



Time Series Analysis and Forecasting

Chapter 9: Prophet and TBATS



Daniel Traian PELE

Bucharest University of Economic Studies

IDA Institute Digital Assets

Blockchain Research Center

AI4EFin Artificial Intelligence for Energy Finance

Romanian Academy, Institute for Economic Forecasting

MSCA Digital Finance

Learning Objectives

By the end of this chapter, you will be able to:

- Handle time series with multiple seasonal patterns
- Use Facebook Prophet for flexible forecasting with holidays
- Apply TBATS models for complex seasonality
- Compare and select between modern forecasting methods



Outline

Multiple Seasonalities

TBATS Model

Facebook Prophet

Comparison and Guidelines

Case Study

AI Use Case

Quiz

Summary



The Problem: Complex Seasonal Patterns

Real-World Examples

- **Hourly electricity demand:** Daily + Weekly + Annual patterns
- **Website traffic:** Daily + Weekly + Holiday effects
- **Retail sales:** Weekly + Monthly + Annual + Holiday effects
- **Call center volume:**
 - ▶ Hourly + Daily + Weekly patterns

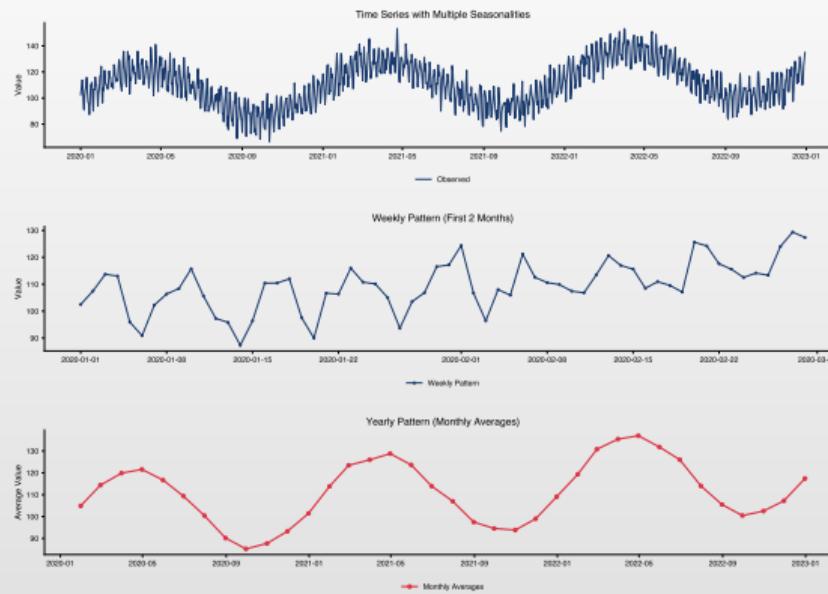
SARIMA Limitation

Standard SARIMA($p, d, q)(P, D, Q)_s$ handles only **one** seasonal period s .

For hourly data with daily AND weekly patterns, we need $s_1 = 24$ and $s_2 = 168$.



Example: Hourly Data with Multiple Seasonalities



Q TSA_ch9_multiple_seasonality



Solutions for Multiple Seasonalities

Traditional Approaches

- Fourier terms:** Add sin/cos regressors
- Dummy variables:** Many parameters
- Nested models:** Complex specification

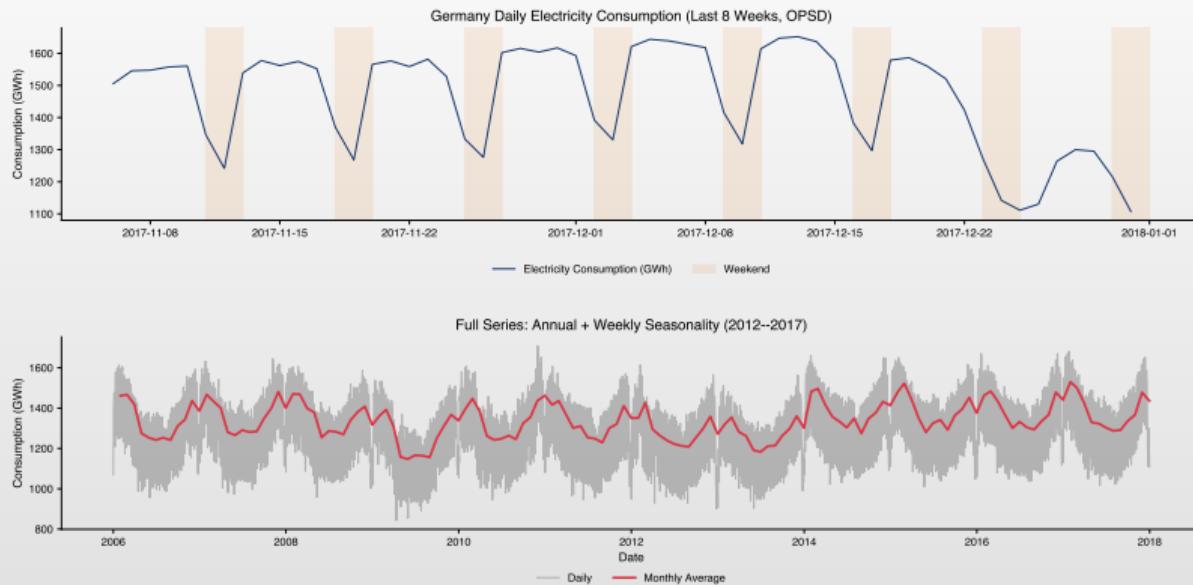
Modern Approaches

- TBATS:** Automatic, handles many periods
- Prophet:** Flexible, interpretable
- Neural methods:**
 - ▶ Deep learning

Method	Max Seasonalities	Interpretable
SARIMA	1	Yes
Fourier + ARIMA	Multiple	Moderate
TBATS	Multiple	Moderate
Prophet	Multiple	Yes



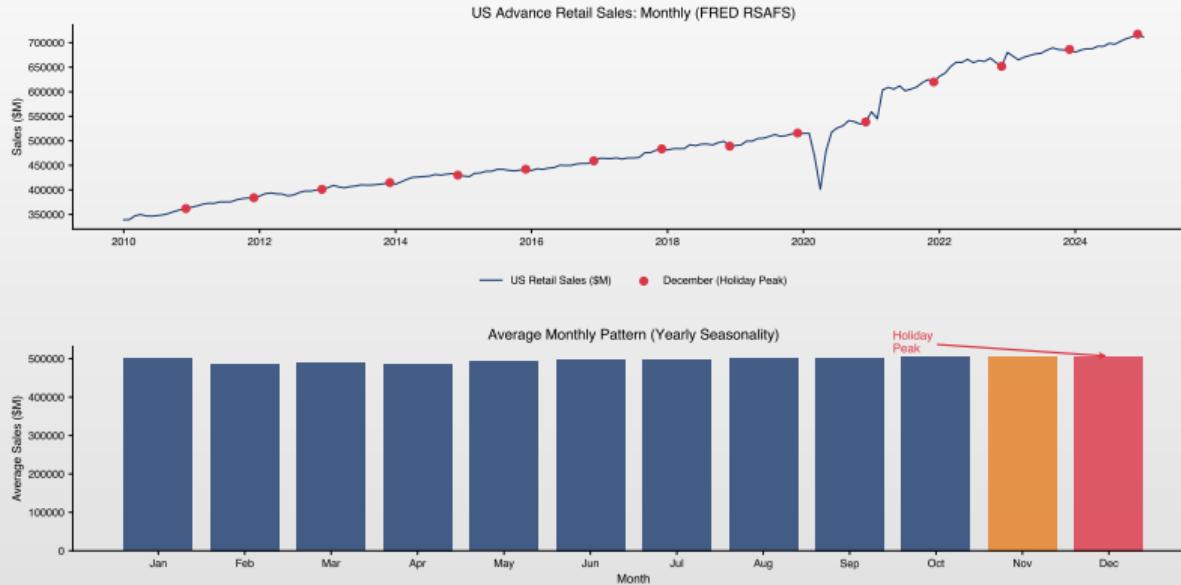
Real Example: Electricity Demand



Q TSA_ch9_electricity_demand



Real Example: Retail Sales with Holidays



Q TSA_ch9_retail_sales



Researcher Spotlight: Rob J. Hyndman



*1967
W Wikipedia

Biography

- Australian statistician, Professor at Monash University
- One of the most influential researchers in time series forecasting
- Creator of the widely-used `forecast` package for R
- Editor-in-Chief of the *International Journal of Forecasting* (2005–2018)

Key Contributions

- **TBATS model** (2011) — trigonometric Box-Cox ARMA with multiple seasonal periods
- **ETS framework** — exponential smoothing state space models with automatic selection
- **forecast package** for R — the standard toolkit for time series forecasting
- **Hierarchical forecasting** and forecast reconciliation methods

TBATS: What Does It Stand For?

TBATS Components

- T** Trigonometric seasonality using Fourier terms
- B** Box-Cox transformation for variance stabilization
- A** ARMA errors for remaining autocorrelation
- T** Trend component (possibly damped)
- S** Seasonal components (multiple allowed)

Key Innovation: Trigonometric Seasonality

$$s_t^{(i)} = \sum_{j=1}^{k_i} \left[s_j^{(i)} \cos\left(\frac{2\pi j t}{m_i}\right) + s_j^{*(i)} \sin\left(\frac{2\pi j t}{m_i}\right) \right]$$

m_i = seasonal period, k_i = number of harmonics



Box-Cox Transformation

Definition 1 (Box-Cox Transformation)

The Box-Cox transformation with parameter ω is defined as:

$$y_t^{(\omega)} = \begin{cases} \frac{y_t^\omega - 1}{\omega} & \text{if } \omega \neq 0 \\ \ln(y_t) & \text{if } \omega = 0 \end{cases}$$

Purpose

- Variance stabilization:** Makes variance constant over time
- Normalization:** Reduces skewness in the data
- Common values: $\omega = 0$ (log), $\omega = 0.5$ (square root), $\omega = 1$ (no transform)



TBATS Model Structure

State Space Representation

$$y_t^{(\omega)} = \ell_{t-1} + \phi b_{t-1} + \sum_{i=1}^T s_{t-m_i}^{(i)} + d_t \quad (1)$$

$$\ell_t = \ell_{t-1} + \phi b_{t-1} + \alpha d_t, \quad b_t = \phi b_{t-1} + \beta d_t \quad (2)$$

- $y_t^{(\omega)}$: Box-Cox transformed observation
- ℓ_t : local level (smoothed mean)
- b_t : trend with damping $\phi \in (0, 1)$
- $s_t^{(i)}$: i -th seasonal component
- d_t : ARMA(p, q) error process
- α, β : smoothing parameters



TBATS: Trigonometric Seasonality State Evolution

Definition 2 (Trigonometric State-Space Recursion)

For each seasonal component with period m_i and k_i harmonics, define states:

$$\begin{pmatrix} s_{j,t}^{(i)} \\ s_{j,t}^{*(i)} \end{pmatrix} = \begin{pmatrix} \cos(\lambda_j) & \sin(\lambda_j) \\ -\sin(\lambda_j) & \cos(\lambda_j) \end{pmatrix} \begin{pmatrix} s_{j,t-1}^{(i)} \\ s_{j,t-1}^{*(i)} \end{pmatrix} + \begin{pmatrix} \gamma_1^{(i)} \\ \gamma_2^{(i)} \end{pmatrix} d_t$$

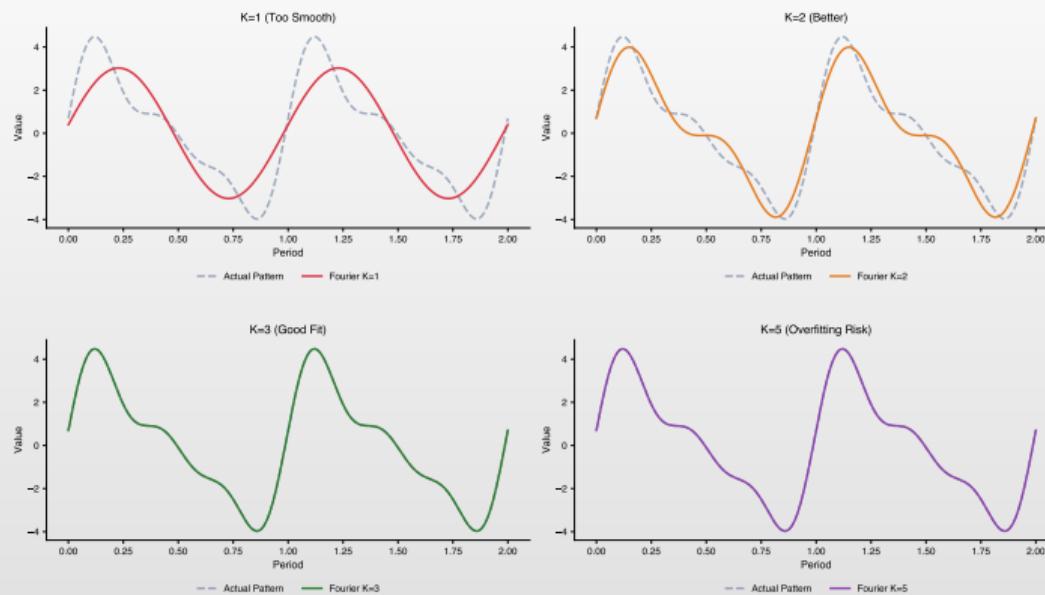
where $\lambda_j = \frac{2\pi j}{m_i}$ is the j -th harmonic frequency.

Interpretation

- The rotation matrix preserves the periodic structure
- Total seasonal: $s_t^{(i)} = \sum_{j=1}^{k_i} s_{j,t}^{(i)}$
- Parameters: $2k_i$ states per seasonal period



Fourier Approximation of Seasonality



Q TSA_ch9_fourier_approximation



TBATS: Choosing the Number of Harmonics

Why Fourier/Trigonometric Terms?

1. **Parsimonious:** $2k$ parameters vs m dummy variables
2. **Smooth:** Captures smooth seasonal patterns naturally
3. **Flexible:** Number of harmonics k controls complexity
4. **Non-integer periods:** Can handle $s = 365.25$ for daily data

Low k (few harmonics)

- Smooth pattern
- Fewer parameters
- May miss sharp peaks

High k (many harmonics)

- Can capture any pattern
- More parameters ($2k$ total)
- Maximum useful: $k \leq \lfloor m/2 \rfloor$



TBATS: Key Features

Automatic Model Selection

TBATS automatically determines:

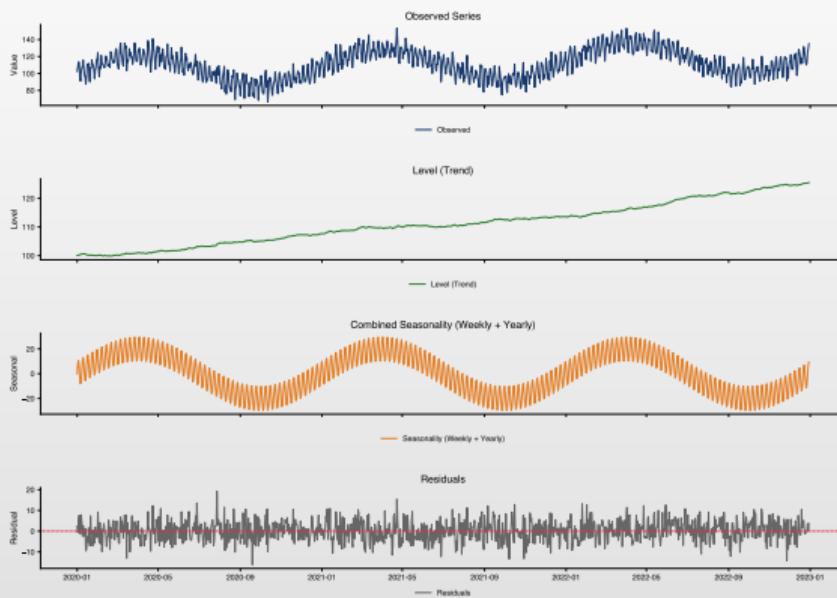
- Box-Cox parameter ω for variance stabilization
- Number of harmonics k_i for each seasonal period
- ARMA orders (p, q) for residual autocorrelation
- Damped vs non-damped trend specification

BATS vs TBATS

- BATS**: Traditional seasonal states (dummy variables)
- TBATS**: Trigonometric (Fourier) seasonal representation
- TBATS more parsimonious for long seasonal periods



TBATS Decomposition Example



 TSA_ch9_tbats_decomposition



TBATS: Advantages and Limitations

Advantages

- Handles **multiple** seasonal periods
- Automatic** model selection
- Handles **non-integer** periods (365.25)
- Box-Cox** for heteroskedasticity
- Good for **high-frequency** data

Limitations

- Computationally intensive**
- No external regressors**
- Less **interpretable** than Prophet
- Can be **slow** for very long series
- Requires **sufficient data per season**



Prophet: Overview

What is Prophet?

Forecasting procedure developed by Facebook (Meta) in 2017 for **business time series**:

- Strong seasonal effects (daily, weekly, yearly)
- Holiday effects and trend changes (changepoints)
- Handles missing data and outliers

Key Philosophy: “Analyst-in-the-loop”

Designed for analysts with domain knowledge but without time series expertise.



Prophet Model Structure

Decomposition Approach

Prophet uses an **additive decomposition**:

$$y(t) = g(t) + s(t) + h(t) + \varepsilon_t$$

$g(t)$: Trend

- Linear or logistic
- Automatic changepoints
- Growth saturation

$s(t)$: Seasonality

- Fourier series
- Multiple periods
- Custom seasonality

$h(t)$: Holidays

- Country holidays
- Custom events
- Window effects



Prophet: Seasonality Component

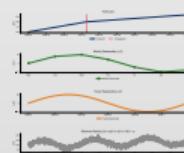
Fourier Series Representation

$$s(t) = \sum_{n=1}^N \left[a_n \cos\left(\frac{2\pi n t}{P}\right) + b_n \sin\left(\frac{2\pi n t}{P}\right) \right]$$

Default Settings

Seasonality	Period	Fourier Order
Yearly	365.25 days	10
Weekly	7 days	3
Daily	1 day	4

Higher N = more flexibility but risk of overfitting



Prophet: Trend Component

Linear Trend with Changepoints

$$g(t) = (k + \mathbf{a}(t)^T \boldsymbol{\delta}) \cdot t + (m + \mathbf{a}(t)^T \boldsymbol{\gamma})$$

- k : base growth rate (slope)
- $\boldsymbol{\delta} = (\delta_1, \dots, \delta_S)$: slope changes at S changepoints
- $\mathbf{a}(t) \in \{0, 1\}^S$: indicator if changepoint s is active at time t

Logistic Growth

For saturating trends:

$$g(t) = \frac{C(t)}{1 + e^{-(k + \mathbf{a}(t)^T \boldsymbol{\delta})(t - m - \mathbf{a}(t)^T \boldsymbol{\gamma})}}$$

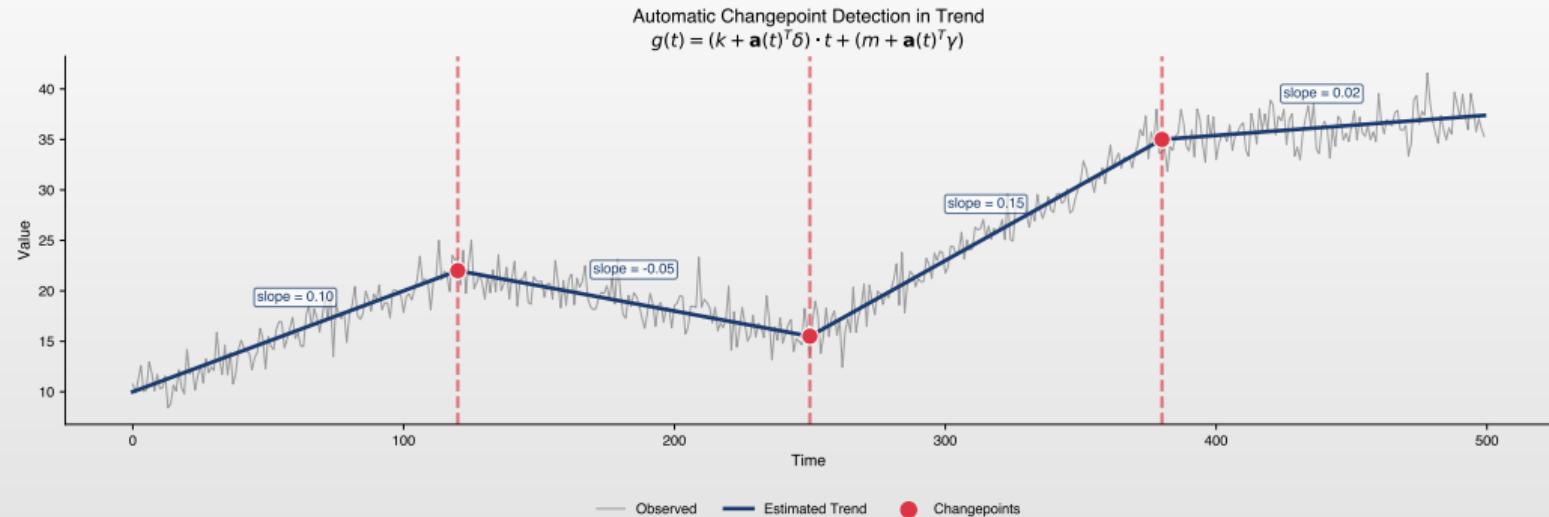
$C(t)$ = time-varying carrying capacity

Continuity Constraint

The offset $\gamma_j = -s_j \cdot \delta_j$ ensures $g(t)$ is continuous at each changepoint s_j .



Trend Changepoint Detection



Q TSA_ch9_changepoint_detection



Prophet: Holiday Effects

Holiday Model

$$h(t) = Z(t) \cdot \kappa$$

where $Z(t)$ is an indicator matrix for holidays and κ are holiday effects.

Built-in Features

- 60+ countries supported
- Custom holiday definitions
- Window effects (before/after)

Holiday Types

- National holidays
- Religious observances
- Business events



Prophet: Customization Options

Seasonality Customization

- Add custom seasonal periods (monthly, quarterly)
- Control Fourier order for each seasonality
- Enable/disable default seasonalities

External Regressors

Prophet supports adding external variables:

- Weather data, promotions, special events
- Binary or continuous regressors
- Automatic regularization



Prophet: Uncertainty Quantification

Bayesian Framework

Prophet uses a **Laplace prior** on changepoint magnitudes:

$$\delta_j \sim \text{Laplace}(0, \tau), \quad \tau = \text{changepoint_prior_scale}$$

Smaller τ = sparser, smaller changepoints (more regularization).

Sources of Uncertainty

1. **Trend:** Future changepoints
2. **Seasonality:** Coefficient variance
3. **Observation:** Residual noise σ^2

Prediction Intervals

- MAP estimation for point forecasts
- Monte Carlo sampling for intervals
- Default: 80% credible interval



Prophet: Tuning Parameters

Key Parameters

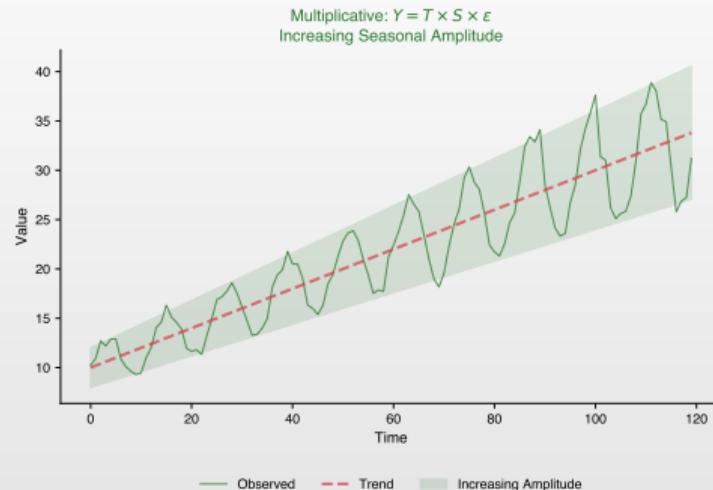
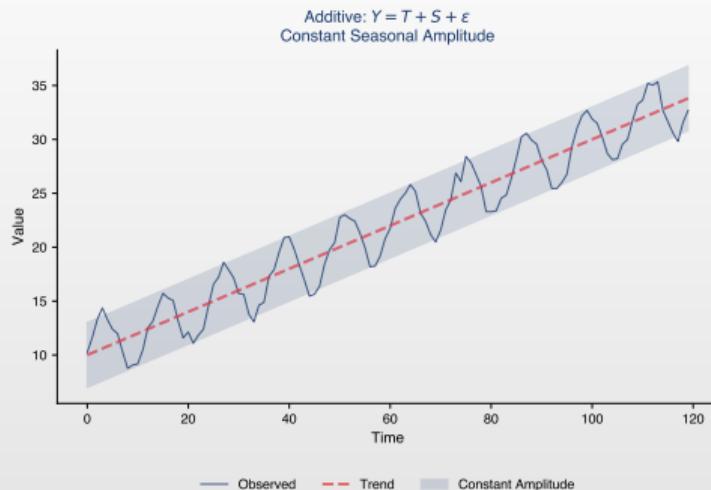
Parameter	Effect
changepoint_prior_scale	Trend flexibility (default: 0.05)
seasonality_prior_scale	Seasonality flexibility (default: 10)
holidays_prior_scale	Holiday effect size (default: 10)
seasonality_mode	'additive' or 'multiplicative'
changepoint_range	Portion of history for changepoints

Practical Tips

- Overfitting trend?** Decrease `changepoint_prior_scale`
- Underfitting seasonality?** Increase `seasonality_prior_scale`
- Seasonal amplitude varies?** Use `seasonality_mode='multiplicative'`



Additive vs Multiplicative Seasonality



Q TSA_ch9_additive_vs_multiplicative



Prophet: Advantages and Limitations

Advantages

- Easy to use:** Minimal tuning needed
- Interpretable:** Clear decomposition
- Handles missing data well**
- Holiday effects built-in**
- Multiple seasonalities**
- External regressors supported**
- Fast fitting**

Limitations

- Not ARIMA-based:** No autocorrelation modeling
- Daily data focus:** Less suited for very high frequency
- Trend assumptions:** Linear/logistic may not fit
- No built-in CV:** Must implement manually
- Overfitting risk with many seasonalities**



TBATS vs Prophet: Head-to-Head

Feature	TBATS	Prophet
Multiple seasonalities	Yes (automatic)	Yes (manual or auto)
Holiday effects	No	Yes (built-in)
External regressors	No	Yes
Trend changepoints	No (smooth)	Yes (automatic)
Missing data	Interpolation needed	Handles natively
Interpretability	Moderate	High
Computation speed	Slow	Fast
High-frequency data	Good	Moderate
Non-integer periods	Yes (e.g., 365.25)	Yes
Uncertainty intervals	Yes	Yes



When to Use Each Model

Use TBATS when:

- High-frequency data
- Multiple seasonal periods
- No external regressors
- Automatic selection preferred

Use Prophet when:

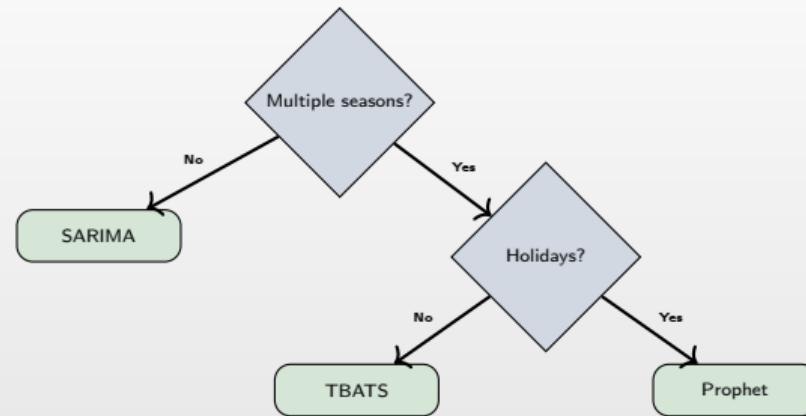
- Business forecasting
- Holiday effects important
- Trend has changepoints
- External regressors available

General Guideline

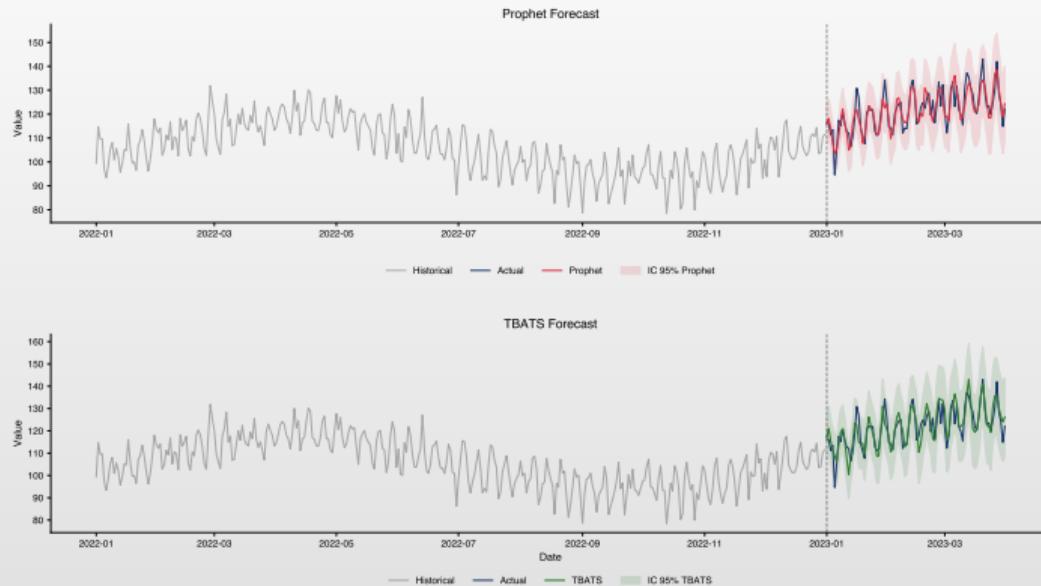
Prophet: business applications with daily data
TBATS: technical applications with high-frequency data



Decision Flowchart



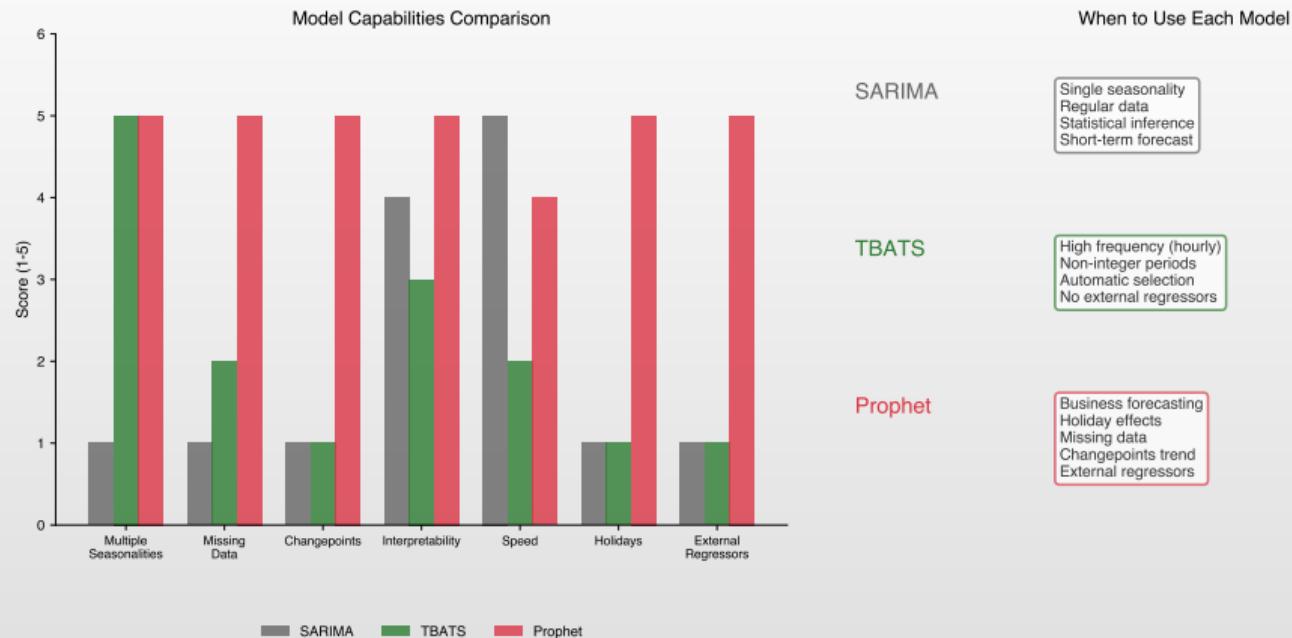
Prophet vs TBATS: Forecast Comparison



Q TSA_ch9_prophet_vs_tbats



Model Selection Guide



Evaluation Metrics

Definition 3 (Forecast Accuracy Metrics)

Let y_t denote actual values, \hat{y}_t forecasts, and n the forecast horizon:

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{t=1}^n (y_t - \hat{y}_t)^2} \quad (\text{penalizes large errors})$$

$$\text{MAE} = \frac{1}{n} \sum_{t=1}^n |y_t - \hat{y}_t| \quad (\text{robust to outliers})$$

$$\text{MAPE} = \frac{100\%}{n} \sum_{t=1}^n \left| \frac{y_t - \hat{y}_t}{y_t} \right| \quad (\text{scale-free})$$

Coverage

For prediction intervals $[\hat{y}_t^L, \hat{y}_t^U]$, coverage rate is the proportion of actual values falling within the interval. Target: match the nominal level (e.g., 80%).



Case Study: Energy Demand Forecasting

Problem

Forecast hourly electricity demand with:

- Daily pattern:** Peak at noon and evening
- Weekly pattern:** Lower on weekends
- Annual pattern:** Higher in summer (AC) and winter (heating)
- Holiday effects:** Lower demand on holidays

Approach

1. Try TBATS with periods [24, 168, 8766]
2. Try Prophet with daily, weekly, yearly seasonality + holidays
3. Compare using cross-validation



Case Study: Results

Model Comparison

Model	MAPE	RMSE	Coverage
SARIMA (daily only)	8.5%	450 MW	75%
TBATS	4.2%	220 MW	82%
Prophet	4.8%	250 MW	85%
Prophet + holidays	3.9%	200 MW	88%

Key Finding

Multiple seasonality models significantly outperform single-seasonality SARIMA.



AI Exercise: Critical Thinking

Prompt to test in ChatGPT / Claude / Copilot

"I have 3 years of hourly energy consumption data. Use Facebook Prophet to forecast the next week. Include holidays and special events. Give me complete Python code."

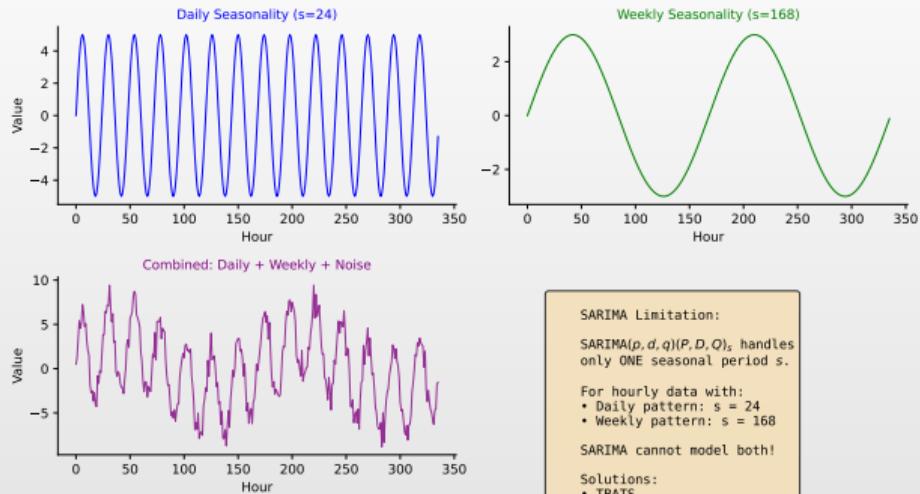
Exercise:

1. Run the prompt in an LLM of your choice and critically analyze the response.
2. Does Prophet automatically detect multiple seasonalities (daily, weekly)?
3. How are holidays specified? Country-specific or custom events?
4. Does it use cross-validation with cutoffs (performance metrics)?
5. Would TBATS be more appropriate for this frequency? Why or why not?

Warning: AI-generated code may run without errors and look professional. *That does not mean it is correct.*



Quiz 1: Multiple Seasonality



Question: Why can't standard SARIMA(p, d, q)(P, D, Q) $_s$ model hourly electricity data with both daily and weekly patterns?

Answer: SARIMA handles only **one** seasonal period s . Cannot set $s = 24$ and $s = 168$ simultaneously.



Quiz 2: TBATS Components

TBATS: What Does It Stand For?

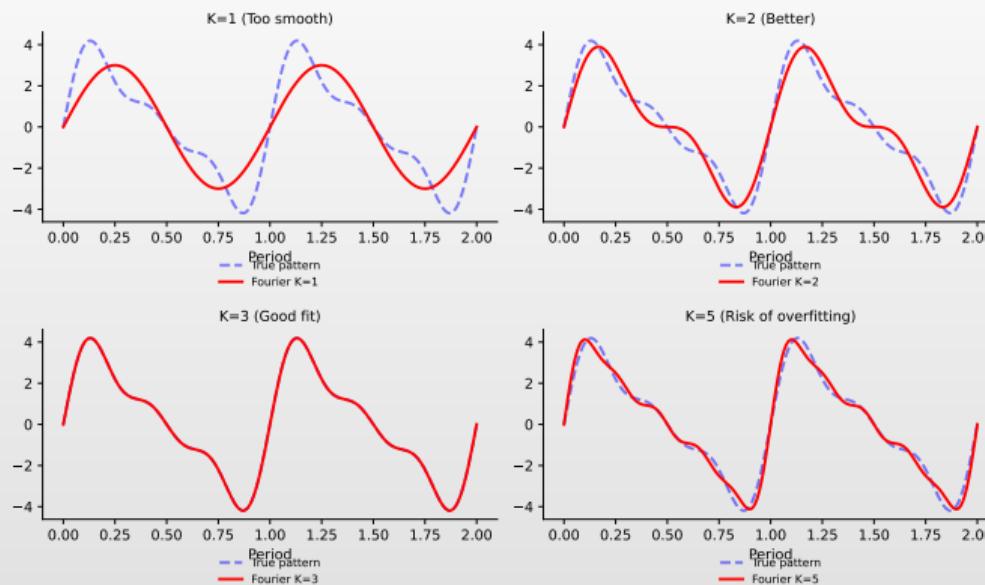
T	Trigonometric	Fourier terms for seasonality $\sum[a_n \cos(\frac{2\pi n t}{m}) + b_n \sin(\frac{2\pi n t}{m})]$
B	Box-Cox	Variance stabilization $y^{(\omega)} = (y^\omega - 1)/\omega$
A	ARMA	Error autocorrelation $\phi(L)d_t = \theta(L)\varepsilon_t$
T	Trend	Level + slope (possibly damped) $\ell_t = \ell_{t-1} + \phi b_{t-1}$
S	Seasonal	Multiple seasonal periods m_1, m_2, \dots, m_T

Question: What does each letter in TBATS represent?

Answer: Trigonometric seasonality, Box-Cox transformation, ARMA errors, Trend, Seasonal components.



Quiz 3: Fourier Harmonics

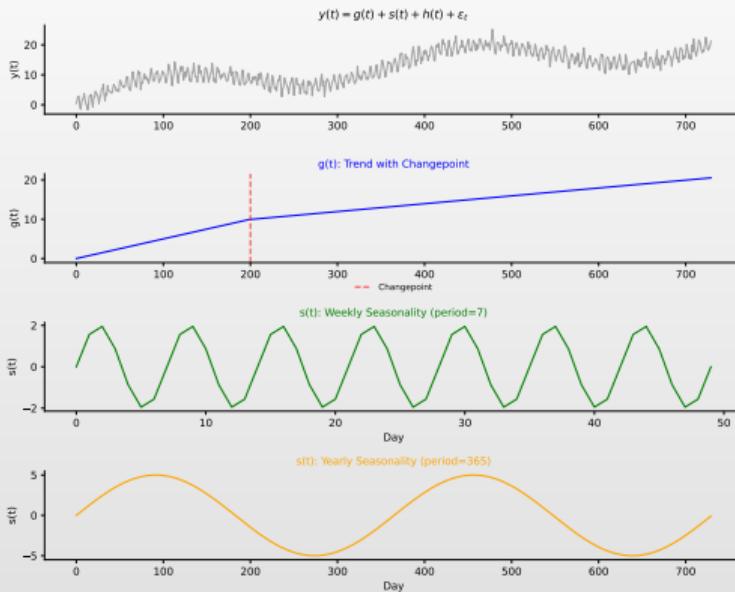


Question: What happens when we increase the number of Fourier harmonics K ?

Answer: Higher K captures more complex patterns but increases overfitting risk.



Quiz 4: Prophet Decomposition



Question: What are the main components in Prophet's model $y(t) = g(t) + s(t) + h(t) + \varepsilon_t$?

Answer: $g(t)$ = trend with changepoints, $s(t)$ = seasonality, $h(t)$ = holiday effects.



Quiz 5: Model Comparison

TBATS vs Prophet: Head-to-Head Comparison

Feature	TBATS	Prophet
Multiple seasonalities	Yes (automatic)	Yes (manual/auto)
Holiday effects	No	Yes (built-in)
External regressors	No	Yes
Trend changepoints	No (smooth)	Yes (automatic)
Missing data	Needs interpolation	Handles natively
Interpretability	Moderate	High
Computation speed	Slow	Fast
High-frequency data	Good	Moderate
Non-integer periods	Yes (e.g., 365.25)	Yes
Best for	Technical/high-freq	Business/daily

Question: What key features does Prophet have that TBATS lacks?

Answer: Holiday effects, external regressors, trend changepoints, native missing data handling.



Key Takeaways

What We Learned

- TBATS handles multiple seasonalities with Fourier terms and Box-Cox transformation
- Prophet provides interpretable decomposition with trend changepoints and holiday effects
- Both methods scale better than SARIMA for high-frequency and complex seasonal data

Important

Choose Prophet for business forecasting with holidays and interpretability needs. Use TBATS for automatic modeling of high-frequency data. Always validate with time series cross-validation—never standard k-fold!



Questions?

Questions?

Next Steps:

- Practice with the Jupyter notebook
- Try Prophet on your own data
- Explore NeuralProphet for deep learning extension



Bibliography I

Prophet

- Taylor, S.J., & Letham, B. (2018). Forecasting at Scale, *The American Statistician*, 72(1), 37–45.
- Harvey, A.C. (1989). *Forecasting, Structural Time Series Models and the Kalman Filter*, Cambridge University Press.

TBATS and Exponential Smoothing

- De Livera, A.M., Hyndman, R.J., & Snyder, R.D. (2011). Forecasting Time Series with Complex Seasonal Patterns Using Exponential Smoothing, *JASA*, 106(496), 1513–1527.
- Hyndman, R.J., Koehler, A.B., Ord, J.K., & Snyder, R.D. (2008). *Forecasting with Exponential Smoothing: The State Space Approach*, Springer.
- Taylor, J.W. (2003). Short-term Electricity Demand Forecasting Using Double Seasonal Exponential Smoothing, *Journal of the Operational Research Society*, 54(8), 799–805.



Bibliography II

Forecasting Comparisons and Competitions

- Makridakis, S., Spiliotis, E., & Assimakopoulos, V. (2020). The M4 Competition, *International Journal of Forecasting*, 36(1), 54–74.
- Hyndman, R.J., & Athanasopoulos, G. (2021). *Forecasting: Principles and Practice*, 3rd ed., OTexts.
- Petropoulos, F., et al. (2022). Forecasting: Theory and Practice, *International Journal of Forecasting*, 38(3), 845–1054.

Online Resources and Code

- **Quantlet:** <https://quantlet.com> → Code repository for statistics
- **Quantinar:** <https://quantinar.com> → Learning platform for quantitative methods
- **GitHub TSA:** <https://github.com/QuantLet/TSA> → Python code for this course



Thank You!

Questions?

Course materials available at: <https://danpele.github.io/Time-Series-Analysis/>

