eds.), vol. 34, pp. 22419–22430, Curran Associates, Inc., 2021.

[65] W. Chen, W. wu Wang, B. Peng, Q. Wen, T. Zhou, and L. Sun, "Learning to rotate: Quaternion transformer for complicated periodical time series forecasting," *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2022.

[66] Y. Liu, H. Wu, J. Wang, and M. Long, "Non-stationary transformers: Exploring the stationarity in time series forecasting," in *Neural Information Processing Systems*, 2022.

[67] T. Zhou, Z. Ma, Q. Wen, X. Wang, L. Sun, and R. Jin, "FEDformer: Frequency enhanced decomposed transformer for long-term series forecasting," in *Proceedings of the 39th International Conference on Machine Learning* (K. Chaudhuri, S. Jegelka, L. Song, C. Szepesvari, G. Niu, and S. Sabato, eds.), vol. 162 of *Proceedings of Machine Learning Research*, pp. 27268–27286, PMLR, 17–23 Jul 2022.

[68] R.-G. Cirstea, C. Guo, B. Yang, T. Kieu, X. Dong, and S. Pan, "Triformer: Triangular, variable-specific attentions for long sequence multivariate time series forecasting-full version," *ArXiv*, vol. abs/2204.13767, 2022.

[69] S. Liu, H. Yu, C. Liao, J. Li, W. Lin, A. X. Liu, and S. Dustdar, "Pyraformer: Low-complexity pyramidal attention for long-range time series modeling and forecasting," in *International Conference on Learning Representations*, 2022.

[70] S. LI, X. Jin, Y. Xuan, X. Zhou, W. Chen, Y.-X. Wang, and X. Yan, "Enhancing the locality and breaking the memory bottleneck of transformer on time series forecasting," *ArXiv*, vol. abs/1907.00235, 2019.

[71] S. Wu, X. Xiao, Q. Ding, P. Zhao, Y. Wei, and J. Huang, "Adversarial sparse transformer for time series forecasting," in *Neural Information Processing Systems*, 2020.

[72] B. Peters, V. Niculae, and A. F. T. Martins, "Sparse sequence-to-sequence models," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics* (A. Korhonen, D. Traum, and L. Màrquez, eds.), (Florence, Italy), pp. 1504–1519, Association for Computational Linguistics, July 2019.

[73] Y. Lin, I. Koprinska, and M. Rana, "Ssdnet: State space decomposition neural network for time series forecasting," *2021 IEEE International Conference on Data Mining (ICDM)*, pp. 370–378, 2021.

[74] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu, "Lightgbm: a highly efficient gradient boosting decision tree," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, (Red Hook, NY, USA), p. 3149–3157, Curran Associates Inc., 2017.

[75] P. Montero-Manso, G. Athanasopoulos, R. Hyndman, and T. Talagala, "Fforma: Feature-based forecast model averaging," *International Journal of Forecasting*, vol. 36, pp. 86–92, Jan. 2020.

[76] B. Tang and D. S. Matteson, "Probabilistic transformer for time series analysis," in *Advances in Neural Information Processing Systems* (M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, eds.), vol. 34, pp. 23592–23608, Curran Associates, Inc., 2021.

[77] D. Dickey and W. Fuller, "Distribution of the estimators for autoregressive time series with a unit root," *JASA. Journal of the American Statistical Association*, vol. 74, 06 1979.

[78] W. Kirch, ed., *Pearson's Correlation Coefficient*, pp. 1090–1091. Dordrecht: Springer Netherlands, 2008.

[79] L. St»hle and S. Wold, "Analysis of variance (anova)," *Chemometrics and Intelligent Laboratory Systems*, vol. 6, no. 4, pp. 259–272, 1989.

[80] S. S. SHAPIRO and M. B. WILK, "An analysis of variance test for normality (complete samples)†," *Biometrika*, vol. 52, pp. 591–611, 12 1965.

[81] H. Levene, "Robust tests for equality of variances," in *Journal of the American Statistical Association*, 1961.

[82] W. H. Kruskal and W. A. Wallis, "Use of ranks in one-criterion variance analysis," *Journal of the American Statistical Association*, vol. 47, no. 260, pp. 583–621, 1952.

[83] A. de Myttenaere, B. Golden, B. Le Grand, and F. Rossi, "Mean absolute percentage error for regression models," *Neurocomputing*, vol. 192, p. 38–48, June 2016.

[84] Y. Dodge, *Mean Squared Error*, pp. 337–339. New York, NY: Springer New York, 2008.

[85] E. Zivot and J. Wang, *Rolling Analysis of Time Series*, pp. 299–346. New York, NY: Springer New York, 2003.

[86] S. Krstanovic and H. Paulheim, "Ensembles of recurrent neural networks for robust time series forecasting," in *Artificial Intelligence XXXIV* (M. Bramer and M. Petridis, eds.), (Cham), pp. 34–46, Springer International Publishing, 2017.

[87] P. Goodwin and R. Lawton, "On the asymmetry of the symmetric mape," *International Journal of Forecasting*, vol. 15, no. 4, pp. 405–408, 1999.

[88] C. J. Willmott and K. Matsuura, "Advantages of the mean absolute error (mae) over the root mean square error (rmse) in assessing average model performance," *Climate Research*, vol. 30, no. 1, pp. 79–82, 2005.

[89] G. M. LJUNG and G. E. P. BOX, "On a measure of lack of fit in time series models," *Biometrika*, vol. 65, pp. 297–303, 08 1978.

[90] G. E. P. Box and D. A. Pierce, "Distribution of residual autocorrelations in autoregressive-integrated moving average time series models," *Journal of the American Statistical Association*, vol. 65, no. 332, pp. 1509–1526, 1970.

[91] R. J. Hyndman and Y. Khandakar, "Automatic time series forecasting: The forecast package for r," *Journal of Statistical Software*, vol. 27, pp. 1–22, 2008.

[92] M. Ainsworth and Y. Shin, "Plateau phenomenon in gradient descent training of relu networks: Explanation, quantification, and avoidance," *SIAM Journal on Scientific Computing*, vol. 43, no. 5, pp. A3438–A3468, 2021.

[93] D. H. Ackley, G. E. Hinton, and T. J. Sejnowski, "A learning algorithm for boltzmann machines," *Cogn. Sci.*, vol. 9, pp. 147–169, 1985.

[94] H. Wu, B. Xiao, N. C. F. Codella, M. Liu, X. Dai, L. Yuan, and L. Zhang, "Cvt: Introducing convolutions to vision transformers," *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 22–31, 2021.

[95] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2015.

[96] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[97] J. Gao, X. Song, Q. Wen, P. Wang, L. Sun, and H. Xu, "Robusttad: Robust time series anomaly detection via decomposition and convolutional neural networks," *ArXiv*, vol. abs/2002.09545, 2020.

[98] D. S. Park, W. Chan, Y. Zhang, C.-C. Chiu, B. Zoph, E. D. Cubuk, and Q. V. Le, "Specaugment: A simple data augmentation method for automatic speech recognition," in *Interspeech*, 2019.

[99] B. Wu, C. Xu, X. Dai, A. Wan, P. Zhang, Z. Yan, M. Tomizuka, J. Gonzalez, K. Keutzer, and P. Vajda, "Visual transformers: Token-based image representation and processing for computer vision," 2020.

[100] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255, Ieee, 2009.

[101] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," *CoRR*, vol. abs/2103.14030, 2021.

[102] R. Wightman, H. Touvron, and H. Jegou, "Resnet strikes back: An improved training procedure in timm," in *NeurIPS 2021 Workshop on ImageNet: Past, Present, and Future*.

[103] R. Godahewa, C. Bergmeir, G. I. Webb, R. J. Hyndman, and P. Montero-Manso, "Monash time series forecasting archive," in *Neural Information Processing Systems Track on Datasets and Benchmarks*, 2021. forthcoming.

[104] Z. Liu, Y. Wang, S. Vaidya, F. Ruehle, J. Halverson, M. Soljačić, T. Y. Hou, and M. Tegmark, "Kan: Kolmogorov-arnold networks," 2024.

A. Objectius de desenvolupament sostenible

En setembre de 2015, l'Organització de Nacions Unides van adoptar un conjunt d'objectius comuns a escala global per al desenvolupament que s'haurien de complir abans de l'any 2030. Aquesta Agenda 2030 compta amb dèsset punts i s'espera que el seu compliment ajude a millorar la vida de les persones a tot el món.

La UPV, com a institució socialment compromesa, té molt present el compliment d'aquesta agenda, per la qual cosa no podríem tancar la memòria d'aquest projecte sense considerar com podem afectar el seu compliment.

Així, dels dèsset objectius que conformen l'acord internacional, considerem que el nostre projecte afecta fonamentalment a huit d'ells, tal i com es pot veure a la taula B.1.

Objetivos de Desarrolo Sostenibles.	**Alt**	**Mitjà**	**Baix**	**No Procedeix**
ODS 1. **Fi de la pobresa.**				**X**
ODS 2. **Fam zero.**				**X**
ODS 3. **Salut i benestar.**				**X**
ODS 4. **Educació de qualitat**				**X**
ODS 5. **Igualtat de gènere.**				
ODS 6. **Aigua neta i sanejament.**	**X**			
ODS 7. **Energia assequible i no contaminant.**				**X**
ODS 8. **Treball decent i creixement econòmic.**		**X**		
ODS 9. **Indústria, innovació i infraestructures.**		**X**		
ODS 10. **Reducció de desigualtats.**				**X**
ODS 11. **Ciutats i comunitats sostenibles.**				**X**
ODS 12. **Producció i consum responsables.**	**X**			
ODS 13. **Acció pel clima.**	**X**			
ODS 14. **Vida submarina.**	**X**			
ODS 15. **Vida d'ecosistemes terrestres.**	**X**			
ODS 16. **Pau, justícia i institucions sòlides.**				**X**
ODS 17. **Aliances per assolir objectius.**		**X**		

Table A.1.: implicació del projecte amb els Objectius de Desenvolupament Sostenibles.

Aquest projecte es fundamenta en el reciclatge i la reutilització d'envasos. Com a tal un dels objectius clau de l'ODS en els que participa és en el dotzé, ja que redueix la generació de residus tals com plàstics o conglomerats (en el cas dels palets) i ajuda a mantindre un ús eficient dels envasos presents. Així, aquesta reducció ja no només de residus sinó també de producció, ajuda a la lluita contra el canvi climàtic, alineant-se amg l'objectiu tretze.

La reducció d'aquests residus també evita l'acumulació de plàstics en mars i oceans (complint així amb l'objectiu catorze) així com en la terra (de manera que s'alinea amb l'objectiu quinze).

No només això, sinó que el nostre projecte s'encarrega d'optimitzar la neteja de les caixes mitjançant una correcta predicció, per la qual cosa, també ajuda a reduir el consum d'aigua, així que també està molt relacionat amb el sisé objectiu dels ODS.

D'altra banda, aquest projecte sorgeix d'una aliança entre l'administració pública (UPV) i el sector privat (Logifruit), fomentant també l'ODS dèsset.

L'aplicació d'aquest model podria arribar a millorar l'eficiència de les plantes de neteja de caixes de Logifruit, per la qual cosa, en grau més baix, també s'està treballant per assolir el huité punt dels ODS.

Finalment, també participem en el nové objectiu de desenvolupament solidari, puix estem treballant per tindre una indústria cada vegada més sostenible amb el medi ambient.