Time Series Forecasting: From VAR to Transformers and Diffusion

A Half-Day Short Course — Instructor Outline

Audience. Graduate students, ML engineers, and data scientists comfortable with basic deep learning (backpropagation, embeddings, attention at a high level).
Format. Lecture with live demos and a hands-on lab. Five modules of ∼50–60 minutes, one 10-minute break before the diffusion module.

Learning Objectives

By the end of the course, participants will be able to:

1. Specify and estimate classical multivariate models (VAR, VARMA), and interpret impulse-response functions, Granger-causality tests, and cointegration/error-correction models.
2. Formulate point and probabilistic forecasting problems and choose appropriate evaluation metrics (MASE, sMAPE, CRPS, quantile loss).
3. Explain self-attention and the transformer block from first principles, describe how transformers were adapted to time series (Informer, Autoformer, PatchTST, iTransformer), and explain why simple linear baselines forced a rethink of the field.
4. Describe how time series foundation models (Chronos-2, TimesFM 2.5, Moirai-2, Toto, TimeGPT, Lag-Llama) tokenize time and deliver zero-shot forecasts, and when to use them versus task-specific models.
5. Derive the intuition behind denoising diffusion models (forward noising, reverse denoising, the ϵ-prediction objective, conditioning and guidance) and explain how TimeGrad, CSDI, SSSD, TimeDiff, and TSDiff apply them to probabilistic forecasting.
6. Run a zero-shot foundation model and a diffusion forecaster on a real dataset, backtest them correctly, and select a model class for a given production scenario.

1 Module 1 (0:00–0:50) — Classical Multivariate Models

Time	Topic	Notes
0:00–0:08	Welcome, roadmap, objectives	Poll the room; from univariate AR to multivariate VAR; the three questions (prediction, propagation, equilibrium).
0:08–0:20	VAR(p): definition and estimation	$\boldsymbol{y}_t = \sum_i A_i \boldsymbol{y}_{t-i} + \boldsymbol{\varepsilon}_t$, $\boldsymbol{\varepsilon}_t \sim \text{iid}\,\mathcal{N}(0, W)$; equation-by-equation OLS; stability via roots outside the unit circle; vector MA(∞) representation.
0:20–0:28	VARMA models	Adding MA terms $B_j \boldsymbol{\varepsilon}_{t-j}$; assumptions (no serial correlation, conditional homoskedasticity); parsimony vs harder estimation.
0:28–0:36	Impulse-response analysis	Invert to MA(∞); $B_{ij}^{(h)}$ = effect on $y_{t+h}^{(i)}$ of a unit shock to $\varepsilon_t^{(j)}$; reading an IRF plot.
0:36–0:43	Granger causality	Does x's past help predict y? $H_0 : b_1 = \cdots = b_p = 0$; F-/Wald test; predictive, not structural.
0:43–0:50	Unit roots, cointegration, ECM	$I(1)$ and spurious regression; cointegration $\boldsymbol{\beta}^\top \boldsymbol{y}_t \sim I(0)$; ECM $\Delta \boldsymbol{y}_t = -\alpha \boldsymbol{\beta}^\top \boldsymbol{y}_{t-1} + \sum_j B_j \Delta \boldsymbol{y}_{t-j} + \boldsymbol{\varepsilon}_t$; **bridge:** from A_i to attention.

2 Module 2 (0:50–1:50) — The Transformer Era

Time	Topic	Notes
0:50–0:58	The forecasting problem & baselines	Point vs probabilistic; horizons, covariates; ARIMA/ETS, Theta, gradient-boosted trees; M4/M5 lessons — never skip the seasonal-naive baseline.
0:58–1:12	**Transformer basics (math)**	Self-attention derived step by step: $Q = XW_Q$, $K = XW_K$, $V = XW_V$; scores $S = QK^\top/\sqrt{d_k}$ and the $\sqrt{d_k}$ variance argument; row-wise softmax; $Z = AV$; causal masking. Multi-head with $d_k = d/h$; pre-norm layer; sinusoidal positional encoding; $\mathcal{O}(L^2 d)$ cost.
1:12–1:20	Why time series stress-test transformers	Quadratic cost; weak point-wise tokens; channel mixing vs independence; distribution shift and RevIN.
1:20–1:30	The efficiency race (2019–2022)	Informer (ProbSparse), Autoformer (decomposition + auto-correlation), FEDformer (frequency domain).
1:30–1:40	The DLinear shock (2022)	A one-layer linear model beat them on long-horizon benchmarks. Diagnosis: weak tokens, leaky benchmarks, error accumulation.
1:40–1:50	The fix: patching	PatchTST (patches as tokens + channel independence), iTransformer (attend across variates).

3 Module 3 (1:50–2:45) — Time Series Foundation Models

Time	Topic	Notes
1:50–2:00	The zero-shot paradigm	From "train a model per series" to "select a pre-trained model"; analogy to LLMs; LOTSA-scale corpora (∼27B observations); synthetic pretraining data.
2:00–2:14	Tokenizing time	Two camps: quantize values into a vocabulary (Chronos-T5 — forecasting as language modeling) vs real-valued patch embeddings (TimesFM, Moirai, Chronos-2). Output heads: autoregressive sampling, direct quantiles, parametric mixtures, generative heads.
2:14–2:28	The model zoo (2026)	Chronos-2 (Amazon, encoder-only, multivariate + covariates), TimesFM 2.5 (Google, ∼200M params, 16k context), Moirai-2 (Salesforce, decoder-only, any-variate) and Moirai-MoE, Toto (Datadog, observability), TimeGPT (Nixtla, API), Lag-Llama, Time-MoE, Sundial (flow-matching head).
2:28–2:38	Do they actually work?	GIFT-Eval / fev-bench leaderboards; zero-shot often beats tuned statistical models; fine-tuning gains modest and dataset-dependent; weaknesses: covariates, irregular sampling, domain shift, benchmark contamination.
2:38–2:45	**Live demo**	Chronos-2 zero-shot on an energy-demand series in ∼10 lines of Python; inspect calibration vs a seasonal-naive baseline.

Break (2:45–2:55)

4 Module 4 (2:55–3:55) — Diffusion Models for Probabilistic Forecasting

Time	Topic	Notes
2:55–3:02	Why generative models?	Forecasts are distributions; quantile heads lack coherent joint trajectories; scenario generation, risk, rare events.
3:02–3:18	**Diffusion basics (math)**	Forward chain $q(x^{(n)} \mid x^{(n-1)}) = \mathcal{N}(\sqrt{1-\beta_n}\, x^{(n-1)}, \beta_n I)$; $\alpha_n, \bar{\alpha}_n$ and the closed form $x^{(n)} = \sqrt{\bar{\alpha}_n} x^{(0)} + \sqrt{1-\bar{\alpha}_n}\, \epsilon$; Gaussian posterior $q(x^{(n-1)} \mid x^{(n)}, x^{(0)})$; reverse p_θ with the μ_θ reparameterization; variational bound collapsing to the simplified ϵ-loss $\|\epsilon - \epsilon_\theta\|^2$; score-based/SDE view; sampling (DDPM vs DDIM/DPM-Solver); conditioning and classifier-free guidance.
3:18–3:28	TimeGrad (2021)	RNN encodes history; conditional diffusion head generates the next step; applied autoregressively. Strength: coherent multivariate samples. Weakness: slow, error accumulation.
3:28–3:38	CSDI and SSSD	Imputation-style masking — generate the whole future segment non-autoregressively conditioned on the observed past; CSDI uses 2-D attention, SSSD swaps in S4 state-space layers.
3:38–3:47	TimeDiff, TSDiff	TimeDiff: time-series-aware conditioning (future mixup, AR initialization) for fast non-AR forecasting. TSDiff: one unconditional model + self-guidance for forecasting, imputation, refinement, and synthetic data.
3:47–3:55	The frontier	Multi-scale (MG-TSD, mr-Diff), non-stationary diffusion (NsDiff), latent and multimodal diffusion, few-step samplers and flow matching; convergence with foundation models (e.g. Sundial).

5 Module 5 (3:55–4:45) — Hands-On Lab, Evaluation, and Practice

Time	Topic	Notes
3:55–4:22	**Hands-on lab (6 coding exercises)**	`exercises.py` (numpy + matplotlib, CPU-only): (1) build/load the dataset and split; (2) classical VAR(1) — OLS recovers A, Granger F-test, impulse response $B^{(h)} = A^h$; (3) seasonal-naive + AutoETS baselines with a backtest-based 80% interval, compute MASE; (4) self-attention from scratch (Q, K, V, softmax, causal mask, multi-head); (5) diffusion forward process via the closed-form q plus a reverse-step stub; (6) sample paths, CRPS, and coverage. Stretch: beat the CRPS with Chronos-2 zero-shot.
4:22–4:32	Evaluating properly	MASE/sMAPE for points, CRPS/quantile loss and coverage for distributions; rolling-origin backtesting; leakage traps (scaling on the full series, overlapping windows, benchmark contamination in pretrained models).
4:32–4:39	Choosing a model	Decision guide: few series + interpretability → classical; many related series + covariates → GBDT or PatchTST; cold start → foundation model zero-shot; risk-sensitive scenarios → diffusion or generative head; latency-critical → linear/distilled.
4:39–4:45	Open problems + Q&A	Irregular and multimodal data, very long horizons, calibration under drift, agentic forecasting pipelines, benchmark hygiene.

Lab Environment

- Python $\geq$ 3.10; one GPU helpful but not required (small Chronos variants run on CPU).
- Core exercises need only `numpy` and `matplotlib` (`exercises.py`, CPU-only). Stretch goals: `pip install chronos-forecasting gluonts torch statsforecast matplotlib`
- Datasets: `electricity` (GluonTS), an M5 sample, or any participant-supplied CSV.

Core Reading List

1. Lütkepohl, *New Introduction to Multiple Time Series Analysis*, 2005; Hamilton, *Time Series Analysis*, 1994 (VAR/VARMA, IRF).
2. Granger, *Investigating Causal Relations. . .* , Econometrica 1969; Engle & Granger, *Co-integration and Error Correction*, Econometrica 1987; Johansen 1991.
3. Vaswani et al., *Attention Is All You Need*, NeurIPS 2017.
4. Zhou et al., *Informer*, AAAI 2021; Wu et al., *Autoformer*, NeurIPS 2021; Zhou et al., *FEDformer*, ICML 2022.
5. Zeng et al., *Are Transformers Effective for Time Series Forecasting?* (DLinear), AAAI 2023.
6. Nie et al., *PatchTST*, ICLR 2023; Liu et al., *iTransformer*, ICLR 2024.

7. Ansari et al., *Chronos: Learning the Language of Time Series*, 2024; Chronos-2 technical report, 2025.

8. Das et al., *TimesFM*, ICML 2024; Woo et al., *Moirai*, ICML 2024; Moirai 2.0 report, 2026.

9. Rasul et al., *TimeGrad*, ICML 2021; Tashiro et al., *CSDI*, NeurIPS 2021; Alcaraz & Strodthoff, *SSSD*, 2022.

10. Shen & Kwok, *TimeDiff*, ICML 2023; Kollovieh et al., *TSDiff*, NeurIPS 2023.

11. Su et al., *Diffusion Models for Time Series Forecasting: A Survey*, 2025; Meijer & Chen, *The Rise of Diffusion Models in Time-Series Forecasting*, 2024.