








Energy Markets Quantitative Analysis

Lecture 7: Machine Learning for Energy Markets



Prof. dr. Daniel Traian PELE

Bucharest University of Economic Studies, Department of Statistics and Econometrics
Romanian Academy, Institute for Economic Forecasting
IDA Institute Digital Assets ▪ AI4EFin ▪ MSCA Digital Finance

Learning objectives

- Set up an honest machine learning (ML) experiment: training, validation and test samples in time order, and only the information known at 12:00 Central European Time (CET) on the day before delivery.
- Build and tune a deep neural network (DNN) and gradient-boosted trees for the 24 day-ahead prices of Romania
  - Try a pretrained foundation model.
- Compare them with the benchmarks of Lecture 2 on the same test year, with the mean absolute error (MAE), the root mean squared error (RMSE), the relative MAE (rMAE) and the Diebold–Mariano (DM) test
  - Combine machine learning with the statistical models.
- Explain the forecasts, predict price spikes and negative prices, and recognise the pitfalls that make a model look better than it is.

Abbreviations (1/2)

AI	artificial intelligence
ARIMA	autoregressive integrated moving average
ARX	autoregressive model with exogenous variables
AUC	area under the ROC curve
AW	averaging across calibration windows
AW-ARX	ARX averaged across calibration windows
AW-LEAR	LEAR averaged across calibration windows
CET	Central European Time
CPU	central processing unit (processor core)
DM	Diebold–Mariano test
DNN	deep neural network
DST	daylight saving time
ENTSO-E	European Network of Transmission System Operators for Electricity
ETC	exchange-traded commodity
EU	European Union
EUA	EU emission allowance (1 t CO_2)
EUR	euro
FWER	familywise error rate
GBM	gradient boosting machine (boosted trees)
GW	gigawatt
ISE	Fraunhofer Institute for Solar Energy Systems
LASSO	least absolute shrinkage and selection operator

Abbreviations (2/2)

LEAR	LASSO-estimated autoregression
LLM	large language model
MAD	median absolute deviation
MAE	mean absolute error
ML	machine learning
MLP	multilayer perceptron (feed-forward neural network)
MW	megawatt
MWh	megawatt-hour
OPCOM	Romanian electricity and gas market operator
PZU	Piața pentru Ziua Următoare (day-ahead market)
RC	Reality Check (data-snooping test of White, 2000)
ReLU	rectified linear unit
RES	renewable energy sources (here: wind + solar)
rMAE	relative MAE (model MAE / naive MAE)
RMSE	root mean squared error
RO	Romania (bidding zone)
ROC	receiver operating characteristic
SHAP	SHapley Additive exPlanations
SMAPE	symmetric mean absolute percentage error
SPA	superior predictive ability (test of Hansen, 2005)
TSO	transmission system operator
XGBoost	eXtreme Gradient Boosting (library)

Data: the Lecture 2 sample and test year, with an added validation year

Element	Choice in this lecture
Target	hourly day-ahead price of Romania (RO), Piața pentru Ziua Următoare (PZU), run by the Romanian electricity and gas market operator (OPCOM)
Inputs (as in Lecture 2)	prices up to $d-1$; load, wind + solar, hydro, nuclear, net imports up to $d-2$, from the European Network of Transmission System Operators for Electricity (ENTSO-E); carbon: European Union (EU) emission allowance (EUA), via an exchange-traded commodity (ETC)
Training	from 8 Jan 2022; grows by one month at each refit
Validation year	1 Nov 2024 – 31 Oct 2025: used only to choose the settings of the models
Test year	1 Nov 2025 – 31 Oct 2026, 8 760 hours, models refitted monthly

- **Data up to 31 Oct 2026**, downloaded 4 Nov 2026 from energy-charts.info (Fraunhofer ISE) and Yahoo Finance.
- **Benchmarks**: the forecasts of Lecture 2, read from its files
  - Naive model; autoregressive model with exogenous variables (ARX); least absolute shrinkage and selection operator (LASSO)
  - LASSO-estimated autoregression (LEAR); ARX averaged across calibration windows (AW-ARX), LEAR likewise (AW-LEAR); ensemble of 4 models

Experiment yourself: notebook §1

Outline

- Machine-learning models and their evaluation
- Deep neural networks
- Gradient boosting and foundation models
- Model comparison, combination and computational cost
- Explaining the forecasts
- Spikes, negative prices and a case study
- Evaluation errors that overstate accuracy

Flexibility and variance of machine-learning models

- **In one sentence:** A flexible model can learn thresholds and interactions that LEAR misses, but it can also learn the noise of the training years.

$$\hat{\theta} = \arg\min_{\theta} \underbrace{\frac{1}{n}\sum_{i=1}^{n} \ell\Big(Y_i, f_\theta(\mathbf{x}_i)\Big)}_{\text{fit to the training days}} + \underbrace{\lambda\,\Omega(\theta)}_{\text{penalty}}$$

- **where**
  - f_θ = the model with parameters θ
  - $\mathbf{x}_i$ = inputs known at gate closure
  - Y_i = price
  - ℓ = loss, here the absolute error
  - Ω = complexity, λ = its weight.

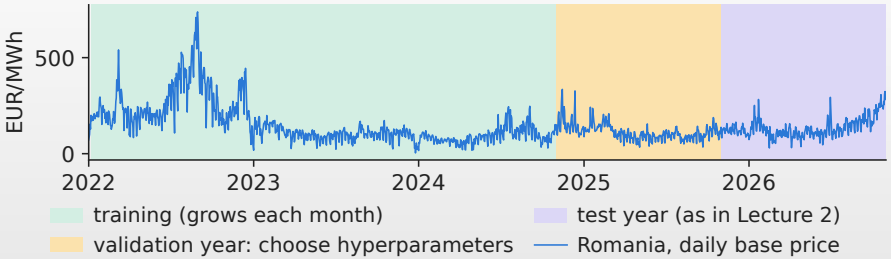
- **How flexible are our models?**
  - LEAR: at most 247 coefficients per hour
  - Our DNN: 141 336 parameters, for 1 727 training days at the last refit
- **Consequence**
  - The training error says nothing about the future; only data the model has not seen can judge it

- **In practice:** ask a vendor how many parameters the model has per day of history, and on which days its error was measured.

(Hastie, Tibshirani & Friedman, 2009; Lago et al., 2021, APEN)

Experiment yourself: notebook §1

Three samples in time order: training, validation, test



Source: energy-charts.info (Fraunhofer ISE): OPCOM prices. Data: daily base price, 2022-01-01 to 2026-10-31.

- **Validation year**: 25 settings per model compared, refitted every 3 months
  - The setting with the lowest MAE is frozen before the test year starts
- **Test year**: models refitted at the start of each month on all earlier days
  - Never used to choose anything; the same year as in Lecture 2, so the errors are directly comparable

In practice: treat the test year like a sealed envelope, opened once, after every choice is made.

(Bergmeir & Benítez, 2012, Inf. Sci.; Lago et al., 2021, APEN)

EMQA_L7_split_ml

Hyperparameters are chosen by random search on the validation year

- **In one sentence:** Settings such as the number of neurons or of trees are not estimated by the model; we try several and keep the one that forecasts best on unseen days.

$$\gamma^* = \arg \min_{\gamma \in \{\gamma_0, \ldots, \gamma_{24}\}} \frac{1}{24\,|\mathcal{V}|} \sum_{d \in \mathcal{V}} \sum_{h=0}^{23} \left| P_{d,h} - \hat{P}_{d,h}(\gamma) \right|$$

- **where**
  - γ = one setting (vector of hyperparameters)
  - γ_0 = the setting of Lecture 2, the others drawn at random
  - $\mathcal{V}$ = days of the validation year
  - $\hat{P}_{d,h}(\gamma)$ = forecast with setting γ.

- **Why random, not a full grid**
  - Only a few settings matter; random draws cover each of them better with the same budget
- **Our budget**
  - 24 random settings + the Lecture 2 setting, for each of the two models
  - 7.3 minutes on one core of the central processing unit (CPU)

- **In practice:** a modest, documented search is enough
  - A search without a separate validation sample is not a search but a leak.

(Bergstra & Bengio, 2012, JMLR)

Experiment yourself: notebook §2

Outline

- Machine-learning models and their evaluation
- Deep neural networks
- Gradient boosting and foundation models
- Model comparison, combination and computational cost
- Explaining the forecasts
- Spikes, negative prices and a case study
- Evaluation errors that overstate accuracy

Structure of a feed-forward neural network

- **In one sentence:** Each neuron bends a linear combination of the inputs; the output layer gives the 24 prices of day d at once.

$$\mathbf{z} = \phi\big(W_1 \mathbf{x}_d + \mathbf{b}_1\big), \qquad \hat{\mathbf{Y}}_d = W_2\,\mathbf{z} + \mathbf{b}_2 \in \mathbb{R}^{24}$$
$$\phi(u) = \max(0, u) \quad \text{or} \quad \phi(u) = \ln(1 + e^u)$$

- **where**
  - $\mathbf{x}_d$ = inputs of day d (prices, load, wind + solar, fundamentals, weekday)
  - W_1, W_2 = weights, $\mathbf{b}_1, \mathbf{b}_2$ = intercepts
  - ϕ = rectified linear unit (ReLU) or its smooth version (softplus)
  - $\hat{\mathbf{Y}}_d$ = asinh-scaled prices, as in Lecture 2.

- **Our network** (chosen on the validation year)
  - 251 inputs, one hidden layer of 512 neurons, softplus, 24 outputs
  - 141 336 parameters; two-layer networks did worse on validation

- **Better than in Lecture 2**
  - The multilayer perceptron (MLP) of Lecture 2 had MAE 25.54; ours has 22.95 EUR/MWh

- **In practice:** a network is a regression with many more coefficients: it needs the same data discipline, not less.

(Lago, Marcjasz, De Schutter & Weron, 2021, APEN)

Experiment yourself: notebook §2

Training: stochastic gradient descent with early stopping

- **In one sentence:** The weights move a little in the direction that lowers the error on a random batch of days; we stop when the error on later days stops falling.

$$\mathcal{L}(\theta) = \frac{1}{24\,|B|} \sum_{d \in B} \sum_{h=0}^{23} \left| Y_{d,h} - \hat{Y}_{d,h}(\theta) \right|$$

$$\theta \leftarrow \theta - \eta\, \widehat{\nabla \mathcal{L}}$$

- **where**
  - B = a mini-batch of 64 days
  - η = learning rate ($3 \cdot 10^{-4}$)
  - $\widehat{\nabla \mathcal{L}}$ = gradient rescaled by the Adam method
  - One epoch = one pass over all training days.

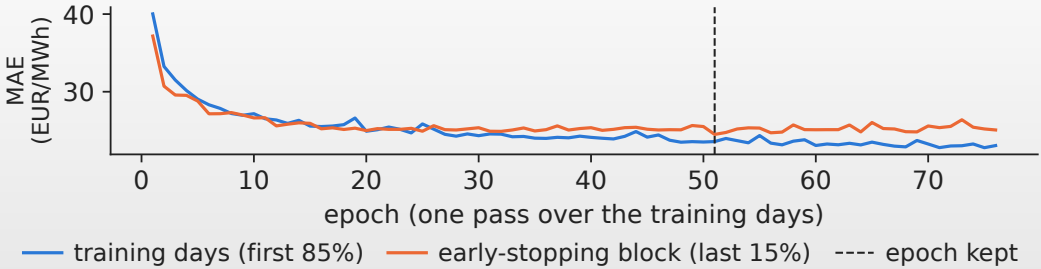
- **Early stopping**
  - Train on the first 85% of the training days, watch the error on the last 15%
  - Keep the best epoch, then refit on all days for that many epochs
- **On our data**
  - Epochs kept: median 56, from 23 to 92 (12 refits × 4 seeds)
  - Dropout: during training each neuron is switched off at random with probability 0.1

- **In practice:** the stopping rule uses only past days: it is part of the model, so it is checked like the rest.

(Kingma & Ba, 2015, ICLR; Srivastava et al., 2014, JMLR)

Experiment yourself: notebook §2

Early stopping based on the error on recent days



Source: energy-charts.info (Fraunhofer ISE); Yahoo Finance (EUA). Data: chosen DNN, refit for 2026-10, seed 1.

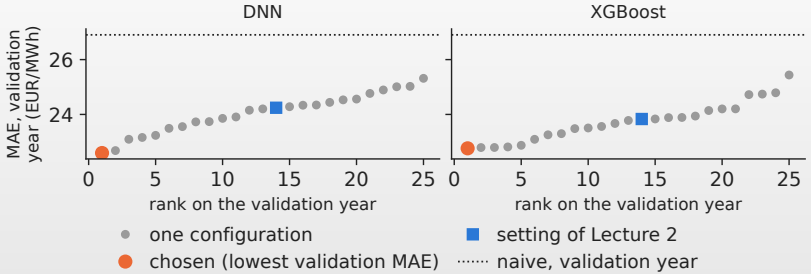
- Refit for Oct 2026: the error on the last 15% of days is lowest at epoch 51
  - 24.50 EUR/MWh; training stops at epoch 76
- After that point the network fits the past better and the recent days worse: it starts to memorise.

- **In practice:** ask the vendor for this curve: without it, the network may simply have memorised the past.

(own computation; chosen DNN, seed 1)

EMQA_L7_learning_curve

Random search cuts the validation error of both models



Source: energy-charts.info (Fraunhofer ISE); Yahoo Finance (EUA). Data: validation year 2024-11-01 to 2025-10-31, quarterly refits.

- **DNN**: best setting 22.59 EUR/MWh; Lecture 2 setting 24.24, rank 14 of 25
  - **eXtreme Gradient Boosting (XGBoost)**: 22.76 against 23.83 (rank 14)
- All 25 settings of both models beat the naive model on the validation year (26.90 EUR/MWh).

- **In practice:** budget one validation year for tuning any machine-learning model.

(own computation; validation year 1 Nov 2024 – 31 Oct 2025)

EMQA_L7_search

Outline

- Machine-learning models and their evaluation
- Deep neural networks
- Gradient boosting and foundation models
- Model comparison, combination and computational cost
- Explaining the forecasts
- Spikes, negative prices and a case study
- Evaluation errors that overstate accuracy

A regression tree cuts the inputs into boxes and predicts one price per box

- **In one sentence:** A tree asks yes/no questions; each final leaf holds one value, so thresholds and interactions come for free.

$$f(\mathbf{x}) = \sum_{j=1}^{J} w_j \mathbb{1}\{\mathbf{x} \in R_j\}$$

$$\text{Gain} = \frac{G_L^2}{H_L + \lambda} + \frac{G_R^2}{H_R + \lambda} - \frac{(G_L + G_R)^2}{H_L + H_R + \lambda} - \gamma$$

- **where**
  - R_j = leaf j, w_j = its value
  - G, H = sums of the 1st and 2nd derivatives of the loss, left (L) and right (R)
  - λ, γ = penalties
  - The split with the largest gain is kept.

- **On our data** (refit for Oct 2026)
  - First question of the first tree: price of $d - 1$, same hour, below 119.5 EUR/MWh?
  - 233 leaves; the model adds up 1 500 such trees

- **In practice:** a tree speaks the language of a trading desk: above a level yesterday, a premium tonight.

(Chen & Guestrin, 2016, KDD; Hastie et al., 2009)

Experiment yourself: notebook §3

Gradient boosting: each new tree corrects the remaining errors

⊡ **In one sentence:** Fit a small tree to what is still wrong, add it with a small weight, repeat a thousand times.

$$F_m(\mathbf{x}) = F_{m-1}(\mathbf{x}) + \eta\, f_m(\mathbf{x}), \qquad w_j^* = -\frac{G_j}{H_j + \lambda}$$

$$\min \sum_i \ell\big(Y_i, F_M(\mathbf{x}_i)\big) + \sum_m \Big(\gamma J_m + \tfrac{\lambda}{2} \sum_j w_{mj}^2\Big)$$

⊡ **where**

- F_m = forecast after m trees
- f_m = tree m, with J_m leaves
- η = learning rate
- w_j^* = optimal value of leaf j (Appendix B)
- ℓ = absolute error.

⊡ **Chosen on the validation year**

- 1 500 trees of depth 8, $\eta = 0.02$, each tree on a fraction 0.8 of the rows

⊡ **Test year**

- MAE 23.26 EUR/MWh; untuned gradient boosting machine (GBM) of Lecture 2: 24.05 (DM $p < 0.001$)
- AW-LEAR (22.53) is still more accurate (DM $p = 0.016$)

⊡ **In practice:** tuning matters more than the software: the same kind of trees gained almost one euro per MWh from good settings.

(Friedman, 2001, Ann. Stat.; Chen & Guestrin, 2016, KDD)

Experiment yourself: notebook §3

Foundation models forecast Romanian prices with no training at all

- **In one sentence:** Chronos models were trained by Amazon on large collections of time series; we give them only the recent prices and ask for the next 24 hours (zero-shot).

$$\left(\hat{q}^{\tau}_{d,h}\right)^{\tau=0.1,\,0.5,\,0.9}_{h=0,\ldots,23} = g\left(P_{t-2047},\ldots,P_t\right)$$

- **where**
  - $P_{t-2047},\ldots,P_t$ = the last 2048 hourly prices, up to the end of $d-1$ (85 days)
  - g = the pretrained network, used as it is
  - $\hat{q}^{\tau}$ = forecast quantiles
  - The point forecast is the median $\hat{q}^{0.5}$.

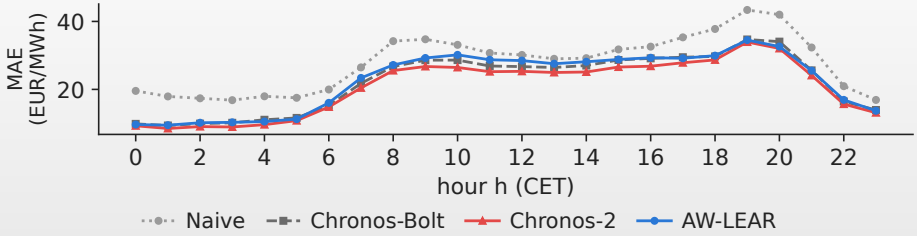
- **Test year**, prices only, no fundamentals
  - Chronos-2: MAE 20.82; Chronos-Bolt: 22.15; AW-LEAR: 22.53 EUR/MWh
  - Chronos-2 more accurate than AW-LEAR: DM $p < 0.001$; in all 24 hours
- **Robustness**
  - Validation year: Chronos-2 21.05, best tuned DNN 22.59
  - Published in Oct 2025, before the test year

- **In practice:** test a foundation model before building your own network: it is cheap, and here it was the most accurate single model.

(Ansari et al., 2024, TMLR; Ansari et al., 2025, arXiv)

Experiment yourself: notebook §4

Chronos-2 is more accurate than AW-LEAR in every hour of the day



Source: energy-charts.info (Fraunhofer ISE): OPCOM prices. Data: test year 2025-11-01 to 2026-10-31; zero-shot, context = last 2048 hourly prices.

- Chronos-Bolt, smaller and older, is level with AW-LEAR (DM $p = 0.220$).
- The 10%–90% band of Chronos-2 holds 76.5% of the prices, not 80%: too narrow.

- **In practice:** use the median
  - Recalibrate the bands (Lecture 2) before sizing risk with them.

(own computation, test year 1 Nov 2025 – 31 Oct 2026)

EMQA_L7_foundation_hour

AIDA 2026 keynote: deep networks, LEAR and zero-shot models

- **R. Weron, keynote** (4 June 2026, slides)
  - Model evolution: ARX, averaging across windows, LEAR, deep and interpretable networks
  - Pretrained zero-shot models against LEAR and distributional DNN models, test 2021–2025
- **Deep and interpretable networks**
  - Open epftoolbox benchmark: the DNN had a lower rMAE than LEAR in all five markets (Lago et al., 2021)
  - NBEATSx: trend, seasonality and exogenous blocks make the network readable (Olivares et al., 2023)
- **Zero-shot results** (RMSE, as printed)
  - 2022: LEAR + conformal prediction 45.71, Chronos 2 41.07, TabPFN 3.0 ts 34.39
  - 2025: 20.38, 20.58, 17.66; a TabPFN version was the best in every year
- **Link with our test year**
  - Chronos-2, zero-shot, is our most accurate single model (MAE 20.82 vs 22.53 for AW-LEAR)
  - Rankings move: TabPFN 2.0 had RMSE 68.85 in 2022, 17.52 in 2025

In practice: keep LEAR as the benchmark and test zero-shot models next to it, over several years.

(AIDA 2026 slides, Bucharest; papers: Lago et al., 2021, APEN; Olivares et al., 2023, IJF)

AIDA 2026: four generations of models on 27 European price series

- **A.-V. Andrei et al., talk** (3 June 2026, slides)
  - 11 models, four generations: econometric, e.g. autoregressive integrated moving average (ARIMA); deep learning; transformers; pretrained large language models (LLM): Chronos, TimesFM
- **Data and walk-forward validation**
  - Wholesale prices of 27 European Union (EU) countries, Ember (from ENTSO-E)
  - 96 steps in, 96 out; training 2000 hours (83 days), test 500 hours (20 days)
  - Symmetric mean absolute percentage error (SMAPE); score $(100 \times \text{SMAPE} + \text{RMSE})/2$
- **Results** (rank among the 11 models)
  - PatchTST (a transformer) is first or second in all 27 countries
  - Romania: PatchTST 1, TimesFM 2, Chronos 8, ARIMA 9
  - Germany: PatchTST 1, TimesFM 5, Chronos 10
  - Gains are model-specific, not uniform across the 4th generation
- **Link with this lecture**
  - On Romanian prices, Chronos-2 (2025) is our best single model (MAE 20.82)

In practice: compare model generations with walk-forward validation on several markets before choosing one.

(AIDA 2026 slides, Bucharest; paper: Andrei et al., 2024, arXiv)

Outline

- Machine-learning models and their evaluation
- Deep neural networks
- Gradient boosting and foundation models
- Model comparison, combination and computational cost
- Explaining the forecasts
- Spikes, negative prices and a case study
- Evaluation errors that overstate accuracy

Model ranking: Chronos-2 is the best single model

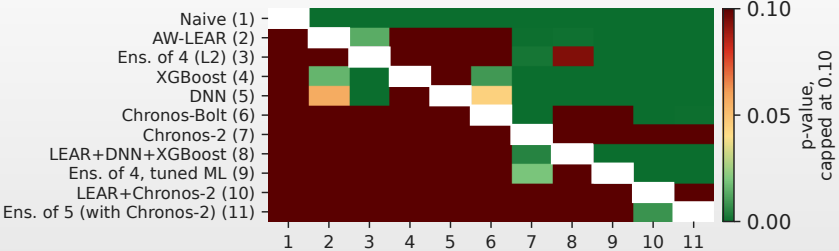
Model	MAE	RMSE	rMAE	vs AW-LEAR, %	DM p
Naive	27.82	44.54	1.000	−23.5	–
AW-ARX	22.30	36.34	0.802	+1.0	0.203
AW-LEAR	22.53	36.35	0.810	+0.0	–
Ensemble of 4 (Lecture 2)	22.13	35.76	0.795	+1.8	0.014
GBM (Lecture 2)	24.05	37.73	0.864	−6.7	1.000
MLP (Lecture 2)	25.54	40.05	0.918	−13.3	1.000
XGBoost, tuned	23.26	36.94	0.836	−3.2	0.984
DNN, tuned	22.95	36.42	0.825	−1.8	0.944
Chronos-Bolt, zero-shot	22.15	35.65	0.796	+1.7	0.220
Chronos-2, zero-shot	20.82	34.32	0.748	+7.6	<0.001
AW-LEAR + DNN + XGBoost	21.97	35.42	0.790	+2.5	0.001
AW-ARX + AW-LEAR + DNN + XGBoost	21.69	35.25	0.779	+3.8	<0.001
AW-LEAR + Chronos-2	20.81	34.11	0.748	+7.6	–
Ensemble of 5 (with Chronos-2)	21.12	34.50	0.759	+6.3	<0.001

- Root mean squared error (RMSE); relative MAE (rMAE) = MAE / MAE of the naive model; green = better, per column.
- DM p: one-sided test that the model is more accurate than AW-LEAR (daily mean absolute error, Newey–West variance, as in Lecture 2).

(own computation, test year 1 Nov 2025 – 31 Oct 2026)

Experiment yourself: notebook §5

DM tests: combinations with ML significantly outperform AW-LEAR



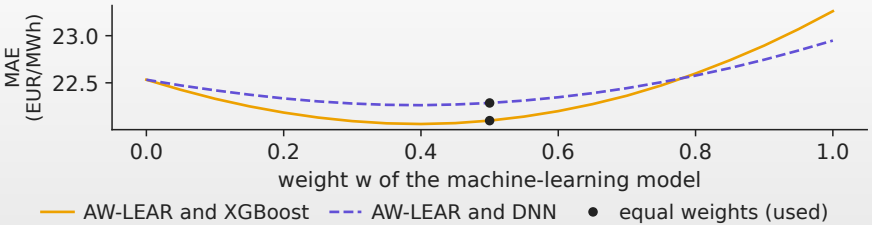
Source: energy-charts.info (Fraunhofer ISE); Yahoo Finance (EUA). Data: test year 2025-11-01 to 2026-10-31, 365 days; one-sided p-values.

- AW-LEAR + DNN + XGBoost beats AW-LEAR ($p = 0.001$)
  - Adding AW-ARX gives MAE 21.69, better than the Lecture 2 ensemble ($p < 0.001$).
- Adding AW-LEAR to Chronos-2 changes nothing ($p = 0.491$): on the hard days the two models make similar errors.

(own computation; one-sided DM tests)

EMQA_L7_dm_matrix

Forecast averaging and the differences between model errors



Source: energy-charts.info (Fraunhofer ISE); Yahoo Finance (EUA). Data: test year 2025-11-01 to 2026-10-31; illustration, the weight is not chosen on the test year.

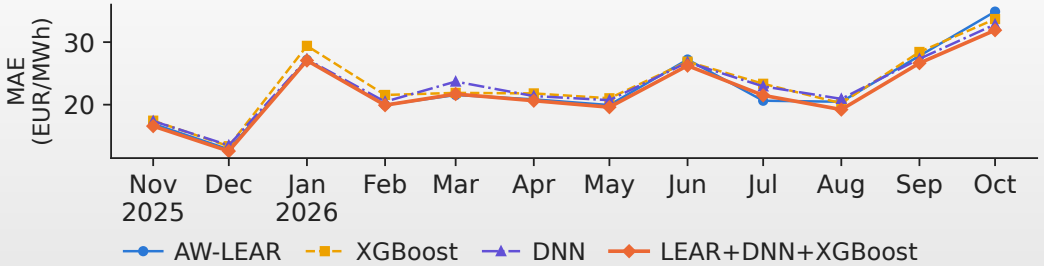
- Var $\left(\frac{e_1+e_2}{2}\right) = \frac{1}{4}(\sigma_1^2 + \sigma_2^2 + 2\rho\sigma_1\sigma_2)$: with correlation $\rho < 1$, the average errs less (e = error, σ = its standard deviation).
- Error correlation with AW-LEAR: 0.94 (DNN), 0.89 (XGBoost)
  - Equal weights do almost as well as the best weight seen afterwards (0.40).

- **In practice:** average with equal weights before estimating clever weights.

(Bates & Granger, 1969; own computation)

EMQA_L7_combo_weight

The combination outperforms AW-LEAR in 10 of 12 months



Source: energy-charts.info (Fraunhofer ISE); Yahoo Finance (EUA). Data: test year 2025-11-01 to 2026-10-31.

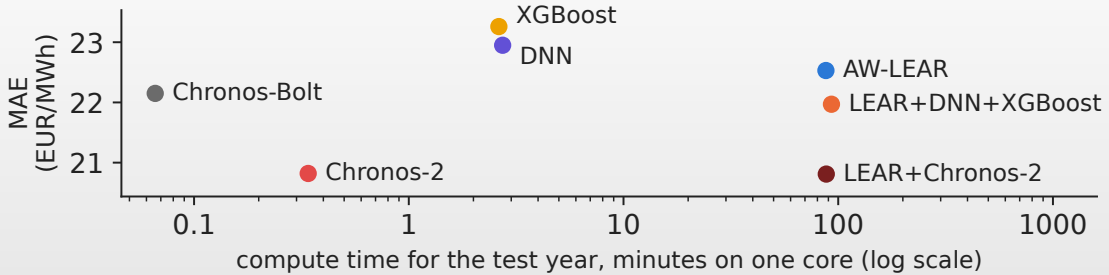
- Alone, XGBoost and the DNN each beat AW-LEAR in only 3 months
  - Averaged with it, in 10.
- All models err more in the same months (January, June, October): the hard days are hard for everyone.

- **In practice:** report accuracy month by month to the risk committee, not only the annual average.

(own computation, test year 1 Nov 2025 – 31 Oct 2026)

EMQA_L7_monthly_ml

Computational cost of daily LEAR and of the ML models



Source: energy-charts.info (Fraunhofer ISE); Yahoo Finance (EUA). Data: test year 2025-11-01 to 2026-10-31; one processor core.

- Test year, one core: AW-LEAR 87.6 min (365 days × 3 windows × 24 h)
  - XGBoost 2.6 min, DNN 2.7 min, Chronos-2 0.3 min
- Hidden costs: the search (7.3 min), the pretraining of Chronos, the people who run the pipeline.

- **In practice:** compare models on error per euro of total cost of ownership, not on error alone.

(own computation; times measured on one core of an 18-core laptop)

EMQA_L7_cost_gain

Outline

- Machine-learning models and their evaluation
- Deep neural networks
- Gradient boosting and foundation models
- Model comparison, combination and computational cost
- Explaining the forecasts
- Spikes, negative prices and a case study
- Evaluation errors that overstate accuracy

Shapley values split each forecast into the contributions of its inputs

- **In one sentence:** The contribution of an input is its average effect when it is added to the model, over all orders in which the inputs could be added.

$$\phi_j = \sum_{S \not\ni j} \frac{|S|!\,(p-|S|-1)!}{p!} \Big[v(S \cup \{j\}) - v(S) \Big]$$

$$\hat{f}(\mathbf{x}) = \phi_0 + \sum_j \phi_j$$

- **where**
  - ϕ_j = SHapley Additive exPlanations (SHAP) value of input j
  - S = a subset of the other inputs
  - $v(S)$ = expected forecast when only S is known
  - ϕ_0 = average forecast
  - $p = 20$ inputs.

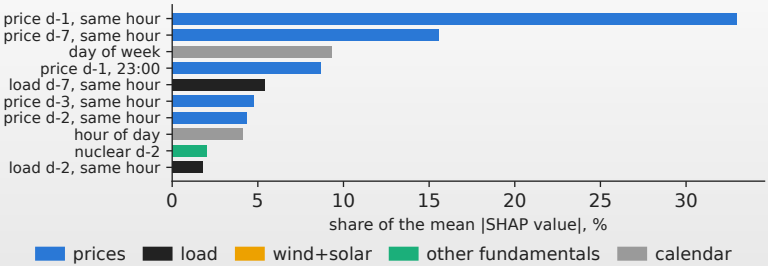
- **Why Shapley values**
  - The only fair split in the game-theory sense; they add up exactly to the forecast
- **On our data**
  - XGBoost, 8 760 test hours; largest gap between $\hat{f}$ and $\phi_0 + \sum \phi_j$: 0.000009
  - Past prices carry 69.3% of the total contribution

- **In practice:** ask for the explanation of each alarm, not only for a global ranking.

(Shapley, 1953; Lundberg et al., 2020, Nat. Mach. Intell.)

Experiment yourself: notebook §6

Main input of the tree models: yesterday's price at the same hour



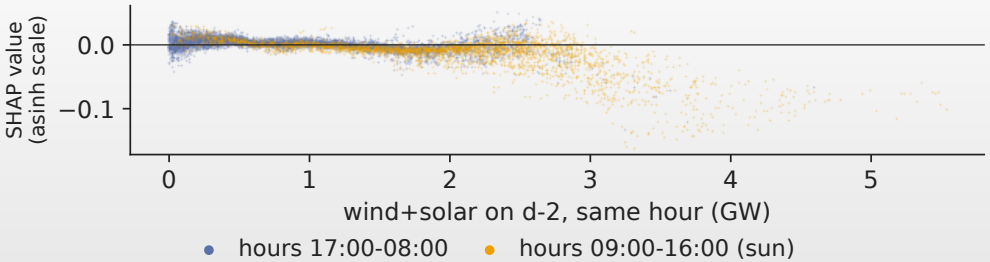
Source: energy-charts.info (Fraunhofer ISE); Yahoo Finance (EUA). Data: test year 2025-11-01 to 2026-10-31, 8760 hours.

- Top inputs: price $d-1$ same hour 32.9%, price $d-7$ same hour 15.6%, day of week 9.3% of the mean $|\phi|$.
- Load 8.5%, wind + solar 2.5%: known only up to $d-2$, they are old news at gate closure.

- **In practice:** buy better inputs first: the load and wind forecasts of the transmission system operator (TSO).

(own computation, test year 1 Nov 2025 – 31 Oct 2026) EMQA_L7_shap_global

Wind and solar lower the price only in the hours when they are large



Source: energy-charts.info (Fraunhofer ISE); Yahoo Finance (EUA). Data: test year 2025-11-01 to 2026-10-31, 8760 hours.

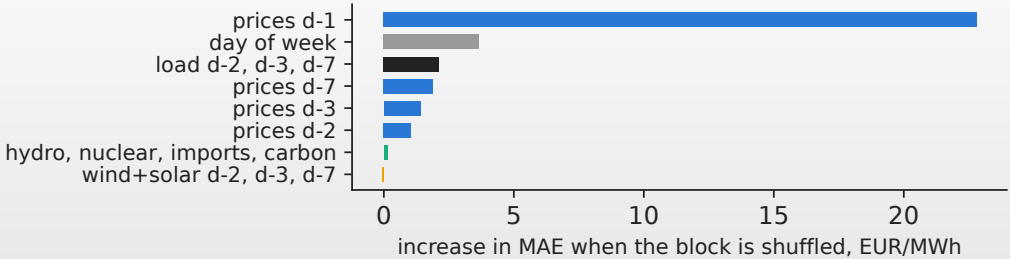
- Each point is one test hour
  - The more wind + solar on $d-2$ at that hour, the more negative its contribution (correlation -0.64)
- The effect appears above about 2 gigawatts (GW): a threshold that a linear model cannot express.

- **In practice:** a solar developer can read here the price discount that its own output helps to create.

(own computation, test year 1 Nov 2025 – 31 Oct 2026)

EMQA_L7_shap_res

The neural network relies even more on yesterday's prices



Source: energy-charts.info (Fraunhofer ISE); Yahoo Finance (EUA). Data: test year 2025-11-01 to 2026-10-31.

- Permutation importance I_b = rise in MAE when block b is shuffled across test days
  - 5 permutations, valid for any model
- Prices of $d-1$: +22.82 EUR/MWh
  - Day of week +3.66, load +2.12, wind + solar −0.08

- **In practice:** monitor the price feed of $d-1$ first: the network fails without it.

(Breiman, 2001, Mach. Learn.; own computation)

EMQA_L7_perm_dnn

Machine learning in energy finance: a systematic literature review

- **R. Clodnitchi, talk** (3 June 2026, slides)
  - Systematic review of artificial intelligence (AI) in energy finance (Anagnoste et al., 2025, *Energy Strategy Reviews*)
  - About 700 Web of Science articles screened, 232 peer-reviewed kept (2020–2025), on six pillars of energy finance
- **Where AI is used** (share of papers using AI/ML)
  - Pricing mechanisms 100%, energy derivatives 100%, energy and financial markets 97.5%, risk management 93.0%
  - Green finance 60.4%; corporate finance of energy firms 0% (panel econometrics only)
- **What is missing**
  - Explainable AI: 19.6% of the derivatives papers, 0–5% in the other pillars
  - Neural networks dominate: 88.2% of the derivatives papers, 72.1% in risk management
  - Proposed next step: explainable AI and causal machine learning for regulators and policy evaluation
- **Link with this lecture**
  - The SHAP and permutation charts of this section are the kind of explanation the review finds rare

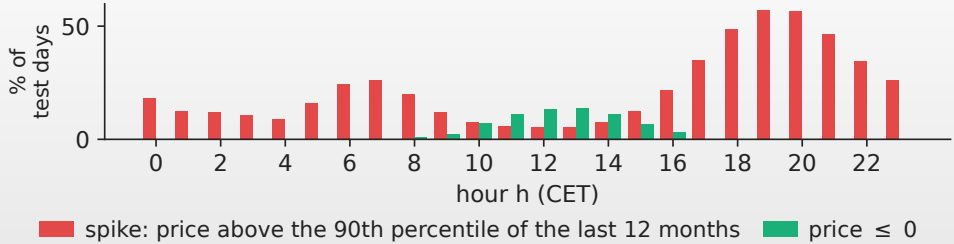
- **In practice:** ask for the explanation together with the forecast: regulators and boards will ask for it.

(AIDA 2026 slides, Bucharest; paper: Anagnoste et al., 2025, ESR)

Outline

- Machine-learning models and their evaluation
- Deep neural networks
- Gradient boosting and foundation models
- Model comparison, combination and computational cost
- Explaining the forecasts
- Spikes, negative prices and a case study
- Evaluation errors that overstate accuracy

Spikes come in the evening, negative prices at midday



Source: energy-charts.info (Fraunhofer ISE): OPCOM prices. Data: test year 2025-11-01 to 2026-10-31.

- **Spike**: price above θ, the 90th percentile of the hourly prices of the 12 months before the delivery month
  - θ from 166.8 to 202.4 EUR/MWh; 22.1% of test hours are spikes, most at 19:00 (57.0% of days)
- **Price $\leq$ 0**: 255 hours on 53 days of the test year, mostly around 13:00.

- **In practice:** a battery charges at midday and discharges at 19:00–20:00: the two events are its business model.

(own computation, test year 1 Nov 2025 – 31 Oct 2026) EMQA_L7_events_hour

Spike classifiers and spike detection from price forecasts

- **In one sentence:** A classifier gives the probability that an hour is a spike; we judge how well it ranks the hours and how honest its probabilities are.

$$\hat{\pi}_{d,h} = \Pr\left(P_{d,h} > \theta_d \mid \mathbf{x}_{d,h}\right)$$

$$\text{Brier} = \frac{1}{n}\sum_{d,h}\left(\hat{\pi}_{d,h} - S_{d,h}\right)^2$$

- **where**
  - $S_{d,h} = 1$ if the hour is a spike, 0 otherwise
  - Precision = share of alerts that were spikes
  - Recall = share of spikes that got an alert
  - F1 = their harmonic mean
  - Area under the curve (AUC) = probability that a spike hour gets a higher score than a normal hour.

- **On our data**: XGBoost and logistic classifiers, refitted monthly
  - Spikes in 22.1% of test hours, 14.3% in the last training sample
  - Brier score: XGBoost 0.072; constant share of spikes 0.172
- **Benchmarks**
  - Persistence: share of spikes at the same hour in the last 7 days
  - The AW-LEAR price forecast itself, compared with θ

- **In practice:** judge an alert by its probabilities, not only by how many spikes it catches.

(Brier, 1950; Fawcett, 2006)

Experiment yourself: notebook §7

Alerts on the test year: AW-LEAR matches a dedicated classifier

Model	AUC	Precision	Recall	F1	Brier	AUC	F1
XGBoost	0.953	0.826	0.685	0.749	0.072	0.980	0.571
Logistic regression	0.953	0.847	0.641	0.730	0.072	0.980	0.581
Persistence, 7 days	0.918	0.761	0.681	0.719	0.083	0.847	0.205
AW-LEAR forecast $> \theta$	0.959	0.851	0.733	0.788	–	0.983	0.558

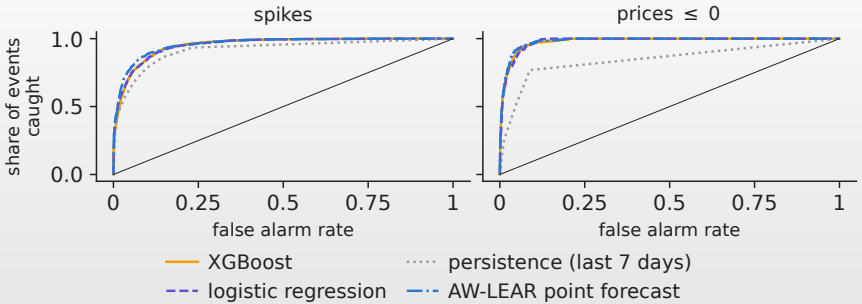
- Columns 2–6: spikes (1 938 hours); columns 7–8: prices ≤ 0 (255 hours). Alert when probability > 0.5; AW-LEAR: alert when its price forecast exceeds θ (or is ≤ 0).
- For negative prices, XGBoost catches only 0.467 of the events at threshold 0.5: these hours are rare and depend on tomorrow's weather.

- **In practice:** buy a spike-alert service only if it beats the spike signal already hidden in your price forecast.

(own computation, test year 1 Nov 2025 – 31 Oct 2026)

Experiment yourself: notebook §7

Persistence detects spikes, but not negative prices



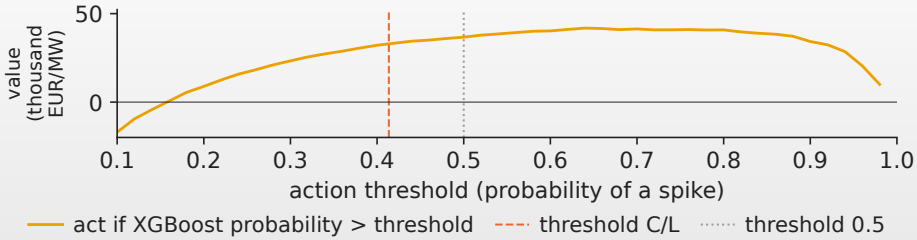
Source: energy-charts.info (Fraunhofer ISE); Yahoo Finance (EUA). Data: test year 2025-11-01 to 2026-10-31, 8760 hours.

- Receiver operating characteristic (ROC) curves
  - Spikes: AUC from 0.918 (persistence) to 0.959 (AW-LEAR), since the evening spike repeats day after day.
- Negative prices: persistence 0.847 against 0.980 for XGBoost
  - They depend on tomorrow's sun and wind, not on yesterday.

(own computation, test year 1 Nov 2025 – 31 Oct 2026)

EMQA_L7_roc

Cost–loss rule: act when the spike probability exceeds C/L



Source: energy-charts.info (Fraunhofer ISE); Yahoo Finance (EUA). Data: test year 2025-11-01 to 2026-10-31; 1 MW of flexible load.

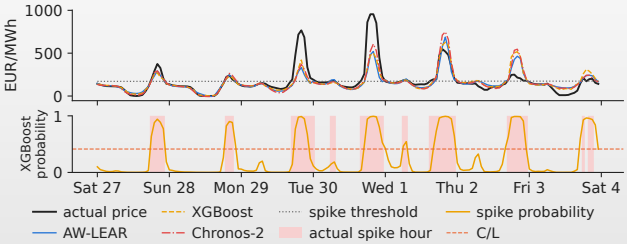
- **Rule**: act if $\hat{\pi}_{d,h} > C/\bar{L} = 30/72.5 = 0.414$
  - C = cost of acting = 30 EUR/MWh (assumed): moving 1 MWh out of an alerted hour earns $P_{d,h} - \theta_d - C$
  - $\bar{L}$ = cost of a missed spike = mean excess over θ in the validation year
- Value per MW and year: 33.0 thousand EUR at C/L, 11.8 with persistence, 62.4 with perfect foresight.

- **In practice:** set the threshold from your costs, not from hindsight (best threshold ex post: 0.64).

(Murphy, 1977, MWR; own computation)

EMQA_L7_decision

Case study, 30 Jun 2026: the models anticipated the spike but underestimated its size



Source: energy-charts.info (Fraunhofer ISE); Yahoo Finance (EUA). Data: 2026-06-27 to 2026-07-03, hourly.

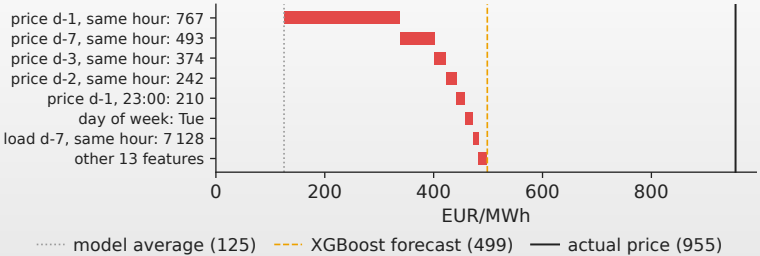
- **Hour 20:00 CET**: 954.59 EUR/MWh, the highest price of the test year
  - Forecasts: AW-LEAR 520, XGBoost 499, DNN 614, Chronos-2 607; spike probability 0.992
- In that week: 47 spike hours, 45 alerts at C/L, 39 of them correct.

- **In practice:** the probability says whether to act, the point forecast how much.

(own computation; Romanian day-ahead prices)

EMQA_L7_case_week

Case study: the trees followed the extreme evening of the day before



Source: energy-charts.info (Fraunhofer ISE); Yahoo Finance (EUA). Data: 2026-06-30, hour 20:00 CET; contributions rescaled to EUR/MWh (Appendix C).

- From 125 (model average) to 499 EUR/MWh: price of $d-1$, same hour, +213
  - Price of $d-7$ +64.
- Trees cannot go above the prices stored in their leaves: extremes are capped.

- **In practice:** show the board this breakdown for every large alert.

(own computation; SHAP values in EUR/MWh, Appendix C) EMQA_L7_shap_local

Research from our group on spike early warning: paper and replication

- **What the paper did** (Andrei, Pele, Cramer, Amza & Diaconu, 2026)
  - Data: 5,000 hours of the Romanian day-ahead market in 2024, with load, renewable and fuel data; spike = price above the 90th percentile
  - Method: random forests and XGBoost explained with SHAP; an early-warning score built from the thresholds of several indicators
  - Result: the previous price and the residual load forecast are the main drivers of spikes
- **Our replication** (case study C7, gate-closure information only)
  - Test period Jul 2025 – Oct 2026; spikes in 22.1% of hours (training: 10.0%)
  - XGBoost AUC 0.946; top drivers: yesterday's and last week's price at the same hour, as on the previous slides
- Paper results as stated in the paper; replication numbers computed by the C7 case notebook on our data.

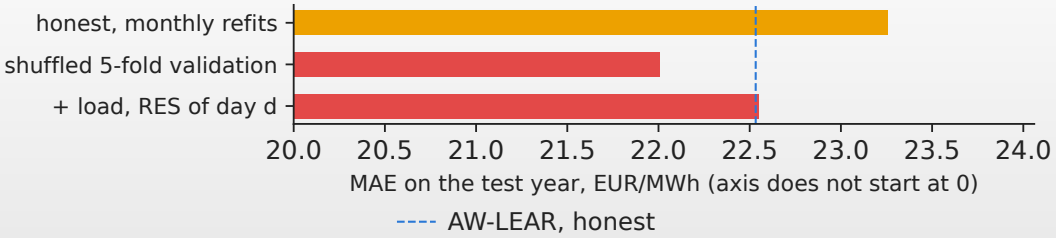
- **In practice:** a residual-load forecast is the input that a spike model needs most and that our public data lack.

(Andrei, Pele, Cramer, Amza & Diaconu, 2026, PICBE; case study C7)

Outline

- Machine-learning models and their evaluation
- Deep neural networks
- Gradient boosting and foundation models
- Model comparison, combination and computational cost
- Explaining the forecasts
- Spikes, negative prices and a case study
- Evaluation errors that overstate accuracy

Look-ahead bias: two invalid evaluations overstate the accuracy of XGBoost



Source: energy-charts.info (Fraunhofer ISE); Yahoo Finance (EUA). Data: test year 2025-11-01 to 2026-10-31.

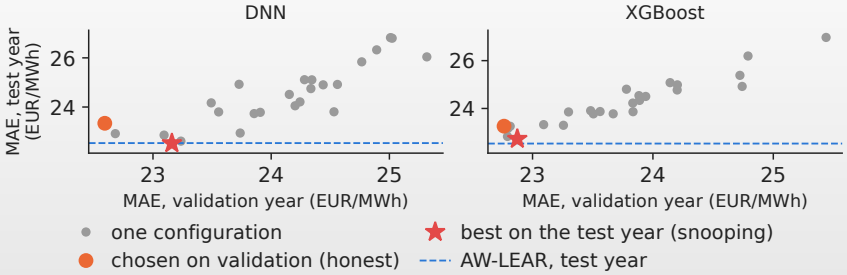
- **Shuffled 5-fold cross-validation**: future days end up in the training sample
  - MAE 22.01 instead of 23.26: 5.4% better, on paper only
- **Load and renewable energy sources (RES) of day d as inputs**: unknown at 12:00 on $d-1$
  - MAE 22.55, an apparent gain of 3.1%: the most that perfect forecasts of these inputs could add

In practice: ask every vendor for a walk-forward test with each input time-stamped at gate closure.

(Bergmeir & Benítez, 2012, Inf. Sci.; own computation)

EMQA_L7_pit_leak

Data snooping: selection on the test year gives a spurious gain over LEAR



Source: energy-charts.info (Fraunhofer ISE); Yahoo Finance (EUA). Data: validation and test year 2025-11-01 to 2026-10-31.

- Honest DNN (chosen on validation, 1 seed): MAE 23.34
  - Best of 25 settings on the test year: 22.52, just below AW-LEAR (22.53).
- Validation works, but not perfectly: Spearman rank correlation 0.86 (DNN), 0.94 (XGBoost).

- **In practice:** ask how many models were tried: the more trials, the more the best one flatters.

(White, 2000, Econometrica; own computation)

EMQA_L7_snooping

The best of many models must be tested against the benchmark

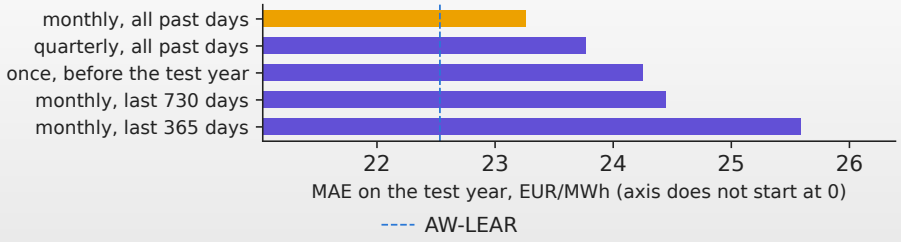
$$\mathrm{RC}_n = \max_{k=1,\dots,m} \sqrt{n}\,\bar{d}_k, \qquad \bar{d}_k = \frac{1}{n}\sum_{t=1}^{n} d_{k,t}$$

- **where**
  - $d_{k,t}$ = benchmark loss minus loss of model k on day t
  - H_0: $E(d_{k,t}) \le 0$ for all k
  - Critical value: joint bootstrap of $(d_{1,t}, \dots, d_{m,t})$, which keeps the correlation between models.
- **Why**: for m independent 5% tests, P(at least one false win) $= 1 - 0.95^m$: 0.401 ($m = 10$), 0.994 ($m = 100$).
- **C.-M. Kuan, 4 June 2026** (slides)
  - Reality Check (RC), White (2000): conservative, poor models added to the set reduce its power (Hansen, 2005)
  - Superior predictive ability (SPA) test, Hansen (2005): recentres the poor models, more powerful
  - Step-RC (Romano & Wolf, 2005), Step-SPA (Hsu, Hsu & Kuan, 2010): name the winners, familywise error rate (FWER) under control
- **On our data**
  - The best of 25 DNN settings on the test year is just below AW-LEAR: the selection RC and SPA correct for

- **In practice:** when shown the best of many models, ask for an RC or SPA p-value against your benchmark.

(AIDA 2026 slides; papers: White, 2000; Hansen, 2005; Romano & Wolf, 2005; Hsu et al., 2010)

Retraining: accuracy loss of 4.2% after a year without updating



Source: energy-charts.info (Fraunhofer ISE); Yahoo Finance (EUA). Data: test year 2025-11-01 to 2026-10-31.

- Fitted once, before the test year: MAE 24.25
  - The loss grows with time (+0.15 EUR/MWh in the first 3 months, +3.70 in the last 3).
- A short rolling window (365 days, MAE 25.59) is worse than the whole history: the trees need many past spikes to learn from.

- **In practice:** put the monthly refit in the operating calendar, with an owner, like any other control.

(own computation, test year 1 Nov 2025 – 31 Oct 2026) EMQA_L7_pit_retrain

Leakage check: forecasts recomputed without future data are unchanged

- **The test**
  - For the first day of each of the 12 test months, all data after 12:00 on $d-1$ are deleted, then replaced by 10 000 EUR/MWh
  - XGBoost, the DNN and the spike classifier are refitted and must give the same forecasts
- **Result**
  - Trees and classifier: identical; DNN: differences below 0.001 EUR/MWh (rounding in parallel arithmetic)
  - Chronos: the context is built only from days before day d, checked in the code
- **Checklist before trusting a model**
  - Every input time-stamped at gate closure
  - Settings chosen on a validation year
  - One test year, opened once, with DM tests against LEAR
  - Monthly refits and a perturbation test like this one

In practice: make this check part of every model review, for statistical and machine-learning models alike.

(Lecture 2; Lago et al., 2021, APEN)

Experiment yourself: notebook §10

Quiz 1: a perfect backtest

- A start-up shows a neural network with rMAE 0.70 on Romania, tested with shuffled 5-fold cross-validation
  - What do you ask for?

Answer

- A walk-forward test on one untouched year, with inputs known at 12:00 on $d-1$: on our data, shuffling alone made XGBoost look 5.4% better.

Quiz 2: choosing the model to deploy

- Chronos-2 has the lowest MAE of the single models
  - AW-LEAR + Chronos-2 has the same MAE
  - Which do you deploy?

Answer

- The combination: same accuracy (DM $p = 0.491$), but if the pretrained model degrades or becomes unavailable, LEAR still carries half of the forecast.

Quiz 3: reading SHAP

- Load has a small SHAP share (8.5%)
  - Can we conclude that load does not affect the price?

Answer

- No: SHAP explains the model, not the market
  - Load enters only with its $d-2$ value, already reflected in yesterday's price.

Quiz 4: an alert threshold

- Moving 1 MWh costs you 60 EUR instead of 30
  - How does the alert threshold change?

Answer

- It doubles: C/L rises from 0.414 to about twice that value, so you act only on the more certain spikes.

Quiz 5: retraining

- Your data team proposes to train the model once a year to save effort
  - What does the test year say?

Answer

- A model frozen for the year was 4.2% less accurate, and the gap widened in the last months
  - A monthly refit of XGBoost takes about 13 seconds.

Wrap-up

- Machine-learning models and their evaluation
- Deep neural networks
- Gradient boosting and foundation models
- Model comparison, combination and computational cost
- Explaining the forecasts
- Spikes, negative prices and a case study
- Evaluation errors that overstate accuracy

- Tuned, the DNN and XGBoost improve on Lecture 2 but do not beat AW-LEAR alone
  - Averaged with it, they do.
- Chronos-2, with no training, is the most accurate single model (MAE 20.82, 7.6% below AW-LEAR)
  - Every model must be explained, validated and refitted.

Research from our group on machine learning for Romanian power prices

- **Early warning of price spikes** (Andrei, Pele, Cramer, Amza & Diaconu, 2026, PICBE): built an explainable artificial intelligence (AI) system for the Romanian day-ahead market
  - The previous price and the residual load drive the spikes — case study C7.
- **Explaining price forecasts across horizons** (Andrei, Pele & Cramer, 2026, PICBE): explained machine-learning forecasts of Romanian hourly prices and showed which drivers matter at each horizon and in each price regime.
- **Four generations of price models** (Andrei, Velev, Toma, Pele & Lessmann, 2024, arXiv 2411.03372): compared four generations of forecasting methods for energy prices, from statistical models to deep learning, as a guide for choosing a model family.
- **Foundation models for tail risk** (Jheng, Tak, Găman, Mazurencu-Marinescu-Pele & Pele, 2026, ECECSR): used pretrained foundation models, without retraining and with conformal calibration, to forecast extreme losses in Central and Eastern European markets.

- Findings as stated in the papers; numbers on the other slides come from our own data.

(Bucharest University of Economic Studies; Romanian Academy, Institute for Economic Forecasting)

Further reading and resources

- **epftoolbox**: the DNN and LEAR of Lago et al. (2021) with open benchmark data — github.com/jeslago/epftoolbox.
- **XGBoost** and **shap** documentation: xgboost.readthedocs.io, shap.readthedocs.io.
- **Chronos** foundation models: github.com/amazon-science/chronos-forecasting.
- **Interpretable Machine Learning** (C. Molnar), free online: christophm.github.io.
- **This course**: code, data, notebooks — github.com/danpele/EMQA.
- **Case study C7** replicates the spike early-warning paper on our data.
- **Experiment yourself**: the notebook reproduces every chart and number of this lecture — notebooks/.

References (1/4)

- Anagnoste, S., Andrei, A.-V., Bolovăneanu, V., Cepoi, C.-O., Clodnitchi, R., Cramer, A.-A., Grecu, R.-A., Lessmann, S., Pele, D. T., Petukhina, A. & Strat, V. A. (2025). The role of AI in (re)shaping energy finance: A systematic literature review. *Energy Strategy Reviews*, 61, 101833. doi:10.1016/j.esr.2025.101833
- Andrei, A.-V., Pele, D.-T. & Cramer, A.-A. (2026). Multi-horizon explainability in energy price forecasting: Disentangling market dynamics and renewable intermittency in Romania's liberalized electricity market. *Proceedings of the International Conference on Business Excellence*, 20(1), 104–117. doi:10.2478/picbe-2026-0014
- Andrei, A.-V., Pele, D.-T., Cramer, A.-A., Amza, A. & Diaconu, D. (2026). Explainable AI-based early warning system for electricity price spikes: Evidence from the Romanian market. *Proceedings of the International Conference on Business Excellence*, 20(1), 4820–4836. doi:10.2478/picbe-2026-0352
- Andrei, A.-V., Velev, G., Toma, F.-M., Pele, D. T. & Lessmann, S. (2024). Energy price modelling: A comparative evaluation of four generations of forecasting methods. arXiv:2411.03372
- Andrei, A.-V., Velev, G., Toma, F.-M., Pele, D. T. & Lessmann, S. (2026). Energy price modelling: A comparative evaluation of four generations of forecasting methods. Talk, AIDA Conference, Bucharest University of Economic Studies, 3 June 2026. github.com/danpele/EMQA/resources/aida2026
- Ansari, A. F., Stella, L., Turkmen, C. 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. et al. (2025). Chronos-2: From univariate to universal forecasting. arXiv:2510.15821
- Bates, J. M. & Granger, C. W. J. (1969). The combination of forecasts. *Journal of the Operational Research Society*, 20(4), 451–468. doi:10.1057/jors.1969.103
- Bergmeir, C. & Benítez, J. M. (2012). On the use of cross-validation for time series predictor evaluation. *Information Sciences*, 191, 192–213. doi:10.1016/j.ins.2011.12.028

References (2/4)

- Bergstra, J. & Bengio, Y. (2012). Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, 13, 281–305. jmlr.org/papers/v13/bergstra12a.html
- Breiman, L. (2001). Random forests. *Machine Learning*, 45(1), 5–32. doi:10.1023/A:1010933404324
- Brier, G. W. (1950). Verification of forecasts expressed in terms of probability. *Monthly Weather Review*, 78(1), 1–3. doi:10.1175/1520-0493(1950)078<0001:VOFEIT>2.0.CO;2
- Chen, T. & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. *Proceedings of the 22nd ACM SIGKDD Conference*, 785–794. doi:10.1145/2939672.2939785
- Clodnitchi, R. (2026). The role of AI in (re)shaping energy finance: A systematic literature review. Talk, AIDA Conference, Bucharest University of Economic Studies, 3 June 2026. github.com/danpele/EMQA/resources/aida2026
- 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
- Fawcett, T. (2006). An introduction to ROC analysis. *Pattern Recognition Letters*, 27(8), 861–874. doi:10.1016/j.patrec.2005.10.010
- Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. *Annals of Statistics*, 29(5), 1189–1232. doi:10.1214/aos/1013203451
- Hansen, P. R. (2005). A test for superior predictive ability. *Journal of Business & Economic Statistics*, 23(4), 365–380. doi:10.1198/073500105000000063

References (3/4)

- Hastie, T., Tibshirani, R. & Friedman, J. (2009). *The Elements of Statistical Learning*, 2nd ed. Springer. doi:10.1007/978-0-387-84858-7
- Hsu, P.-H., Hsu, Y.-C. & Kuan, C.-M. (2010). Testing the predictive ability of technical analysis using a new stepwise test without data snooping bias. *Journal of Empirical Finance*, 17(3), 471–484. doi:10.1016/j.jempfin.2010.01.001
- Jheng, S., Tak, R., Găman, Ş., Mazurencu-Marinescu-Pele, M. & Pele, D. T. (2026). Can foundation models manage risk? Zero-shot VaR and ES forecasting with conformal calibration in CEE markets. *Economic Computation and Economic Cybernetics Studies and Research*, 60(2), 267–290. doi:10.24818/18423264/60.2.26.14
- Kingma, D. P. & Ba, J. (2015). Adam: A method for stochastic optimization. *International Conference on Learning Representations*. arXiv:1412.6980
- Kuan, C.-M. (2026). Large-scale multiple inequality testing without data snooping bias. Keynote, AIDA Conference, Bucharest University of Economic Studies, 4 June 2026. github.com/danpele/EMQA/resources/aida2026
- Lago, J., De Ridder, F. & De Schutter, B. (2018). Forecasting spot electricity prices: Deep learning approaches and empirical comparison of traditional algorithms. *Applied Energy*, 221, 386–405. doi:10.1016/j.apenergy.2018.02.069
- Lago, J., Marcjasz, G., De Schutter, B. & Weron, R. (2021). Forecasting day-ahead electricity prices: A review of state-of-the-art algorithms, best practices and an open-access benchmark. *Applied Energy*, 293, 116983. doi:10.1016/j.apenergy.2021.116983
- Lundberg, S. M., Erion, G., Chen, H. et al. (2020). From local explanations to global understanding with explainable AI for trees. *Nature Machine Intelligence*, 2(1), 56–67. doi:10.1038/s42256-019-0138-9
- Marcjasz, G., Narajewski, M., Weron, R. & Ziel, F. (2023). Distributional neural networks for electricity price forecasting. *Energy Economics*, 125, 106843. doi:10.1016/j.eneco.2023.106843

References (4/4)

- Murphy, A. H. (1977). The value of climatological, categorical and probabilistic forecasts in the cost-loss ratio situation. *Monthly Weather Review*, 105(7), 803–816. doi:10.1175/1520-0493(1977)105<0803:TVOCCA>2.0.CO;2
- Olivares, K. G., Challu, C., Marcjasz, G., Weron, R. & Dubrawski, A. (2023). Neural basis expansion analysis with exogenous variables: Forecasting electricity prices with NBEATSx. *International Journal of Forecasting*, 39(2), 884–900. doi:10.1016/j.ijforecast.2022.03.001
- Romano, J. P. & Wolf, M. (2005). Stepwise multiple testing as formalized data snooping. *Econometrica*, 73(4), 1237–1282. doi:10.1111/j.1468-0262.2005.00615.x
- Shapley, L. S. (1953). A value for n-person games. In *Contributions to the Theory of Games II*, 307–317. Princeton University Press. doi:10.1515/9781400881970-018
- Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I. & Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15, 1929–1958. jmlr.org/papers/v15/srivastava14a.html
- Weron, R. (2014). Electricity price forecasting: A review of the state-of-the-art with a look into the future. *International Journal of Forecasting*, 30(4), 1030–1081. doi:10.1016/j.ijforecast.2014.08.008
- Weron, R. (2026). Machine learning for electricity price forecasting. Keynote, AIDA Conference, Bucharest University of Economic Studies, 4 June 2026. github.com/danpele/EMQA/resources/aida2026
- White, H. (2000). A reality check for data snooping. *Econometrica*, 68(5), 1097–1126. doi:10.1111/1468-0262.00152

Appendix A: back-propagation and the Adam update

$$\frac{\partial \mathcal{L}}{\partial W_2} = \boldsymbol{\delta}\, \mathbf{z}^\top, \quad \boldsymbol{\delta} = \frac{\partial \mathcal{L}}{\partial \hat{\mathbf{Y}}}$$

$$\frac{\partial \mathcal{L}}{\partial W_1} = \left[(W_2^\top \boldsymbol{\delta}) \odot \phi'(\mathbf{u}) \right] \mathbf{x}^\top$$

- Chain rule, from the output back to the inputs
  - For the absolute error $\partial |y - \hat{y}| / \partial \hat{y} = -\operatorname{sign}(y - \hat{y})$.
- $\mathbf{u} = W_1 \mathbf{x} + \mathbf{b}_1$
  - $\odot$ = element-wise product
  - For softplus, $\phi'(u) = 1/(1 + e^{-u})$.

$$m_k = \beta_1 m_{k-1} + (1 - \beta_1) g_k$$

$$v_k = \beta_2 v_{k-1} + (1 - \beta_2) g_k^2$$

$$\theta_k = \theta_{k-1} - \eta \, \frac{m_k / (1 - \beta_1^k)}{\sqrt{v_k / (1 - \beta_2^k)} + \varepsilon}$$

- g_k = gradient on mini-batch k
  - $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\varepsilon = 10^{-8}$ (defaults)
  - Each weight gets its own step size.
- Weight decay adds $\lambda\theta$ to g_k (a quadratic penalty, here $\lambda = 10^{-5}$).

(Kingma & Ba, 2015, ICLR; Hastie et al., 2009)

Appendix B: derivation of the optimal leaf value $-G/(H+\lambda)$

$$\ell\big(Y_i, F_{m-1} + f\big) \approx \ell_i + g_i f(\mathbf{x}_i) + \tfrac{1}{2} h_i f(\mathbf{x}_i)^2$$

$$\sum_j \left[G_j w_j + \tfrac{1}{2}(H_j + \lambda) w_j^2 \right] + \gamma J, \quad G_j = \textstyle\sum_{i \in R_j} g_i$$

$$w_j^* = -\frac{G_j}{H_j + \lambda}, \qquad \text{score} = -\tfrac{1}{2} \sum_j \frac{G_j^2}{H_j + \lambda} + \gamma J$$

- Second-order Taylor expansion of the loss around the current forecast
  - g_i, h_i = first and second derivatives
  - H_j = sum of h_i in leaf j.
- Within a leaf the objective is a parabola in w_j: setting its derivative to zero gives w_j^*.
- The split gain of the regression-tree slide is the score before the split minus the score after it.
- For the absolute error $h_i = 0$
  - XGBoost then sets the leaf values by a line search.

(Chen & Guestrin, 2016, KDD)

Appendix C: SHAP values in EUR/MWh

$$\hat{P} = m + s \sinh\left(\phi_0 + \sum_j \phi_j\right), \qquad P_0 = m + s \sinh \phi_0$$

$$\phi_j^{\text{EUR}} = \phi_j \frac{\hat{P} - P_0}{\sum_k \phi_k} \Rightarrow \sum_j \phi_j^{\text{EUR}} = \hat{P} - P_0$$

- The trees forecast the asinh-scaled price
  - m, s = median and scaled median absolute deviation (MAD) of the training prices.
- SHAP values add up on that scale
  - To report euros, the euro gap $\hat{P} - P_0$ is shared in proportion to ϕ_j.
- Signs and ranking are unchanged
  - The contributions add up exactly to the forecast in EUR/MWh.

(Lundberg et al., 2020, Nat. Mach. Intell.)

Appendix D: search spaces and the chosen settings

DNN setting	Values tried	Chosen
neurons, layer 1	64, 128, 256, 512	512
neurons, layer 2	0, 32, 64, 128, 256	none
activation	ReLU, softplus	softplus
dropout	0.0; 0.1; 0.2; 0.3	0.1
learning rate	$3 \cdot 10^{-4}$; 10^{-3}; $3 \cdot 10^{-3}$	$3 \cdot 10^{-4}$
L1 penalty, inputs	0; 10^{-3}; 10^{-2}; 10^{-1}	0
batch (days)	32, 64, 128	64
load/RES/fund.	yes or no	yes/yes/yes

XGBoost setting	Values tried	Chosen
trees	300, 600, 1000, 1500	1500
learning rate η	0.02; 0.05; 0.10	0.02
depth	3, 4, 5, 6, 8	8
min. leaf weight	1, 5, 20, 50	5
rows per tree	0.6; 0.8; 1.0	0.8
λ	0.1; 1.0; 10.0	0.1
loss	square, abs., Huber	absolute

- 24 random draws per model (fixed seed) + the Lecture 2 setting; chosen = lowest MAE on the validation year.

(Bergstra & Bengio, 2012, JMLR)