Topic here

Valeriia Baranivska
Kyiv Ukraine

Kaunas — 2025

Contents

1 Introduction

The discipline of time series forecasting has evolved from foundational linear statistical models to encompass complex, non-linear machine learning architectures and sophisticated hybrid approaches. Historically, the analytical landscape was dominated by models such as the Autoregressive Integrated Moving Average (ARIMA), which furnish a robust and interpretable framework for capturing the stochastic, linear dependencies inherent in many time-varying processes. However, the increasing complexity of real-world data, characterized by non-linear relationships and intricate patterns, illuminated the limitations of these linear paradigms. This catalyzed the integration of non-linear models from the machine learning domain, most notably Recurrent Neural Networks (RNNs) and their advanced variant, Long Short-Term Memory (LSTM) networks, which are specifically designed to learn from sequential data.

More recently, a contemporary paradigm of hybrid modeling has emerged, seeking to synthesize the discrete strengths of both statistical and machine learning approaches. These models operate on the principle of decomposition, leveraging ARIMA-type models to capture linear structures and employing LSTMs to model the remaining non-linear residuals. This report provides a detailed theoretical and mathematical exposition of this modeling hierarchy, from the fundamental Autoregressive process to advanced hybrid and optimization techniques. It underscores that a profound mathematical understanding of this diverse toolkit is not merely an academic formalism but an essential prerequisite for developing robust, reliable, and high-performing forecasting systems in any quantitative discipline. The critical importance of rigorous model selection and optimization is also addressed, with an examination of hyperparameter tuning methodologies that are crucial for achieving maximal predictive performance.

2 Theory

2.1 Regressive models

2.1.1 Autoregressive processes (AR)

The Autoregressive (AR) model serves as a fundamental building block in time series analysis. It is predicated on the simple yet powerful idea that the current value of a series can be explained as a function of its past values. This "regression on itself" provides a parsimonious way to model the linear dependencies and temporal structure within the data.

Autoregression is a statistical method used for analyzing and predicting time series data. In an autoregressive model, the value of a variable at time t depends linearly on its previous values.

In simple terms, **autoregression means predicting the present based on the past.**

AR(p) Model Definition

An autoregressive model of order p, denoted as AR(p), is defined as:

$$X_t = c + \phi_1 X_{t-1} + \phi_2 X_{t-2} + \cdots + \phi_p X_{t-p} + \varepsilon_t$$

where:

- X_t is the value of the time series at time t
- c is a constant (intercept)
- $\phi_1, \phi_2, \ldots, \phi_p$ are the parameters of the model (coefficients)
- p is the order of the model (number of lagged terms)

- ε_t is white noise (a random error term, assumed to be normally distributed with mean 0 and constant variance)

AR(1) Example

The simplest autoregressive model is AR(1), where $p = 1$:

$$X_t = c + \phi_1 X_{t-1} + \varepsilon_t$$

This means that the current value X_t depends on:

- a constant c
- the previous value X_{t-1} multiplied by the coefficient ϕ_1
- a random noise term ε_t

Stationarity Condition

For an AR(1) model to be stationary (i.e., its statistical properties like mean and variance do not change over time), the absolute value of the coefficient must be less than 1:

$$|\phi_1| < 1$$

More generally, for an AR(p) model to be stationary, the roots of the characteristic equation must lie outside the unit circle in the complex plane.

Estimation of Parameters

The parameters ϕ_i can be estimated using methods such as:

- **Least Squares Estimation**
- **Yule-Walker Equations**
- **Maximum Likelihood Estimation**

Use Cases of Autoregression

- Forecasting stock prices
- Weather prediction
- Modeling economic indicators (like GDP, inflation)
- Any time-dependent signal

Conclusion

Autoregression is a powerful yet simple approach to model time series by leveraging past values. It serves as the foundation for more complex models like ARMA and ARIMA.

2.1.2 Seasonal Autoregressive Integrated Moving Average (SARIMA)

SARIMA models are denoted as:

$$\text{SARIMA}(p, d, q) \times (P, D, Q)_s$$

where:

- (p, d, q) are the non-seasonal components:
  - p = non-seasonal autoregressive order
  - d = non-seasonal differencing order
  - q = non-seasonal moving average order
- $(P, D, Q)_s$ are the seasonal components:
  - P = seasonal autoregressive order
  - D = seasonal differencing order
  - Q = seasonal moving average order
  - s = length of the seasonal period (e.g., 12 for monthly data with yearly seasonality)

General SARIMA Model Formula

Let B be the backshift operator, where $B^k X_t = X_{t-k}$. The SARIMA model can be written as:

$$\Phi_P(B^s)\phi_p(B)(1-B)^d(1-B^s)^D X_t = \Theta_Q(B^s)\theta_q(B)\varepsilon_t$$

where:

- $\phi_p(B)$ is the non-seasonal AR polynomial:
$$\phi_p(B) = 1 - \phi_1 B - \phi_2 B^2 - \cdots - \phi_p B^p$$
- $\theta_q(B)$ is the non-seasonal MA polynomial:
$$\theta_q(B) = 1 + \theta_1 B + \theta_2 B^2 + \cdots + \theta_q B^q$$
- $\Phi_P(B^s)$ is the seasonal AR polynomial:
$$\Phi_P(B^s) = 1 - \Phi_1 B^s - \Phi_2 B^{2s} - \cdots - \Phi_P B^{Ps}$$
- $\Theta_Q(B^s)$ is the seasonal MA polynomial:
$$\Theta_Q(B^s) = 1 + \Theta_1 B^s + \Theta_2 B^{2s} + \cdots + \Theta_Q B^{Qs}$$
- $(1-B)^d$ is the non-seasonal differencing operator
- $(1-B^s)^D$ is the seasonal differencing operator
- ε_t is white noise

Example: SARIMA(1, 1, 1)(1, 1, 1)$_{12}$

This model has:

- Non-seasonal AR(1), differencing(1), MA(1)
- Seasonal AR(1), differencing(1), MA(1) with period 12 (e.g., monthly data with yearly seasonality)

$$(1-\phi_1 B)(1-\Phi_1 B^{12})(1-B)(1-B^{12})X_t = (1+\theta_1 B)(1+\Theta_1 B^{12})\varepsilon_t$$

Steps to Fit a SARIMA Model

1. **Visualize** the time series and identify trends and seasonality.
2. **Difference** the series d times and D seasonal times (if necessary) to make it stationary.
3. **Identify** potential orders (p, q) and (P, Q) using ACF/PACF plots.
4. **Estimate** parameters using Maximum Likelihood Estimation (MLE).
5. **Diagnose** the residuals to check if they resemble white noise.
6. **Forecast** future values using the fitted model.

Conclusion

SARIMA enhances ARIMA by incorporating seasonal patterns directly into the model. It is a robust and versatile approach for time series forecasting, especially when both trend and seasonality are present.

2.1.3 (SARIMAX)

2.2 Machine learning models

2.2.1 Long-short temporary memory (LSTM)

2.2.2 Model tuning (Kernal Tune)

3 Final model (SARIMAX LSTM)

The SARIMA model is great for short- to medium-term forecasting when the data structure remains stable. But when we look far into the future, its limitations become apparent.

- Linearity: SARIMA is a linear model. It assumes that the relationship between past and future values is linear. Weather systems, economies, and other complex processes are deeply non-linear.
- Damping to the mean: When forecasting many steps ahead, SARIMA forecasts inevitably tend to converge to the mean of the series (or its trend). Confidence intervals become extremely wide, and the forecast loses its practical value. It shows the average "climate normal", not the actual future weather.
- Static: The model assumes that the parameters describing the series (coefficients ,) are constant. In the real world, the climate changes, economic conditions evolve, and these parameters can change over time.

Key advantages of LSTMs

- Learning nonlinear dependencies: LSTMs are able to detect complex, non-linear patterns in data that SARIMA simply ignores.
- "Memory: Thanks to its gate architecture, LSTMs can "decide" which information from the past to keep for the long term and which to forget. This allows it to capture dependencies spanning long periods.
- Working with many variables: LSTM can easily add other time series (e.g. solar activity, humidity, data from other weather stations) to improve the forecast. This is called multivariate forecasting.

The idea is to decompose the time series into two components: linear and nonlinear.

$$Y_t = L_t + N_t$$

where:

- L_t – is the linear component (trends, seasonality).
- N_t – is the nonlinear component (complex patterns, randomness).

Algorithm for building a hybrid model

1. Train SARIMAX. We build the best possible SARIMA model for our data (for example, for temperature). We use it to make a forecast. This prediction will be our linear component: $L^t = SARIMA\_forecast(Yt)$.
2. Obtain the residuals. Residuals are the difference between the actual data and the SARIMA forecast. They represent the part of the data that the linear model could not explain.

   $$Nt = Yt - L^t$$

   These residuals contain all the nonlinear information.
3. Train the LSTM on the residuals. Now, we train the LSTM model to predict the residuals from the SARIMA model rather than the data itself. The goal of LSTM is to find hidden patterns in these residuals. We get the forecast of the nonlinear part: $N^t = LSTM\_forecast(Nt)$.
4. Combine the forecasts. The final, improved forecast is the sum of the SARIMA forecast and the LSTM residual forecast.

   $$Y^t = L^t + N^t$$

This approach allows SARIMAX to do what it does best (model linear trends and seasonality) and let LSTM do the hard part - model the rest, i.e., the non-linear dynamics.

4 Modelling. Solution

We try to predict weather with at least with temprature, wind,pressure

The mathematical representation of SARIMA is as follows:

$$(1-\phi_1\beta)(1-\Phi_1\beta^S)(1-\beta)(1-\beta^S)y_t = (1+\theta_1\beta)(1+\Theta_1\beta^S)\varepsilon_t$$

- y_t is the observed time series at time t,
- B is the backward shift operator, representing the lag operator (i.e.,Byt=yt1Byt=yt1)
- ϕ_1 is the non-seasonal autoregressive coefficient,
- Φ_1 is the seasonal autoregressive coefficient,
- θ_1 is the non-seasonal moving average coefficient,
- Θ_1 is the seasonal moving average coefficient,
- s is the seasonal period,
- ε_t is the white noise error term at time t.

Aftter additional research it better to use SARIMAX. The differences only in that could simply add the external regressors as a linear term, just like in a standard regression model.

If look ar the math step-by-step:

1. The Baseline SARIMA Model Equation

The standard SARIMA$(p, d, q)(P, D, Q)_s$ model describes a time series y_t by relating it to its own past values and past error terms. The mathematical representation, using the backshift operator B, is:

$$\phi_p(B)\Phi_P(B^s)(1-B)^d(1-B^s)^D y_t = \theta_q(B)\Theta_Q(B^s)\varepsilon_t$$

Where:

- $\phi_p(B) = (1 - \phi_1 B - \cdots - \phi_p B^p)$ is the non-seasonal autoregressive polynomial.
- $\Phi_P(B^s) = (1 - \Phi_1 B^s - \cdots - \Phi_P B^{Ps})$ is the seasonal autoregressive polynomial.
- $\theta_q(B) = (1 + \theta_1 B + \cdots + \theta_q B^q)$ is the non-seasonal moving average polynomial.
- $\Theta_Q(B^s) = (1 + \Theta_1 B^s + \cdots + \Theta_Q B^{Qs})$ is the seasonal moving average polynomial.
- $(1-B)^d$ and $(1-B^s)^D$ represent the non-seasonal and seasonal differencing operators, respectively.
- ε_t is the white noise error term.

2. Incorporating External Regressors (The 'X' Component)

To account for the influence of external variables, we introduce a linear combination of these regressors into the model. Let's say we have k external time series regressors, denoted as $x_{1,t}, x_{2,t}, \ldots, x_{k,t}$. Each regressor is assigned a coefficient β_i.

The influence of these regressors is captured by the term:

$$\sum_{i=1}^{k} \beta_i x_{i,t} = \beta_1 x_{1,t} + \beta_2 x_{2,t} + \cdots + \beta_k x_{k,t}$$

3. The Complete SARIMAX Equation

The SARIMAX model combines the SARIMA structure for the error terms with the linear regression component for the external variables. The external regressors are added to the right-hand side of the equation.

The final mathematical model is:

$$\phi_p(B)\Phi_P(B^s)(1-B)^d(1-B^s)^D y_t = \sum_{i=1}^{k} \beta_i x_{i,t} + \theta_q(B)\Theta_Q(B^s)\varepsilon_t$$

This equation states that the differenced and autoregressively transformed value of y_t is a function of the external variables plus a structured error term (the moving average part).

4. Application to Weather Forecasting

Let's define the model for predicting temperature, using pressure and wind components as external regressors.

- Target value (Endogenous): $y_{1,t} = Temperature_t$
- External regressor 1: $x_{1,t} = Pressure_t$
- External regressor 2: $x_{1,t} = Wind_t(direction)$
- External regressor 3: $x_{1,t} = S - Wind_t(speed)$

5 Conclusion

Conclusion by full work

This report has detailed the mathematical foundations of a comprehensive suite of time series forecasting models, progressing from the linear elegance of ARIMA to the non-linear power of LSTMs and the synergistic potential of hybrid architectures. The analysis highlights the distinct capabilities and underlying assumptions of each modeling family. The ARIMA framework, culminating in the SARIMAX model, offers a highly interpretable and efficient method for capturing linear dependencies, seasonality, and the influence of external factors. LSTMs provide a robust solution for modeling complex, non-linear dynamics that are beyond the scope of linear models, thanks to their sophisticated gating mechanisms that regulate information flow and memory. Hybrid models, such as the ARIMA-LSTM, represent a pragmatic synthesis, decomposing a time series into its linear and non-linear constituents to be modeled by the most appropriate tool. Ultimately, the

analysis reveals that there is no universally superior model. The optimal choice is contingent upon the specific characteristics of the time series data, the nature of the forecasting problem, and practical constraints such as data availability and computational resources. The journey from a simple AR(1) process to a hybrid model optimized with Bayesian methods demonstrates a clear trade-off between interpretability and complexity. The report reinforces that a systematic, mathematically-grounded approach—encompassing rigorous identification, principled estimation, thorough diagnostic checking, and intelligent optimization—is the cornerstone of developing robust, defensible, and high-performance forecasting solutions in any quantitative field.

Future research

6 References

1. Linear/polynomial regression
2. Autoregressive and time series (1)
3. Autoregressive and time series (2)
4. Predicting climate change using an AR LSTMM
5. Time Series Prediction with LSTM
6. **A Hybrid ARIMA-LSTM Model for Short-Term Vehicle Speed Prediction**
https://www.mdpi.com/1996-1073/17/15/3736
7. **A Prediction Model of Fund Net Value Based on ARIMA-LSTM Hybrid Model**
https://arxiv.org/pdf/2111.15355
8. **A Thorough Introduction to ARIMA Models**
https://medium.com/analytics-vidhya/a-thorough-introduction-to-arima-models-987a24e9ff71
9. **A Tutorial on Bayesian Optimization**
https://arxiv.org/pdf/1807.02811
10. **AR, MA and ARMA models**
https://math.unm.edu/ ghuerta/tseries/week$4_1.pdf$
11. **AR Order Selection with Partial Autocorrelation Sequence**
https://www.mathworks.com/help/signal/ug/ar-order-selection-with-partial-autocorrelation-sequence.html
12. **ARIMA and SARIMAX models with Python**
https://cienciadedatos.net/documentos/py51-arima-sarimax-models-python
13. **ARIMA models**
https://www.stat.berkeley.edu/ ryantibs/timeseries-f23/lectures/arima.pdf
14. **ARIMA models and Box-Jenkins methodology**
https://library.fiveable.me/data-inference-and-decisions/unit-9/arima-models-box-jenkins-methodolog
guide/5DlAsKVJgMNm55u9
15. **Autoregressive (AR) Model for Time Series Forecasting**
https://www.geeksforgeeks.org/data-analysis/autoregressive-ar-model-for-time-series-forecasting/
16. **Autoregressive model**
https://en.wikipedia.org/wiki/Autoregressive$_model$
17. **Autoregressive Models**
https://online.stat.psu.edu/stat501/book/export/html/996
18. **Bayesian Optimization using Gaussian Processes: an introduction**
https://www.miguelgondu.com/blogposts/2023-07-31/intro-to-bo/
19. **Box-Jenkins ARIMA Modeling**
https://johngalt.com/forecasting-methods/box-jenkins-arima-modeling
20. **Box-Jenkins method**

7 Additional

(a) **Link for Git Hub repository -**

(b) **Link for Kaggle Notebook -**