

# Predicció de la Demanda Energètica per Barris de Barcelona per a una Gestió Sostenible dels Recursos.

Aleix Francía Albert

**Resum–** L'electricitat constitueix un element fonamental de la societat contemporània i un indicador del desenvolupament tecnològic.

El repte de garantir un ús sostenible dels recursos es presenta com una prioritat global, on l'optimització esdevé un instrument essencial. D'acord amb l'Agència Internacional de l'Energia (AIE), el consum energètic a l'Unió Europea va disminuir un 3,2% el 2022, la segona més gran des de 2009. Tot i que es preveu una recuperació, els preus negatius de l'electricitat s'han fet més freqüents, reflectint la inestabilitat del mercat i la necessitat d'un avenç administratiu.

Aquest estudi es centra en la predicció de la demanda elèctrica a la ciutat de Barcelona, mitjançant l'ús d'algoritmes d'intel·ligència artificial. Aquestes prediccions i la incertesa associada, proporcionen eines per a l'optimització i la gestió dels recursos.

**Paraules clau–** Eficiència energètica, consum elèctric, predicció de demanda, intel·ligència artificial, volatilitat del mercat, Barcelona.

**Abstract–** Electricity is a fundamental element of contemporary society and an indicator of technological development.

The challenge of ensuring the sustainable use of resources is presented as a global priority, where optimization becomes an essential tool. According to the International Energy Agency (IEA), energy consumption in the European Union decreased by 3.2% in 2022, the second-largest drop since 2009. Although recovery is expected, negative electricity prices have become more common, reflecting market instability and the need for better management.

This study focuses on predicting electricity demand in the city of Barcelona, using artificial intelligence algorithms. These predictions and the associated uncertainty provide tools for the optimization and management of resources.

**Keywords–** Energy efficiency, electricity consumption, demand forecasting, artificial intelligence, market volatility, Barcelona.



---

• *E-mail de contacte:* afranciaa2501@gmail.com

• *Treball tutoritzat per:* Albert Romero Sánchez (Departament d'Informàtica)

# ÍNDEX

|          |  |           |
|----------|--|-----------|
| <b>1</b> | <b>Introducció</b>   | <b>3</b>  |
| <b>2</b> | <b>Conjunt de dades</b>  | <b>3</b>  |
| <b>3</b> | <b>EDA(<i>Exploratory Data Analysis</i>)</b>                       | <b>3</b>  |
| 3.1      | Distribució del consum per codi postal. . . . .                    | 3         |
| 3.2      | Histograma i Tendència Temporal . . . . .                          | 4         |
| 3.3      | Distribució del Consum Mensual i Diari . . . . .                   | 4         |
| 3.4      | Distribució Segons Tram Horari . . . . .                           | 4         |
| 3.5      | Influència de les variables independents . . . . .                 | 4         |
| <b>4</b> | <b>Random Forest Regressor</b>                                     | <b>5</b>  |
| 4.1      | Quantificació de la incertesa associada . . . . .                  | 5         |
| <b>5</b> | <b>Extreme Gradient Boosting</b>                                   | <b>5</b>  |
| 5.1      | Hiperparàmetres Seleccionats . . . . .                             | 5         |
| 5.2      | Selecció de Característiques . . . . .                             | 5         |
| 5.3      | Monte Carlo Ensemble . . . . .                                     | 6         |
| 5.4      | Bootstrap Ensemble . . . . .                                       | 7         |
| <b>6</b> | <b>Xarxa neuronal</b>  | <b>7</b>  |
| 6.1      | Activació . . . . .  | 7         |
| 6.2      | Long Short-Term Memory(LSTM) . . . . .                             | 8         |
| 6.3      | LSTM Bidireccional . . . . .                                       | 8         |
| 6.4      | Mètriques d'avaluació . . . . .                                    | 8         |
| 6.5      | Resultats . . . . .  | 9         |
| 6.6      | Efecte de Batch Normalization . . . . .                            | 9         |
| 6.7      | Efecte de Dropout . . . . .  | 9         |
| 6.8      | Estimació de la Incertesa mitjançant Monte Carlo Dropout . . . . . | 10        |
| <b>7</b> | <b>Conclusions</b>   | <b>10</b> |
| A.1      | Codi . . . . .   | 12        |
| A.2      | Pseudocodi dels diferents algoritmes . . . . .                     | 12        |

# 1 INTRODUCCIÓ

LA creixent demanda energètica, combinada amb la urgència de minimitzar la petjada de carboni, condueix a dissenyar polítiques públiques que vetllin per la transició a un model sostenible. Davant d'aquesta realitat, és de rigor comprendre els factors que determinen la demanda elèctrica, és especialment rellevant a mesura que l'energia continua sent un recurs escàs, degut a la impossibilitat d'emmagatzemar energia de manera eficient.

El consum elèctric domèstic ha patit un augment significatiu a Espanya en les darreres dècades, impulsat per una agrupació de factors econòmics, climàtics i socials. Un dels principals determinants d'aquest increment és la correlació positiva entre l'augment del poder adquisitiu de les llars i la demanda d'electricitat.

Aquest estudi té com a objectiu la predicció de la demanda energètica a Barcelona, desglossada per codis postals, mitjançant l'ús de dades corresponents al període comprès entre els anys 2019 i 2024 i la utilització de sistemes d'intel·ligència artificial. Així mateix, s'analitza la incertesa associada a aquestes prediccions per garantir la fiabilitat dels resultats obtinguts i proporcionar una eina addicional per a la presa de decisions.

L'anàlisi proporciona evidències empíriques que faciliten la comprensió de les dinàmiques energètiques locals, generant així una eina per al disseny d'estratègies adaptades, suposant un avenç cap a una gestió dels recursos més eficient sense comprometre el desenvolupament i les necessitats inherents a la ciutat contemporània.

# 2 CONJUNT DE DADES

L'elaboració del model predictiu s'ha basat en un conjunt de dades multidimensional que recull informació energètica, meteorològica i econòmica referida a la ciutat de Barcelona, al llarg del període comprès entre els anys 2019 i 2024. Aquest conjunt de dades ha estat obtingut a partir de múltiples fonts públiques, consolidant-ne la coherència i la compatibilitat per a l'anàlisi posterior.

- **Consum elèctric:** Obtingut de l'Observatori de l'Energia de Barcelona i del portal de dades obertes de l'Ajuntament de Barcelona [1, 2]. Proporcionen registres desagregats per codis postals i franges horàries.
- **Dades meteorològiques:** Extretes mitjançant l'API d'arxiu meteorològic d'Open Meteo [3]. Inclouen múltiples variables, com la temperatura, la velocitat del vent, la radiació solar i la durada de la insolació.
- **Indicadors econòmics:** Es decideix incorporar la taxa interanual de l'Índex de Producció Industrial (IPI), obtinguda de l'EPData en col·laboració amb l'Institut Nacional d'Estadística (INE) [4].

A continuació, s'esmenten les variables emprades en l'estudi, juntament amb la seva descripció i tipologia, tal com es recull a la Taula 1. Les dades de consum energètic originals es trobaven desagregades per sectors, en conseqüència han estat agregades per tal de considerar el consum total. Aquesta agrupació respon a la necessitat d'obtenir una visió global i integrada del comportament energètic a nivell local.

Paral·lelament, s'ha aplicat un procés de pretractament que ha inclòs l'eliminació de valors nuls, la normalització de les escales i la codificació de les variables categòriques, amb l'objectiu d'optimitzar el rendiment del model.

| Variable              | Descripció                     | Tipus              |
|-----------------------|--------------------------------|--------------------|
| Any                   | Any del registre.              | Numèrica discreta  |
| Mes                   | Mes de l'any.                  | Catègorica ordinal |
| Dia                   | Dia del mes.                   | Catègorica ordinal |
| Tram_Horari           | Franja horària (1–4).          | Catègorica ordinal |
| Codi_Postal           | Zona geogràfica.               | Catègorica nominal |
| Valor                 | Variable objectiu a predir.    | Numèrica contínua  |
| Temperatura_2m        | Temperatura a 2 m.             | Numèrica contínua  |
| Temperatura_Percebuda | Temperatura percebuda.         | Numèrica contínua  |
| Pluja                 | Pluja (sí/no).                 | Binària            |
| Velocitat_Vent_10m    | Velocitat del vent a 10 m.     | Numèrica contínua  |
| Es_De_Dia             | Si és de dia.                  | Binària            |
| Durada_Sol            | Durada de la insolació.        | Numèrica contínua  |
| Radiació_Directa      | Radiació solar directa.        | Numèrica contínua  |
| Dia_Setmana           | Dia de la setmana (0–6).       | Catègorica ordinal |
| Festiu                | Si és festiu.                  | Binària            |
| Taxa_Interanual_IPI   | Índex de producció industrial. | Numèrica contínua  |
| Punt_Rosada_2m        | Punt de rosada a 2 m.          | Numèrica contínua  |

Dimensions del conjunt de dades: 348.030 observacions × 17 variables

TAULA 1: VARIABLES DEL CONJUNT DE DADES AMB DESCRIPCIÓ I TIPUS.

# 3 EDA(Exploratory Data Analysis)

Un cop realitzat el preprocessament de les dades, es desenvolupa una anàlisi exploratòria mitjançant tècniques estadístiques i visuals per comprendre la naturalesa del fenomen, facilitant el posterior desenvolupament del model predictiu. L'objectiu principal és identificar patrons, tendències, valors atípics i correlacions entre les diferents variables independents respecte la variable dependent.

## 3.1 Distribució del consum per codi postal.

Tal com es mostra en el mapa de calor de la Figura 1, el consum energètic es distribueix de manera desigual entre els diferents codis postals de l'àrea de Barcelona. Aquesta variació espacial reflecteix diferències subjacents en la densitat de població, l'ús del sòl o l'activitat econòmica.

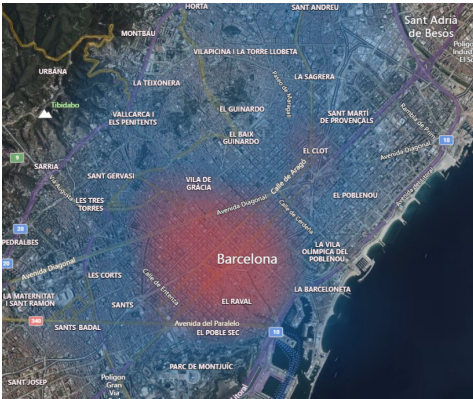


Fig. 1: Distribució espacial del consum energètic a Barcelona.

### 3.2 Histograma i Tendència Temporal

L'histograma de la Figura 2 i el diagrama de caixes de la Figura 3 evidencien una tendència sostinguda a la reducció del consum energètic anual. Aquesta evolució pot atribuir-se a factors estructurals com la implementació de polítiques públiques orientades a l'eficiència energètica, l'augment de la sensibilització ambiental i la renovació tecnològica d'infraestructures i equips. Especialment en entorns densament poblats com l'àrea metropolitana de Barcelona, aquestes mesures tenen un efecte acumulatiu rellevant sobre la demanda. Tanmateix, és fonamental considerar l'impacte de variables externes i esdeveniments disruptius, com canvis econòmics, oscil·lacions climàtiques o la pandèmia de la, que poden introduir anomalies temporals en la tendència observada.

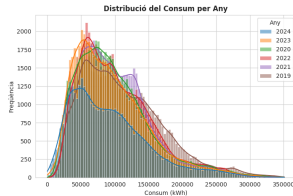


Fig. 2: Historiograma.

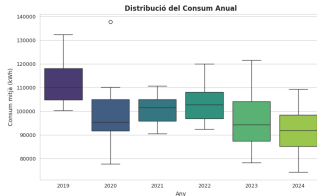


Fig. 3: Consum anual.

### 3.3 Distribució del Consum Mensual i Diari

Com es mostra a la Figura 4, el consum energètic mensual presenta un patró clar i recurrent que influeix significativament en la demanda global. En canvi, el consum diari a la Figura 5 sembla més variable i no segueix una tendència de distribució consistent.

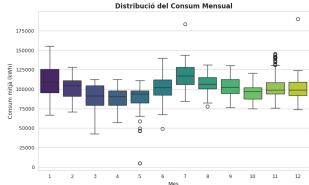


Fig. 4: Consum Mensual.

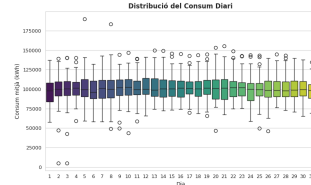


Fig. 5: Consum Diari.

### 3.4 Distribució Segons Tram Horari

El període horari és una variable que influeix directament en la demanda energètica, principalment determinada per patrons d'activitat socioeconòmica. El consum tendeix a arribar al seu màxim entre les **12:00–18:00**, i el període de **00:00–06:00** presenta el consum més baix, amb una activitat mínima i poca variabilitat.

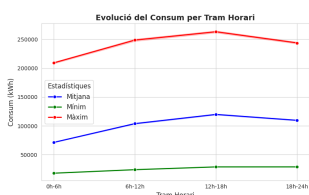


Fig. 6: Evolució per tram.

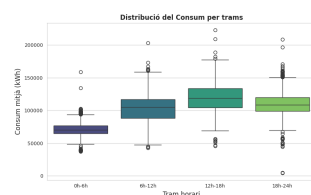


Fig. 7: Diagrama de caixes.

### 3.5 Influència de les variables independents

Per comprendre la influència de les variables independents sobre la dependent, és clau analitzar les seves interrelacions mitjançant coeficients de correlació, que indiquen la força i direcció d'aquestes associacions.

- **Pearson:** Mesura la relació lineal entre dues variables. La fórmula per al coeficient de correlació de Pearson ( $r$ ) és:

$$r = \frac{\sum (X_i - \bar{X})(Y_i - \bar{Y})}{\sqrt{\sum (X_i - \bar{X})^2 \sum (Y_i - \bar{Y})^2}}$$

on  $X_i$  i  $Y_i$  són les observacions de les dues variables, i  $\bar{X}$  i  $\bar{Y}$  són les mitjanes.

- **Spearman:** Mesura la relació monòtona (quan dues variables evolucionen en la mateixa direcció), però no necessàriament de manera lineal. La fórmula per al coeficient de correlació de Spearman ( $\rho$ ) és:

$$\rho = 1 - \frac{6 \sum d_i^2}{n(n^2 - 1)}$$

on  $d_i$  és la diferència entre els rangs de les dues variables, i  $n$  és el nombre d'observacions.

- **Correlació de Distància (DCOR):** Mesura tot tipus de dependència entre dues variables, ja sigui lineal o no lineal. La fórmula per a la correlació de distància és:

$$DCOR(X, Y) = \frac{\sqrt{\sum_{i,j} (d(X_i, X_j) - d(Y_i, Y_j))^2}}{n}$$

on  $d(X_i, X_j)$  és la distància entre les observacions de  $X$ , i  $d(Y_i, Y_j)$  és la distància entre les observacions de  $Y$ , i  $n$  és el nombre d'observacions.

El següent mapa de calor, representat a la Taula 2, mostra els coeficients de correlació entre les variables independents i la demanda energètica:

| Variable              | Pearson | Spearman | Distance Correlation |
|-----------------------|---------|----------|----------------------|
| És_De_Dia             | 0.205   | 0.213    | 0.907                |
| Tram_Horari           | 0.258   | 0.287    | 0.283                |
| Durada_Sol            | 0.217   | 0.229    | 0.220                |
| Radiació_Directa      | 0.197   | 0.240    | 0.201                |
| Codi_Postal           | 0.037   | -0.029   | 0.126                |
| Temperatura_2m        | 0.133   | 0.121    | 0.124                |
| Festiu                | -0.054  | -0.049   | 0.115                |
| Velocitat_Vent_10m    | 0.102   | 0.117    | 0.109                |
| Temperatura_Percebuda | 0.111   | 0.100    | 0.106                |
| Dia_Setmana           | -0.111  | -0.099   | 0.102                |
| Pluja                 | 0.013   | 0.016    | 0.095                |
| Any                   | -0.084  | -0.091   | 0.078                |
| Taxa_Interanual_IPI   | 0.008   | -0.007   | 0.053                |
| Punt_Rosada_2m        | 0.044   | 0.039    | 0.050                |
| Mes                   | 0.019   | 0.018    | 0.042                |
| Dia                   | 0.001   | 0.001    | 0.006                |

TAULA 2: COMPARACIÓ DE PEARSON, SPEARMAN I LA CORRELACIÓ DE DISTÀNCIA AMB EL VALOR

## 4 RANDOM FOREST REGRESSOR

El **RandomForestRegressor** és un model de regressió que utilitza múltiples arbres de decisió generats mitjançant la tècnica de **bagging**, on cada arbre es crea a partir d'un subconjunt aleatori de les dades i característiques. La predicció final és la mitjana de les prediccions dels arbres generats.

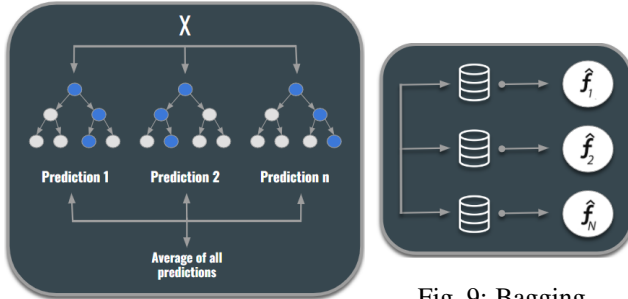


Fig. 9: Bagging.

Fig. 8: RandomForestRegressor.

És un model adient per a l'estudi, ja que entrenant cada arbre amb subconjunts diferents de les dades minimitza la dependència entre ells, augmentant la capacitat de generalització. A més, captura relacions no lineals i gestiona eficaçment dades sorolloses o *outliers*.

### 4.1 Quantificació de la incertesa associada

El model quantifica la incertesa en cada predicció, determinada mitjançant el càlcul de la desviació estàndard del conjunt de prediccions dels arbres de decisió individuals, reflectint la variabilitat en els resultats del model i així facilitant el desenvolupament d'estratègies adaptatives.

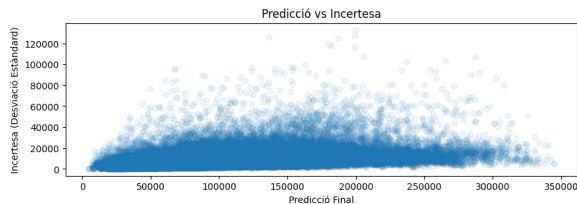


Fig. 10: Relació entre les prediccions finals i la incertesa associada (desviació estàndard) per a cada observació.

## 5 EXTREME GRADIENT BOOSTING

XGBoost[5] és una tècnica d'aprenentatge automàtic d'arbre de decisions augmentat amb gradient (GBDT) escalable i distribuïda, pertany a la família dels models d'*ensemble learning*. Els **GBDT** entrenen de manera seqüencial un conjunt d'arbres de decisió poc profunds, amb cada iteració utilitzant els residus d'error del model anterior, els arbres nous s'optimitzen per corregir els errors dels arbres previs. La predicció final és una suma ponderada de totes les prediccions de l'arbre.

Tant el **Random Forest** com el **GBDT** construeixen models formats per diversos arbres de decisió, però la diferència rau en com es construeixen i combinen els arbres.

El terme *gradient boosting* fa referència al procés de millorar un model feble mitjançant la combinació d'altres models febles per generar-ne un de fort.

Aquest mètode optimitza funcions objectiu per al model posterior, millorant-lo mitjançant la segona derivada (*Taylor expansion*) de la funció de pèrdua, fet que el fa més ràpid en comparació amb les tècniques de gradient boosting tradicionals.

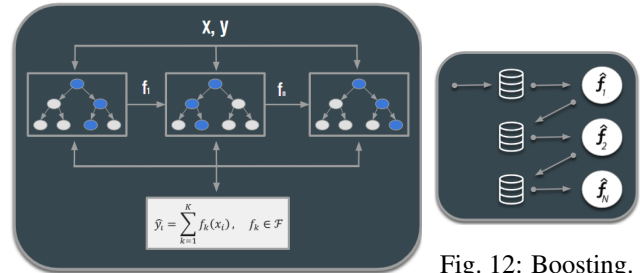


Fig. 11: XGBoost.

Fig. 12: Boosting.

### 5.1 Hiperparàmetres Seleccionats

Els hiperparàmetres controlen el procés d'aprenentatge d'un model. En el cas d'XGBoost, determinen el creixement dels arbres i l'optimització del model.

Es defineix una graella de hiperparàmetres per explorar diferents combinacions de valors i trobar la configuració que obté els resultats més adients per al conjunt de dades. Aquesta tasca es realitza mitjançant la tècnica d'optimització **GridSearchCV**[6], ja utilitzada prèviament. A més, s'aplica regularització amb **Elastic Net**.

- **reg lambda:** Aplica una penalització quadràtica als pesos dels arbres per reduir la variabilitat i evitar pesos excessivament grans:

$$\text{Regularització L2} = \lambda \sum_{i=1}^n w_i^2$$

- **reg alpha:** Aplica una penalització absoluta als pesos, promovent la dispersió en el model:

$$\text{Regularització L1} = \alpha \sum_{i=1}^n |w_i|$$

- **Elastic Net:** Combina penalitzacions L1 (Lasso) i L2 (Ridge):

$$\text{Minimitzar} \quad \text{Funció de Pèrdua} + \lambda_1 \sum |w_i| + \lambda_2 \sum w_i^2$$

### 5.2 Selecció de Característiques

Amb l'objectiu de reduir el nombre de variables i optimitzar el rendiment del model, s'ha aplicat la tècnica de selecció recursiva de característiques amb validació creuada (RFECV). Aquest mètode elimina iterativament les característiques menys rellevants, utilitzant la validació creuada per determinar el nombre òptim de variables que maximitza la mètrica de rendiment, en aquest cas, el coeficient de determinació  $R^2$  del model XGBoost.

Aquesta tècnica resulta més eficient computacionalment que la selecció exhaustiva, ja que no avalua totes les combinacions possibles, sinó que elimina progressivament característiques fins a trobar el subconjunt més adequat.



El procés de selecció es detallat en el pseudocodi 1 que es presenta a continuació. Aquest esquema il·lustra els passos principals per optimitzar el conjunt de característiques.

#### Algorithm 1 Pseudocodi simplificat de RFECV

```

0: Inicialitzar el conjunt de característiques  $F$ .
0: Inicialitzar la millor puntuació  $best\_score \leftarrow 0$ 
0: while no es compleix el criteri de parada do
0:   for cada fold en validació creuada do
0:     Entrenar el model amb les característiques  $F$  en
       les dades d'entrenament
0:     Avaluar el model en les dades de validació
0:   end for
0:   Calcular la puntuació mitjana  $score$ 
0:   if  $score > best\_score$  then
0:      $best\_score \leftarrow score$ 
0:     Guardar el conjunt actual  $F$  com el millor conjunt
0:   end if
0:   Eliminar les característiques menys importants de  $F$ 
0: end while
0: Retornar el millor conjunt de característiques. =0

```

S'han avaluat tres configuracions de XGBoost per mesurar l'impacte de l'optimització: model base, model ajustat amb GridSearchCV i model combinat amb GridSearchCV i selecció de característiques RFECV. La Taula 3 mostra els resultats obtinguts.

| Configuració                     | MAPE (0-100) ↓ | $R^2$ (0-1) ↑ |
|----------------------------------|----------------|---------------|
| XGBoost (sense optimització)     | 2.50           | 0.94          |
| XGBoost amb GridSearchCV         | <b>1.71</b>    | <b>0.97</b>   |
| XGBoost amb GridSearchCV + RFECV | 1.82           | 0.97          |

TAULA 3: COMPARACIÓ DE MÈTRIQES PER AL MODEL XGBOOST SOTA DIFERENTS CONFIGURACIONS.

Aquest procés de selecció de característiques s'ha implementat utilitzant la biblioteca `scikit-learn`, que proporciona eines eficaces i àmpliament reconegudes, entre una d'elles la selecció recursiva de característiques (RFECV). Mitjançant l'ús de RFECV, s'ha pogut identificar de manera sistemàtica les característiques més rellevants, eliminant progressivament les variables menys informatives durant el procés de validació creuada. Aquesta metodologia no ha millorat la precisió del model, sinó que ha reduït la complexitat. Per tant no es farà ús en posteriors apartats.

### 5.3 Monte Carlo Ensemble

El mètode *Monte Carlo* fa referència a una classe d'algorismes computacionals que utilitzen mostreig aleatori repetit, resolent tasques deterministes complexes, amb tècniques probabilístiques. El nom *Monte Carlo* prové del famós casino de Mònaco i va ser popularitzat per Nicholas Metropolis i Stanislaw Ulam en el seu treball de 1949 [7, 8].

Un principi fonamental que recolza aquesta metodologia és la *Llei dels Grans Nombres*, que estableix que, a mesura que augmenta el nombre de mostres, la mitjana de les mostres es va acostant al valor esperat. Aquest principi és essencial per al mètode de Monte Carlo, ja que justifica l'ús d'un nombre gran de simulacions per obtenir una estimació precisa.

A les Figures 13 i 14, es mostra clarament com, a mesura que s'augmenta la mida de la mostra, la distribució resultant es va ajustant cada cop més a la distribució normal real.

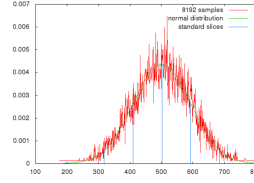


Fig. 13: Distribució normal amb ~8.000 mostres [9].

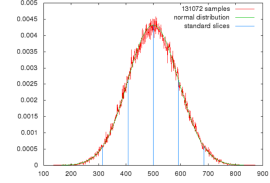


Fig. 14: Distribució normal amb ~131.000 mostres [9].

En aquest estudi, el mètode **Monte Carlo** s'aplica com a tècnica estadística per estimar distribucions de probabilitat mitjançant un *ensemble* de models XGBoost entrenats amb diferents llavors aleatòries (`random_state`). Aquest procediment, descrit al pseudocodi de la Secció A.2, genera una distribució de prediccions per entrada. La mitjana d'aquestes prediccions proporciona l'estimació puntual, mentre que la seva desviació estàndard permet quantificar la incertesa associada, reflectint la variabilitat del model i la fiabilitat de les prediccions.

• **Predicció Puntual:**  $\hat{y} = \frac{1}{N} \sum_{i=1}^N \hat{y}_i$

• **Incertesa (Desviació Estàndard):**

$$\sigma = \sqrt{\frac{1}{N} \sum_{i=1}^N (\hat{y}_i - \hat{y})^2}$$

• **Interval de Confiança:**  $CI_{95\%} = [\hat{y}_{2.5\%}, \hat{y}_{97.5\%}]$

En aquest estudi, el mètode **Monte Carlo** s'aplica com a tècnica estadística per estimar distribucions de probabilitat mitjançant un *ensemble* de models XGBoost entrenats amb diferents llavors aleatòries (`random_state`).

Aquest procediment, descrit al pseudocodi de la Secció A.2, genera una distribució de prediccions per entrada. La mitjana d'aquestes prediccions proporciona l'estimació puntual, mentre que la seva desviació estàndard permet quantificar la incertesa associada. Aquesta aproximació facilita una anàlisi més robusta en contextos on la incertesa és un factor crític. Han sigut assolits uns resultats de **1.74** de MAPE i **0.97** de  $R^2$  sobre el conjunt de test.

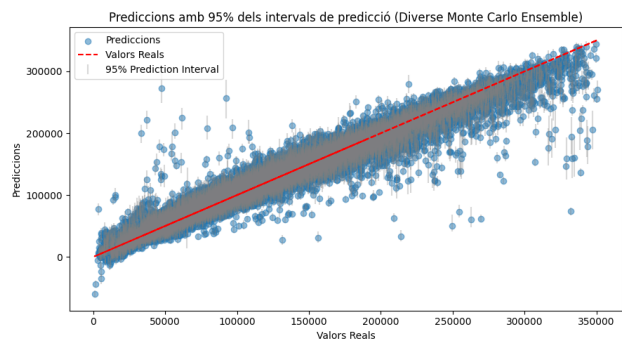


Fig. 15: Prediccions d'ensemble de Monte Carlo amb intervals de confiança del 95%.

## 5.4 Bootstrap Ensemble

*Bagging (Bootstrap Aggregating)*[10] és una tècnica estadística que construeix múltiples models a partir de subconjunts aleatoris de les dades originals. L'objectiu d'aquesta tècnica és quantificar la incertesa associada a les prediccions.

A partir d'un conjunt d'entrenament  $D$  de mida  $n$ , es generen  $m$  nous conjunts de dades  $D_i$  de mida  $n'$  mitjançant mostreig aleatori, amb o sense reposició. Per a  $n' = n$  i un  $n$  gran, s'espera que aproximadament un  $(1 - 1/e) \approx 63.2\%$  de les mostres siguin úniques, mentre que la resta seran duplicades—aquesta és la *bootstrap sample*[11].

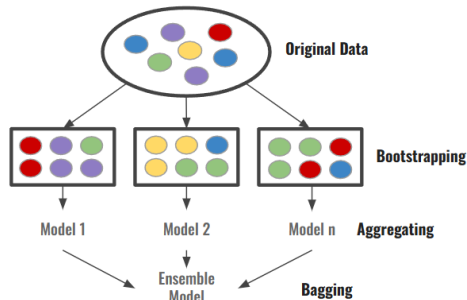


Fig. 16: Il·lustració de l'agregació per bootstrap.

Cada conjunt bootstrap es fa servir per entrenar un model independent. Per a la predicció, els resultats de tots els models són combinats, obtenint una estimació agregada. Permetent obtenir tant una predicció puntual com una estimació d'incertesa mitjançant la distribució dels resultats dels models, de manera similar al **Monte Carlo Ensemble** anteriorment descrit.

Aquesta tècnica permet construir intervals de confiança per reflectir la incertesa modelada, proporcionant un marc per a la presa de decisions basades en la incertesa. Han sigut assolits uns resultats de **1.8** de MAPE i **0.94** de  $R^2$  sobre el conjunt de test.

## 6 XARXA NEURONAL

Una xarxa neuronal artificial[12] és un paradigma d'aprenentatge i processament automàtic inspirat en la forma en què funciona el sistema nerviós dels animals. Es tracta d'un sistema d'interconnexió de neurones en una xarxa que col·labora per produir un estímul de sortida. Seguint l'enfocament connexionista, la xarxa aprèn patrons complexos ajustant aquests pesos mitjançant algorismes d'entrenament com el **backpropagation**.

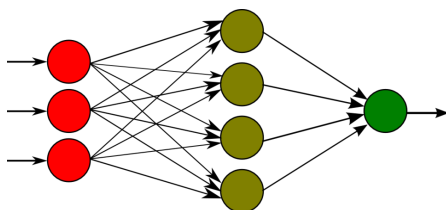


Fig. 17: **Xarxa neuronal artificial**. Perceptró simple amb 3 neurones d'entrada, 4 neurones en la seva capa oculta i una neurona de sortida [12].

Els models, han sigut generats utilitzant un model base que es mostra a la Taula 4, adaptat del treball presentat a [13]. El model ha estat ajustat per adaptar-se a les característiques específiques del conjunt de dades utilitzat en aquest estudi. Per capturar millor els patrons temporals presents a les dades, va ser incorporat unitats de *Long Short-Term Memory (LSTM)* [14] a l'arquitectura.

| Layer           | Input Shape    | Output Shape   | Param # |
|-----------------|----------------|----------------|---------|
| InputLayer      | (None, 15, 1)  | (None, 15, 1)  | 0       |
| LSTM            | (None, 15, 1)  | (None, 15, 64) | 16,896  |
| GlobalAvgPool1D | (None, 15, 64) | (None, 64)     | 0       |
| Dense(ReLU)     | (None, 64)     | (None, 256)    | 16,640  |
| BatchNorm       | (None, 256)    | (None, 256)    | 1,024   |
| Dropout(0.19)   | (None, 256)    | (None, 256)    | 0       |
| Dense(ReLU)     | (None, 256)    | (None, 32)     | 8,224   |
| BatchNorm       | (None, 32)     | (None, 32)     | 128     |
| Dropout(0.5)    | (None, 32)     | (None, 32)     | 0       |
| Dense(Linear)   | (None, 32)     | (None, 1)      | 33      |

TAULA 4: XARXA NEURONAL DE REFERÈNCIA.

### 6.1 Activació

L'activació transforma l'entrada d'una neurona en una sortida mitjançant una funció d'activació, introduint no linearitat a la xarxa neuronal. Per al cas d'estudi, diverses variants de la funció ReLU seran utilitzades, inicialment implementada en el model.

- **ReLU (Rectified Linear Unit):**

$$f(x) = \max(0, x)$$

- **LeakyReLU (Leaky Rectified Linear Unit):**

$$f(x) = \begin{cases} x, & \text{si } x > 0 \\ \alpha x, & \text{si } x < 0 \end{cases}$$

- **ELU (Exponential Linear Unit):**

$$f(x) = \begin{cases} x, & \text{si } x > 0 \\ \alpha(e^x - 1), & \text{si } x < 0 \end{cases}$$

Com es mostra a la Taula 5, segons l'estudi [15], cada funció d'activació mostra un comportament diferent dependent del conjunt de dades utilitzat.

| Accuracy results comparison over CIFAR100 dataset |              |              |              |              |              |
|---|--------------|--------------|--------------|--------------|--------------|
| Activation  | MobileNet    | GoogleNet    | ResNet50     | SENet18      | DenseNet121  |
| ELU   | 61.97 ± 0.24 | 72.57 ± 1.76 | 71.41 ± 1.63 | 71.27 ± 0.58 | 72.06 ± 1.93 |
| ReLU  | 61.33 ± 0.34 | 74.05 ± 1.69 | 71.96 ± 0.94 | 70.45 ± 0.73 | 72.99 ± 1.35 |
| LReLU   | 61.13 ± 0.41 | 63.79 ± 2.38 | 72.77 ± 0.49 | 70.58 ± 0.45 | 73.33 ± 1.25 |
| Training time comparison over CIFAR100 dataset    |              |              |              |              |              |
| Activation  | MobileNet    | GoogleNet    | ResNet50     | SENet18      | DenseNet121  |
| ELU   | 00:31:05     | 04:57:37     | 03:36:47     | 01:13:25     | 04:08:39     |
| ReLU  | 00:31:22     | 04:55:10     | 03:32:30     | 01:15:33     | 04:15:06     |
| LReLU   | 00:31:48     | 05:01:30     | 03:33:00     | 01:18:38     | 04:14:09     |

TAULA 5: COMPARACIÓ SOBRE CIFAR100 ADAPTADA DE "ACTIVATION FUNCTIONS IN DEEP LEARNING: A COMPREHENSIVE SURVEY AND BENCHMARK"[15].

## 6.2 Long Short-Term Memory(LSTM)

Inicialment, el model va ser entrenat i avaluat utilitzant la funció d'activació *ReLU* (Rectified Linear Unit), coneguda per la seva simplicitat computacional i l'eficiència en la propagació del gradient. A més, *ReLU* accelera el temps d'entrenament en comparació amb altres alternatives, com la funció sigmoide o tanh, degut al seu comportament més simple durant la retropropagació. Al mateix temps, es va incorporar memòria a llarg termini mitjançant *LSTM* (Long Short-Term Memory), una arquitectura específica dins de les xarxes neuronals recurrents (*RNN*), dissenyada per superar les limitacions inherents a les *RNN* tradicionals que sovint presenten dificultats per modelar dependències de llarga durada.

Les *LSTM* es van desenvolupar per abordar les dificultats de les *RNN* clàssiques, que presenten dificultats per retindre informació rellevant durant períodes de temps llargs, degut al fenomen de la desaparició o explosió del gradient. A diferència de les *RNN* convencionals, les *LSTM* introdueixen una estructura basada en portes (*input*, *forget* i *output gates*) que regula el flux d'informació a través de la xarxa, permetent la retenció o el descens d'informació important segons la seva rellevància temporal.

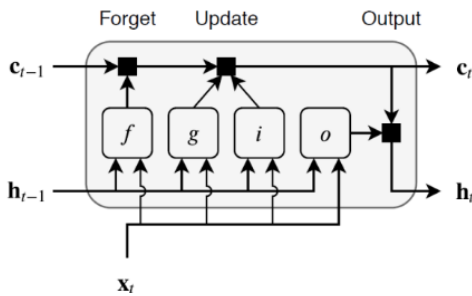


Fig. 18: LSTM.

Aquesta arquitectura facilita l'aprenentatge de seqüències complexes i temporalment llargues, millorant el rendiment del model en tasques com el reconeixement de patrons o la predicció de sèries temporals.

Els LSTM funcionen a través de les següents portes de controlen el flux d'informació:

- **Porta d'oblitament:** Decideix quina informació de la memòria es manté o es descarta, el valor de la porta és una valoració entre 0 (l'informació es descarta) i 1 (la informació es manté).
- **Porta d'entrada:** Decideix quina informació nova s'afegeix a la memòria, Combinant l'*input* actual amb l'estat anterior. La porta d'entrada utilitza una funció d'activació (normalment una **sigmoide**) per controlar l'actualització.

$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

- **Porta de sortida:** Determina quina informació es transmet a la següent capa o com a *output*, també fa servir l'activació **sigmoide**.

L'*LSTM* tradicional processa la seqüència d'entrada en una única direcció (generalment d'esquerra a dreta). L'estat de la xarxa en cada moment depèn de l'entrada i de l'estat anterior.

## 6.3 LSTM Bidireccional

L'*LSTM* **bidireccional** es compon de dues rutes d'entrada paral·leles, cadascuna amb un flux de processament en direccions oposades. Aquesta estructura permet al model obtenir una visió més completa de la seqüència d'entrada.

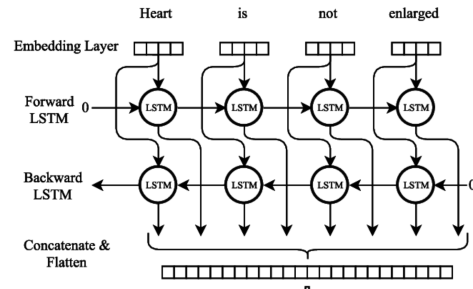


Fig. 19: LSTM Bidireccional.

Per avaluar l'impacte de les arquitectures LSTM bidireccionals en comparació amb les tradicionals, es generarà tant un model **LSTM tradicional** com un model amb estructura bidireccional per cada tipus d'activació esmentat prèviament. Tots els models mantindran la mateixa arquitectura inicial, de manera que podrà ser comparat directament l'efecte de la direccionalitat en el rendiment del model.

## 6.4 Mètriques d'avaluació

Les mètriques que s'usaran durant l'estudi per avaluar com d'adient és el model i el tipus d'activació escollida per la nostra tasca, seran com a pèrdua **MSE**, com a mètrica **MAPE**, i **R<sup>2</sup>**.

- **MSE** (Mean Squared Error): mitjana del quadrat de les diferències entre els valors predits i els reals. Penalitza més les prediccions allunyades. És la mètrica utilitzada com a pèrdua durant l'entrenament dels diferents models.

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2$$

- **MAPE** (Mean Absolute Percentage Error): : és la mitjana dels errors relatius absoluts expressats en percentatge. Es tracta d'una mètrica àmpliament reconeguda per quantificar la desviació entre les prediccions i els valors reals, especialment per la seva fàcil interpretabilitat.

$$MAPE = \frac{1}{n} \sum_{i=1}^n \left| \frac{y_i - \hat{y}_i}{y_i} \right| \times 100$$

- **R<sup>2</sup>** (Coeficient de Determinació): representa la proporció de la variància de la variable dependent que és explicada pel model respecte al total de variància observada. S'interpreta com una mesura del grau d'ajust del model als valors reals. Un valor proper a 1 indica que el model té una alta capacitat explicativa, mentre que valors propers a 0 una explicació molt limitada de la variabilitat.

$$R^2 = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2}$$



## 6.5 Resultats

Com pot ser observat a la Taula 6, els models que fan ús del *LSTM* generalment obtenen millors resultats amb la mateixa funció d'activació. El model amb millor rendiment en el conjunt de proves és l' *LSTM* tradicional amb activació *ELU*. D'altra banda, tot i ser la funció d'activació més costosa computacionalment, *ELU* va destacar per la seva bona relació rendiment-computació en comparació amb altres activacions basades en *ReLU*, seguit de *LeakyReLU*, i finalment *ReLU*, la menys costosa.

| Model         | MAPE (0-100) ↓ | R <sup>2</sup> (0-1) ↑ |
|---------------|----------------|------------------------|
| LSTM ELU      | <b>1.7424</b>  | <b>0.9683</b>          |
| BI LSTM ELU   | 1.7919         | 0.9671                 |
| BI LSTM LEAKY | 1.8283         | 0.9659                 |
| LSTM LEAKY    | 1.8567         | 0.9647                 |
| LSTM RELU     | 2.0296         | 0.9591                 |
| BI LSTM RELU  | 2.2039         | 0.9550                 |

TAULA 6: MÈTRIQES DELS DIFERENTS MODELS, ORDENATS PER MAPE CREIXENT.

Pel que fa a la convergència durant l'entrenament, la corba de pèrdua és suau i mostra una disminució gradual. L'evolució de la pèrdua de validació durant l'entrenament per als diferents models es mostra a la Figura 20.

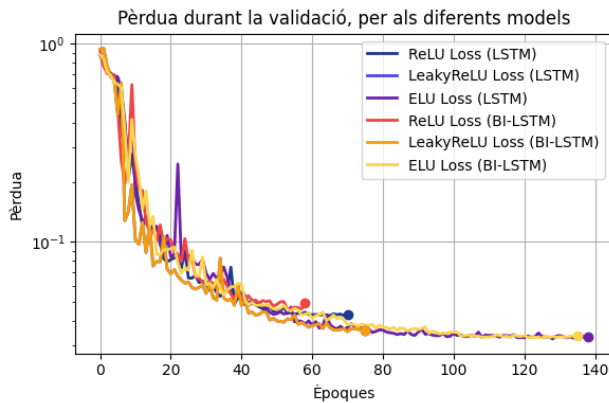


Fig. 20: Pèrdua dels diferents models durant l'entrenament, per al conjunt de validació.

Malgrat la complexitat del model de referència, aquest aconsegueix una convergència primerenca durant les primeres èpoques. Ha sigut aplicada una forta regularització, així evitant el sobreajustament.

En fomentar que el model aprengui els patrons temporals subjacents en lloc de memoritzar les dades d'entrenament. Això millora el rendiment en general, però també provoca una taxa de convergència més lenta durant les etapes posteriors de l'entrenament.

Pel que fa a les dades de prova, tots els models van obtenir millors resultats que durant l'entrenament, mostrant mètriques molt similars a les del conjunt de validació. Això es deu a una regularització forta, com l'ús de capes de *Dropout* i *Batch Normalization*.

## 6.6 Efecte de Batch Normalization

La *Batch Normalization*[16] és una tècnica introduïda per Sergey Ioffe i Christian Szegedy l'any 2015 per millorar el procés d'entrenament de xarxes neuronals, i prevenir l'*explosió del gradient*[17], fenomen que es produeix quan els gradients dels errors són molt significatius i provoquen actualitzacions desmesurades dels pesos del model durant l'entrenament.

Aquesta metodologia consisteix a normalitzar les activacions de les capes per tal que tinguin una mitjana propera a zero i una desviació estàndard pròxima a 1, tret que redueix la variació interna de covariables (*internal covariate shift*) i accelera la convergència del model.[18].

- **Durant l'entrenament:** la capa normalitza la seva sortida utilitzant la mitjana i la desviació estàndard del *batch* actual d'entrades. És a dir, per a cada canal que s'està normalitzant i dificultant la convergència. La capa retorna:

$$\gamma \cdot \frac{\text{batch} - \text{mean}(\text{batch})}{\sqrt{\text{var}(\text{batch}) + \epsilon}} + \beta$$

A continuació es detallen els termes:

- $\epsilon$  és una constant configurable,
- $\gamma$  és un factor d'escalat que s'aprèn (inicial a 1),
- $\beta$  és un factor de desplaçament que s'aprèn (inicial a 0),

- **Durant la validació/test:** La capa normalitza la seva sortida utilitzant una mitjana mòbil de la mitjana i la desviació estàndard dels *batches* observats durant l'entrenament, fent que el model sigui més estable i així aconseguint una millor generalització. Retorna:

$$\gamma \cdot \frac{\text{batch} - \text{self.moving\_mean}}{\sqrt{\text{self.moving\_var} + \epsilon}} + \beta$$

A continuació es detallen els termes:

- **self.moving\_mean** and **self.moving\_var** són variables no entrenables que s'actualitzen cada vegada que la capa és cridada en mode d'entrenament.

La capa només normalitzarà les seves entrades durant la inferència després d'haver estat entrenada amb dades que tinguin estadístiques similars.

## 6.7 Efecte de Dropout

*Dropout*[19] és una tècnica de regularització que tracta de desactivar de manera aleatòria un percentatge de neurones, en cada *forward pass* durant l'entrenament. Fet que obliga al model a evitar la dependència excessiva de certes neurones.

- **Durant l'entrenament:** El model realitza l'entrenament amb una arquitectura "degradada", reduint la seva capacitat i augmentant la pèrdua durant aquest procés.
- **Durant la validació/test:** el Dropout no s'aplica i totes les neurones estan actives, fent que el model funcioni amb el seu potencial total.

## 6.8 Estimació de la Incertesa mitjançant Monte Carlo Dropout

Per avaluar la incertesa associada a les prediccions del model, s'aplica **Monte Carlo Dropout**, mantenint actives les capes de *Dropout* durant la fase d'inferència, com s'observa a la Figura 22, metre que durant l'inferència estàndard es desactiven com es mostra a la Figura 21. Això permet generar múltiples passades estocàstiques amb màscares de *Dropout* diferents, apropant la distribució posterior dels pesos del model.

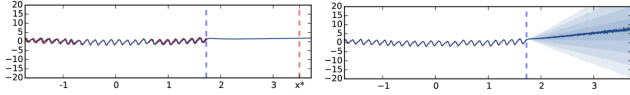


Fig. 21: Dropout estàndard amb pes mitjà [20].

Fig. 22: MC Dropout amb no-linearitats *ReLU* [20].

En un model de xarxa neuronal, la predicció convencional es calcula com:

$$\hat{y} = \hat{f}(x; \theta)$$

En aplicar *Monte Carlo Dropout*, cada passada dona una predicció diferent, i la mitjana i desviació estàndard d'aquestes prediccions proporcionen una estimació de la predicció i la seva incertesa.

El procediment algorítmic és el següent:

### Algorithm 2 Monte Carlo Dropout amb Xarxes Neuronals

**Require:** Model de xarxa neuronal amb *Dropout* (*model*), dades d'entrada *X*, nombre de simulacions *T*

**Ensure:** Mitjana i desviació estàndard per a cada entrada

- 1: Inicialitza una llista buida *f\_preds*
- 2: **for** *t* = 1 a *T* **do**
- 3:    // **Habilita el Dropout durant la inferència**
- 4:    Genera les prediccions: *preds* = *model*(*X*, *training*=True)
- 5:    Afegir *preds* a *f\_preds*
- 6: **end for**
- 7: Calcula la mitjana i desviació estàndard de les prediccions
- 8: **return** *mean\_preds*, *std\_preds* = 0

## 7 CONCLUSIONS

La construcció i avaluació sistemàtica de diversos models predictius utilitzant un conjunt coherent de mètriques quantitatives ha permès una comparació objectiva del seu rendiment. Exclusivament basats en aquests resultats numèrics, els models que van mostrar els nivells més alts d'exactitud predictiva i fiabilitat van ser: la xarxa Long Short-Term Memory (LSTM) amb activació Exponential Linear Unit (ELU) combinada amb Monte Carlo Dropout, el model XGBoost amb Ensemble de Monte Carlo i el RandomForestRegressor. Aquests models van superar consistentment els altres en termes de reducció d'error i capacitats de generalització, tal com es mostra a la Taula 7.

| Model   | MAPE          | $R^2$         |
|---|---------------|---------------|
| <b>Models LSTM</b>                            |               |               |
| LSTM ELU                                      | 1.7424        | <b>0.9683</b> |
| LSTM ELU + Monte Carlo Dropout                | 1.7648        | <b>0.9678</b> |
| LSTM LEAKY                                    | 1.8567        | 0.9647        |
| LSTM RELU                                     | 2.0296        | 0.9591        |
| <b>Models LSTM Bidireccionals</b>             |               |               |
| BI LSTM ELU                                   | 1.7919        | <b>0.9671</b> |
| BI LSTM LEAKY                                 | 1.8283        | 0.9659        |
| BI LSTM RELU                                  | 2.2039        | 0.9550        |
| <b>Models XGBoost</b>                         |               |               |
| XGBoost utilitzant un Ensemble de Monte Carlo | 1.7417        | 0.9665        |
| XGBoost utilitzant un Ensemble de Bootstrap   | 1.8020        | 0.9647        |
| <b>Model Random Forest</b>                    |               |               |
| RandomForestRegressor                         | <b>1.5753</b> | 0.9646        |

TAULA 7: RESULTATS DELS DIFERENTS MODELS SOBRE EL CONJUNT DE TEST.

Tot i que tots els models implementats incorporen mecanismes efectius de regularització, l'**LSTM ELU** amb **Monte Carlo Dropout** ha demostrat un rendiment consistentment robust. Aquest es veu reflectit en la distribució d'errors obtinguda mitjançant el Monte Carlo Dropout i mostrada a la Figura 23, on el LSTM manté una alta resistència davant de variacions en les dades d'entrada.

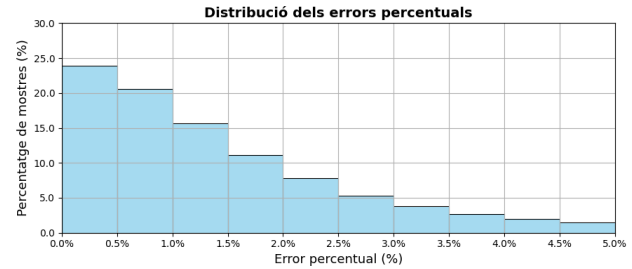


Fig. 23: Distribució de l'error percentual normalitzat en la predicció del model LSTM ELU amb Monte Carlo Dropout.

Malgrat que el **LSTM ELU** amb **Monte Carlo Dropout** destaca per la seva capacitat de modelar el fenomen, cal destacar que el **Random Forest** ha superat en termes de rendiment global com és observat a la Taula 7 i a la comparació de la distribució de l'error percentual 24. Oferint una solució amb una complexitat computacional molt menor.

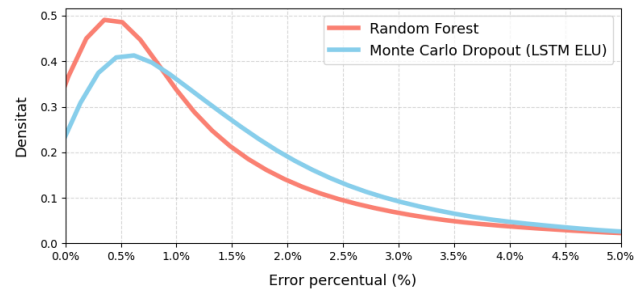


Fig. 24: Comparació de la distribució dels errors percentuals normalitzats entre els models Monte Carlo Dropout (LSTM ELU) i Random Forest.

## AGRAÏMENTS

Vull expressar el meu sincer agraïment a totes les persones que m'han acompanyat i recolzat incondicionalment durant aquest procés, permetent-me dedicar temps de qualitat a la recerca. En especial, agraeixo a Albert Romero Sánchez i Xavier Miquel Armengol per la seva dedicació en introduir-me en aquest àmbit, i reconec especialment l'esforç i guia constant d'Albert Romero durant la tutorització del treball.

## REFERÈNCIES

- [1] A. de Barcelona, “L’observatori de l’energia de barcelona.”
- [2] —, “Consum d’electricitat a barcelona open data.”
- [3] O. Meteo, “Weather data api archive.”
- [4] EPData, “Índex de producció industrial a catalunya (ine).”
- [5] nvidia, “What is xgboost?” 2025.
- [6] P. Banerjee, “Xgboost hyperparameters tuning,” 2020.
- [7] W. contributors, “Nicholas metropolis — wikipedia, the free encyclopedia,” 2025.
- [8] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, “Equation of state calculations by fast computing machines,” Los Alamos Scientific Lab., Los Alamos, NM (United States); Univ. of Chicago, IL (United States), Tech. Rep., 03 1953.
- [9] Wikipedia contributors, “Monte carlo method,” 2024.
- [10] —, “Bootstrap aggregating — Wikipedia, The Free Encyclopedia,” 2024.
- [11] —, “Bootstrapping (statistics) — Wikipedia, The Free Encyclopedia,” 2024.
- [12] Wikipedia, “Xarxa neuronal artificial,” 2025.
- [13] D. Zholtayev, D. Dauletiya, A. Tileukulova, D. Akimbay, M. Nursultan, Y. Bushanov, A. Kuzdeuov, and A. Yeshmukhametov, “Smart pipe inspection robot with in-chassis motor actuation design and integrated ai-powered defect detection system,” *IEEE Access*, vol. 12, pp. 119 520–119 534, 2024.
- [14] GeeksforGeeks, “What is lstm – long short term memory?” 2025.
- [15] S. R. Dubey, S. K. Singh, and B. B. Chaudhuri, “Activation functions in deep learning: A comprehensive survey and benchmark,” *Neurocomputing*, vol. 503, pp. 92–108, 2022.
- [16] K. Team, “Batch normalization layer - keras api documentation,” 2024.
- [17] GeeksforGeeks, “Vanishing and exploding gradients problems in deep learning,” 2020, accessed: 2025-04-30.
- [18] DeepAI, “Batch normalization,” 2024.
- [19] K. Team, “Dropout layer - keras api documentation,” 2024.
- [20] Y. Gal and Z. Ghahramani, “Dropout as a bayesian approximation: Representing model uncertainty in deep learning,” in *International Conference on Machine Learning (ICML)*, 2016, pp. 1050–1059.
- [21] Àrea Metropolitana de Barcelona, “Mapa energètic.”
- [22] G. James, D. Witten, T. Hastie, R. Tibshirani, and J. Taylor, *An Introduction to Statistical Learning: with Applications in Python*. Springer, 2023.
- [23] Laura Melgar García, José Francisco Torres Maldonado, Alicia Troncoso, and José Cristóbal Riquelme Santos, “Técnicas big data para la predicción de la demanda y precio eléctrico,” *Economía Industrial*, 2024.
- [24] N. I. M. S. N. IMS), “The effect of batch size on the generalizability of the convolutional neural networks on a histopathology dataset,” *ScienceDirect*, 2020.
- [25] S. M. Z. Z. Arjun Ghosh, Nanda Dulal Jana, “Designing optimal convolutional neural network architecture using differential evolution algorithm,” *CellPress*, 2022.
- [26] Keras Documentation, “Reduce learning rate on plateau callback,” 2025.
- [27] R. Tavenard, *Deep Learning Basics (lecture notes)*, 2023.
- [28] P. Li, “Optimization algorithms for deep learning,” 2017.
- [29] S. M. Z. Z. Arjun Ghosh, Nanda Dulal Jana, “Energy consumption prediction using modified deep cnn-bi lstm with attention mechanism,” *Heliyon*, 2025.
- [30] xgboosting, “Estimating the uncertainty or confidence of an xgboost model,” 2024.
- [31] J. Brownlee, “How to tune the number and size of decision trees with xgboost in python,” 2020.
- [32] GeeksforGeeks, “Confidence intervals for xgboost,” 2024.
- [33] M. G.-T. F. M.- A. T. Federico Divina, José Francisco Torres Maldonado, “Spanish short-term electric energy consumption forecasting,” 2020.
- [34] L. Yang, H. Ma, Y. Zhang, and S. Li, “Research on energy consumption prediction of public buildings based on improved support vector machine,” in *2023 35th Chinese Control and Decision Conference (CCDC)*, 2023, pp. 2699–2704.
- [35] J. Li, X. Jiang, L. Shao, H. Liu, C. Chen, G. Wang, and D. Du, “Energy consumption data prediction analysis based on eemd-arma model,” in *2020 IEEE International Conference on Mechatronics and Automation (ICMA)*, 2020, pp. 1338–1342.

- [36] M. A. N. M. Asri, N. Zaini, and M. F. A. Latip, “Development of an lstm-based model for energy consumption prediction with data pre-analysis,” in *2021 11th IEEE International Conference on Control System, Computing and Engineering (ICCSCE)*, 2021, pp. 228–233.
- [37] K. Yang, X. Yang, L. Zhou, L. Li, and X. Wang, “Regional energy prediction model based on cdc model,” in *2022 IEEE 5th Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC)*, vol. 5, 2022, pp. 10–14.
- [38] N. I. N. Komori, N. Zaini, and M. F. A. Latip, “Classification and profiling of electricity consumption patterns using bayesian networks,” in *2022 IEEE Symposium on Industrial Electronics Applications (ISIEA)*, 2022, pp. 1–6.
- [39] J. Brownlee, “Time series forecasting with lstms,” 2020.
- [40] P. Kotschieder, M. Fiterau, A. Criminisi, and S. R. Bulò, “Deep neural decision forests,” in *2015 IEEE International Conference on Computer Vision (ICCV)*, 2015, pp. 1467–1475.
- [41] K. Amasyali and N. M. El-Gohary, “A review of data-driven building energy consumption prediction studies,” *Renewable and Sustainable Energy Reviews*, vol. 81, pp. 1192–1205, 2018.
- [42] F. S. Lars Siemer and D. Kleinhans, “Cost-optimal operation of energy storage units: Benefits of a problem-specific approach,” *Journal of Energy Storage* 6, 2016.
- [43] K. Biel and C. H. Glock, “Systematic literature review of decision support models for energy-efficient production planning,” *Computers Industrial Engineering*, vol. 101, pp. 243–259, 2016.
- [44] Y. Chen, B. shen, A. Ali, and S. reyes, “Econometric analysis of factors influencing electricity consumption in spain: Implications for policy and pricing strategies,” *Heliyon*, vol. 10, no. 17, p. e36217, 2024.
- [45] AFAcoding, “Energy demand prediction for neighborhoods in barcelona for sustainable resource management,” 2025. [Online]. Available: <https://github.com/AFAcoding/Energy-Demand-Prediction>
- [46] S. Raschka, “mlxtend: Machine Learning Extensions,” <https://rasbt.github.io/mlxtend/>, 2025.

## APÈNDIX

### A.1 Codi

Per a més detalls sobre el procés d’anàlisi, el repositori de GitHub [45] proporciona el codi complet.

### A.2 Pseudocodi dels diferents algoritmes

---

#### Algorithm 3 Monte Carlo Ensemble amb XGBoost

---

**Require:** Dades d’entrenament  $(X, y)$ , N de models  $n$

**Ensure:** Llista de models XGBoost entrenats

```

1: Inicialitzar una llista buida models
2: for  $i = 1$  fins a  $n$  do
3:   // Es genera un model XGBoost en cada iteració
4:   Fixar la llavor aleatòria a  $i$  // Garantim diversitat
5:   Inicialitzar un model XGBoost.
6:   Entrenar el model amb  $(X, y)$ 
7:   Afegir el model entrenat a la llista models
8: end for
9: return models = 0

```

---



---

#### Algorithm 4 Bootstrap Ensemble amb XGBoost

---

**Require:** Dades d’entrenament  $(X, y)$ , N de models  $n$

**Ensure:** Llista de models XGBoost entrenats amb mostres bootstrap

```

1: Inicialitzar una llista buida models
2: for  $i = 1$  fins a  $n$  do
3:   // Es genera un model XGBoost per cada mostra
4:   Generar una mostra aleatòria amb reemplaçament dels índexs de  $X$  (bootstrap)
5:   Construir  $X_{boot}, y_{boot}$  amb les mostres seleccionades
6:   Inicialitzar un model XGBoost.
7:   Entrenar el model amb  $(X_{boot}, y_{boot})$ 
8:   Afegir el model entrenat a la llista models
9: end for
10: return models = 0

```

---



---

#### Algorithm 5 Monte Carlo Dropout amb xarxes neuronals

---

**Require:** Model neuronal amb Dropout (model), dades d’entrada  $X$ , nombre de simulacions  $n$

**Ensure:** Mitjana i desviació estàndard de les prediccions

```

1: Inicialitzar una llista buida f_preds
2: for  $i = 1$  fins a  $n$  do
3:   // Es repeteix la inferència amb Dropout actiu.
4:   Generar prediccions del model: preds = model(X, training=True)
5:   Convertir preds a numpy i afegir a f_preds
6: end for
7: Convertir f_preds a una matriu numpy
8: Calcular la mitjana de les prediccions per mostra
9: Calcular la desviació estàndard per mostra
10: return mean_preds, std_preds = 0

```

---