Nonlinear Time Series Modeling Using Bernstein Polynomials and Bayesian Inference

Anonymous authors
Paper under double-blind review

Abstract

Modeling and forecasting nonlinear time series presents a highly non-trivial challenge to both statistical and neural network methods. This is particularly true with time series from chaotic systems with very few methods being able to provide accurate predictions along with uncertainty quantification. This article taps into fundamental theoretical properties of Bernstein polynomials to model temporal dependence structures present in nonlinear time series. The new Bernstein Polynomial Autoregressive (BPAR) model is seen to have architectural similarity to a single layer perceptron or a single hidden layer artificial neural network. Unlike deep multilayer neural network forecasters, the BPAR retains a single structured layer, making the model comparatively simple while allowing lag-specific Bernstein coefficient blocks to indicate which past observations are most responsible for prediction. Bayesian statistical inference is applied and this produces uncertainty quantification on model parameters and on predictions. Carefully designed shrinkage prior distributions are utilized in the Bayesian inference, and they assist in maintaining overall model parsimony. An extensive empirical study involving several real and simulated nonlinear time series data is conducted along with suitable model residual diagnostic checks. Results demonstrate, especially with chaotic and highly nonlinear time series, superior forecasting performance of the BPAR relative to existing statistical and neural network methods, and also point to the unique ability of BPAR to provide uncertainty quantification in these data settings.

1 Introduction

Nonlinear time series modeling must reconcile three goals that are often in tension: flexible approximation of nonlinear temporal dependence, model interpretability, and statistically meaningful uncertainty quantification with diagnosis. Classical statistical time series models such as ARIMA and SARIMA remain valuable because they are parsimonious and diagnostically transparent, but they are built around linear dependence after transformation and differencing (Box et al., 2015; Shumway & Stoffer, 2017). Related forecasting frameworks such as exponential smoothing, dynamic regression, vector autoregressions, and grouped time series methods remain firmly rooted in this statistical tradition (Hyndman & Athanasopoulos, 2021). Artificial neural network-based models offer far greater flexibility for nonlinear data, yet that flexibility is often purchased at the cost of black-box learned features, heavier tuning burdens, and less direct uncertainty interpretation. The difficulty is especially sharp in chaotic and strongly nonlinear systems (Lele, 1994; Mukherjee et al., 1997), where short-memory linear dependence structure can coexist with smooth nonlinear temporal dependence.

Recent machine learning work has pushed highly nonlinear time series forecasting in several important directions. Long Short-Term Memory (LSTM) networks were introduced to mitigate unstable or vanishing gradient flow in recurrent backpropagation (Rumelhart et al., 1986; Hochreiter & Schmidhuber, 1997), and recurrent autoregressive neural forecasters remain standard flexible baselines for nonlinear time series prediction (Salinas et al., 2017). Multilayer perceptron (MLP) forecasters use lagged observations as inputs to a feedforward network; their role as flexible nonlinear approximators is classical (Hornik et al., 1989), and MLP variants have also been studied in chaotic time series prediction problems (Stamatis et al., 1999). Basis-style residual networks and attention-based models have further expanded the neural forecasting toolkit through

interpretable block decompositions and longer-range temporal modeling (Oreshkin et al., 2020; Lim et al., 2021; Zhou et al., 2021; Wu et al., 2021). Spectral and Koopman-style methods provide expressive decompositions for nonlinear temporal dynamics (Lange et al., 2021). Generative forecasting models have also shown strong probabilistic performance on chaotic time series benchmarks, especially when the objective is designed for calibration rather than adversarial training (Pacchiardi et al., 2024). Recent studies further emphasize how training horizon, latent representation, and model family affect performance on canonical dynamical systems ranging from periodic to chaotic regimes (Aceituno et al., 2025; Kucukahmetler et al., 2026). Chaotic dynamical systems arising in settings such as astrophysics, biochemistry, and climatology, see Gilpin (2023a) for an example, provide a useful yet inherently challenging benchmark for comparing models that differ not only in predictive accuracy but also in interpretability (Gilpin, 2023b). Taken together, this literature demonstrates the strength of artificial neural network-based forecasters, but it also underscores persistent challenges: the fitted mechanisms are rarely interpretable at the time lag response level, hyperparameter tuning can be substantial, and calibrated uncertainty is usually tied to comparatively complex learning pipelines.

The main idea of this article comes from Bernstein polynomials and their ability to approximate continuous functions defined on a compact interval (Lorentz, 1986). The newly proposed Bernstein Polynomial Autoregressive Model, abbreviated as BPAR, captures temporal dependence in nonlinear time series using these Bernstein polynomials. The resulting model architecture resembles a one-layer artificial neural network with a structured basis dictionary, yet the fitted effects remain directly interpretable lag by lag through the Bernstein coefficients. Compared to deep or multilayer neural network forecasters, this single structured layer avoids learned hidden representations and substantially reduces architectural choices such as the number of layers, hidden units, and activation functions. At the same time, each lag has its own Bernstein coefficient block, so posterior summaries of these blocks can indicate which past observations contribute most strongly to prediction. BPAR is particularly natural for chaotic time series in which the next observation is often a smooth nonlinear function of recent history. An additive extension, called AR-BPAR in this paper, augments the Bernstein model branch with a linear autoregressive component. This autoregressive component captures linear dependence, while the Bernstein branch captures nonlinear effects, and this is seen to be useful in modeling time series data with a mix of linear and nonlinear temporal dependence.

Two Bayesian inference strategies are studied in this paper. First, a variational Bayesian approach (Graves, 2011; Blundell et al., 2015) that learns model parameters without knowledge of the exact posterior distribution. Second, an exact Bayesian inference method wherein Gaussian priors along with an inverse-gamma variance prior leads to analytical expressions for the posterior distributions; Section 2.3 compares these two inference routes. These two Bayesian routes involve carefully designed shrinkage prior distributions on certain model parameters and this assists in controlling the total number of model parameters. Posterior credible intervals then help determine the statistical significance of all model parameters and this assists in model pruning. The BPAR architecure resembles a single layer artificial neural network. A notable advantage of the variational Bayesian approach, over the exact Bayesian approach, is that it can cater to the addition of more layers in the model architecture.

The main contributions are summarized as follows.

- The new BPAR framework provides a highly flexible model class for modeling and forecasting a wide range of nonlinear time series. An extension to include a linear autoregressive model term further allows the model class to adapt to time series with mixed linear and nonlinear temporal dependence.
- Viewed as a hybrid between statistical time series models and artificial neural network models for nonlinear time series forecasting, the BPAR enables model interpretability and predictive uncertainty quantification via Bayesian inference. BPAR is better than classical statistical models such as ARIMA/SARIMA by accommodating highly nonlinear temporal dynamics. Simultaneously, its single layer Bernstein polynomial structure avoids the typical hyperparameter selection, such as number of hidden layers/nodes and activation function, challenges faced by artificial neural network models, while retaining lag-level summaries of which past observations drive prediction.
- Two Bayesian inference approaches are explored: exact Bayesian inference and variational Bayesian inference. The former brings computational advantages while the latter brings more flexibility in terms of prior distribution selection and number of layers in the model architecture. Shrinkage prior

distributions are proposed and these help control the overall size of the mode parameter vector. Posterior credible intervals on all parameters enable a systematic approach to model pruning.
- An extensive empirical study spanning simulated examples, chaotic time series benchmarks and real nonlinear time series is conducted, and the advantages of the BPAR model are discussed.

2 Methodology

In this section, the Bernstein Polynomial Autoregressive Model (BPAR) and its additive autoregressive extension (AR-BPAR) are described. We start with the approximation theory on Bernstein polynomials, then introduce our new models, and finally outline the Bayesian inference procedures.

2.1 Approximation theory using Bernstein Polynomials

For $z \in [0, 1]$, the Bernstein basis polynomial of degree m is written as

$$B_{k,m}(z) = \binom{m}{k} z^k (1-z)^{m-k}, \qquad k = 0, \ldots, m.$$

The key property of interest is that Bernstein polynomials well approximate continuous functions on a compact interval. More precisely, let $f : [0, 1] \to \mathbb{R}$ be continuous. Its Bernstein polynomial approximation of degree m is

$$(\mathcal{B}_m f)(z) = \sum_{k=0}^{m} f\left(\frac{k}{m}\right) B_{k,m}(z). \tag{1}$$

The Bernstein approximation theorem states that

$$\lim_{m\to\infty} (\mathcal{B}_m f)(z) = f(z)$$

uniformly on $[0, 1]$; see Feller (1966), Lorentz (1986), and Beals (2004). This result provides the mathematical motivation and foundation for representing the nonlinear impact of past values of a time series on its future.

To visualize this approximation theorem, consider the continuous function

$$g(z) = 3z + 3\sin(4\pi z) + 2\cos(5\pi z), \qquad z \in [0, 1].$$

Figure 1 plots approximations of function $g(\cdot)$ with Bernstein polynomials at degrees $m \in \{5, 10, 20, 40, 80, 160, 320\}$. The low-degree curves smooth out local oscillations, while the high-degree curves nearly overlap the true function. It is precisely this property that is used to build our BPAR model in Section 2.2 and beyond. Specifically, each time-lagged observation of the time series is modeled to influence its future observations by a continuous function that is then approximated by Bernstein polynomials.

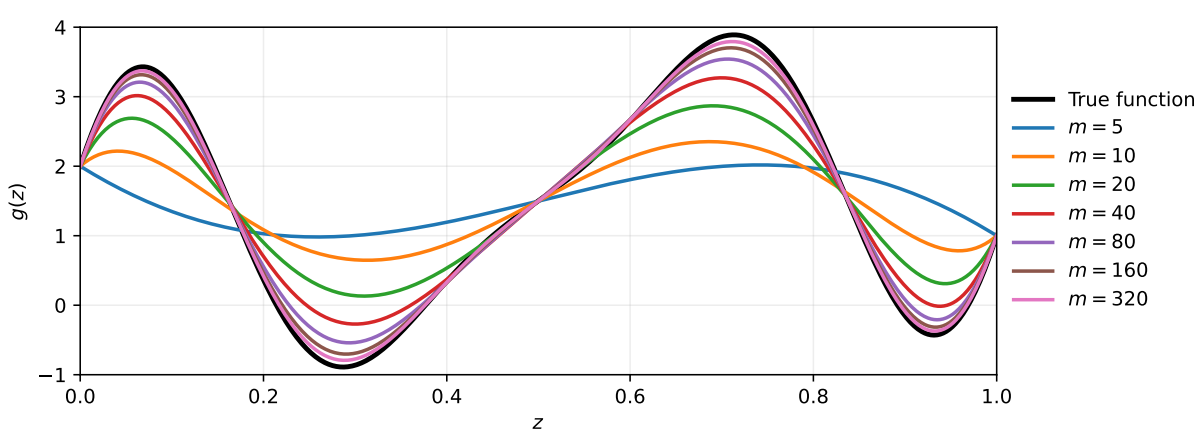

Figure 1: Approximations of the function $g(z) = 3z + 3\sin(4\pi z) + 2\cos(5\pi z)$ on $[0, 1]$ at varying Bernstein polynomial degree m.

2.2 Bernstein Polynomial Autoregression (BPAR)

Let x_t be the observed nonlinear time series. The BPAR model of order d_B is written as

$$x_t = \alpha + \beta \sum_{j=1}^{d_B} \sum_{k=0}^{m} w_{j,k} B_{k,m}(z_{t-j}) + \varepsilon_t, \;\; \varepsilon_t \sim \mathcal{N}(0, \sigma^2), \tag{2}$$

where the coefficients $\alpha, \beta \in \mathbb{R}$, and m is the Bernstein polynomial degree. We denote this model by BPAR(d_B). The injective map $h: \mathbb{R} \to [0,1]$ yields $z_{t-j} = h(x_{t-j})$, and the nonlinear temporal autocorrelations are captured via the Bernstein polynomial representation using the sequence of random variables $\{z_t\}$. The sequence of random variables $\{\epsilon_t\}$ denotes the error component.

The BPAR(d_B) model's parameter vector is written as $\theta = (\alpha, \beta, \mathbf{w}^\top)^\top$, with the Bernstein weight coefficients $\mathbf{w} = (w_{1,0}, \ldots, w_{1,m}, \ldots, w_{d_B,0}, \ldots, w_{d_B,m})^\top$. We then define

$$\mathbf{b}_m(z) = [B_{0,m}(z), \ldots, B_{m,m}(z)]^\top, \;\; \phi_t = \left[\mathbf{b}_m(z_{t-1})^\top, \ldots, \mathbf{b}_m(z_{t-d_B})^\top\right]^\top.$$

Then, the BPAR(d_B) model from (2) can be expressed as

$$\mu_t(\theta) = \alpha + \beta \phi_t^\top \mathbf{w}, \quad x_t = \mu_t(\theta) + \varepsilon_t. \tag{3}$$

The BPAR model architecture is illustrated in Figure 2. This is seen to resemble a single layer artificial neural network with the Bernstein polynomials playing the role of pseudo-activation functions. This single layer representation is simpler than deep multilayer neural network forecasters because the nonlinear transformations are supplied by a fixed Bernstein basis rather than by several learned hidden layers. The coefficient block $\mathbf{w}_j = (w_{j,0}, \ldots, w_{j,m})^\top$ controls the nonlinear contribution of time lag j, hence posterior means and credible intervals for $\mathbf{w}_j$ provide a direct way to assess which time lags drive predictions. With such a structure in place, the BPAR caters to Bayesian statistical inference and this is described in Section 2.3. This inferential method enables uncertainty quantification regarding the model parameters and also on the time series' predictions. Our extensive empirical study in Section 3 shows superior forecasting performance of the BPAR model over classical linear time series models when the time series data is highly nonlinear, for example chaotic time series. Further, BPAR's performance is seen to be on par or better than its artificial neural network competitors such as LSTM and MLP.

Remark 2.1. Prediction intervals using nonlinear autoregressive models have been developed and studied in the statistical literature; see Franke et al. (2002); Pan & Politis (2016); Wu & Politis (2024) for examples. The model assumption in these works is that $x_t = m(x_{t-1}, x_{t-2}, \ldots, m_{t-p}) + \epsilon_t$, for an order p model wherein $m(\cdot)$ is a possibly nonlinear function. If $m(\cdot)$ is assumed to be smooth, the problem then becomes a nonparametric autoregressive model with $m(\cdot)$ being estimated using kernel methods. In practice, when implementing their methods, $m(\cdot)$ is estimated using kernels or assumed to be known. Our BPAR(d_B) model in (2) is more flexible than the above described setup as it models the nonlinear influence of each time-lagged version of the time series via the z_t random variables in (2). As evidenced by the results in our empirical study in Section 3, this previous point is seen to be true when dealing with chaotic time series. Further, the Bernstein polynomials help avoid the difficulty in selecting the unknown $m(\cdot)$ function.

2.2.1 Extension to AR-BPAR

The BPAR model from Section 2.2 is very well suited for highly nonlinear or chaotic time series data. There are many examples of real time series data displaying a mix of linear and nonlinear temporal autocorrelations, with some examples wherein the linear effects are much more prominent than the nonlinear effects. To accommodate such types of time series data, we present the AR(d_L)-BPAR(d_B) model below which extends the BPAR(d_B) model. Specifically, we have

$$x_t = \alpha + \sum_{i=1}^{d_L} \rho_i x_{t-i} + \sum_{j=1}^{d_B} \sum_{k=0}^{m} w_{j,k} B_{k,m}(z_{t-j}) + \varepsilon_t, \;\; \varepsilon_t \sim \mathcal{N}(0, \sigma^2), \tag{4}$$

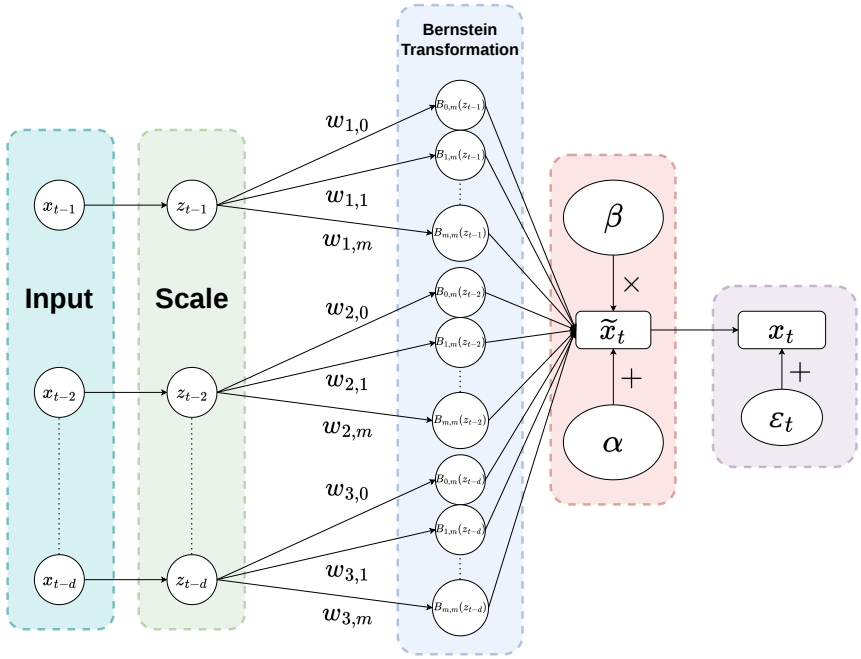

Figure 2: BPAR(d_B) model architecture.

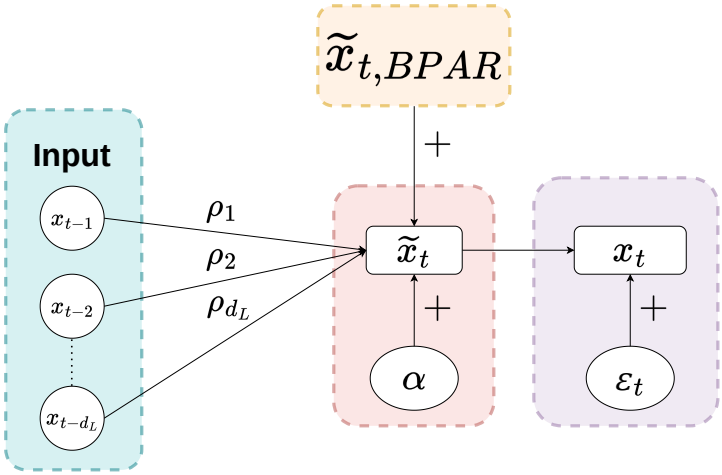

Figure 3: AR(d_L)-BPAR(d_B) model architecture. Note that $\tilde{x}_{t,BPAR}$ is the $\tilde{x}$ from the BPAR(d_B) architecture in Figure 2.

where d_L is the linear autoregressive model order. The AR(d_L)-BPAR(d_B) can also be written as

$$\begin{aligned} x_t &= \mathbf{f}_t^\top \theta + \varepsilon_t, \\ \theta &= (\alpha, \rho_1, \ldots, \rho_{d_L}, w_{1,0}, \ldots, w_{1,m}, \ldots, w_{d_B,0}, \ldots, w_{d_B,m})^\top, \\ \mathbf{f}_t &= [1,\, x_{t-1}, \ldots, x_{t-d_L},\, B_{0,m}(z_{t-1}), \ldots, B_{m,m}(z_{t-d_B})]^\top. \end{aligned} \tag{5}$$

Figure 3 shows the architecture of the AR(d_L)-BPAR(d_B) model. The empirical study in Section 3 provides evidence on the usefulness of this extension when dealing with time series data that include a mix of linear and nonlinear temporal dependence.

2.3 Bayesian Inference

Let θ denote the vector of model parameters and let $K_\theta = \dim(\theta)$ denote its size. For the BPAR(d_B) model from Section 2.2, $\theta = (\alpha, \beta, \mathbf{w}^\top)^\top$ and $K_\theta = 2+d_B(m+1)$. Here, the $d_B(m+1)$-dimensional vector $\mathbf{w}$ denotes the Bernstein weight coefficients; see Figure 2. For AR(d_L)-BPAR(d_B) from Section 2.2.1, $\theta = (\alpha, \rho^\top, \mathbf{w}^\top)^\top$ with $K_\theta = 1+d_L+d_B(m+1)$. Here, $\rho = (\rho_1, \rho_2, \ldots, \rho_{d_L})$ denotes the coefficients corresponding to the linear autoregressive part of the model; see Figure 3. We now detail two inference approaches, namely variational Bayesian inference and exact Bayesian inference.

2.3.1 Bayesian variational inference

Bayesian variational inference is a popular estimation method used in machine learning models wherein the exact posterior distributions become intractable Hoffman et al. (2013); Blei et al. (2017). Another notable advantage is the avoidance of MCMC algorithms to draw samples from the posterior distributions as this can become computationally expensive. With the BPAR(d_B) model from Section 2.2, the Gaussian prior distribution specification on the parameter vector θ is written as

$$p(\theta) = \mathcal{N}(m_0, V_0), \tag{6}$$

where V_0 is the prior diagonal covariance matrix and m_0 is the prior mean vector. The intercept parameter α and the scale parameter β are assigned fixed weakly informative prior variances, while the Bernstein weights are assigned lag- and Bernstein degree-adaptive shrinkage variances. More explicitly,

$$m_0 = \left(m_{0,\alpha}, m_{0,\beta}, \mathbf{0}^\top_{d_B(m+1)}\right)^\top, \qquad V_0 = \operatorname{diag}\left(\tau_\alpha^2, \tau_\beta^2, v^{\mathrm{B}}_{1,0}, \ldots, v^{\mathrm{B}}_{d_B,m}\right),$$

with

$$v^{\mathrm{B}}_{j,k} = \operatorname{Var}(w_{j,k}) = \frac{\tau_{\mathrm{B}}^2}{j^{\gamma_{\mathrm{B},\ell}}(k+1)^{\gamma_{\mathrm{B},m}}}, \qquad j = 1, \ldots, d_B, \quad k = 0, \ldots, m.$$

Here τ_α^2 and τ_β^2 do not depend on lag or the Bernstein polynomial degree m, and they control only the global location and scale of the nonlinear response. In contrast, $v^{\mathrm{B}}_{j,k}$ decreases with lag index j and Bernstein basis index k when $\gamma_{\mathrm{B},\ell}$ or $\gamma_{\mathrm{B},m}$ are positive. Therefore the shrinkage mechanism here penalizes selecting a high model order d_B and/or high Bernstein polynomial degree m. To simplify technical exposition, we explain the variational inference method under the Gaussian prior family. Note that with the variational inference framework other prior probability distributions can also be used.

The posterior distribution is approximated by a multivariate Normal distribution that is written as

$$q(\theta) = \mathcal{N}(m_q, \Sigma_q), \qquad \Sigma_q = L_q L_q^\top, \tag{7}$$

where m_q is a learnable mean vector, and L_q is a learnable lower-triangular Cholesky factor. Sampling from this distribution is implemented through the reparameterization

$$\theta = m_q + L_q \xi, \qquad \xi \sim \mathcal{N}(0, I), \tag{8}$$

so that stochastic gradients can be propagated through Monte Carlo draws of the parameters. Next, the Kullback–Leibler divergence between the prior and posterior distributions is expressed as

$$\mathrm{KL}(q(\theta) \,\|\, p(\theta)) = \frac{1}{2}\left[\operatorname{tr}\!\left(V_0^{-1}\Sigma_q\right) + (m_q - m_0)^\top V_0^{-1}(m_q - m_0) - K_\theta + \log\frac{\det V_0}{\det \Sigma_q}\right], \tag{9}$$

where K_θ is the dimension of θ. The training objective function is the negative evidence lower bound given by

$$\mathcal{L}_{\mathrm{VI}} = -\mathbb{E}_{q(\theta)}\left[\log p(\mathbf{x} \mid \theta, \sigma^2)\right] + \lambda\, \mathrm{KL}(q(\theta) \,\|\, p(\theta)), \tag{10}$$

where λ is the KL-weight used in training. The gradients of (10) are obtained via backpropagation; see (Rumelhart et al., 1986; Goodfellow et al., 2016) for details. After minimizing the objective function in

(10), the optimized variational mean and covariance are denoted by $\hat{m}_q$ and $\hat{\Sigma}_q$, respectively. Thus, for any particular model parameter θ_j, the posterior marginal distribution can be written as

$$q(\theta_j) \approx \mathcal{N}(\hat{m}_{q,j}, \hat{\Sigma}_{q,jj}),$$

where $\hat{m}_{q,j}$ and $\hat{\Sigma}_{q,jj}$ are estimated by the backpropagation procedure used to minimize (10). This then yields an approximate $100(1-\alpha)\%$ posterior credible interval as

$$\theta_j \in \hat{m}_{q,j} \pm z_{1-\alpha/2}\sqrt{\hat{\Sigma}_{q,jj}}.$$

2.3.2 Exact Bayesian Inference for AR-BPAR

With the AR(d_L)-BPAR(d_B) model from Section 2.2.1, a sample size of n and row-aligning f_t from (5) across the t index produces a design matrix $H \in \mathbb{R}^{n \times K_\theta}$. The model can be then rewritten as

$$\mathbf{x} = H\theta + \boldsymbol{\varepsilon}, \qquad \boldsymbol{\varepsilon} \sim \mathcal{N}(0, \sigma^2 I_n). \tag{11}$$

The conjugate prior specification here is

$$\theta \mid \sigma^2 \sim \mathcal{N}(m_0, \sigma^2 V_0), \tag{12}$$

$$\sigma^2 \sim \text{InvGamma}(a_0, b_0), \tag{13}$$

where V_0 is a diagonal matrix, and

$$m_0 = \left(m_{0,\alpha}, \mathbf{0}_{d_L}^\top, \mathbf{0}_{d_B(m+1)}^\top\right)^\top, \qquad V_0 = \text{diag}\left(\tau_\alpha^2, v_1^{\text{AR}}, \ldots, v_{d_L}^{\text{AR}}, v_{1,0}^{\text{B}}, \ldots, v_{d_B,m}^{\text{B}}\right),$$

where

$$v_i^{\text{AR}} = \frac{\tau_{\text{AR}}^2}{i^{\gamma_{\text{AR}}}}, \qquad v_{j,k}^{\text{B}} = \frac{\tau_{\text{B}}^2}{j^{\gamma_{\text{B},\ell}}(k+1)^{\gamma_{\text{B},m}}}.$$

Equivalently,

$$\text{Var}(\alpha \mid \sigma^2) = \sigma^2\tau_\alpha^2, \qquad \text{Var}(\rho_i \mid \sigma^2) = \sigma^2 v_i^{\text{AR}}, \qquad \text{Var}(w_{j,k} \mid \sigma^2) = \sigma^2 v_{j,k}^{\text{B}}.$$

In other words, the above hierarchy means that the intercept α receives a fixed weakly informative prior variance, while the AR coefficients and Bernstein weights can be shrunk by their lag orders (d_L, d_B) and, for the Bernstein block, also by the polynomial degree m.

To derive the posterior distribution in closed form, the Gaussian data likelihood is written as

$$p(\mathbf{x} \mid \theta, \sigma^2, H) \propto (\sigma^2)^{-n/2} \exp\left\{-\frac{1}{2\sigma^2}(\mathbf{x} - H\theta)^\top(\mathbf{x} - H\theta)\right\}, \tag{14}$$

and the Gaussian prior density is

$$p(\theta \mid \sigma^2) \propto (\sigma^2)^{-K_\theta/2} \exp\left\{-\frac{1}{2\sigma^2}(\theta - m_0)^\top V_0^{-1}(\theta - m_0)\right\}. \tag{15}$$

Conditional on σ^2, the posterior density of θ is proportional to the product of (14) and (15). The quadratic part involving θ is

$$\begin{aligned} Q(\theta) &= (\mathbf{x} - H\theta)^\top(\mathbf{x} - H\theta) + (\theta - m_0)^\top V_0^{-1}(\theta - m_0) \\ &= \theta^\top(H^\top H + V_0^{-1})\theta - 2\theta^\top(H^\top\mathbf{x} + V_0^{-1}m_0) + \mathbf{x}^\top\mathbf{x} + m_0^\top V_0^{-1}m_0. \end{aligned} \tag{16}$$

Let

$$V_n = (V_0^{-1} + H^\top H)^{-1}, \qquad m_n = V_n(V_0^{-1}m_0 + H^\top\mathbf{x}).$$

Then $V_n^{-1}m_n = V_0^{-1}m_0 + H^\top \mathbf{x}$, and completing the square gives

$$Q(\theta) = (\theta - m_n)^\top V_n^{-1}(\theta - m_n) + \mathbf{x}^\top \mathbf{x} + m_0^\top V_0^{-1} m_0 - m_n^\top V_n^{-1} m_n. \tag{17}$$

The final three terms do not depend on θ. Combining this completed square with the inverse-gamma prior for σ^2 gives the posterior updates

$$V_n = (V_0^{-1} + H^\top H)^{-1}, \tag{18}$$

$$m_n = V_n(V_0^{-1}m_0 + H^\top \mathbf{x}), \tag{19}$$

$$a_n = a_0 + \frac{n}{2}, \tag{20}$$

$$b_n = b_0 + \frac{1}{2}\left(\mathbf{x}^\top \mathbf{x} + m_0^\top V_0^{-1} m_0 - m_n^\top V_n^{-1} m_n\right). \tag{21}$$

Hence,

$$\theta \mid \sigma^2, \mathbf{x} \sim \mathcal{N}(m_n, \sigma^2 V_n), \qquad \sigma^2 \mid \mathbf{x} \sim \mathrm{InvGamma}(a_n, b_n). \tag{22}$$

Parameter uncertainty is quantified using the posterior distribution in (22). Marginally, after integrating out σ^2, each model parameter satisfies

$$\theta_j \mid \mathbf{x} \sim t_{2a_n}\left(m_{n,j}, \frac{b_n}{a_n}(V_n)_{jj}\right),$$

where the second argument is again written in variance form. Hence, a Bayesian $100(1-\alpha)\%$ posterior credible interval for θ_j is

$$m_{n,j} \pm t_{2a_n,\,1-\alpha/2}\sqrt{\frac{b_n}{a_n}(V_n)_{jj}}.$$

The Student-t posterior predictive distribution follows by integrating over both θ and σ^2. For a new feature vector $\mathbf{f}_*$ and future observation x_*,

$$x_* \mid \theta, \sigma^2, \mathbf{x} \sim \mathcal{N}(\mathbf{f}_*^\top \theta, \sigma^2).$$

Integrating out $\theta \mid \sigma^2, \mathbf{x} \sim \mathcal{N}(m_n, \sigma^2 V_n)$ gives

$$x_* \mid \sigma^2, \mathbf{x} \sim \mathcal{N}\big(\mathbf{f}_*^\top m_n, \sigma^2\left(1 + \mathbf{f}_*^\top V_n \mathbf{f}_*\right)\big). \tag{23}$$

The factor $\mathbf{f}_*^\top V_n \mathbf{f}_*$ is the posterior coefficient-uncertainty contribution, while the leading 1 is new observation noise. Finally, integrating out $\sigma^2 \mid \mathbf{x} \sim \mathrm{InvGamma}(a_n, b_n)$ gives

$$x_* \mid \mathbf{x} \sim t_{2a_n}\left(\mathbf{f}_*^\top m_n,\ \frac{b_n}{a_n}\left(1 + \mathbf{f}_*^\top V_n \mathbf{f}_*\right)\right), \tag{24}$$

where the first argument is the degrees of freedom and the second scale argument is written in variance form. Therefore, the posterior predictive mean is $\mathbf{f}_*^\top m_n$ when $a_n > 1/2$, and the predictive variance is

$$\frac{b_n}{a_n - 1}\left(1 + \mathbf{f}_*^\top V_n \mathbf{f}_*\right), \qquad a_n > 1.$$

The main advantage of this exact Bayesian estimation route is that posterior shrinkage, parameter uncertainty, and predictive uncertainty are all obtained in closed form without the iterative optimization procedure done in the variational inference in Section 2.3.1.

2.4 Model hyperparameter selection

With the BPAR(d_B) model from Section 2.2, let η_B denote a hyperparameter configuration and this can be written as

$$\eta_{\mathrm{B}} = (d_B, m, \tau_{\mathrm{B}}, \gamma_{\mathrm{B}}, \lambda, \eta_{\mathrm{opt}}),$$

where m is the Bernstein polynomial degree, τ_{B} and γ_{B} are the shrinkage parameters, λ is the variational KL weight, and η_{opt} collects optimizer settings such as learning rate and weight decay. With the AR(d_L)-BPAR(d_B) model from Section 2.2.1, the hyperparameter configuration has the form

$$\eta_{\mathrm{AB}} = (d_L, d_B, m, \tau_{\mathrm{AR}}, \tau_{\mathrm{B}}, \gamma_{\mathrm{AR}}, \gamma_{\mathrm{B}}),$$

where $\{\tau_{\mathrm{AR}}, \gamma_{\mathrm{AR}}\}$ are the shrinkage parameters corresponding to the linear autoregressive part of the model, $\{\tau_{\mathrm{B}}, \gamma_{\mathrm{B}}\}$ are the shrinkage parameters corresponding to the nonlinear part of the model.

Let $\mathcal{G}$ be the finite grid of candidate configurations. For each $\eta \in \mathcal{G}$, the model is fit on the training split and evaluated on the validation split. The validation error is

$$R_{\mathrm{val}}(\eta) = \left[\frac{1}{|\mathcal{V}|} \sum_{t \in \mathcal{V}} \left(x_t - \hat{x}_t^{(\eta)} \right)^2 \right]^{1/2}, \tag{25}$$

where $\mathcal{V}$ is the validation index set. Residual temporal autocorrelations are summarized during grid selection by the validation Ljung–Box test p-value denoted by $P_{\mathrm{LB}}(\eta)$. Recall that the Ljung–Box test has a null hypothesis that the residuals are a white noise time series without any remaining temporal correlations. Smaller $R_{\mathrm{val}}(\eta)$ indicates better validation accuracy, while larger $P_{\mathrm{LB}}(\eta)$ indicates weaker evidence of residual temporal autocorrelations.

The implemented hyperparameter selection rule gives first priority to the condition in (25), and then second priority to the Ljung–Box p-values $P_{\mathrm{LB}}(\eta)$. With a small choice for δ, we define the near-best hyperparameter selection set as

$$\mathcal{G}_\delta = \left\{ \eta \in \mathcal{G} : R_{\mathrm{val}}(\eta) \leq (1+\delta) \min_{\eta' \in \mathcal{G}} R_{\mathrm{val}}(\eta') \right\}.$$

The selected hyperparameter configuration is then defined as

$$\hat{\eta} = \arg\max_{\eta \in \mathcal{G}_\delta} P_{\mathrm{LB}}(\eta). \tag{26}$$

Another way to understand our procedure above is that the final hyperparameter configuration choice never sacrifices more than a fraction δ of the best validation error, and it prefers the near-best configuration with the most satisfactory residuals.

3 Empirical study

We test the performance of the new methods over a wide range of simulated, benchmark and real time series datasets. Models are trained and tuned on the training and validation sets, and the main paper reports held-out test set performance comparisons. Forecast accuracy is summarized by root mean squared error (RMSE) and mean absolute error (MAE). Residual Ljung–Box diagnostics are used as a secondary check for remaining temporal dependence (Ljung & Box, 1978). Lower RMSE and MAE are better, while larger Ljung–Box p-values indicate weaker signal of leftover autocorrelation. As competitors we consider the linear autoregressive model often abbreviated as AR(p), its extension SARIMA that includes trend and seasonal terms, and two artificial neural network models: the MLP and LSTM. For the injective map h used in the model definitions in Sections 2.2 and 2.2.1, we explored two choices: min–max scaling based on the observations range and a logistic sigmoid function. These two choices yielded very similar results, and the reported results are based on the first choice. The empirical study includes nonlinear time series simulation schemes, a few well-known chaotic time series examples, and two real nonlinear time series. All datasets use chronological train/validation/test splits. Models A–C in Section 3.1.1 use 180/60/60 observations split. In Section 3.1.2 with chaotic time series benchmarks, the Ricker, Logistic, and Gauss maps use 300/100/100 observations as the split, and the Mackey–Glass uses a 1400/300/300 split. In Section 3.2 with the two real datasets, we use a 720/144/144 observations split. The main paper reports held-out test forecasting results, while Appendix A provides a more detailed report on each dataset. Specifically, the appendix includes time series plots of the train/validation/test splits, tables with precise RMSE, MAE, and Ljung–Box p-values, forecast figures with posterior predictive intervals, and full train/validation/test comparison tables.

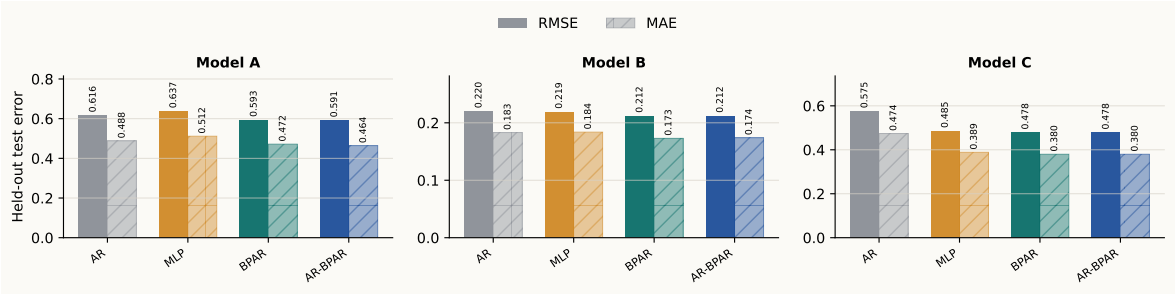

Figure 4: Held-out test RMSE and MAE for Models A–C. The competing methods are AR, MLP, BPAR, and AR-BPAR.

3.1 Simulated and chaotic benchmark datasets

3.1.1 Simulation schemes

The first set of experiments uses three nonlinear autoregressive simulation schemes adapted from the bootstrap prediction study of Wu & Politis (2024). These three schemes, called Models A–C, are given below.

$$\begin{aligned}
&\text{Model A:} \quad x_t = 0.1x_{t-1}I(x_{t-1} \le 0) + 0.8x_{t-1}I(x_{t-1} > 0) + \varepsilon_t, \\
&\text{Model B:} \quad x_t = \{0.1x_{t-1} + 0.5e^{-x_{t-1}^2}\varepsilon_t\}I(x_{t-1} \le 0) + \{0.8x_{t-1} + 0.5e^{-x_{t-1}^2}\varepsilon_t\}I(x_{t-1} > 0), \\
&\text{Model C:} \quad x_t = 0.2 + \log(0.5 + |x_{t-1}|) + \varepsilon_t,
\end{aligned} \tag{27}$$

where $\varepsilon_t \sim \mathcal{N}(0, 0.5^2)$. Figure 4 shows the held-out test RMSE and MAE comparison across competing methods in each panel. The AR(d_L)-BPAR(d_B) model reveals slightly better performance than its competitors, with the difference being most evident in Model C. Appendix A.1 reports the full results, and we witness comparable performance between all methods for Models A and B, but better results under the new AR(d_L)-BPAR(d_B) model for Model C. The mixed results can be attributed to the extent of nonlinearity present in the time series, with Model C being the most nonlinear. The next section includes examples of highly nonlinear time series, also called chaotic time series.

3.1.2 Chaotic time series benchmarks

Multiple real world processes arising from energy systems, atmospheric dynamics and financial markets are viewed as nonlinear dynamical systems Kui & Lai (2025). The time series data from these applications are highly nonlinear and present a difficult forecasting challenge. To evaluate the performance of our new methods, we consider four chaotic time series benchmarks from Python's `dysts` library Gilpin (2023a). We now provide details of the four time series along with the forecast comparison.

Ricker map: $x_{t+1} = x_t \exp(a - x_t)$,

Logistic map: $x_{t+1} = rx_t(1 - x_t), \quad r = 3.9$,

Gauss map: $x_{t+1} = \exp(-ax_t^2) + b, \quad a = 4.9, \; b = -0.5$,

Mackey–Glass delay system: $dx(t)/dt = \beta x(t-\tau)/(1 + x(t-\tau)^\eta) - \gamma x(t), \quad \beta = 0.2, \; \gamma = 0.1, \quad \eta = 10, \quad \tau = 17$.

The Ricker panel in Figure 5 shows that our new BPAR model is the best performing method by test RMSE and test MAE. Appendix A.2.1 provides more detailed results on RMSE, MAE, Ljung–Box residual p-values, the training/validation/test split plot of the time series, forecast comparison, and other performance related summaries. A similar story is seen with the Logistic map, Gauss map, and Mackey–Glass delay system; see Figure 5. Additional method-specific results, including LSTM where it was run, remain in the appendix tables. More detailed performance results on the Logistic and Gauss maps can be found in Appendix A.2.2, and results on the Mackey–Glass delay system can be found in Appendix A.2.3.

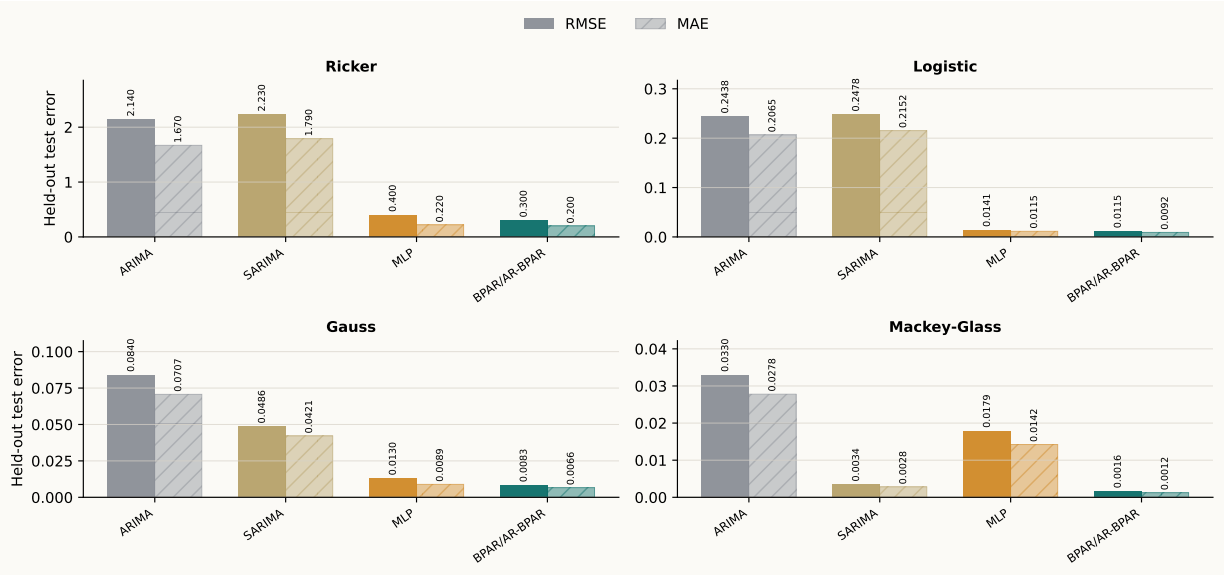

Figure 5: Held-out test RMSE and MAE summaries for the chaotic time series benchmark datasets. Each panel compares the common set of methods available for all four chaotic benchmarks: ARIMA, SARIMA, MLP, and BPAR/AR-BPAR.

Observe that the chaotic time series benchmarks chosen here are more nonlinear than the simulation schemes discussed in Section 3.1.1. A central feature in chaotic time series models is that the influence of the past on the future is assumed to be smooth; see Lele (1994); Mukherjee et al. (1997) for a more technical view of chaotic time series. Our proposed BPAR model from Section 2.2 is tailored for such situations as Bernstein polynomials well approximate bounded continuous functions. This is resulting in the superior forecast performance of the BPAR model seen in this section. A key advantage of the proposed method is that unlike much of existing statistical literature on nonlinear time series models, see Tiao & Tsay (1994); Pan & Politis (2016); Wu & Politis (2024) for examples, our modeling and forecasting approach does not require knowledge of the functional form that defines how the past influences the future of the time series. While artificial neural networks have been utilized for forecasting such highly nonlinear time series, see Kui & Lai (2025) for a recent example, our approach enables model tractability and provides prediction uncertainty quantification.

3.2 Real datasets

We consider two real time series data examples that have, in the past, been modeled and predicted as nonlinear time series. The first is the Ontario electricity price series, drawn from public reports maintained by the Independent Electricity System Operator (IESO)[1] (Independent Electricity System Operator, 2026). Ontario wholesale electricity prices are volatile, spike-prone, and have previously been modeled using nonlinear autoregressive neural networks and neural network price-spike forecasting methods (Andalib & Atry, 2009; Sandhu et al., 2016). The second is the Beijing $PM_{2.5}$ pollution series from the UCI Machine Learning Repository[2] (Chen, 2015). This hourly dataset records $PM_{2.5}$ measurements from the U.S. Embassy in Beijing together with meteorological covariates from Beijing Capital International Airport during 2010–2014. Beijing $PM_{2.5}$ has also been studied as a nonlinear and volatile forecasting problem using deep learning and hybrid time-series models (Yang et al., 2021; Zhao et al., 2022).

In terms of forecasting performance, for the Ontario electricity prices time series, Figure 6 shows that the LSTM and our new AR-BPAR model are the two best performing models with both showing comparable

[1] `https://www.ieso.ca/en/power-data/data-directory`
[2] `https://archive.ics.uci.edu/dataset/381/beijing+pm2+5+data`

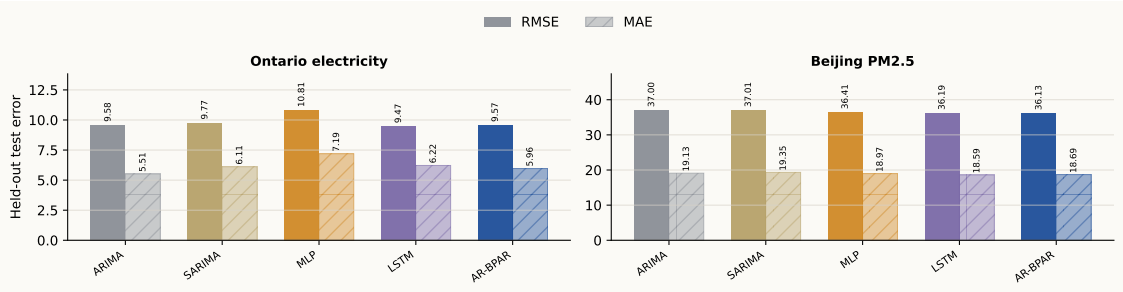

Figure 6: Held-out test RMSE and MAE summaries for the real-data benchmarks. Both panels use the same five methods: ARIMA, SARIMA, MLP, LSTM, and AR-BPAR.

results. A similar story is seen with the Beijing $PM_{2.5}$ time series. Detailed performance results on these two real time series datasets can be found in Appendix A.3.

4 Conclusion

This article introduces a new class of nonlinear time series models by tapping into fundamental properties of Bernstein polynomials. The newly proposed BPAR, and its extension to AR-BPAR, modeling framework are highly flexible in modeling a wide variety of nonlinear time series data. They serve as a useful statistical middle ground between rigid linear time series models and fully black-box neural network forecasters. Bayesian inference enables uncertainty quantification on model parameters and also on the predictions. Shrinkage prior distributions are placed on certain model parameters that control the overall size of the model parameter vector. Posterior credible intervals on each model parameter makes model pruning possible. With similarities seen between the new BPAR and a single layer artificial neural network, the foundational work in this article can lead to extensions involving additional hidden layers in the BPAR model architecture. In such situations, the variational Bayesian inference approach can be utilized to quantify prediction uncertainty.

Results from extensive empirical experiments point to the usefulness of the new models. With time series data arising from chaotic systems, the new BPAR model substantially outperforms classical statistical models. In these data settings, the BPAR also does better than artificial neural network competitors. Overall, the empirical findings make a strong case for the new nonlinear time series modeling framework using Bernstein polynomials, especially when time lag-level interpretation and uncertainty quantification are central scientific goals.

References

Pau Vilimelis Aceituno, Jack William Miller, Noah Marti, Youssef Farag, and Victor Boussange. Temporal horizons in forecasting: A performance-learnability trade-off. *Transactions on Machine Learning Research*, 2025. URL `https://openreview.net/forum?id=BeudQIxT1R`.

Arash Andalib and Farid Atry. Multi-step ahead forecasts for electricity prices using NARX: A new approach, a critical analysis of one-step ahead forecasts. *Energy Conversion and Management*, 50(3):739–747, 2009. doi: 10.1016/j.enconman.2008.09.040.

Richard Beals. *Analysis: An Introduction*. Cambridge University Press, 2004. ISBN 9780521600470.

David M. Blei, Alp Kucukelbir, and Jon D. McAuliffe. Variational inference: A review for statisticians. *Journal of the American Statistical Association*, 112(518):859–877, 2017. doi: 10.1080/01621459.2017.1285773.

Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. Weight uncertainty in neural networks. In *Proceedings of the 32nd International Conference on Machine Learning*, pp. 1613–1622, 2015.

George E. P. Box, Gwilym M. Jenkins, Gregory C. Reinsel, and Greta M. Ljung. *Time Series Analysis: Forecasting and Control*. Wiley, 5 edition, 2015.

Song Chen. Beijing PM2.5. UCI Machine Learning Repository, 2015. URL https://archive.ics.uci.edu/dataset/381/beijing+pm2+5+data.

William Feller. *An Introduction to Probability Theory and Its Applications, Volume II*. John Wiley & Sons, 1966.

Jürgen Franke, Jens-Peter Kreiss, and Enno Mammen. Bootstrap of kernel smoothing in nonlinear time series. *Bernoulli*, 8(1):1–37, 2002. doi: 10.2307/3318738.

William Gilpin. Chaos as an interpretable benchmark for forecasting and data-driven modelling, 2023a. URL https://arxiv.org/abs/2110.05266.

William Gilpin. Model scale versus domain knowledge in statistical forecasting of chaotic systems, 2023b. URL https://arxiv.org/abs/2303.08011.

Ian Goodfellow, Yoshua Bengio, Aaron Courville, and Yoshua Bengio. *Deep learning*, volume 1. MIT Press, 2016.

Alex Graves. Practical variational inference for neural networks. In *Advances in Neural Information Processing Systems 24*, 2011. URL https://papers.neurips.cc/paper/4329-practical-variational-inference-for-neural-networks.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997.

Matthew D. Hoffman, David M. Blei, Chong Wang, and John Paisley. Stochastic variational inference. *Journal of Machine Learning Research*, 14(40):1303–1347, 2013. URL https://jmlr.org/papers/v14/hoffman13a.html.

Kurt Hornik, Maxwell Stinchcombe, and Halbert White. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2(5):359–366, 1989.

Rob J. Hyndman and George Athanasopoulos. *Forecasting: Principles and Practice*. OTexts, Melbourne, Australia, 3rd edition, 2021. URL https://OTexts.com/fpp3.

Independent Electricity System Operator. Data Directory, 2026. URL https://www.ieso.ca/en/power-data/data-directory. Accessed May 14, 2026.

Deniz Kucukahmetler, Maximilian Jean Hemmann, Julian Mosig von Aehrenfeld, Maximilian Amthor, Christian Deubel, Nico Scherf, and Diaaeldin Taha. Relative geometry of neural forecasters: Linking accuracy and alignment in learned latent geometry. *Transactions on Machine Learning Research*, 2026. URL https://openreview.net/forum?id=t4stf5Gafz.

Yangyang Kui and Qiang Lai. Deep learning methods for chaotic time series prediction. *Knowledge-Based Systems*, 329:114441, 2025. ISSN 0950-7051. doi: https://doi.org/10.1016/j.knosys.2025.114441. URL https://www.sciencedirect.com/science/article/pii/S0950705125014807.

Henning Lange, Steven L. Brunton, and J. Nathan Kutz. From fourier to koopman: Spectral methods for long-term time series prediction. *Journal of Machine Learning Research*, 22(41):1–38, 2021. URL https://jmlr.org/papers/v22/20-406.html.

Subhash Lele. Estimating functions in chaotic systems. *Journal of the American Statistical Association*, 89 (426):512–516, 1994. doi: 10.1080/01621459.1994.10476769.

Bryan Lim, Sercan O. Arik, Nicolas Loeff, and Tomas Pfister. Temporal fusion transformers for interpretable multi-horizon time series forecasting. *International Journal of Forecasting*, 37: 1748–1764, 2021. doi: 10.1016/j.ijforecast.2021.03.012. URL https://research.google/pubs/temporal-fusion-transformers-for-interpretable-multi-horizon-time-series-forecasting/.

Greta M. Ljung and George E. P. Box. On a measure of lack of fit in time series models. *Biometrika*, 65(2): 297–303, 1978.

George G. Lorentz. *Bernstein Polynomials*. Chelsea Publishing Company, 1986.

Sayan Mukherjee, Edgar Osuna, and Federico Girosi. Nonlinear prediction of chaotic time series using support vector machines. In *Proceedings of the IEEE Workshop on Neural Networks for Signal Processing*, pp. 511–520. IEEE, 1997. doi: 10.1109/NNSP.1997.622408.

Boris N. Oreshkin, Dmitri Carpov, Nicolas Chapados, and Yoshua Bengio. N-beats: Neural basis expansion analysis for interpretable time series forecasting. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=Keah34oiobzl.

Lorenzo Pacchiardi, Rilwan A. Adewoyin, Peter Dueben, and Ritabrata Dutta. Probabilistic forecasting with generative networks via scoring rule minimization. *Journal of Machine Learning Research*, 25(45): 1–64, 2024. URL https://www.jmlr.org/papers/v25/23-0038.html.

Li Pan and Dimitris N. Politis. Bootstrap prediction intervals for linear, nonlinear and nonparametric autoregressions. *Journal of Statistical Planning and Inference*, 177:1–27, 2016. ISSN 0378-3758. doi: https://doi.org/10.1016/j.jspi.2014.10.003. URL https://www.sciencedirect.com/science/article/pii/S037837581400175X.

David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. Learning representations by back-propagating errors. *Nature*, 323:533–536, 1986. doi: 10.1038/323533a0.

David Salinas, Valentin Flunkert, Jan Gasthaus, and Tim Januschowski. Deepar: Probabilistic forecasting with autoregressive recurrent networks, 2017. URL https://arxiv.org/abs/1704.04110.

Harman S. Sandhu, Liping Fang, and Ling Guan. Forecasting day-ahead price spikes for the ontario electricity market. *Electric Power Systems Research*, 141:450–459, 2016. doi: 10.1016/j.epsr.2016.08.005.

Robert H. Shumway and David S. Stoffer. *Time Series Analysis and Its Applications: With R Examples*. Springer, Cham, Switzerland, 4 edition, 2017. ISBN 978-3-319-52451-1. doi: 10.1007/978-3-319-52452-8.

N. Stamatis, D. Parthimos, and T. M. Griffith. Forecasting chaotic cardiovascular time series with an adaptive slope multilayer perceptron neural network. *IEEE Transactions on Biomedical Engineering*, 46 (12):1441–1453, 1999. doi: 10.1109/10.804572.

George C. Tiao and Ruey S. Tsay. Some advances in non-linear and adaptive modelling in time-series. *Journal of Forecasting*, 13(2):109–131, 1994. doi: https://doi.org/10.1002/for.3980130206. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/for.3980130206.

Haixu Wu, Jiehui Xu, Jianmin Wang, and Mingsheng Long. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. In *Advances in Neural Information Processing Systems*, volume 34, pp. 22419–22430, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/bcc0d400288793e8bdcd7c19a8ac0c2b-Abstract.html.

Kejin Wu and Dimitris N. Politis. Bootstrap prediction inference of nonlinear autoregressive models. *Journal of Time Series Analysis*, 45(5):800–822, 2024. doi: 10.1111/jtsa.12739.

Jie Yang, Rui Yan, Mingyue Nong, Jiaqiang Liao, Feipeng Li, and Wei Sun. PM2.5 concentrations forecasting in beijing through deep learning with different inputs, model structures and forecast time. *Atmospheric Pollution Research*, 12(9):101168, 2021. doi: 10.1016/j.apr.2021.101168.

Lingxiao Zhao, Zhiyang Li, and Leilei Qu. Forecasting of beijing PM2.5 with a hybrid ARIMA model based on integrated AIC and improved GS fixed-order methods and seasonal decomposition. *Heliyon*, 8(12): e12239, 2022. doi: 10.1016/j.heliyon.2022.e12239.

Table 1: Held-out test set prediction comparison for Models A–C. LB refers to the residual Ljung–Box test p-value.

Model	Method	Test RMSE	Test MAE	LB
A	AR	0.616	0.488	0.01
A	MLP	0.637	0.512	0.00
A	BPAR	0.593	0.472	0.02
A	AR-BPAR	0.591	0.464	0.02
B	AR	0.220	0.183	0.31
B	MLP	0.219	0.184	0.32
B	BPAR	0.212	0.173	0.21
B	AR-BPAR	0.212	0.174	0.22
C	AR	0.575	0.474	0.90
C	MLP	0.485	0.389	0.63
C	BPAR	0.478	0.380	0.61
C	AR-BPAR	0.478	0.380	0.61

Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wancai Zhang. Informer: Beyond efficient transformer for long sequence time-series forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 11106–11115, 2021. doi: 10.1609/aaai.v35i12.17325. URL `https://aaai.org/papers/11106-informer-beyond-efficient-transformer-for-long-sequence-time-series-forecasting/`.

A Complete empirical study results

This appendix collects the dataset-specific details omitted from the main text. The results are organized in the same order for the three empirical blocks: a held-out test prediction table, a compact held-out test prediction figure with posterior prediction intervals where available, and then the detailed train/validation/test displays and comparison tables. The purpose is to preserve the complete experimental record while keeping the main paper focused on held-out test-set comparisons.

A.1 Nonlinear simulation models

The simulation formulas are given in Section 3; this appendix keeps the supporting numerical results grouped with the corresponding prediction displays.

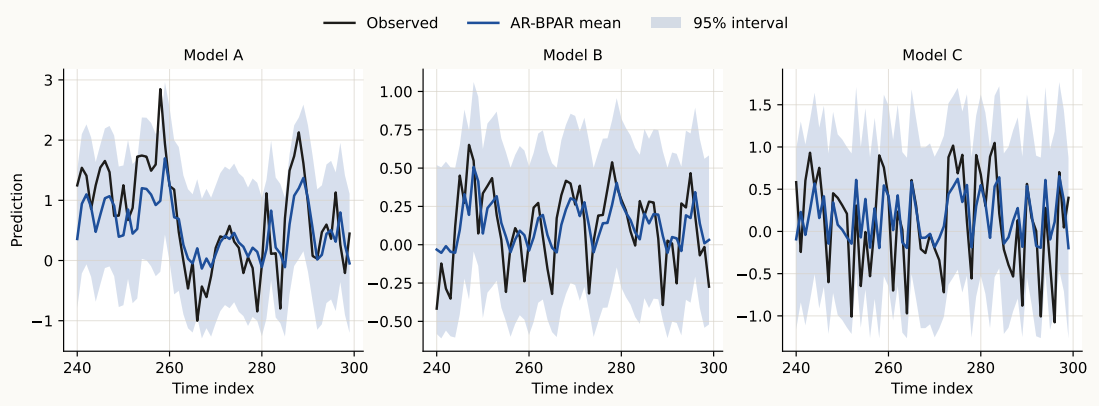

Figure 7: Held-out test set predictions for Models A–C using the selected BPAR/AR-BPAR specification. Panels are ordered left-to-right as Models A, B, and C. Each panel shows the observed test series, the BPAR/AR-BPAR posterior mean, and the 95% posterior prediction interval.

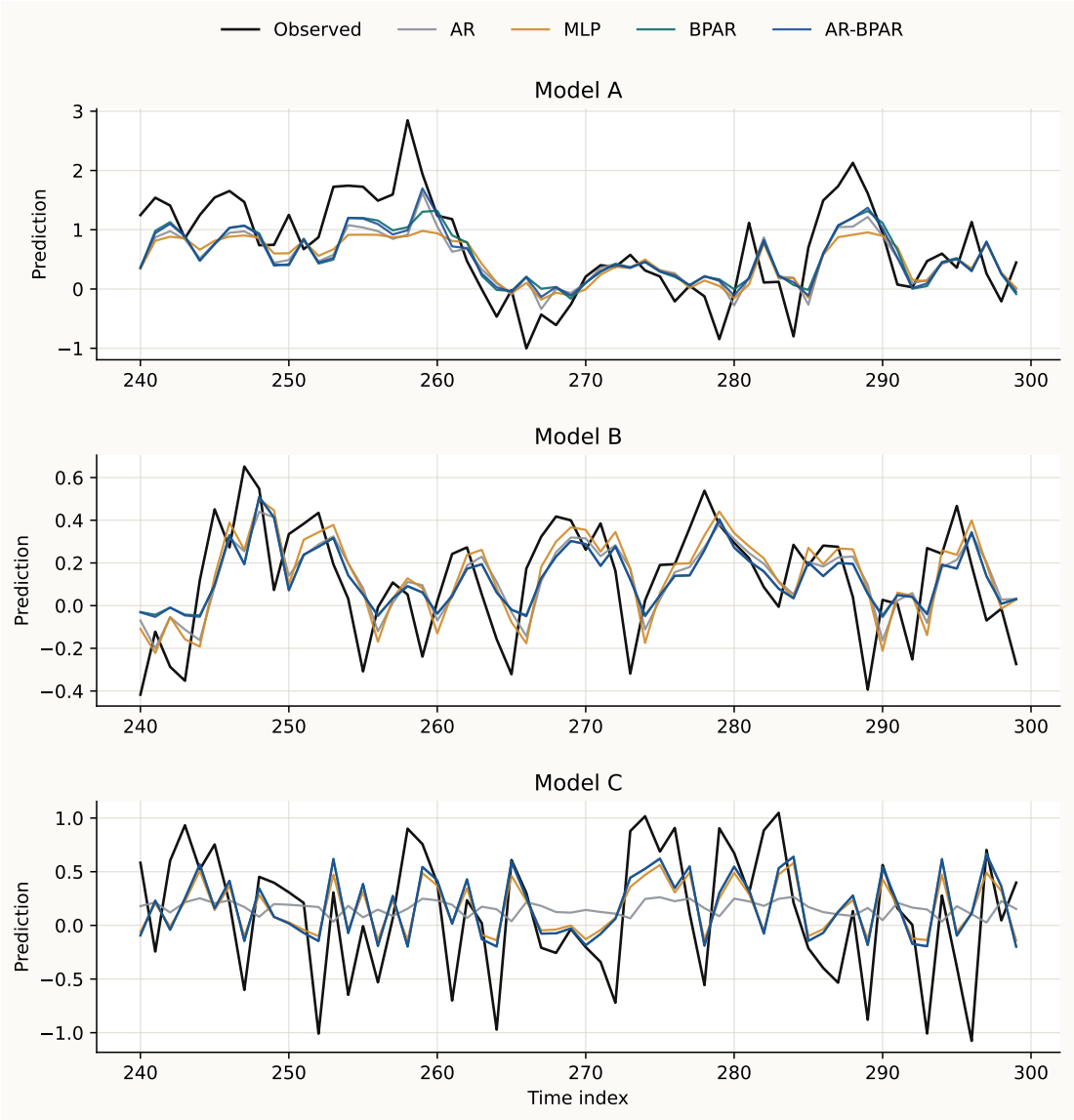

Figure 8: Held-out test set prediction comparison for Models A–C.

A.2 Chaotic time series benchmarks

Table 2: Held-out test set prediction comparison for the chaotic time series benchmarks. LB refers to the residual Ljung–Box test p-value.

Benchmark	Method	Test RMSE	Test MAE	LB
Ricker	ARIMA	2.14	1.67	0.03
Ricker	SARIMA	2.23	1.79	0.00
Ricker	MLP	0.40	0.22	0.06
Ricker	BPAR	0.30	0.20	0.22
Logistic	ARIMA	0.2438	0.2065	0.32
Logistic	SARIMA	0.2478	0.2152	0.09
Logistic	MLP	0.0141	0.0115	0.62
Logistic	LSTM	0.0174	0.0143	0.08
Logistic	BPAR	0.0115	0.0092	3.6×10^{-4}
Gauss	ARIMA	0.0840	0.0707	6.1×10^{-20}
Gauss	SARIMA	0.0486	0.0421	0.43
Gauss	MLP	0.0130	0.0089	0.03
Gauss	BPAR	0.0083	0.0066	2.2×10^{-5}
Mackey–Glass	AR-BPAR (Bayesian)	0.00162	0.00123	3.03×10^{-59}
Mackey–Glass	AR-BPAR (Neural)	0.00619	0.00455	2.09×10^{-66}
Mackey–Glass	ARIMA	0.03297	0.02781	5.08×10^{-201}
Mackey–Glass	SARIMA	0.00343	0.00280	2.85×10^{-82}
Mackey–Glass	MLP	0.01790	0.01422	7.18×10^{-96}
Mackey–Glass	LSTM	0.01811	0.01528	1.06×10^{-98}

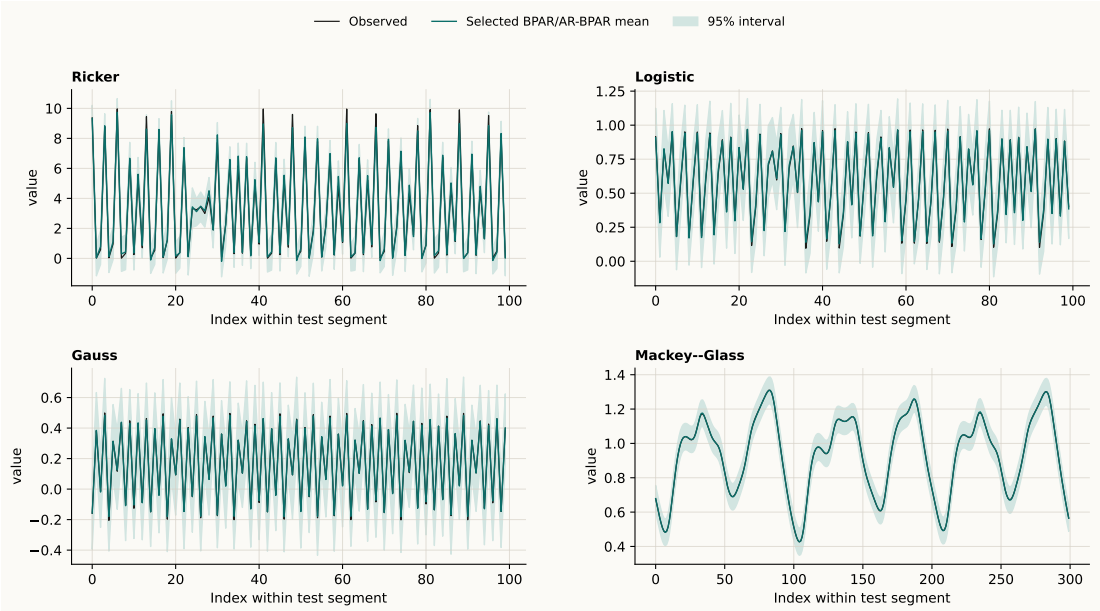

Figure 9: Held-out test set predictions for the chaotic time series benchmarks. Panels are ordered left-to-right, top-to-bottom as Ricker, Logistic, Gauss, and Mackey–Glass. Each panel shows the observed test set series, the selected BPAR/AR-BPAR posterior mean, and the 95% posterior prediction interval.

A.2.1 Ricker map

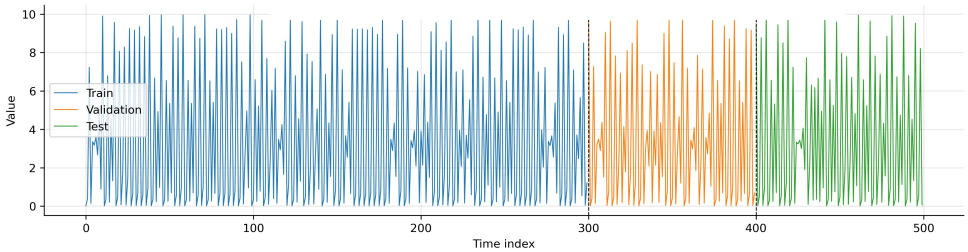

Figure 10: Ricker map time series with indications (vertical dashed lines) of the chronological train/validation/test split.

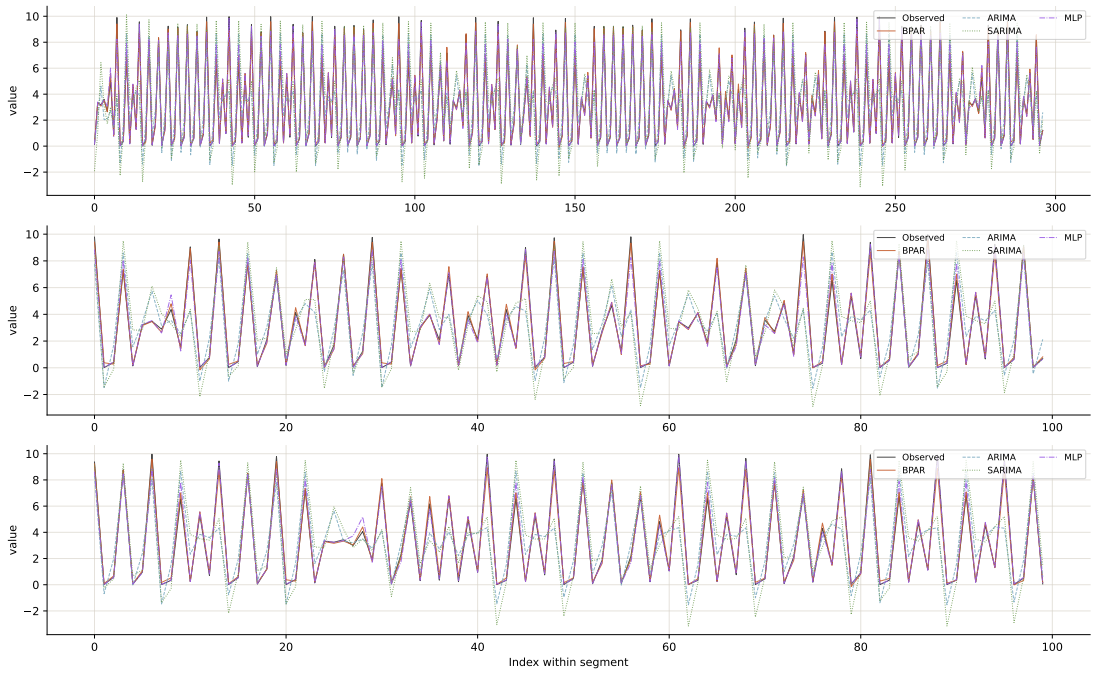

Figure 11: Train, validation, and test set predictions comparison for the Ricker map time series.

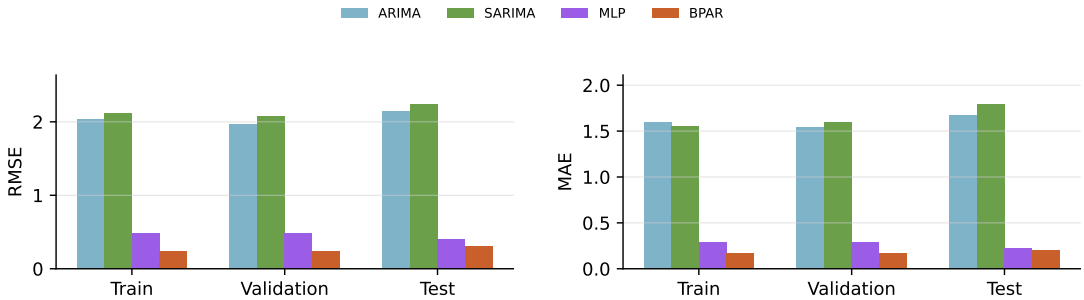

Figure 12: RMSE and MAE comparison across the train, validation, and test splits for the Ricker map time series.

Table 3: RMSE and MAE comparison for the Ricker map time series across train, validation, and test splits.

Split	ARIMA RMSE	SARIMA RMSE	MLP RMSE	BPAR RMSE	ARIMA MAE	SARIMA MAE	MLP MAE	BPAR MAE
Train	2.035675	2.112786	0.479972	0.233081	1.600501	1.549507	0.284966	0.164067
Validation	1.962163	2.069753	0.486570	0.231865	1.540349	1.598570	0.287815	0.173453
Test	2.137523	2.234414	0.403244	0.299752	1.667278	1.788471	0.223882	0.202838

A.2.2 Logistic and Gauss maps

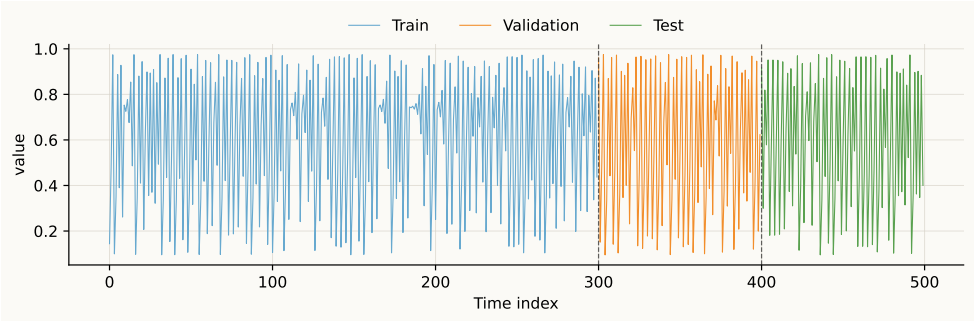

Figure 13: Logistic map time series with indications (vertical dashed lines) of the chronological train/validation/test split.

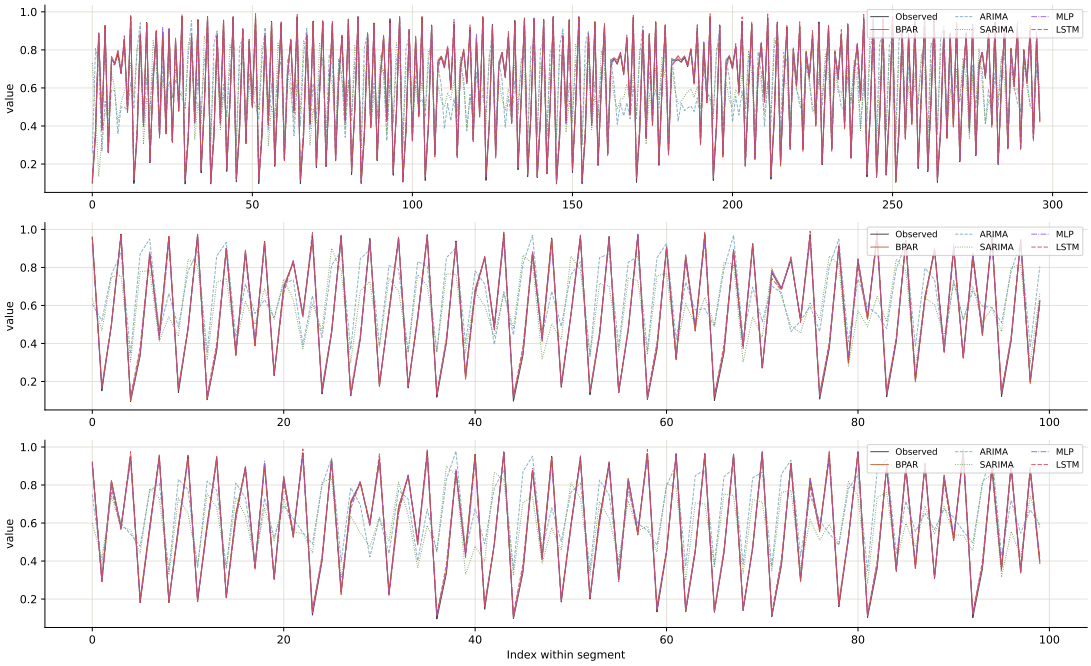

Figure 14: Train, validation, and test set predictions comparison for the Logistic map time series.

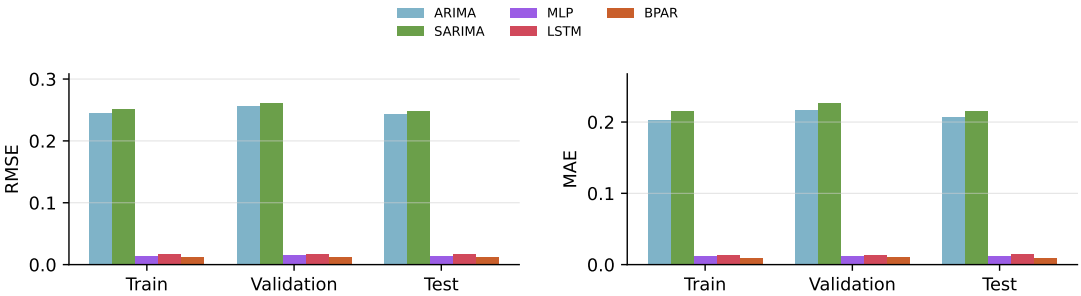

Figure 15: RMSE and MAE comparison across the train, validation, and test splits for the Logistic map benchmark.

Table 4: RMSE and MAE comparison for the Logistic map time series across train, validation, and test splits.

Split	ARIMA RMSE	SARIMA RMSE	MLP RMSE	LSTM RMSE	BPAR RMSE	ARIMA MAE	SARIMA MAE	MLP MAE	LSTM MAE	BPAR MAE
Train	0.244600	0.251541	0.014209	0.017371	0.011690	0.202602	0.215852	0.011462	0.013876	0.009486
Val	0.256764	0.261324	0.014816	0.017327	0.012412	0.216424	0.226765	0.011868	0.013984	0.010059
Test	0.243788	0.247817	0.014067	0.017386	0.011470	0.206514	0.215189	0.011513	0.014250	0.009212

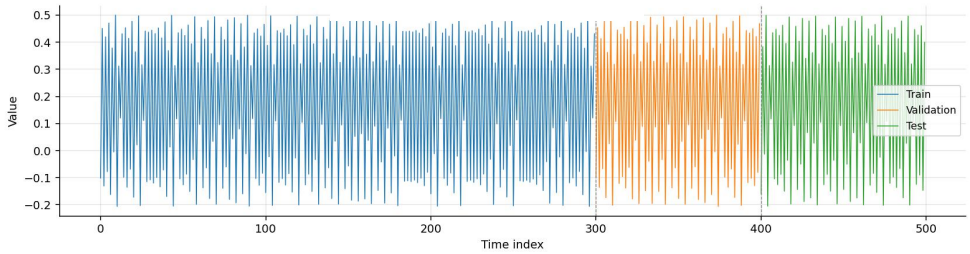

Figure 16: Gauss map time series with indications (vertical dashed lines) of the chronological train/validation/test split.

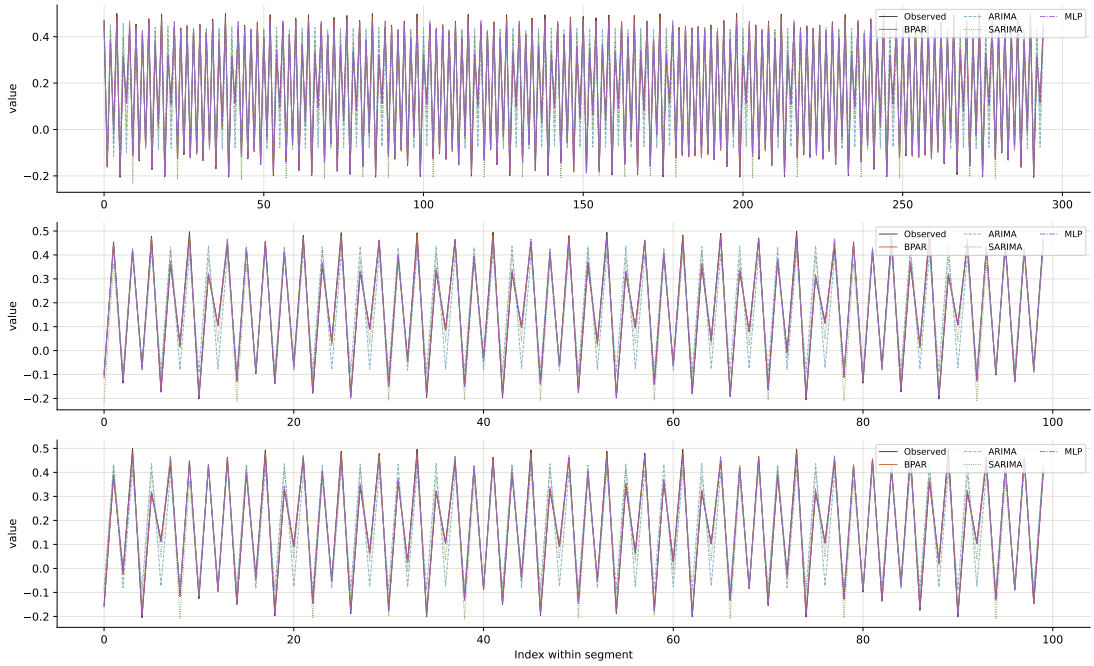

Figure 17: Train, validation, and test set predictions comparison for the Gauss map time series.

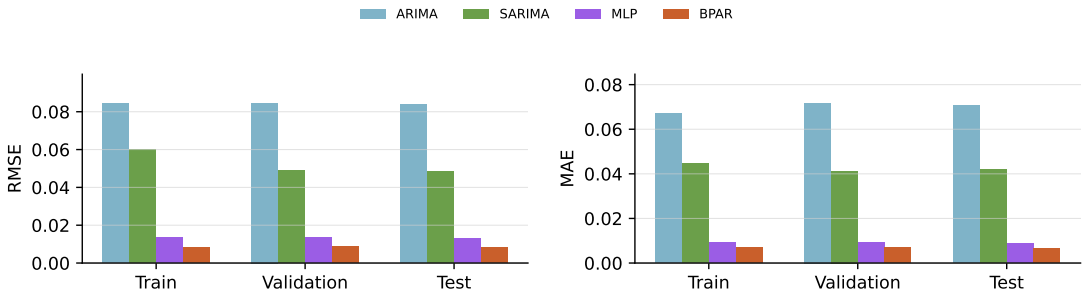

Figure 18: RMSE and MAE comparison across the train, validation, and test splits for the Gauss map benchmark.

Table 5: RMSE and MAE comparison for the Gauss map time series across train, validation, and test splits.

Split	ARIMA RMSE	SARIMA RMSE	MLP RMSE	BPAR RMSE	ARIMA MAE	SARIMA MAE	MLP MAE	BPAR MAE
Train	0.084599	0.060093	0.013300	0.008423	0.066850	0.044778	0.009104	0.006914
Val	0.084605	0.049018	0.013261	0.008704	0.071754	0.041287	0.009150	0.007036
Test	0.084016	0.048566	0.012987	0.008275	0.070664	0.042094	0.008936	0.006567

A.2.3 Mackey–Glass delay system

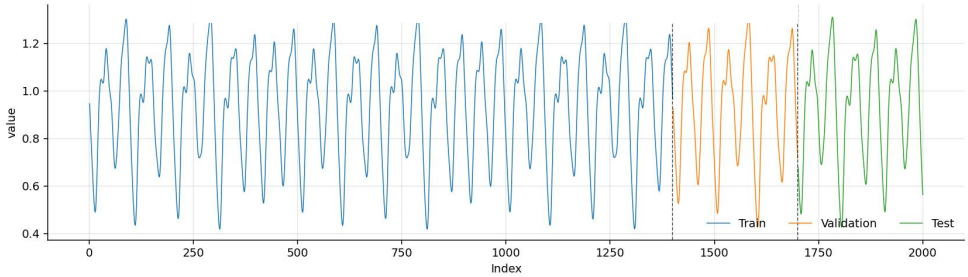

Figure 19: Mackey–Glass time series with indications (vertical dashed lines) of the chronological train/validation/test split.

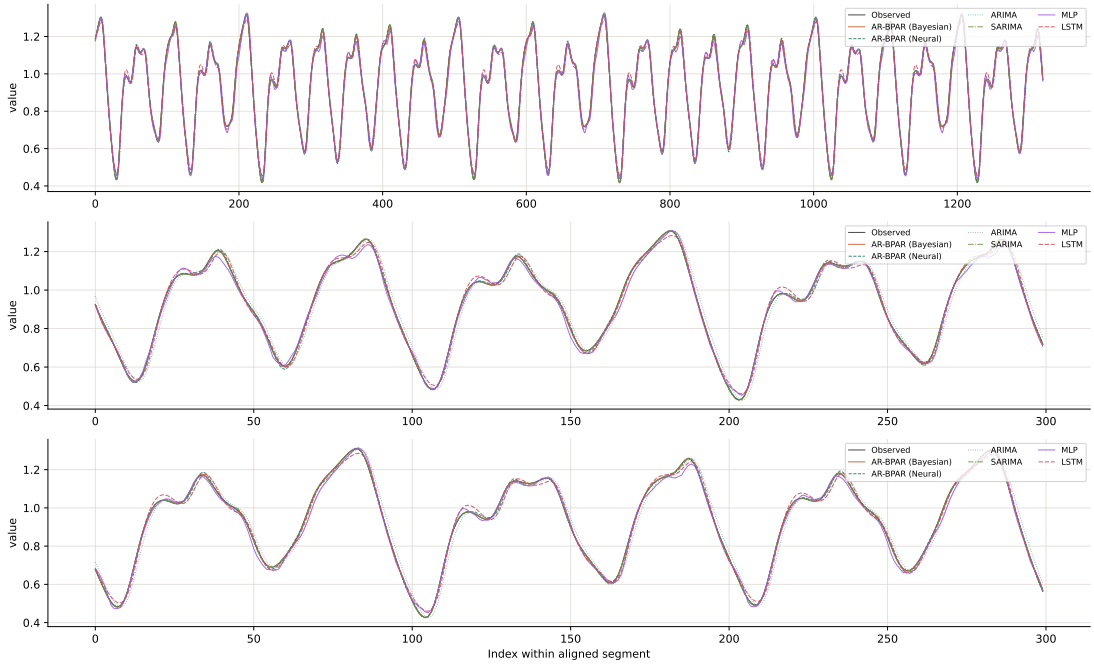

Figure 20: Train, validation, and test set predictions comparison for the Mackey-Glass time series.

Table 6: RMSE and MAE comparison for the Mackey-Glass time series across train, validation, and test splits.

Split	Bayes AR-BPAR RMSE	Neural AR-BPAR RMSE	ARIMA RMSE	SARIMA RMSE	MLP RMSE	LSTM RMSE	Bayes AR-BPAR MAE	Neural AR-BPAR MAE	ARIMA MAE	SARIMA MAE	MLP MAE	LSTM MAE
Train	0.00173	0.00654	0.04157	0.04950	0.01894	0.01868	0.00130	0.00478	0.02818	0.00557	0.01472	0.01570
Validation	0.00164	0.00611	0.03314	0.00336	0.01808	0.01794	0.00126	0.00459	0.02817	0.00273	0.01438	0.01517
Test	0.00162	0.00619	0.03297	0.00343	0.01790	0.01811	0.00123	0.00455	0.02781	0.00280	0.01422	0.01528

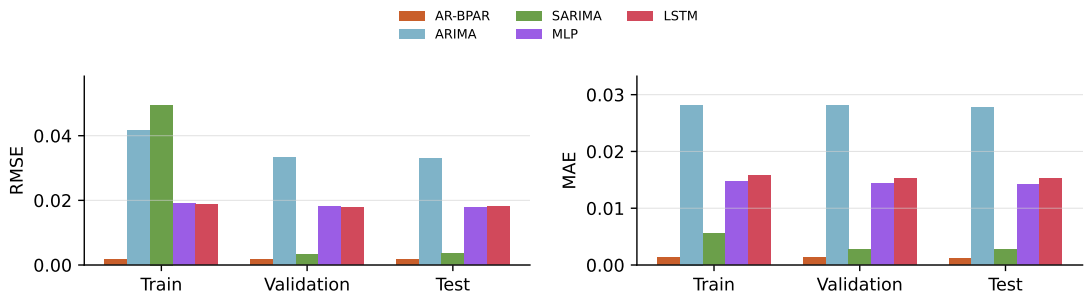

Figure 21: RMSE and MAE comparison across the train, validation, and test splits for the Mackey–Glass benchmark.

A.3 Real data time series

Table 7: Held-out test set prediction comparison for the real data time series. LB refers to the residual Ljung–Box test p-value.

Dataset	Method	Test RMSE	Test MAE	Test LB p
Ontario electricity	AR-BPAR (Bayesian)	9.78	6.27	0.39
Ontario electricity	AR-BPAR (Neural)	9.57	5.96	0.77
Ontario electricity	ARIMA	9.58	5.51	0.79
Ontario electricity	SARIMA	9.77	6.11	0.23
Ontario electricity	MLP	10.81	7.19	0.98
Ontario electricity	LSTM	9.47	6.22	0.92
Beijing $PM_{2.5}$	ARIMA	37.00	19.13	0.26
Beijing $PM_{2.5}$	SARIMA	37.01	19.35	0.31
Beijing $PM_{2.5}$	MLP	36.41	18.97	0.34
Beijing $PM_{2.5}$	LSTM	36.19	18.59	0.61
Beijing $PM_{2.5}$	AR-BPAR (Bayesian)	36.13	18.69	0.59

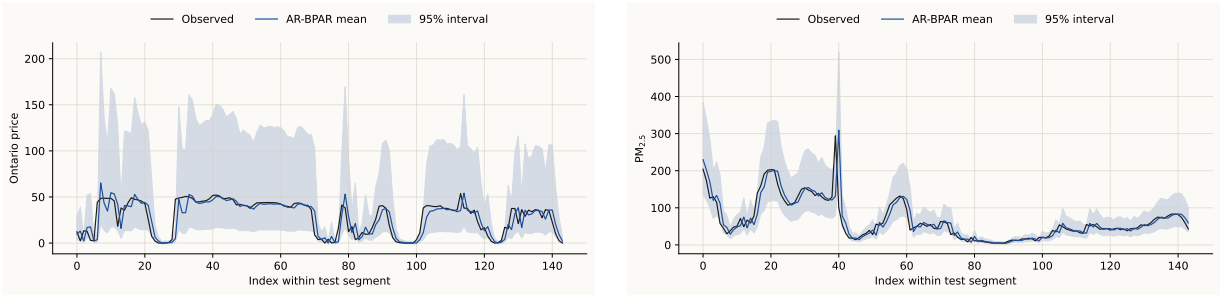

Figure 22: Held-out test set predictions along with their posterior prediction intervals. The left panel is Ontario electricity prices, and the right panel is Beijing $PM_{2.5}$.

A.3.1 Ontario day-ahead electricity prices

Table 8: Prediction comparison for the Ontario electricity price time series across train, validation, and test splits.

Split	Bayes AR-BPAR RMSE	Bayes AR-BPAR MAE	Neural AR-BPAR RMSE	Neural AR-BPAR MAE	ARIMA RMSE	ARIMA MAE	SARIMA RMSE	SARIMA MAE	MLP RMSE	MLP MAE	LSTM RMSE	LSTM MAE
Train	38.76	10.16	39.91	9.85	38.98	9.73	42.36	10.38	37.38	9.88	39.45	10.14
Validation	14.48	8.74	14.15	8.51	14.47	8.61	14.40	8.78	14.00	9.02	13.86	8.56
Test	9.78	6.27	9.57	5.96	9.58	5.51	9.77	6.11	10.81	7.19	9.47	6.22

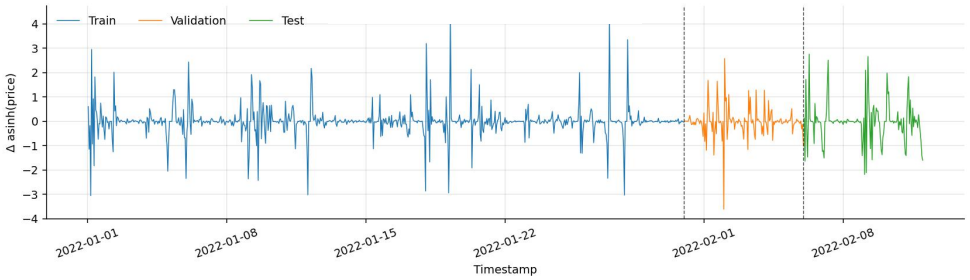

Figure 23: Ontario electricity price time series with indications (vertical dashed lines) of train/validation/test splits.

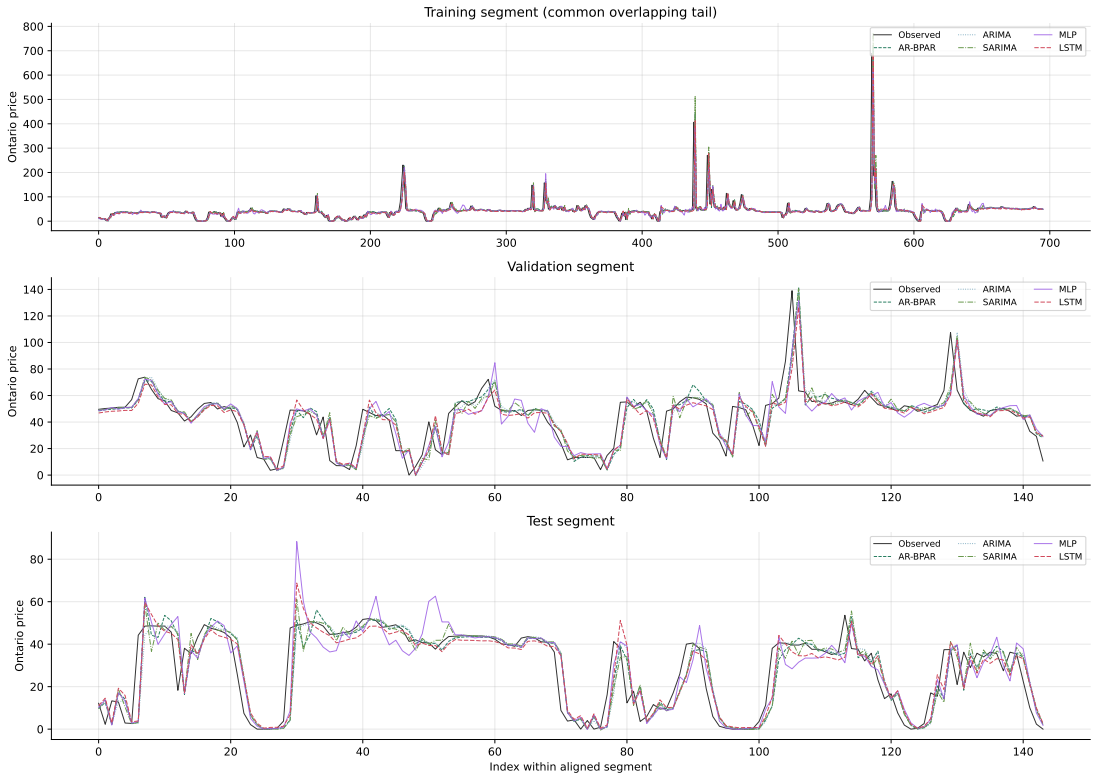

Figure 24: Train, validation, and test set predictions comparison for the Ontario electricity price time series.

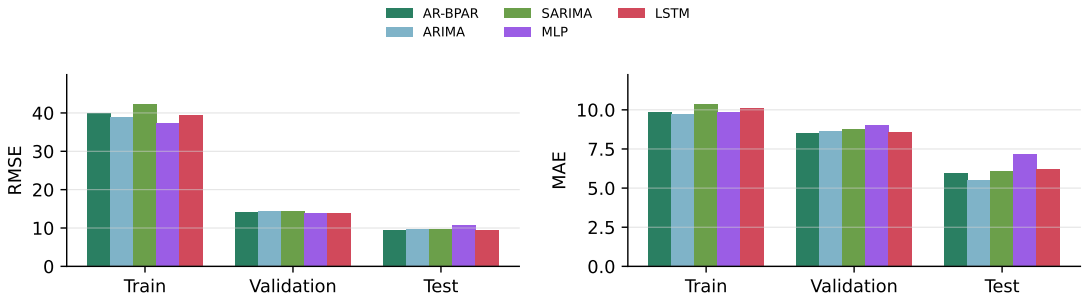

Figure 25: RMSE and MAE comparison across the train, validation, and test splits for the Ontario electricity price time series.

A.3.2 Beijing $PM_{2.5}$

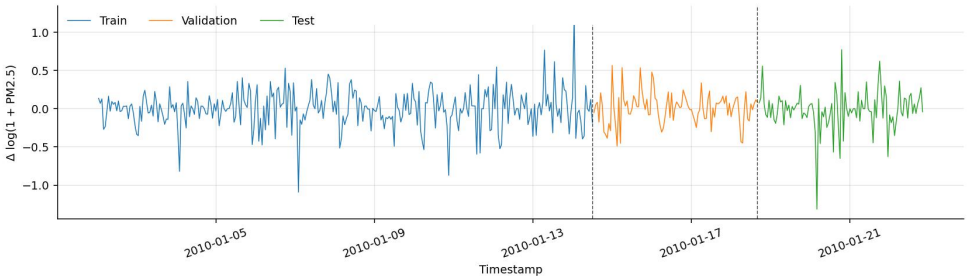

Figure 26: $PM_{2.5}$ time series with indications (vertical dashed lines) of train/validation/test splits.

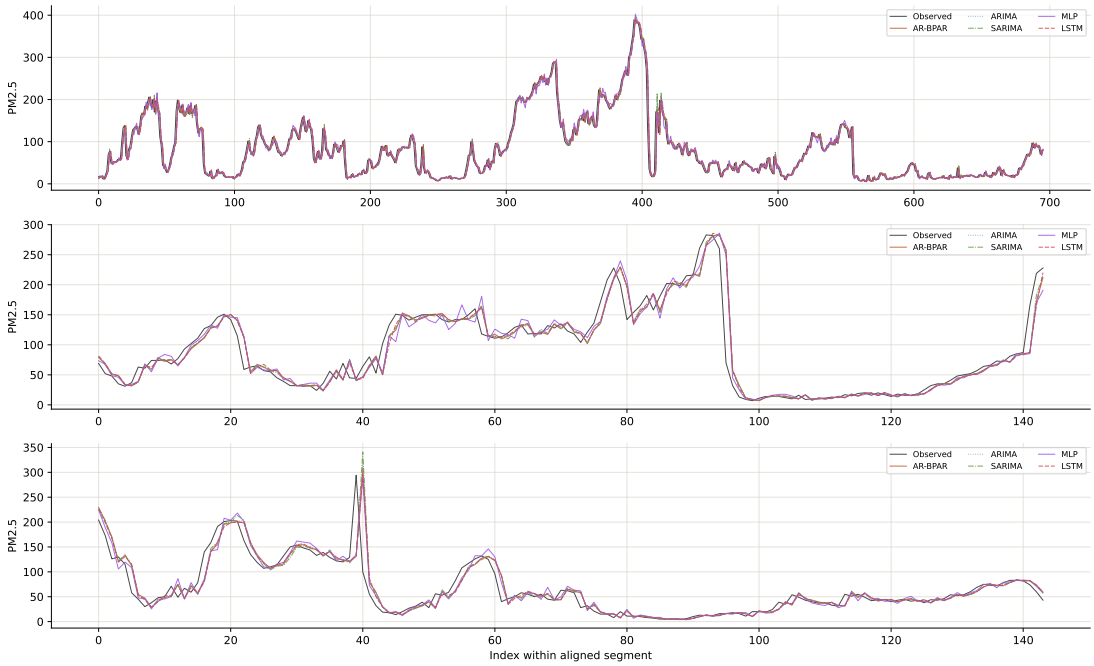

Figure 27: Train, validation, and test set predictions comparison for the $PM_{2.5}$ time series.

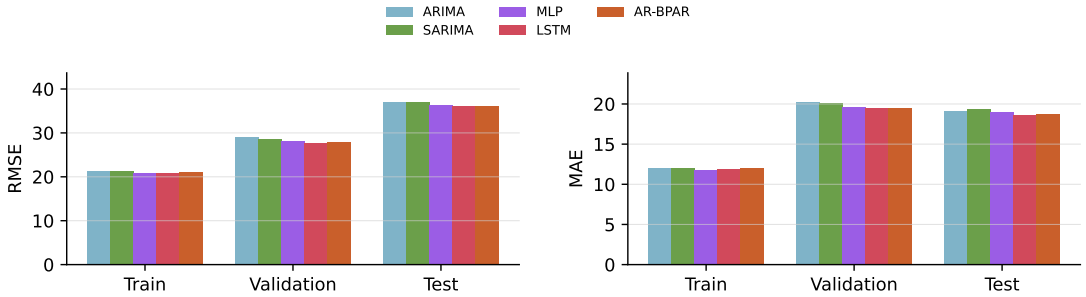

Figure 28: RMSE and MAE comparison across the train, validation, and test splits for the log-differenced $PM_{2.5}$ time series.

Table 9: Prediction comparison for the Ontario electricity price time series across train, validation, and test splits.

Split	ARIMA RMSE	SARIMA RMSE	MLP RMSE	LSTM RMSE	AR-BPAR RMSE	ARIMA MAE	SARIMA MAE	MLP MAE	LSTM MAE	AR-BPAR MAE
Train	21.40	21.36	20.80	20.79	21.06	12.02	11.98	11.77	11.90	12.02
Validation	29.03	28.66	28.23	27.72	27.85	20.23	20.05	19.59	19.46	19.46
Test	37.00	37.01	36.41	36.19	36.13	19.13	19.35	18.97	18.59	18.69