








Serii de timp

Capitolul 11: Foundation models pentru serii de timp



Daniel Traian PELE

Academia de Studii Economice din București
IDA Institute Digital Assets
Blockchain Research Center
AI4EFin Artificial Intelligence for Energy Finance
Academia Română, Institutul de Prognoză Economică
MSCA Digital Finance

Acronime folosite în acest material (1/2)

- **AI**: Artificial Intelligence (inteligență artificială)
- **AIC**: Akaike Information Criterion (criteriul informațional Akaike)
- **API**: Application Programming Interface (interfață de programare a aplicațiilor)
- **AR**: AutoRegressive (model) — model autoregresiv
- **ARIMA**: AutoRegressive Integrated Moving Average (model) — model autoregresiv integrat cu medie mobilă
- **BNR**: Banca Națională a României
- **BY-NC**: Attribution–NonCommercial (Creative Commons licence terms) — atribuire, utilizare necomercială (condițiile licenței Creative Commons)
- **CPU**: Central Processing Unit (procesorul central)
- **CRPS**: Continuous Ranked Probability Score (scorul de probabilitate continuu pe ranguri)
- **ENTSO-E**: European Network of Transmission System Operators for Electricity (rețeaua europeană a operatorilor de transport și de sistem pentru energie electrică)
- **ETS**: Error, Trend, Seasonality (exponential smoothing state space models) — eroare, trend, sezonalitate (modele de netezire exponențială)
- **FRED**: Federal Reserve Economic Data (baza de date economice a Rezervei Federale din St. Louis)

Acronime folosite în acest material (2/2)

- **GARCH**: Generalised AutoRegressive Conditional Heteroskedasticity (heteroscedasticitate condiționată autoregresivă generalizată)
- **GIFT**: General Time Series Forecasting Model Evaluation (GIFT-Eval benchmark) — benchmark-ul general de evaluare a modelelor de prognoză (GIFT-Eval)
- **GPT**: Generative Pre-trained Transformer (Transformer generativ pre-antrenat)
- **GPU**: Graphics Processing Unit (procesorul grafic)
- **IAPC**: Indicele armonizat al prețurilor de consum
- **INS**: Institutul Național de Statistică
- **LLM**: Large Language Model (model lingvistic de mari dimensiuni)
- **LOTSA**: Large-scale Open Time Series Archive (Moirai pretraining corpus) — arhiva deschisă de serii de timp pe care a fost pre-antrenat Moirai
- **MAE**: Mean Absolute Error (eroarea absolută medie)
- **MASE**: Mean Absolute Scaled Error (eroarea absolută medie scalată)
- **PIB**: Produsul intern brut
- **RMSE**: Root Mean Squared Error (rădăcina erorii pătratice medii)
- **SUA**: Statele Unite ale Americii
- **VAR**: Vector AutoRegression (model vector autoregresiv)
- **WQL**: Weighted Quantile Loss (pierderea cuantilă ponderată)

Întrebarea de azi și traseul

- **Întrebarea**: poate un singur model mare, pre-antrenat pe multe serii, să prognozeze o serie pe care nu a văzut-o la fel de bine ca un model construit pentru acea serie?
  - până acum (capitolele 0–10): un model pentru fiecare serie, estimat pe acea serie (ETS, ARIMA, GARCH, VAR)
  - Capitolul 9: modele globale de învățare automată, antrenate pe seriile unui singur set de date
- **Traseul** capitolului (studiu individual)
  - ideea de foundation model; Transformer-ul și atenția, pe scurt
  - de la numere la tokeni: scalare, cuantizare, patching; principalele familii de modele
  - prognozele probabiliste și scorurile lor; un test corect pe serii din România și din SUA
  - leakage-ul și contaminarea; situațiile în care aceste modele ajută și cele în care nu ajută

Rezultatele învățării

- Definiți un foundation model, prognoza zero-shot și fine-tuning-ul
- Explicați cum devine o serie intrarea unui Transformer: scalarea prin medie, cuantizarea, patching-ul
- Descrieți Chronos, TimesFM, Moirai, Lag-Llama și TimeGPT și spuneți prin ce diferă
- Evaluați o prognoză cuantilă cu pierderea pinball, CRPS, MASE și acoperirea unui interval
- Rulați un model deschis pe un CPU, comparați-l corect cu prognoza sezonieră naivă, ETS și ARIMA și recunoașteți contaminarea

Noțiuni necesare azi

- Reperele de prognoză și evaluarea lor (capitolele 0 și 4)
  - prognoza sezonieră naivă $\hat{y}_{T+h} = y_{T+h-m}$: prognoza repetă valoarea din același sezon al ciclului anterior
  - T: ultima perioadă observată; h: orizontul; m: lungimea sezonului (12 pentru date lunare, 168 pentru date orare cu ciclu săptămînal); notația $\hat{}$ indică o prognoză
  - ETS; validarea încrucișată cu origine mobilă; MAE, RMSE, MASE; testul Diebold–Mariano
- ARIMA și ARIMA sezonier (capitolele 3 și 4); noțiuni de bază de învățare automată (Capitolul 9)
  - eșantioanele de antrenare, validare și test; o rețea neuronală ca funcție flexibilă cu mulți parametri
  - modelele globale: un model antrenat pe multe serii
- Cuantilele: q_τ este valoarea sub care variabila se află cu probabilitatea τ
  - $\tau \in (0, 1)$: nivelul; $q_{\mathbf{0,5}}$ este mediana, $q_{\mathbf{0,1}}$ și $q_{\mathbf{0,9}}$ delimitează un interval central de 80%

Trei moduri de a prognoza multe serii

- Modele **locale**
  - cîte un model pentru fiecare serie, estimat doar pe acea serie
  - ETS, ARIMA: puțini parametri, interpretabile; 10 000 de serii înseamnă 10 000 de estimări
- Modele **globale** (Capitolul 9)
  - un model antrenat pe toate seriile unui set de date
  - exemplu: DeepAR (Salinas et al. (2020)), o rețea recurentă antrenată pe toate produsele unui comerciant
- Modele de tip **foundation**
  - un model pre-antrenat o singură dată pe un corpus foarte larg de serii din multe domenii
  - folosit apoi pe serii noi și seturi de date noi, fără reestimare
  - termenul provine de la Bommasani et al. (2021): modele antrenate pe date largi, la scară mare, adaptabile la multe sarcini

Zero-shot, fine-tuning și prognoza din context

- **Contextul**
  - istoricul recent $y_{T-C+1}, \ldots, y_T$ transmis modelului în momentul prognozei (C = lungimea contextului)
- Prognoza **zero-shot**
  - ponderile pre-antrenate rămîn fixe
  - se folosește doar contextul seriei noi
  - niciun parametru nu se estimează pe seria-țintă: prognoza durează o fracțiune de secundă
- **Fine-tuning**
  - antrenarea suplimentară a ponderilor pre-antrenate pe datele-țintă
  - poate ajuta la serii neobișnuite; cere un eșantion de validare și mai mult calcul
- Rezultatul este de obicei o mulțime de **cuantile** $\hat{q}_{0,1}; \ldots; \hat{q}_{0,9}$ pentru fiecare pas viitor, nu un singur număr

Pre-antrenarea la scară mare



Un centru de date Google, The Dalles, Oregon

Foto: Visitor7 (2011); CC BY-SA 3.0; Wikimedia Commons

- Corpusurile de pre-antrenare amestecă serii publice din multe domenii
  - energie, transport, comerț, vreme, trafic web, finanțe și economie
  - plus serii sintetice: amestecuri aleatoare de serii reale și serii generate din procese gaussiene (Chronos)
- Scara: TimesFM a fost pre-antrenat pe circa 10^{11} puncte (Das et al. (2024))
- Pre-antrenarea se face o singură dată, de dezvoltator, pe multe procesoare grafice (GPU)
  - utilizatorul doar rulează modelul: versiunile mici rulează pe un CPU obișnuit

Recapitulare: ideea de foundation model

- Modelele locale, globale și de tip foundation diferă prin datele pe care se antrenează
- Zero-shot: ponderi fixe, doar contextul seriei noi
- Rezultatul este o mulțime de cuantile: o prognoză probabilistă

Transformer-ul (Vaswani et al., 2017)

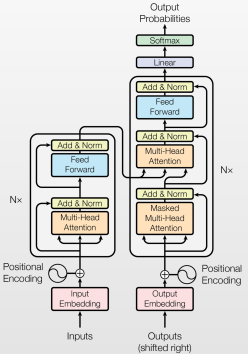


Diagrama originală: encoder (stînga), decoder (dreapta)

Figura: Vaswani et al. (2017), Google; CC BY-SA 4.0; Wikimedia Commons

- Vaswani et al. (2017): o rețea pentru secvențe construită pe **atenție**, fără recurență
  - creată pentru traducere; stă la baza majorității modelelor mari de limbaj (LLM)
- Intrarea: o secvență de **tokeni**, fiecare transformat într-un vector (embedding)
- Arhitectura **encoder–decoder**
  - **encoder-ul** citește toată intrarea
  - **decoder-ul** produce ieșirea pas cu pas
  - modelele **doar cu decoder** (GPT, TimesFM, Lag-Llama) prognozează tokenul următor din toți cei anteriori
- Fiecărui vector i se adaugă informația despre poziție, deoarece atenția ignoră ordinea

Self-attention (1/2): formula

- Fiecare vector de intrare x_i produce trei vectori, prin trei transformări liniare învățate la antrenare
  - o interogare $q_i = W_Q x_i$ (ce caută poziția i), o cheie $k_i = W_K x_i$ (ce oferă poziția i), o valoare $v_i = W_V x_i$ (ce transmite poziția i)
  - W_Q, W_K, W_V: matrice de ponderi învățate la antrenare, aceleași pentru toate pozițiile
- Ieșirea în poziția i este o medie ponderată a valorilor tuturor pozițiilor
  - $\text{out}_i = \sum_j w_{ij} v_j, \qquad w_{ij} = \frac{\exp(q_i^\top k_j / \sqrt{d})}{\sum_l \exp(q_i^\top k_l / \sqrt{d})}$
- Notațiile
  - j, l: indici care parcurg toate pozițiile de intrare; w_{ij}: ponderea pe care poziția i o dă poziției j
  - $q_i^\top k_j$: produsul scalar dintre interogare și cheie, mare cînd cei doi vectori au aceeași direcție (scorul de similaritate)
  - d: lungimea vectorilor-cheie; împărțirea la $\sqrt{d}$ menține scorurile la o mărime moderată
  - exponențiala face toate ponderile pozitive, iar numitorul face ca ponderile poziției i să aibă suma 1

Self-attention (2/2): forma matricială și intuiția

- Forma matricială a aceluiași calcul: $\text{softmax}(QK^\top/\sqrt{d})\,V$
  - Q, K, V: matrice ale căror linii sînt vectorii q_i, k_i, v_i; $QK^\top$ conține toate scorurile $q_i^\top k_j$
  - softmax: aplicată pe fiecare linie, transformă scorurile în ponderile w_{ij} de pe slide-ul anterior
- Intuiția pentru serii: pasul de azi se poate uita la aceeași oră de săptămîna trecută și îi poate da o ponderе mare
  - ponderile sînt învățate din date, nu fixate dinainte, ca lagurile unui AR(p)
- Costul: n intrări cer n^2 comparații
  - un context de 2048 de valori dă circa 4 milioane de comparații în fiecare strat de atenție

Exemplu rezolvat: ponderile atenției

- O interogare și trei chei; scorurile $q^\top k_j/\sqrt{d} = 2, 1, 0$; valorile $v_j = 10, 20, 30$
- Pasul 1: exponențiala: e^2, e^1, e^0
- Pasul 2: împărțirea la sumă: ponderile 0,67; 0,24; 0,09 (au suma 1)
- Pasul 3: ieșirea $= \sum_j w_j v_j = 14{,}2$
  - scorul cel mai mare primește cea mai mare parte din pondere, dar fiecare valoare contribuie
- Un foundation model suprapune multe astfel de straturi, fiecare cu mai multe capete de atenție

Recapitulare: Transformer-ul

- Atenția: fiecare pas este o medie ponderată, învățată, a tuturor pașilor
- Arhitecturi encoder–decoder (Chronos) și arhitecturi doar cu decoder (TimesFM, Lag-Llama)
- Costul crește cu pătratul numărului de intrări

Reprezentarea seriilor numerice ca date de intrare

- Seriile diferă prin unități și niveluri: consumul în GW, un indice în jur de 100, un curs de schimb în jur de 5
  - un singur model pentru toate are nevoie de o scală comună
- Trei soluții folosite în practică
  - **scalare + cuantizare**: fiecare valoare devine un token dintr-un vocabular fix (Chronos)
  - **patching**: grupuri de valori consecutive devin un vector de intrare (TimesFM, Moirai, Chronos-Bolt)
  - **cifrele ca text**: un LLM general citește numerele ca șiruri de caractere (Gruver et al. (2023))
- În toate cazurile, prognozele se transformă înapoi în unitățile inițiale

Scalarea prin medie și cuantizarea (Chronos) (1/2)

- **Scalarea prin medie**
  - fiecare valoare a contextului se împarte la media valorilor absolute ale contextului
  - $\tilde{x}_t = x_t / s, \qquad s = \frac{1}{C} \sum_{t=1}^{C} |x_t|$
  - x_t: a t-a valoare a contextului, $t = 1, \ldots, C$; C: lungimea contextului; s: scala; $\tilde{x}_t$: valoarea scalată
  - contextul scalat are valori de ordinul unității, oricare ar fi unitățile
- **Cuantizarea**
  - fiecare valoare scalată este înlocuită cu numărul unui interval
  - intervalul $[-15, 15]$ este împărțit în 4093 de intervale egale, de lățime 0,0073; tokenul lui $\tilde{x}_t$ este numărul celui mai apropiat centru
  - plus tokeni speciali (completare, sfîrșit de secvență)

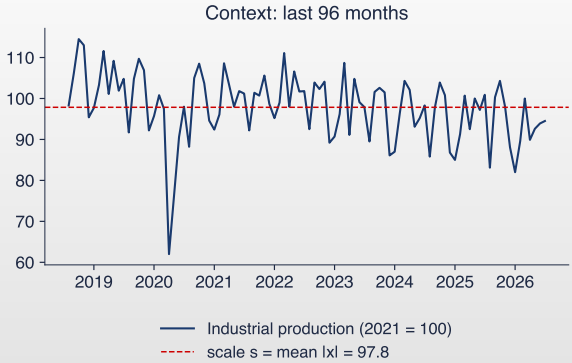
Scalarea prin medie și cuantizarea (Chronos) (2/2)

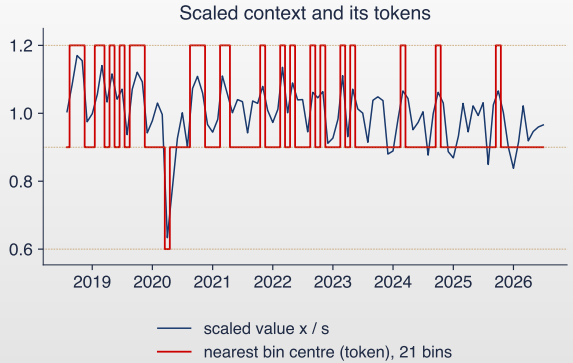
- Antrenarea: un model de limbaj (T5) învață să prognozeze tokenul următor
  - funcția de pierdere: cross-entropy $-\log \hat{p}(z_{t+1})$, unde z_{t+1} este tokenul următor efectiv, iar $\hat{p}(z_{t+1})$ probabilitatea atribuită lui de model
  - este 0 cînd această probabilitate este 1 și crește cînd probabilitatea scade
- Prognoza, în trei pași
  - se eșantionează din model multe traiectorii de tokeni
  - fiecare token se înlocuiește cu centrul intervalului său și se înmulțește cu s
  - se citesc cuantilele valorilor eșantionate la fiecare pas viitor
- Limita: valorile scalate în afara intervalului $[-15, 15]$ nu pot fi produse

Exemplu rezolvat: scalarea și tokenii

- Contextul: rata șomajului din România (%, ajustată sezonier, Eurostat), martie 2026–august 2026
  - $x = 6{,}5;\ 6{,}8;\ 6{,}7;\ 6{,}8;\ 6{,}4;\ 6{,}4$
- Pasul 1: scala $s =$ media lui $|x_t| = 6{,}60$
- Pasul 2: valorile scalate $x_t/s = 0{,}985;\ 1{,}030;\ 1{,}015;\ 1{,}030;\ 0{,}970;\ 0{,}970$
- Pasul 3: cele mai apropiate centre (de la 0 la 4092): tokenii 2180, 2187, 2184, 2187, 2178, 2178
  - valorile egale dau tokeni egali; o variație de 0,1 puncte mută tokenul cu mai multe intervale
- Modelul nu vede niciodată nivelul de 6,6%: doar forma relativă la scală

Scalarea și cuantizarea pe date reale (1/2)





Scalarea și cuantizarea pe date reale (2/2)

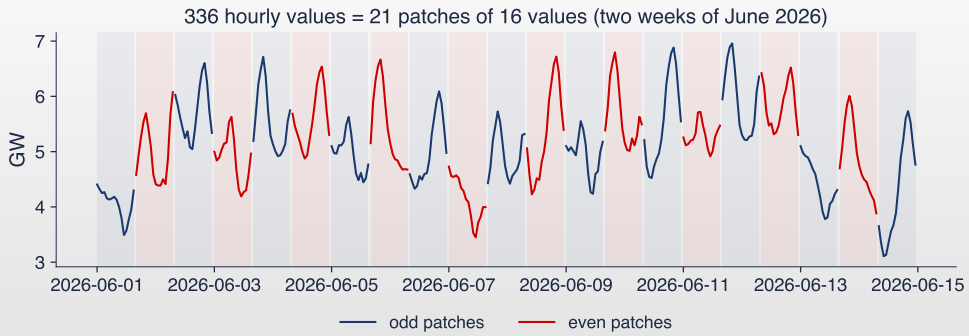
- Producția industrială a României (Eurostat, 2021 = 100), ultimele 96 de luni; scala $s = 97{,}8$
- Dreapta: contextul scalat și tokenii lui pe o grilă grosieră de 21 de intervale (Chronos folosește 4093)
- După scalare, toate valorile sînt între 0,63 și 1,17: cîteva procente din intervalul $[-15, 15]$
- Pe grila grosieră, cele 96 de luni folosesc doar 3 tokeni: căderea din aprilie 2020 (carantina) este un token separat
- Cu 4093 de intervale, grila este suficient de fină pentru a păstra forma sezonieră
- O valoare extremă mărește s și micșorează toate celelalte valori scalate: valorile aberante din context contează

TSA_ch11_tokens_scores

Patching-ul

- **Patch**
  - un bloc de P valori consecutive, transformat într-un vector de intrare de o rețea mică
  - PatchTST (Nie et al. (2023)): „o serie de timp valorează cît 64 de cuvinte”
- Două avantaje
  - mai puține intrări: 2048 de valori cu $P = 16$ dau 128 de patch-uri, deci atenția cere 128^2 în loc de 2048^2 comparații
  - fiecare intrare poartă o formă locală (o pantă, un vîrf), nu un singur număr zgomotos
- Folosit de TimesFM (patch-uri de intrare de 32), Moirai (mărimea patch-ului după frecvență), Chronos-Bolt și Chronos-2 (16)
  - și ieșirea poate fi un patch întreg de valori viitoare, ceea ce este mai rapid decît token cu token

Patching-ul consumului de energie electrică al României



- 336 de valori orare (două săptămîni din iunie 2026, ENTSO-E), împărțite în 21 de patch-uri de cîte 16 valori

TSA_ch11_tokens_scores

Interpretarea patching-ului

- O zi are 24 de ore, deci un patch de 16 acoperă două treimi dintr-o zi: patch-urile nu se aliniază cu ciclul zilnic
- Atenția leagă patch-urile: creșterea de dimineață de luni poate fi pusă în legătură cu cea de lunea trecută
- Un ciclu săptămînal (168 de ore) este vizibil doar dacă contextul acoperă cel puțin o săptămînă: 11 patch-uri

Recapitulare: de la numere la tokeni

- Scalarea prin medie elimină unitățile; cuantizarea transformă valorile în tokeni (Chronos)
- Patching-ul grupează valorile: secvențe mai scurte, forme locale
- Prognoza se transformă înapoi în unitățile inițiale

Chronos, Chronos-Bolt și Chronos-2 (Amazon)



Amazon Spheres, Seattle

Foto: Andy Li (2018); CC0; Wikimedia Commons

- **Chronos** (Ansari et al. (2024))
  - T5 encoder–decoder pe valori scalate și cuantizate
  - eșantionează traiectorii de tokeni
- **Chronos-Bolt** (noiembrie 2024)
  - patch-uri de 16 valori și ieșirea directă a cuantilelor 0,1; . . . ; 0,9
  - dimensiunile folosite aici: tiny, 9 milioane, și small, 48 de milioane de parametri; context de pînă la 2048 de valori
- **Chronos-2** (Ansari et al. (2025), octombrie 2025)
  - 119 milioane de parametri, 21 de niveluri de cuantile, covariate și grupuri de serii înrudite
- Ponderi deschise (licența Apache-2.0); pachetul Python `chronos-forecasting`; toate rulează pe un CPU

TimesFM (Google) și Lag-Llama

- **TimesFM** (Das et al. (2024))
  - Transformer doar cu decoder, patch-uri de intrare de 32 și de ieșire de 128 de valori
  - pre-antrenat pe serii reale (Google Trends, accesări de pagini Wikipedia) și sintetice; circa 200 de milioane de parametri
  - ponderi deschise (Apache-2.0); versiunile ulterioare adaugă ieșiri cuantile
- **Lag-Llama** (Rasul et al. (2023))
  - doar cu decoder, construit pe arhitectura Llama a modelelor de limbaj
  - intrările: valorile seriei la laguri alese după frecvență (de exemplu 7 zile, 12 luni, 24 de ore), plus variabile de calendar
  - ieșirea: parametrii unei distribuții Student-t pentru pasul următor; eșantionarea dă traiectorii
  - mic și deschis (Apache-2.0); gîndit pentru fine-tuning

Moirai (Salesforce) și TimeGPT



Salesforce Tower, San Francisco

Foto: Saggittarius A (2020); CC BY-SA 4.0; Wikimedia Commons

- **Moirai** (Woo et al. (2024))
  - model de prognoză „universal”
  - orice număr de variabile într-o singură secvență; mărimea patch-ului aleasă după frecvență
  - ieșirea: un amestec de distribuții; corpusul LOTSA, circa 27 de miliarde de observații
  - ponderi deschise, cu licență necomercială (CC BY-NC 4.0)
- **TimeGPT** (Garza et al. (2023))
  - model închis, disponibil doar printr-un API cu plată
  - ponderile și datele de antrenare nu pot fi inspectate; versiunea se poate schimba; datele pleacă de pe calculatorul dumneavoastră
  - nu îl folosim în acest curs: rezultatele nu ar putea fi reproduse

Comparația familiilor de modele

Model	Arhitectura	Intrarea	Ieșirea	Acces
Chronos (2024)	encoder–decoder (T5)	tokeni (intervale)	traiectorii eșantionate	Apache-2.0
Chronos-Bolt (2024)	encoder–decoder	patch-uri de 16	9 cuantile	Apache-2.0
Chronos-2 (2025)	encoder	patch-uri + covariate	21 de cuantile	Apache-2.0
TimesFM (2024)	decoder	patch-uri de 32	patch-uri de 128	Apache-2.0
Moirai (2024)	encoder	patch-uri, orice variabile	amestec	CC BY-NC 4.0
Lag-Llama (2023)	decoder (Llama)	valori la laguri fixe	Student-t	Apache-2.0
TimeGPT (2023)	encoder–decoder (puține detalii)	–	puncte, intervale	API cu plată

- O altă direcție dă cifrele unor LLM-uri generale (Gruver et al. (2023)); Tan et al. (2024) arată că partea de model de limbaj adaugă cost, dar nu și precizie

Recapitulare: familiile de modele

- Chronos: tokeni; Chronos-Bolt și Chronos-2: patch-uri și cuantile directe
- TimesFM și Lag-Llama: doar cu decoder; Moirai: orice număr de variabile
- Ponderile deschise fac rezultatele reproductibile; un API cu plată, nu

Prognozele cuantile

- O **prognoză probabilistă** dă distribuția lui y_{T+h}, condiționată de datele pînă la T
  - foundation models o raportează prin cuantile $\hat{q}_\tau$ la nivelurile $\tau = 0{,}1; \ldots; 0{,}9$
- Prognoza punctuală: mediana $\hat{q}_{\mathbf{0{,}5}}$
- Intervalul central de 80%: $[\hat{q}_{\mathbf{0{,}1}}, \hat{q}_{\mathbf{0{,}9}}]$; un **fan chart** desenează mai multe astfel de benzi
- Două calități de verificat
  - **calibrarea**: intervalul de 80% conține circa 80% din valorile observate
  - **finețea** (sharpness): dintre prognozele calibrate, cea mai îngustă este mai bună (Gneiting și Raftery (2007))
- Pentru ETS și ARIMA folosim cuantilele distribuției Normale, $\hat{q}_\tau = \hat{y}_{T+h} + z_\tau \hat{\sigma}_h$
  - z_τ: cuantila de nivel τ a distribuției Normale standard ($z_{\mathbf{0{,}9}} = 1{,}28$); $\hat{\sigma}_h$: abaterea standard estimată a erorii de prognoză la orizontul h

Pierderea pinball

- Pierderea **pinball** (cuantilă) evaluează o cuantilă prognozată $\hat{q}_\tau$ față de valoarea observată y
  - $L_\tau(y, \hat{q}_\tau) = \tau\,(y - \hat{q}_\tau)$ dacă $y \geq \hat{q}_\tau$ și $(1-\tau)(\hat{q}_\tau - y)$ altfel
- Notațiile
  - τ: nivelul cuantilei; $\hat{q}_\tau$: cuantila prognozată; y: valoarea observată ulterior
- Interpretarea
  - pierderea este 0 cînd $\hat{q}_\tau = y$ și crește liniar cu distanța dintre ele
  - o valoare observată peste cuantilă primește ponderea τ, una sub cuantilă, ponderea $1-\tau$
  - valoarea ei așteptată este minimă în cuantila adevărată: raportarea cuantilei adevărate este strategia optimă

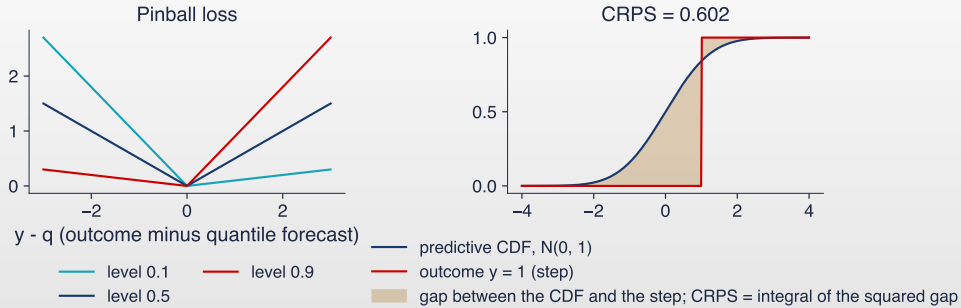
Scorul CRPS

- **CRPS** (continuous ranked probability score, scorul de probabilitate continuu pe ranguri, Matheson și Winkler (1976)) evaluează o întreagă distribuție prognozată față de valoarea observată y
  - $\text{CRPS}(F, y) = \int (F(z) - \mathbf{1}\{z \geq y\})^{\mathbf{2}}\, dz = 2 \int_{\mathbf{0}}^{\mathbf{1}} L_\tau(y, F^{-\mathbf{1}}(\tau))\, d\tau$
- Notațiile
  - F: funcția de repartiție prognozată; z: variabila de integrare, care parcurge toate valorile posibile
  - $\mathbf{1}\{z \geq y\}$: indicatorul, 1 dacă $z \geq y$ și 0 altfel (funcția de repartiție a unei prognoze care pune toată masa în y)
  - $F^{-\mathbf{1}}(\tau)$: cuantila de nivel τ a lui F; a doua formă este pierderea pinball mediată pe toate nivelurile
- Interpretarea
  - CRPS ≥ 0, în unitățile lui y; 0 doar pentru o prognoză care pune toată masa în valoarea observată; mai mic înseamnă mai bine
  - pentru o prognoză punctuală este $|y - \hat{y}|$: CRPS generalizează MAE

Pierderea cuantilă ponderată (WQL)

- **WQL** (weighted quantile loss, pierderea cuantilă ponderată)
  - CRPS aproximat cu cele 9 niveluri 0,1; ...; 0,9 și împărțit la mărimea seriei
  - $\text{WQL} = \frac{1}{9} \sum_{\tau} \frac{2 \sum_t L_\tau(y_t, \hat{q}_{\tau,t})}{\sum_t |y_t|}$
- Notațiile
  - τ: parcurge cele 9 niveluri; t: parcurge toți pașii de prognoză de la toate originile
  - $\hat{q}_{\tau,t}$: cuantila de nivel τ prognozată pentru pasul t; y_t: valoarea observată la pasul t
- Interpretarea
  - numitorul $\sum_t |y_t|$ elimină unitățile: se pot compara serii cu niveluri diferite
  - 0 pentru o prognoză perfectă; mai mic înseamnă mai bine; folosit de Chronos și de GIFT-Eval

Pierderea pinball și CRPS



- Stînga: L_τ în funcție de $y - \hat{q}$ pentru $\tau = 0{,}1;\ 0{,}5;\ 0{,}9$. Dreapta: prognoza N(0, 1), valoarea observată $y = 1$
- $\mathrm{N}(\mu,\ \sigma^2)$: distribuția Normală cu media μ și varianța σ^2

TSA_ch11_tokens_scores

Interpretarea scorurilor

- La $\tau = 0{,}9$, o valoare peste cuantilă costă de 9 ori mai mult decît una sub ea: cuantila de 0,9 ar trebui depășită doar în 10% din cazuri
- La $\tau = 0{,}5$, pierderea este jumătate din eroarea absolută: mediana este evaluată ca o prognoză punctuală
- CRPS pentru N(0, 1): 0,234 dacă $y = 0$, 0,602 dacă $y = 1$; N(0, 4) în $y = 1$ dă 0,663: o bandă mai largă este penalizată chiar dacă acoperă valoarea
- CRPS răsplătește în același timp calibrarea și finețea

Exemplu rezolvat: pierderile pinball

- Prognoza ratei șomajului pentru luna viitoare: $\hat{q}_{0,1} = 6{,}5$, $\hat{q}_{0,5} = 7{,}0$, $\hat{q}_{0,9} = 7{,}6$; valoarea observată $y = 7{,}2$
- $\tau = 0{,}1$: y este deasupra, pierderea $0{,}1 \cdot (7{,}2 - 6{,}5) = 0{,}07$
- $\tau = 0{,}5$: y este deasupra, pierderea $0{,}5 \cdot (7{,}2 - 7{,}0) = 0{,}10$
- $\tau = 0{,}9$: y este dedesubt, pierderea $(1 - 0{,}9) \cdot (7{,}6 - 7{,}2) = 0{,}04$
- CRPS aproximat cu aceste trei niveluri: $\frac{2}{3}$ (suma pierderilor) $= 0{,}140$ puncte procentuale
  - valoarea observată este în intervalul de 80% [6,5; 7,6]: în scorul de acoperire (două slide-uri mai departe) contează ca acoperită

Precizia punctuală: MASE

- **MASE** (mean absolute scaled error, eroarea absolută medie scalată, Hyndman și Koehler (2006))
  - MAE al prognozei împărțit la MAE în eșantion al prognozei sezoniere naive
  - $\text{MASE} = \dfrac{\frac{1}{h}\sum_{j=1}^{h} |y_{T+j} - \hat{y}_{T+j}|}{\frac{1}{T-m}\sum_{t=m+1}^{T} |y_t - y_{t-m}|}$
- Notațiile
  - numărătorul: eroarea absolută medie a prognozelor $\hat{y}_{T+j}$ pe cei h pași de după originea T
  - numitorul: eroarea absolută medie a prognozei sezoniere naive y_{t-m} în interiorul eșantionului, $t = m+1, \ldots, T$; m: lungimea sezonului
- Interpretarea
  - sub 1: mai bun decît prognoza sezonieră naivă în eșantion; peste 1: mai slab; fără unități
  - exemplu: MAE 0,30, MAE naiv în eșantion 0,25, deci MASE = 1,2

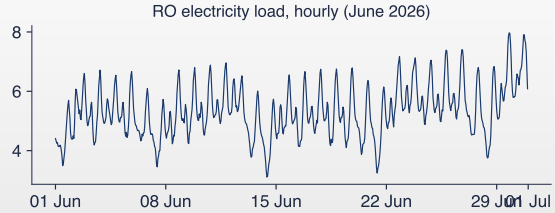
Acoperirea și scorurile relative

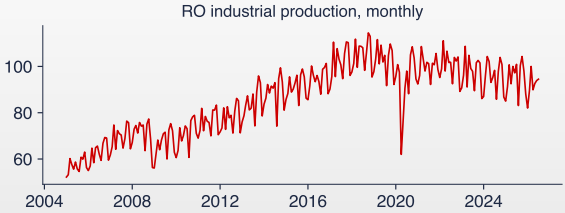
- **Acoperirea**
  - proporția valorilor observate din intervalul de 80%
  - Cov $= \frac{1}{n}\sum_{i=1}^{n} \mathbf{1}\{\hat{q}_{0,1}^{(i)} \leq y_i \leq \hat{q}_{0,9}^{(i)}\}$; n: numărul de prognoze; $\hat{q}_{\tau}^{(i)}$: cuantila prognozei i; $\mathbf{1}\{\cdot\}$: 1 dacă valoarea observată este în interval, 0 altfel
  - ținta este 80%; 7 din 10 în interval înseamnă 70%: intervale puțin prea înguste
- **Scorurile relative**
  - MASE sau WQL al unui model împărțit la cel al prognozei sezoniere naive, pe aceleași origini
  - sub 1: modelul este mai bun decît prognoza sezonieră naivă
  - mediate pe cele S serii cu media geometrică $(r_1 r_2 \cdots r_S)^{1/S}$, r_k: raportul seriei k; astfel 2 și 0,5 se compensează

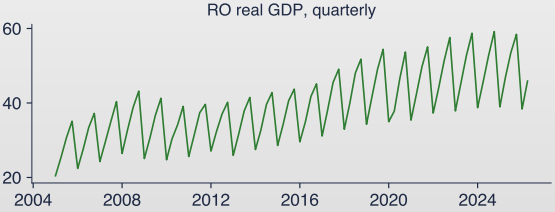
Recapitulare: scorurile

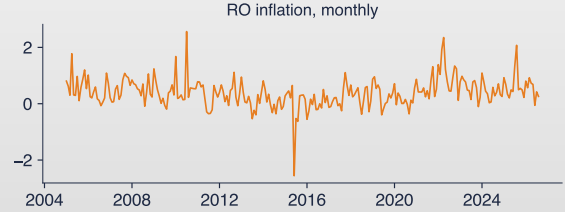
- Pierderea pinball pentru fiecare cuantilă; CRPS pentru întreaga distribuție; WQL ca aproximare scalată
- MASE pentru mediană; acoperirea pentru calibrare
- Raportați scorurile relativ la prognoza sezonieră naivă

Șapte serii pentru test (1/3)

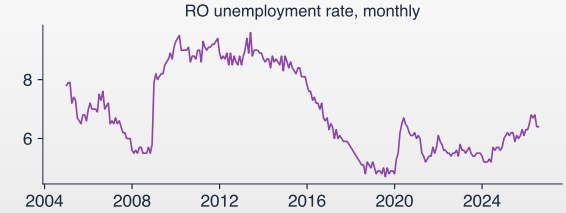


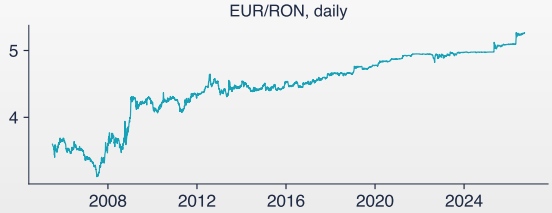


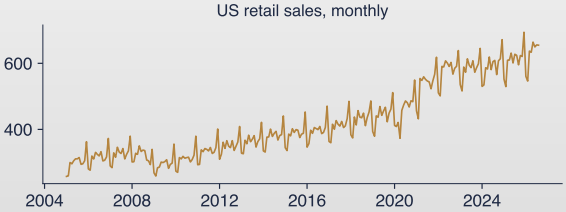




Șapte serii pentru test (2/3)





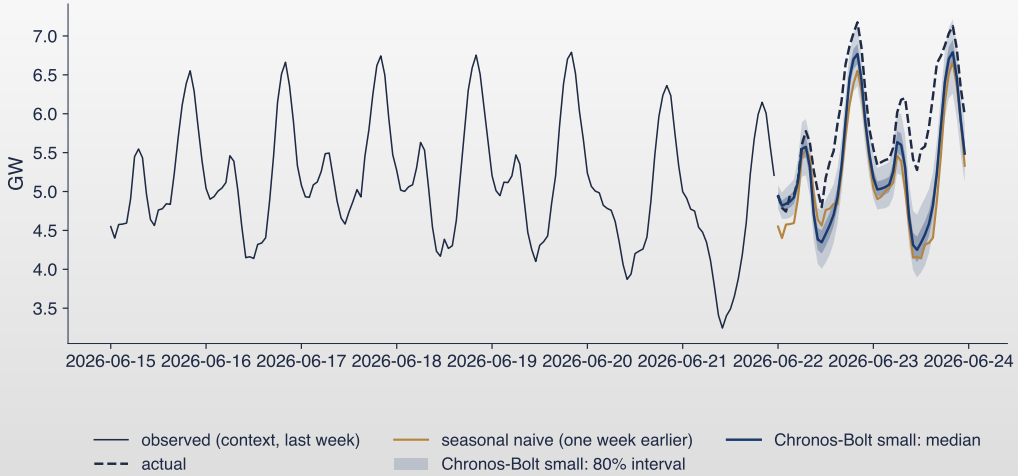


Șapte serii pentru test (3/3)

- România (ENTSO-E)
  - consumul orar de energie electrică
- România (Eurostat)
  - producția industrială
  - PIB-ul real, neajustat sezonier
  - inflația IAPC; rata șomajului
- EUR/RON (cursul de referință BNR)
- Vînzările cu amănuntul din SUA (FRED RSXFSN)
- Sezonalitate puternică: consumul (cicluri zilnice și săptămînale), producția industrială și vînzările din SUA (anuale), PIB-ul (trimestrial)
- Aproape de un mers aleator: EUR/RON și rata șomajului; zgomotoasă și cu memorie scurtă: inflația lunară
- Rupturi: pandemia din 2020 (producția industrială, PIB-ul, vînzările); creșterea inflației din 2022
- Lungimi de la 126 de trimestre (PIB) la 39 408 ore (consumul): cantități foarte diferite de context

TSA_ch11_tokens_scores

Consumul de energie electrică al României, prognoză pe 48 de ore (1/2)



Consumul de energie electrică al României, prognoză pe 48 de ore (2/2)

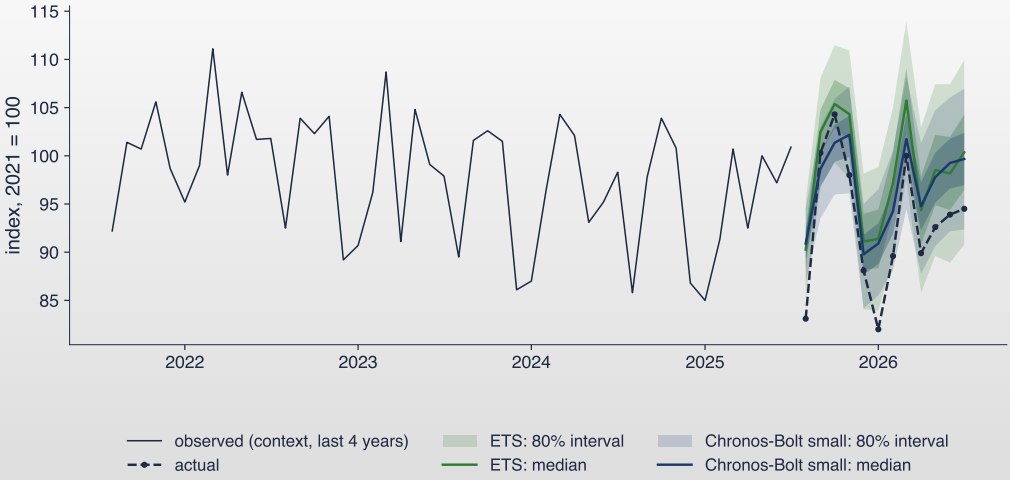
- Chronos-Bolt small, zero-shot, context de 2048 de ore (85 de zile); originea 22 iunie 2026, ora 00:00; benzi: intervalele centrale de 50% și 80%
- Modelul reproduce forma zilnică (minimul de noapte, vîrful de seară) fără nicio estimare pe această serie
- MASE: Chronos-Bolt 1,31, prognoza sezonieră naivă 1,61: mai bun decît repetarea săptămînii trecute
- Dar consumul efectiv rămîne peste prognoză toată ziua: doar 29% din cele 48 de ore cad în banda de 80%
- Modelul nu primește temperatura sau calendarul: o schimbare a vremii pe care ultimele 85 de zile nu o anunță nu poate fi anticipată
- O singură origine dovedește puțin: urmează testul sistematic

TSA_ch11_zero_shot

Exemplu rezolvat: evaluarea unei ore

- Ora 24 a prognozei consumului (22 iunie 2026, ora 23:00): $\hat{q}_{0,1} = 5{,}1837$, $\hat{q}_{0,5} = 5{,}4763$, $\hat{q}_{0,9} = 5{,}7758$ GW; valoarea observată 5,8033 GW
- Toate cele trei cuantile sînt sub valoarea observată, deci fiecare pierdere este $\tau\,(y - \hat{q}_\tau)$
  - $\tau = 0{,}1$: 0,062; $\tau = 0{,}5$: 0,163; $\tau = 0{,}9$: 0,025 GW
- Valoarea observată este peste $\hat{q}_{0,9}$: indicatorul de acoperire este 0 pentru această oră
- Mediate pe multe ore și origini, aceste pierderi dau WQL din secțiunea următoare

Producția industrială a României, prognoză pe 12 luni (1/2)

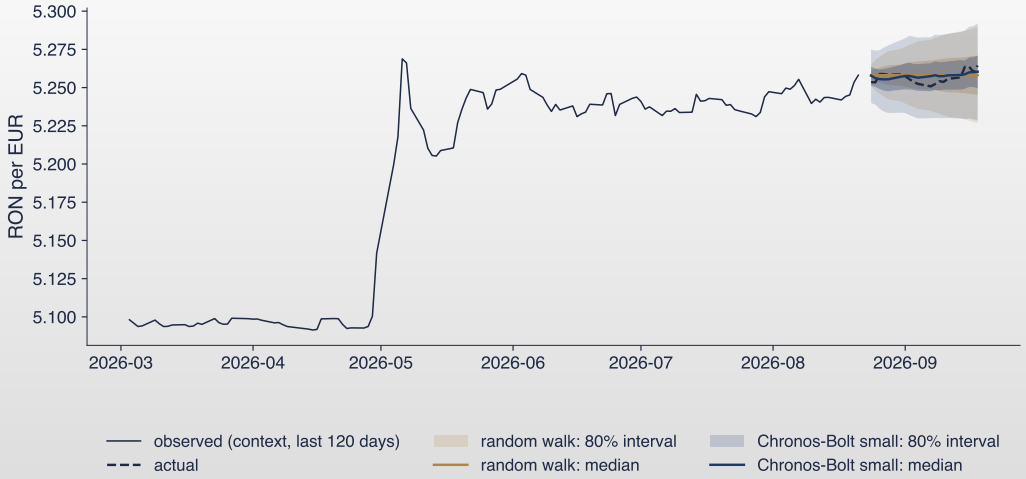


Producția industrială a României, prognoză pe 12 luni (2/2)

- Originea august 2025, prognoze pînă în iulie 2026; Chronos-Bolt small (zero-shot) și ETS (estimat pe serie), ambele cu benzi de 80%
- Ambele modele copiază tiparul sezonier (scăderile din august și decembrie) și păstrează nivelul recent
- MASE: Chronos-Bolt 1,04, ETS 1,20, prognoza sezonieră naivă 0,67: pentru această origine cîștigă cea mai simplă prognoză
- Producția slabă din 2026 nu este anunțată de context; benzile de 80% acoperă 83% (Chronos-Bolt) și 83% (ETS) din luni
- Benzile Chronos-Bolt sînt mai înguste decît cele ale ETS la orizonturi lungi

TSA_ch11_zero_shot

EUR/RON, prognoză pe 20 de zile (1/2)



EUR/RON, prognoză pe 20 de zile (2/2)

- Cursul de referință BNR; originea 24 august 2026, ultima valoare dinaintea ei 5,2581; Chronos-Bolt small comparat cu mersul aleator cu benzi Normale
- Mediana după 20 de zile este 5,2604, practic ultima valoare, 5,2581: modelul a învățat să se comporte ca un mers aleator
- Lățimea benzii de 80% la 20 de zile: 0,064 (Chronos-Bolt) și 0,063 (mersul aleator): aproape aceeași incertitudine
- MAE pe cele 20 de zile: 0,0032 față de 0,0035: practic egalitate
- Un model nu poate extrage ce trecutul nu conține: cursurile de schimb rămîn aproape imprevizibile (Capitolul 1)

TSA_ch11_zero_shot

Recapitulare: prognozele zero-shot

- Fără estimare, modelul copiază ciclurile și nivelurile din context
- Nu poate anticipa ce contextul nu arată (căldura, o recesiune)
- Pentru un mers aleator, prognoza lui coincide practic cu cea a mersului aleator

Protocolul testului

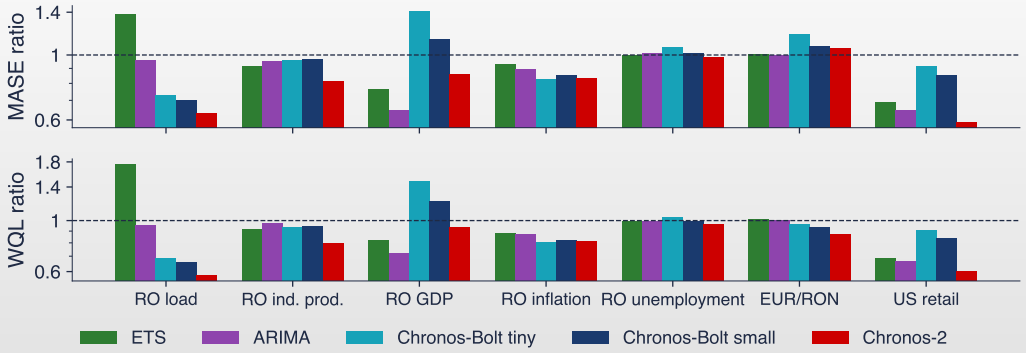
- **Origine mobilă** (Capitolul 4)
  - la fiecare origine, fiecare model prognozează h pași doar din datele pînă la origine
  - consumul: $h = 48$ de ore, origini săptămînale din iulie 2025 pînă în iunie 2026; seriile lunare: $h = 12$, origini lunare din 2016; PIB-ul: $h = 4$, din 2012; EUR/RON: $h = 20$ de zile, la fiecare 20 de zile din 2016
  - 708 origini în total; contextul: tot trecutul, cel mult 2048 de observații
- **Reperele**, reestimate la fiecare origine
  - prognoza sezonieră naivă (prognoza naivă pentru șomaj și EUR/RON)
  - ETS cu trend amortizat și sezonalitate aditivă; ARIMA cu ordinele alese o singură dată după AIC, înainte de prima origine (consumul: ARIMA(2,0,2) pe diferențele săptămînale)
- **Foundation models**, zero-shot
  - Chronos-Bolt tiny și small, Chronos-2
- Aceleași origini, același context, același orizont, aceleași scoruri: MASE, WQL, acoperirea

Rezultate: WQL relativ la prognoza sezonieră naivă

Seria	Origini	ETS	ARIMA	Bolt tiny	Bolt small	Chronos-2
Consumul RO (orar)	52	1,75	0,96	0,69	0,66	0,58
Producția industrială RO	116	0,92	0,97	0,94	0,95	0,79
PIB-ul real RO	55	0,82	0,72	1,48	1,22	0,94
Inflația RO	117	0,88	0,87	0,81	0,82	0,81
Șomajul RO	117	1,00	0,99	1,03	0,99	0,96
EUR/RON (zilnic)	134	1,01	1,01	0,96	0,93	0,87
Vînzările cu amănuntul SUA	117	0,69	0,66	0,90	0,84	0,60
Media geometrică		0,97	0,87	0,95	0,90	0,78

- Sub 1: mai bun decît prognoza sezonieră naivă; fiecare valoare este WQL al modelului împărțit la cel al prognozei sezoniere naive pe aceleași origini

MASE și WQL pe serii și modele



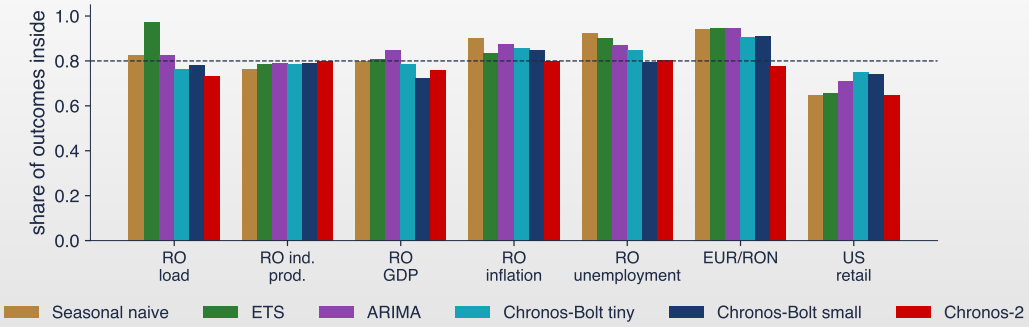
- Rapoarte față de prognoza sezonieră naivă (scală logaritmică; linia punctată este 1)

TSA_ch11_benchmark

Interpretarea testului

- Consumul orar: foundation models cîștigă clar (raportul WQL 0,58 pentru Chronos-2, 0,66 pentru Bolt small, 0,96 pentru ARIMA); ETS, doar cu sezonul zilnic, eșuează (1,75)
- PIB-ul României (serie trimestrială scurtă): ARIMA este cel mai bun (0,72); Chronos-Bolt este mai slab decît prognoza sezonieră naivă (1,22)
- Șomajul: toate modelele sînt aproape de 1; EUR/RON: aproape de 1, cu excepția Chronos-2 (WQL 0,87), a cărui mediană nu este totuși mai bună decît prognoza naivă (MASE 1,05)
- Media geometrică a WQL: Chronos-2 0,78, ARIMA 0,87, Bolt small 0,90, Bolt tiny 0,95, ETS 0,97; Chronos-2 este cel mai bun pe 5 din 7 serii
- Un ARIMA bine specificat este la fel de bun ca foundation models mici; un cîștigător unic pe toate seriile nu există

Acoperirea intervalelor de 80%



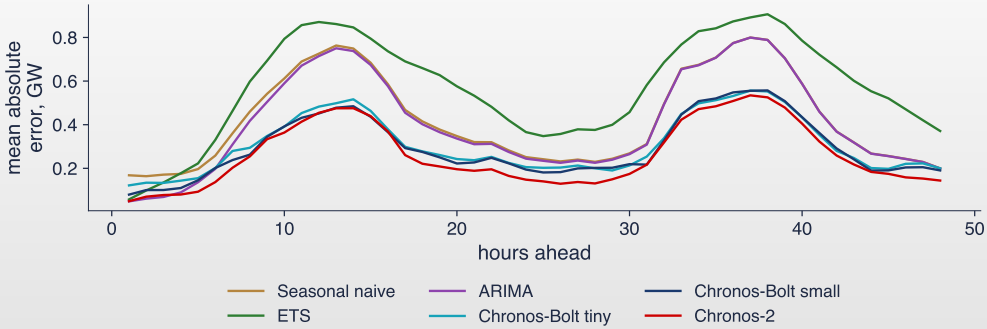
- Proporția valorilor observate din $[\hat{q}_{0,1}, \hat{q}_{0,9}]$, pe toate originile și toți pașii; linia punctată este ținta de 0,8

TSA_ch11_benchmark

Interpretarea acoperirii

- Acoperirea medie: prognoza sezonieră naivă 83%, ETS 84%, ARIMA 84%, Bolt small 80%, Chronos-2 76%
- Chronos-2 are cel mai bun WQL, dar intervale prea înguste: fine, dar nu complet calibrate
- Vînzările din SUA: fiecare model acoperă mult sub 80% (de la 64,7% pentru prognoza sezonieră naivă și 65,0% pentru Chronos-2 pînă la 75,2% pentru Bolt tiny): șocul din 2020 și saltul din 2021 îi surprind pe toți
- Benzile ETS și ARIMA presupun erori Normale și variață constantă: prea largi în perioadele calme, prea înguste în crize

Consumul României: eroarea în funcție de orizont



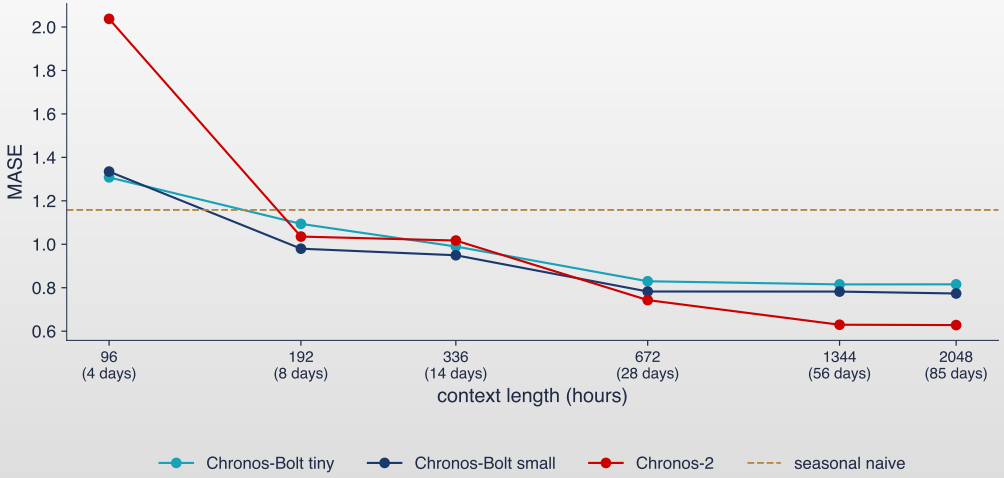
⊡ Eroarea absolută medie (GW) a prognozei mediane pentru fiecare dintre cele 48 de ore, pe 52 de origini săptămînale

TSA_ch11_benchmark

Interpretarea erorii în funcție de orizont

- Prima oră: ARIMA 0,05 și Chronos-2 0,05 GW, mult mai bine decît prognoza sezonieră naivă, 0,17: ultima observație contează cel mai mult
- După o zi, ARIMA revine la nivelul prognozei naive (0,24 față de 0,25 GW): corecția lui se stinge
- Chronos-2 își păstrează avantajul pe tot orizontul (0,14 GW la 48 de ore, față de 0,20)
- ETS este bun cîteva ore, apoi cel mai slab: un sezon zilnic nu poate reprezenta weekendul

Lungimea necesară a contextului (1/2)



Lungimea necesară a contextului (2/2)

- MASE pentru consumul României cu contexte de 96 pînă la 2048 de ore; 26 de origini săptămînale (ultima jumătate de an); linia punctată este prognoza sezonieră naivă
- Cu 96 de ore (4 zile), fiecare model este mai slab decît prognoza sezonieră naivă: Chronos-Bolt small 1,33, Chronos-2 2,04, naiv 1,16
- De la 192 de ore (peste o săptămînă), ciclul săptămînal intră în context și eroarea scade
- Chronos-2 se îmbunătățește pînă la 1344 de ore (0,63); Chronos-Bolt se stabilizează de la 672 de ore (0,78)
- Regula: dați modelului mai multe cicluri sezoniere complete; un model zero-shot doar repetă ce vede

TSA_ch11_context_contamination

Dimensiunea și precizia



⊡ Media geometrică, pe cele șapte serii, a raportului WQL față de prognoza sezonieră naivă, în funcție de numărul de parametri

TSA_ch11_context_contamination

Interpretarea dimensiunii și a costului

- Parametri: Bolt tiny 9 milioane, Bolt small 48 de milioane, Chronos-2 119 milioane; ETS și ARIMA: cîțiva pe serie
- Modelele mai mari obțin aici scoruri mai bune, dar Chronos-2 diferă și prin arhitectură și date de antrenare: dimensiunea singură nu este cauza; ARIMA (0,87) se situează între Bolt small și Chronos-2
- Timpul pentru o prognoză a consumului pe procesorul unui laptop: Bolt small 19 ms, Chronos-2 43 ms, ARIMA 114 ms, ETS 411 ms
- Inferența zero-shot este ieftină; costul a fost plătit o singură dată, la pre-antrenare

Benchmark-uri de referință

- **M3** (Makridakis și Hibon (2000))
  - 3003 serii
  - metodele sofisticate nu le-au bătut în medie pe cele simple
- **M4** (Makridakis et al. (2020))
  - 100 000 de serii
  - au cîștigat combinațiile și un hibrid de netezire exponențială și rețea neuronală
- **Arhiva Monash** (Godahewa et al. (2021))
  - seturi publice de date cu orizonturi fixe și rezultate de referință
- **GIFT-Eval** (Aksu et al. (2024))
  - multe domenii și frecvențe, MASE și CRPS, un clasament public al foundation models
  - separă datele de test de un corpus de pre-antrenare, pentru a limita contaminarea
- Studii critice: modele liniare simple au bătut primele modele de prognoză cu Transformer (Zeng et al. (2023)); componentele LLM nu adaugă precizie (Tan et al. (2024))

Utilitatea foundation models: avantaje și limite

- **Ajută**
  - multe serii, puțin timp pentru fiecare; sezonalitate puternică și regulată; contexte lungi (consum, vînzări, trafic)
  - un reper rapid și puternic înaintea unui model dedicat; ieșire probabilistă fără cost suplimentar
- **Nu ajută**
  - serii apropiate de un mers aleator (cursuri de schimb, randamente): nu este nimic de învățat din trecut
  - serii scurte (cîteva zeci de trimestre), unde un ARIMA atent specificat este mai bun
  - cînd contează factorii explicativi (temperatura, prețurile, politicile), iar aceștia nu sînt în context
  - cînd modelul trebuie explicat parametru cu parametru sau se pune o întrebare cauzală
- Comparați întotdeauna cu prognoza sezonieră naivă, ETS și ARIMA pe aceleași origini; testați diferențele cu Diebold–Mariano (Diebold și Mariano (1995))

Recapitulare: evaluarea

- Chronos-2 este cel mai bun în medie, mai ales pentru consumul orar; ARIMA este aproape și cîștigă la PIB
- Pe seriile apropiate de un mers aleator, niciun model nu bate prognoza naivă
- Intervalele foundation models tind să fie prea înguste; contextul trebuie să acopere sezoanele

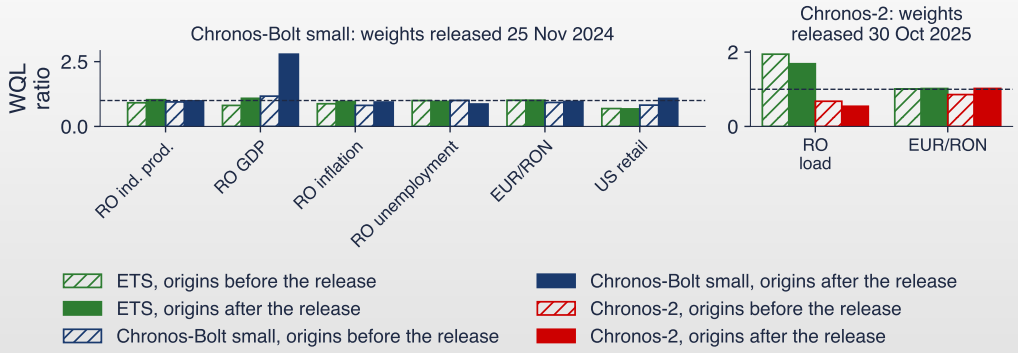
Leakage-ul în propriul cod

- **Leakage**
  - informația din perioada de test ajunge în prognoze (Hewamalage et al. (2023))
- Greșeli tipice
  - scalarea sau standardizarea cu media întregului eșantion
  - alegerea ordinelor ARIMA, a lungimii contextului sau a versiunii modelului după ce ați văzut erorile de test
  - folosirea datelor revizuite (PIB) ca și cum ar fi fost cunoscute la origine
  - împărțirea aleatoare a unei serii de timp în antrenare și test (Capitolul 9)
- În acest capitol: fiecare context se termină la origine; ordinele ARIMA au fost fixate înaintea primei origini; Chronos scalează doar cu contextul
  - problema rămasă: PIB-ul și producția de la Eurostat sînt ultima versiune revizuită, nu date în timp real

Contaminarea benchmark-urilor

- **Contaminarea**
  - seria de test (sau serii puternic corelate cu ea) a fost în corpusul de pre-antrenare
  - scorul măsoară atunci memoria, nu prognoza: o distorsiune optimistă
- Chronos raportează două benchmark-uri: în domeniu (seturi folosite la antrenare) și zero-shot (seturi nefolosite) (Ansari et al. (2024))
  - serii publice precum M4 sau datele macro din SUA sînt probabil în multe corpusuri; FRED este public, la fel și Eurostat
- Remedii
  - testați doar pe date publicate după data lansării ponderilor
  - benchmark-uri live: prognoze depuse înainte ca rezultatul să existe
  - raportați corpusul de antrenare și data lansării fiecărui model folosit

O verificare a contaminării: înainte și după lansare



- Raportul WQL față de prognoza sezonieră naivă pentru originile dinainte (haşurat) și de după (plin) lansarea ponderilor; ETS pe aceleași origini, ca termen de comparație

TSA_ch11_context_contamination

Interpretarea verificării contaminării

- Chronos-2 pe EUR/RON: 0,86 înainte de lansare (124 de origini), 1,02 după (10 origini): avantajul dispare
- Chronos-2 pe consum: 0,68 înainte, 0,54 după: cîștigul este real, și pe date pe care nu le-a putut vedea
- Chronos-Bolt small, vînzările din SUA: 0,82 înainte, 1,08 după, în timp ce ETS este stabil (0,69; 0,68)
- Atenție: 3–35 de origini după lansare sînt puține; și perioada de după diferă (2025–2026); o scădere este un semnal de alarmă, nu o dovadă
- Chronos-2 a fost lansat în octombrie 2025: rezultatele lui lunare de mai sus folosesc doar origini dinaintea acestei date

Recapitulare: leakage-ul și contaminarea

- Leakage-ul este sub controlul dumneavoastră: folosiți doar datele disponibile la fiecare origine
- Contaminarea nu este: corpusul de pre-antrenare poate conține seria dumneavoastră
- Verificați rezultatele pe date publicate după lansarea modelului

Contribuția posibilă a AI

- **Cod**
  - o primă versiune a unui backtest cu origine mobilă, a pierderii pinball și a WQL, a unui fan chart
- **Lectură**
  - un rezumat al articolului unui model (arhitectura, corpusul, licența, data lansării)
- **Explorare**
  - rularea mai multor modele deschise pe multe serii românești de la Eurostat sau INS
- Exemplu de prompt
  - `Write Python code that downloads Romanian monthly industrial production (Eurostat sts_inpr_m, not seasonally adjusted), forecasts 12 months ahead from monthly origins since 2016 with amazon/chronos-bolt-small (package chronos-forecasting, CPU), ETS and the seasonal naive, and reports MASE, the weighted quantile loss and the coverage of the 80% interval.`

Verificări necesare

- Pachetul și numele modelelor: `BaseChronosPipeline`, `amazon/chronos-bolt-small`; o clasă sau un nume de model inventat eșuează imediat
- Contextul se termină la origine: nicio scalare, ordonare sau calibrare cu date de test
- Nivelurile cuantilelor: Chronos-Bolt dă doar 0,1–0,9; o „cuantilă de 1%” obținută din el este o extrapolare
- Reperele: prognoza sezonieră naivă, ETS și ARIMA pe aceleași origini; scoruri relative la prognoza naivă
- Data lansării ponderilor comparată cu perioada de test; afirmațiile de tip „cel mai bun model” au nevoie de o sursă
- Fiecare articol citat: trebuie să existe; verificați numărul arXiv sau DOI-ul

Idei de reținut

- Un foundation model este pre-antrenat o singură dată pe multe serii și prognozează serii noi zero-shot, doar din context
- Seriile devin intrări ale unui Transformer prin scalare, cuantizare (Chronos) sau patching (TimesFM, Moirai, Chronos-Bolt)
- Ieșirile sînt cuantile: evaluați-le cu pierderea pinball, CRPS sau WQL și verificați acoperirea
- Pe cele șapte serii: cîștiguri mari la consumul orar, niciunul la seriile apropiate de un mers aleator, pierderi la PIB-ul trimestrial scurt; ARIMA rămîne un reper puternic
- Contaminarea poate face scorurile publicate să pară mai bune decît sînt: testați pe date publicate după lansarea modelului

Formule de reținut

Mărimea	**Formula**
Atenția	$\mathrm{softmax}(QK^\top/\sqrt{d})\,V$
Scalarea prin medie	$\tilde{x}_t = x_t/s, \quad s = \frac{1}{C}\sum_{t=\mathbf{1}}^{C} \lvert x_t\rvert$
Pierderea pinball	$L_\tau(y,\hat{q}_\tau) = \max\{\tau(y-\hat{q}_\tau),\ (\tau-1)(y-\hat{q}_\tau)\}$
CRPS	$\int (F(z) - \mathbf{1}\{z \geq y\})^{\mathbf{2}}\, dz = 2\int_{\mathbf{0}}^{\mathbf{1}} L_\tau(y, F^{-\mathbf{1}}(\tau))\, d\tau$
WQL	$\frac{\mathbf{1}}{\mathbf{9}} \sum_\tau 2\sum_t L_\tau(y_t,\hat{q}_{\tau,t}) \big/ \sum_t \lvert y_t\rvert$
MASE	$\frac{\mathbf{1}}{h}\sum_j \lvert y_{T+j} - \hat{y}_{T+j}\rvert \Big/ \frac{\mathbf{1}}{T-m}\sum_{t>m} \lvert y_t - y_{t-m}\rvert$
Acoperirea	$\frac{\mathbf{1}}{n}\sum_i \mathbf{1}\{\hat{q}^{(i)}_{\mathbf{0},\mathbf{1}} \leq y_i \leq \hat{q}^{(i)}_{\mathbf{0},\mathbf{9}}\}$, ținta 0,8

Autoevaluare (1/2)

- **Întrebare**: care este diferența dintre prognoza zero-shot și fine-tuning?
  - **Răspuns**: zero-shot păstrează fixe ponderile pre-antrenate și folosește doar contextul; fine-tuning-ul antrenează în continuare ponderile pe datele-țintă
- **Întrebare**: un context este 10, 20, 30. Cît sînt s și valorile scalate în Chronos?
  - **Răspuns**: $s = 20$; valorile scalate 0,5; 1; 1,5
- **Întrebare**: $\hat{q}_{0,9} = 5$ și $y = 6$. Cît este pierderea pinball?
  - **Răspuns**: $0{,}9 \cdot (6 - 5) = 0{,}9$
- **Întrebare**: cîte patch-uri de 16 dă un context de 512 valori?
  - **Răspuns**: 32

Autoevaluare (2/2)

- **Întrebare**: un interval de 80% acoperă 95% din valorile observate. Este un rezultat bun?
  - **Răspuns**: nu: intervalele sînt prea largi (lipsește finețea); CRPS penalizează acest lucru
- **Întrebare**: de ce nu poate un foundation model să bată mersul aleator pe EUR/RON?
  - **Răspuns**: variațiile cursului sînt aproape imprevizibile pe baza trecutului; niciun model nu poate extrage informație pe care contextul nu o conține
- **Întrebare**: un model lansat în 2025 obține scoruri foarte bune pe date de test din 2018–2023. Ce verificați?
  - **Răspuns**: dacă acele serii au fost în corpusul lui de pre-antrenare; repetați testul pe date publicate după lansare
- **Idee de proiect**: comparați Chronos-Bolt, Chronos-2 și ARIMA pe toate seriile lunare ale României dintr-un set de date Eurostat, înainte și după datele de lansare
- Urmează: Capitolul 12, analiza spectrală

Bibliografie (1/3)

- Aksu, T., Woo, G., Liu, J., Liu, X., et al. (2024). GIFT-Eval: A benchmark for general time series forecasting model evaluation. arXiv:2410.10393
- Ansari, A. F., Stella, L., Turkmen, C., Zhang, X., et al. (2024). Chronos: Learning the language of time series. *Transactions on Machine Learning Research*. arXiv:2403.07815
- Ansari, A. F., Shchur, O., Küken, J., Auer, A., et al. (2025). Chronos-2: From univariate to universal forecasting. arXiv:2510.15821
- Bommasani, R., Hudson, D. A., Adeli, E., Altman, R., et al. (2021). On the opportunities and risks of foundation models. arXiv:2108.07258
- Das, A., Kong, W., Sen, R., & Zhou, Y. (2024). A decoder-only foundation model for time-series forecasting. *Proceedings of the 41st International Conference on Machine Learning (ICML)*. arXiv:2310.10688
- Diebold, F. X., & Mariano, R. S. (1995). Comparing predictive accuracy. *Journal of Business & Economic Statistics*, 13(3), 253–263. doi:10.1080/07350015.1995.10524599
- Garza, A., Challu, C., & Mergenthaler-Canseco, M. (2023). TimeGPT-1. arXiv:2310.03589
- Gneiting, T., & Raftery, A. E. (2007). Strictly proper scoring rules, prediction, and estimation. *Journal of the American Statistical Association*, 102(477), 359–378. doi:10.1198/016214506000001437

Bibliografie (2/3)

- Godahewa, R., Bergmeir, C., Webb, G. I., Hyndman, R. J., & Montero-Manso, P. (2021). Monash time series forecasting archive. *NeurIPS Track on Datasets and Benchmarks*. arXiv:2105.06643
- Gruver, N., Finzi, M., Qiu, S., & Wilson, A. G. (2023). Large language models are zero-shot time series forecasters. *Advances in Neural Information Processing Systems*, 36. arXiv:2310.07820
- Hewamalage, H., Ackermann, K., & Bergmeir, C. (2023). Forecast evaluation for data scientists: Common pitfalls and best practices. *Data Mining and Knowledge Discovery*, 37(2), 788–832. doi:10.1007/s10618-022-00894-5
- Huang, C., & Petukhina, A. (2022). *Applied Time Series Analysis and Forecasting with Python*. Springer. doi:10.1007/978-3-031-13584-2
- Hyndman, R. J., & Athanasopoulos, G. (2021). *Forecasting: Principles and Practice* (3rd ed.). OTexts. otexts.com/fpp3
- Hyndman, R. J., & Koehler, A. B. (2006). Another look at measures of forecast accuracy. *International Journal of Forecasting*, 22(4), 679–688. doi:10.1016/j.ijforecast.2006.03.001
- Makridakis, S., & Hibon, M. (2000). The M3-Competition: Results, conclusions and implications. *International Journal of Forecasting*, 16(4), 451–476. doi:10.1016/S0169-2070(00)00057-1
- Makridakis, S., Spiliotis, E., & Assimakopoulos, V. (2020). The M4 Competition: 100,000 time series and 61 forecasting methods. *International Journal of Forecasting*, 36(1), 54–74. doi:10.1016/j.ijforecast.2019.04.014

Bibliografie (3/3)

- Matheson, J. E., & Winkler, R. L. (1976). Scoring rules for continuous probability distributions. *Management Science*, 22(10), 1087–1096. doi:10.1287/mnsc.22.10.1087
- Nie, Y., Nguyen, N. H., Sinthong, P., & Kalagnanam, J. (2023). A time series is worth 64 words: Long-term forecasting with Transformers. *International Conference on Learning Representations (ICLR)*. arXiv:2211.14730
- Rasul, K., Ashok, A., Williams, A. R., Ghonia, H., et al. (2023). Lag-Llama: Towards foundation models for probabilistic time series forecasting. arXiv:2310.08278
- Salinas, D., Flunkert, V., Gasthaus, J., & Januschowski, T. (2020). DeepAR: Probabilistic forecasting with autoregressive recurrent networks. *International Journal of Forecasting*, 36(3), 1181–1191. doi:10.1016/j.ijforecast.2019.07.001
- Tan, M., Merrill, M. A., Gupta, V., Althoff, T., & Hartvigsen, T. (2024). Are language models actually useful for time series forecasting? *Advances in Neural Information Processing Systems*, 37. arXiv:2406.16964
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., et al. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*, 30. arXiv:1706.03762
- Woo, G., Liu, C., Kumar, A., Xiong, C., Savarese, S., & Sahoo, D. (2024). Unified training of universal time series forecasting Transformers. *Proceedings of the 41st International Conference on Machine Learning (ICML)*. arXiv:2402.02592
- Zeng, A., Chen, M., Zhang, L., & Xu, Q. (2023). Are Transformers effective for time series forecasting? *Proceedings of the AAAI Conference on Artificial Intelligence*, 37(9), 11121–11128. doi:10.1609/aaai.v37i9.26317