CROSS-VALIDATION TECHNIQUES FOR AUTO-REGRESSIVE TIME SERIES

XUAN LI

Abstract. This report is based on the paper "A Note on the Validity of Cross-Validation for Evaluating Autoregressive Time Series Prediction" (Bergmeir et al., 2018). There are three sections in this report. The first section provides a comprehensive summary of the paper (Bergmeir et al., 2018), presenting the key findings and some general perspectives toward understanding the work. The second section presents two possible research extension proposals beyond the original paper. The final section includes an extensional study we explore. In the study, we aim to mimic similar experiments using autoregressive models with an added lasso regression term, and we compare different cross-validation methods (including a blocked CV method) with out-of-sample estimation in terms of estimating the prediction error. Additionally, the proof of Theorem 1 in the original paper in this extensional scenario will be discussed from a general perspective.

Contents

Date: September 24, 2024.

1. Summary of the Original Paper

The original paper demonstrates the validity of using $K-fold$ cross-validation (CV) methods for evaluating auto-regressive time series models (Bergmeir et al., 2018). Cross-validation is a widely used method in machine learning regression and classification problems (Hastie et al., 2009). However, for time series data, CV is often considered problematic due to the serial correlation and non-stationarity in the dependent data setting (Bergmeir & Benitez, 2012). The authors demonstrated that cross-validation methods can be valid and effective for auto-regressive models under certain conditions (Bergmeir et al., 2018). The results are shown through both the theoretical proof and practical simulations.

The paper builds on the existing literatures on for time series data model fitting techniques. Traditionally, out-of-sample (OOS) evaluation is preferred for time series due to concerns about using the future data to predict the past data, and issues with correlation in errors (Bergmeir et al., 2018). In OOS, a block of data at the end of the set is left out as the testing data, and the remaining data is used as the training data. There are many studies that demonstrate this procedure (Tashman, 2000). Previous studies have also proposed different modifications to CV to adapt for the dependent data setting (Burman & Nolan, 1992; Burman et al., 1994; Gyorfi et al., 1989), such as $h-block$ CV (Burman et al., 1994). In $h-block$ CV, h data points preceding and following the testing data are left out in the training set due to avoiding dependency. The authors of the original paper (Bergmeir et al., 2018) referred to this type of CV as non-dependent cross-validation $nonDepCV$. However, this procedure often leads to inefficient use of data. The authors present their work as a shift from the conventional thinking by demonstrating that the standard $K-fold$ CV is not only valid but also helpful for auto-regressive models under certain conditions. In this case, the $K-fold$ CV can be used without modification, i.e., future data can be used as training data and past data can be used as testing data during this procedure. This is a key contribution of the paper. It challenges the conventional thinking that CV is not appropriate for time series data.

The main idea is that the error result we get by performing cross validation methods on a data set $\{y_t\}_{t=1}^{T}$, denoted as $\hat{PE}$, can approximate the prediction error when forecasting on a future data set $\{y_t\}_{T+1}^{n}$, denoted as PE.

1.1. **Theoretical Work.**

The original paper provides a theoretical proof by showing that the prediction error on the in-set data performed by CV is a consistent estimator

for the prediction error on the out-set data (the unseen future data). Without loss of generality, the paper's theoretical proof focuses on the leave-one-out CV ($LOOCV$) scene since generalisation to the $K-fold$ CV is straightforward.

Let $y_1, y_2, ..., y_n$ be a set of observation data from a stationary process. Let's consider a purely auto-regressive model of order P, i.e., AR(p), consider the following nonlinear regression model

$$y_t = g(x_t, \theta) + \epsilon_t \tag{1.1}$$

where ϵ_t is the regression error term, x_t consists the lagged values of y_t and θ is the parameter vector with all the coefficients values, and g is the function of the lagged values of y_t up to p^{th} order. g is a continuous and differentiable function with respect to θ for all $x_t = (y_{t-1}, y_{t-2}, ..., y_{t-p})'$. The estimation of $\hat{\theta}$ is the argument that minimises the objective function

$$Q(\theta) = \sum_{t=p+1}^{n} (y_t - g(x_t, \theta))^2 \tag{1.2}$$

Now suppose that $\{\tilde{y}_t\}_{t=1}^m$ is another set of observations that has the same distribution as $\{y_t\}_{t=1}^n$, which can be the future data

$$\tilde{y}_t = g(\tilde{x}_t, \theta) + \tilde{\epsilon}_t \tag{1.3}$$

The prediction error is defined as

$$PE = E(\tilde{y} - g(\tilde{x}, \hat{\theta})^2 \tag{1.4}$$

where $\hat{\theta}$ here is the estimate minimizing the objective function $\tilde{Q}(\theta) = \sum_{t=p+1}^{m} (\tilde{y}_t - g(\tilde{x}_t, \theta))^2$. In the paper, the authors consider estimating PE on dataset $\{\tilde{y}_t\}_{t=1}^m$ by performing cross-validation on $\{y_t\}_{t=1}^n$. In the $LOOCV$ scene, the training sample is $\{(x_j, y_j); j = p+1, ..., n, j \neq t\}$ and the test sample is $\{(x_t, y_t)\}$. The estimator of PE (denoted by $\hat{PE}$) is defined as

$$\hat{PE} = \frac{1}{n-p} \sum_{t=p+1}^{n} \left(y_t - g(x_t, \hat{\theta_{-t}}) \right)^2 \tag{1.5}$$

where $\hat{\theta_{-t}}$ is the leave-one-out estimate for θ on $\{(x_j, y_j); j = p+1, ..., n, j \neq t\}$. In order to prove that $\hat{PE}$ approximates PE, the following assumptions are needed.

Assumption 1. The nonlinear AR(p) process that generated $\{y_t\}_{t=1}^n$ is stationary and ergotic.

Assumption 2. $\hat{\theta_{-t}}$ is a consistent estimator of θ.

Assumption 3. The errors are MDS:

(1) $\{\epsilon_t, F_t\}$ form a sequence of martingale differences (MDS) where F_t is the sigma field generated by $\{\epsilon_s, y_s; s \leq t\}$. Note that i.i.d errors are MDS.
(2) $Var(\epsilon_t|x_t) = \sigma^2$ and $E[|\epsilon_t|^{4+\delta} < K$ for some $K < \infty$ and $\delta > 0$.
(3) $\{\epsilon_t\}$ have absolutely continuous distribution with respect to Lebesgue measure.

Note that due to stationarity of $\{y_t\}_{t=1}^n$, conditions for the consistency of $\hat{\theta}$ and $\hat{\theta_{-t}}$ are equivalent. In short, we want to make sure that the time series model is stationary, the estimator is consistent, and the errors are uncorrelated.

The theoretical validity is carried out by proving the following theorem. Compared to the original paper, we have slightly altered the notation for better clarification in the proof, and we also restructured the proof by introducing some additional notation, which makes the logic easier to follow.

Theorem 1. *Suppose that Assumptions 1-3 hold, then we have* $\hat{PE} \xrightarrow{approx.} PE$.

Proof. By expanding the prediction error into integral form, we have

$$PE = \int (\tilde{y} - g(\tilde{x}, \hat{\theta}))^2 d\tilde{F_m}$$

where $\tilde{F_m}$ is the distribution of the process generating $\{\tilde{y_t}\}_{t=1}^m$. Suppose the true model is $\tilde{y} = g(\tilde{x}, \theta) + \tilde{\epsilon}$, where $g(\tilde{x}, \theta)$ represents the true underlying relationship between $\tilde{x}$ and $\tilde{y}$, and $\tilde{\epsilon}$ is the noise. By substituting for $\tilde{y}$, we have

$$PE = \int \left(g(\tilde{x}, \theta) + \tilde{\epsilon} - g(\tilde{x}, \hat{\theta})\right)^2 d\tilde{F_m}$$

By expanding the squared terms and simplification, we have

$$PE = \int \left[(g(\tilde{x}, \theta) - g(\tilde{x}, \hat{\theta}))^2 + \tilde{\epsilon}^2\right] d\tilde{F_m}$$

The authors performed a bias-variance decomposition by subtracting and adding the expected prediction $E[g(\tilde{x}, \hat{\theta})]$, which represents the average prediction over multiple realizations of the training data. Therefore, we have

$$PE = \int \left[\left(g(\tilde{x}, \theta) - E[g(\tilde{x}, \hat{\theta})] + E[g(\tilde{x}, \hat{\theta})] - g(\tilde{x}, \hat{\theta})\right)^2 + \tilde{\epsilon}^2\right] d\tilde{F_m}$$

By expanding the squared terms, we have

$$\begin{aligned} PE = \int \Bigg[&\left(g(\tilde{x}, \theta) - E[g(\tilde{x}, \hat{\theta})]\right)^2 + \left(E[g(\tilde{x}, \hat{\theta})] - g(\tilde{x}, \hat{\theta})\right)^2 \\ &+ 2\left(g(\tilde{x}, \theta) - E[g(\tilde{x}, \hat{\theta})]\right)\left(g(\tilde{x}, \hat{\theta}) - E[g(\tilde{x}, \hat{\theta})]\right) + \tilde{\epsilon}^2\Bigg] d\tilde{F_m} \end{aligned}$$

Notice that the cross-term vanishes when integrated since $E[g(\tilde{x},\hat{\theta}) - E[g(\tilde{x},\hat{\theta}))] = 0$, and hence the equation can be simplified to

$$\begin{aligned} PE &= \int \left[\Big(g(\tilde{x},\theta) - E[g(\tilde{x},\hat{\theta})] \Big)^2 + \Big(E[g(\tilde{x},\hat{\theta})] - g(\tilde{x},\theta) \Big)^2 + \tilde{\epsilon}^2 \right] d\tilde{F}_m \\ &= \underbrace{\int \Big(g(\tilde{x},\theta) - E[g(\tilde{x},\hat{\theta})] \Big)^2 d\tilde{F}_m}_{\text{Bias}} + \underbrace{\int \Big(E[g(\tilde{x},\hat{\theta})] - g(\tilde{x},\hat{\theta}) \Big)^2 d\tilde{F}_m}_{\text{Variance}} + \underbrace{\int \tilde{\epsilon}^2 d\tilde{F}_m}_{\text{noise}} \end{aligned}$$

We expand $\hat{PE}$ in a similar vein.

$$\hat{PE} = \underbrace{\int \Big(g(x,\theta) - E[g(x,\hat{\theta_{-t}})] \Big)^2 dF_n}_{\text{Bias}} + \underbrace{\frac{1}{n-p} \sum_{t=p+1}^{n} \Big(E[g(x_t,\hat{\theta_{-t}})] - g(x_t,\hat{\theta_{-t}}) \Big)^2}_{\text{Variance}} + \underbrace{\int \epsilon^2 dF_n}_{\text{noise}}$$

where F_n is the distribution of the process generating $\{y_t\}_{t=1}^n$. Notice that $\tilde{F_m}$ and F_n are the empirical distributions of 2 different samples from the same underlying process F, i.e $\{\tilde{y_t}\}_{t=1}^m$ and $\{y_t\}_{t=1}^n$ are 2 datasets from the same underlying distribution process. By **Assumption 1** and **Assumption 2**, the underlying distribution process is stationary and ergotic, and both $\hat{\theta}$ and $\hat{\theta_{-t}}$ are consistent estimators. The bias and noise terms in both $\hat{PE}$ and PE are asymptotically identical. As $n, m \to \infty$, we want to show that

(1.6)
$$\frac{1}{n-p} \sum_{t=p+1}^{n} \Big(E[g(x_t,\hat{\theta_{-t}})] - g(x_t,\hat{\theta_{-t}}) \Big)^2 \xrightarrow{Approxi.} \int \Big(E[g(\tilde{x},\hat{\theta})] - g(\tilde{x},\hat{\theta}) \Big)^2 d\tilde{F}_m$$

The key takeaway here is that, ultimately, the variance of the true prediction error denoted by PE is consistently estimated by the variance of the cross-validation prediction error $\hat{PE}$, when the number of data gets big. When n, m becomes big, let $N = max\{n, m\}$, and let $\epsilon_t^E(\hat{\theta_{-t}}) = E[g(x_t,\hat{\theta_{-t}})] - g(x_t,\hat{\theta_{-t}}), \tilde{\epsilon}_t^E(\hat{\theta}) = E[g(\tilde{x}_t,\hat{\theta})] - g(\tilde{x}_t,\hat{\theta})$, we can write the left term in equation 1.6 as

$$\frac{1}{N-p} \sum_{t=p+1}^{N} \big(\epsilon_t^E(\theta_{-t}) \big)^2$$

and similarly we can write the right term as

$$\frac{1}{(N-p)^2} \sum_{t=p+1}^{N} \sum_{j=p+1}^{N} \tilde{\epsilon}_t^E(\hat{\theta}) \tilde{\epsilon}_j^E(\hat{\theta})$$

Our proof is complete if we can show that, as N grows big

$$E\left[\sum_{t=p+1}^{N}\sum_{j=p+1}^{N}\tilde{\epsilon}_t^E(\hat{\theta})\tilde{\epsilon}_j^E(\hat{\theta})\right] \approx E\left[\sum_{t=p+1}^{N}\left(\epsilon_t^E(\theta_{-t})\right)^2\right]$$

which follows by Lemma 1. □

Lemma 1. Suppose that **Assumptions 1-3** hold. Then, there exists an arbitrarily small constant c >0 such that

$$E\left[\sum_{t=p+1}^{N}\sum_{j=p+1}^{N}\tilde{\epsilon}_t^E(\hat{\theta})\tilde{\epsilon}_j^E(\hat{\theta}) - \sum_{t=p+1}^{N}\left(\epsilon_t^E(\theta_{-t})\right)^2\right]^4 \leq cN^4$$

Proof. Note that

$$\begin{aligned}
&\sum_{t=p+1}^{N}\sum_{j=p+1}^{N}\tilde{\epsilon}_t^E(\hat{\theta})\tilde{\epsilon}_j^E(\hat{\theta}) - \sum_{t=p+1}^{N}\left(\epsilon_t^E(\theta_{-t})\right)^2 \\
&= \underbrace{\sum_{t=p+1}^{N}(\tilde{\epsilon}_t^E(\hat{\theta}))^2 - \sum_{t=p+1}^{N}(\epsilon_t^E(\hat{\theta_{-t}}))^2}_{\text{part 1}} + \underbrace{\sum_{t=p+1,t\neq j}^{N}\sum_{j=p+1}^{N}\tilde{\epsilon}_t^E(\hat{\theta})\tilde{\epsilon}_j^E(\hat{\theta})}_{\text{part 2}}
\end{aligned}$$

By adding and subtracting the expectation terms, we can express (part 1) of the above equation as

$$\begin{aligned}
\sum_{t=p+1}^{N}(\tilde{\epsilon}_t^E(\hat{\theta}))^2 - \sum_{t=p+1}^{N}(\epsilon_t^E(\hat{\theta_{-t}}))^2 = &\sum_{t=p+1}^{N}\left((\tilde{\epsilon}_t^E(\hat{\theta}))^2 - E[(\tilde{\epsilon}_t^E(\hat{\theta}))^2]\right) \\
&- \sum_{t=p+1}^{N}\left((\epsilon_t^E(\hat{\theta_{-t}}))^2 - E[(\epsilon_t^E(\hat{\theta_{-t}}))^2]\right) \\
&+ \sum_{t=p+1}^{N}\left(E[(\tilde{\epsilon}_t^E(\hat{\theta}))^2] - E[(\epsilon_t^E(\hat{\theta_{-t}}))^2]\right)
\end{aligned}$$

Notice that summations here are martingale difference sequences (by **Assumption 3**). Therefore by Burkholder's inequality, we have

$$E\left|\sum_{t=p+1}^{N}\left((\tilde{\epsilon}_t^E(\hat{\theta}))^2 - E[(\tilde{\epsilon}_t^E(\hat{\theta}))^2]\right)\right|^4 \leq cE\left|\sum_{t=p+1}^{N}\left((\tilde{\epsilon}_t^E(\hat{\theta}))^2 - E[(\tilde{\epsilon}_t^E(\hat{\theta}))^2]\right)^2\right|^2 \leq cN^2$$

$$E\left|\sum_{t=p+1}^{N}\left((\epsilon_t^E(\hat{\theta_{-t}}))^2 - E[(\epsilon_t^E(\hat{\theta_{-t}}))^2]\right)\right|^4 \leq cE\left|\sum_{t=p+1}^{N}\left((\epsilon_t^E(\hat{\theta_{-t}}))^2 - E[(\epsilon_t^E(\hat{\theta_{-t}}))^2]\right)^2\right|^2 \leq cN^2$$

$$E\left|\sum_{t=p+1}^{N}\left(E[(\tilde{\epsilon}_t^E(\hat{\theta}))^2]-E[(\epsilon_t^E(\hat{\theta_{-t}}))^2]\right)\right|^4 \leq cE\left|\sum_{t=p+1}^{N}\left(E[(\tilde{\epsilon}_t^E(\hat{\theta}))^2]-E[(\epsilon_t^E(\hat{\theta_{-t}}))^2]\right)^2\right|^2 \leq cN^2$$

For the (part 2) of the equation, due to the symmetry of a covariance matrix, we can express it as

$$\sum_{t=p+1,t\neq j}^{N}\sum_{j=p+1}^{N}\tilde{\epsilon}_t^E(\hat{\theta})\tilde{\epsilon}_j^E(\hat{\theta}) = 2\sum_{t=p+2}^{N}\tilde{\epsilon}_t^E(\hat{\theta})\sum_{j=p+1}^{j<t}\tilde{\epsilon}_j^E(\hat{\theta})$$

Notice that sum of the off-diagonal elements of the variance-covariance matrix becomes negligible (the impact of the covariance terms becomes negligible).

$$E\left[\sum_{t=p+1,t\neq j}^{N}\sum_{j=p+1}^{N}\tilde{\epsilon}_t^E(\hat{\theta})\tilde{\epsilon}_j^E(\hat{\theta})\right]^4 = 2E\left[\sum_{t=p+2}^{N}\tilde{\epsilon}_t^E(\hat{\theta})\sum_{j=p+1}^{j<t}\tilde{\epsilon}_j^E(\hat{\theta})\right]^4 \leq cN^4$$

Both $\{\tilde{\epsilon}_t^E\}$ and $\{\epsilon_t^E\}$ are stationary martingale difference sequences, and any linear combination of martingales based on the same filtration is also a martingale. After omitting certain rows and applying the law of iterated expectation, this inequality follows from the proof of Lemma 5.3 in (Burman & Nolan, 1992). And therefore, we have Lemma 1 proved. □

Note that **Errors are uncorrelated** will be the key component for the success of the cross-validation. In fact, as stated by the authors, the proposed CV method works exactly towards ensuring the uncorrelatedness between residuals. Alternatively, we can understand the proof from the perspective of the Glivenko-Cante's theorem in dependent setting. The theorem states that the empirical distribution function converges uniformly to the true distribution function under certain conditions (Bradley, 2005; Doukhan, 1994). In our case, the set of observations $\{y_t\}_{t=1}^n$ is generated from a stationary and ergotic process by **Assumption 1**. Due to stationarity, the dependence between observations decays exponentially (i.e the auto-correlations decay exponentially (Takemura, 2016) for stationary AR processes), and therefore indicates a α-mixing condition (Brockwell & Davis, 1991). This was also mentioned by the authors in the original paper. Therefore, the data process in this case satisfies all conditions needed to apply the dependent version of Glivenko-Cante's theorem. Similar logic follows for $\{\tilde{y}_t\}_{t=1}^m$. Therefore, both F_n and $\tilde{F_m}$ are empirical distribution functions of the observations $\{y_t\}_{t=1}^n$ and $\{\tilde{y}_t\}_{t=1}^m$ from the same underlying AR(p) process that satisfies the stationarity and mixing condition. By Glivenko-Cante's theorem, they converges uniformly to the true distribution function F as the sample size grows large.

1.2. **Generalization of Theorem 1.** Notice that in the proof, we did not rely on any specific properties of the AR(p) model itself. In **Assumption 1**, what truly matters are the stationarity and the alpha-mixing condition. Both of them can be extended to a wide range of time series models. **Assumption 2** ensures that we have consistent estimators for the coefficients terms. And **Assumption 3** (error terms form a martingale difference sequence) plays an important role in the proof as mentioned before. Both **Assumption 2 and 3** do not depend on any specific properties of the AR(p) model, and they can be extended to a wide range of time series models as well.

Therefore, since the proof for **Theorem 1** depends on these general conditions (not on any special properties of the AR(p) model), we can extend the **Theorem 1** to a more general setting. By generalizing **Assumption 1**, we can apply the results of the cross-validation (CV) procedure beyond just the auto-regressive (AR) models. This can include any nonlinear time series models that meet the adjusted **Assumption 1** (i.e., **Assumption 4**), **Assumption 2**, and **Assumption 3**.

Assumption 4. The generalized version of **Assumption 1** for any nonlinear time series process:

(1) **Stationarity**: The time series process should be stationary, so the statistical properties remain consistent over time.
(2) **Alpha-Mixing**: The process should satisfy the alpha-mixing condition, ensuring weak dependence between observations.
(3) **Correct Model Specification**: The model used for cross-validation must be correctly specified, meaning that it matches with the true data-generating process and therefore doesn't introduce bias due to misspecification.

The martingale difference property of the error term in Assumption 3 ensures that the errors are uncorrelated. It is key for the consistency and accuracy of the cross-validation error estimates. If the model is misspecified, it would cause correlations in the errors, which is why we account for this in the updated version of **Assumption 1**, i.e., **Assumption 4**.

In short, with Assumptions **2-4**, we can now apply **Theorem 1** to not only AR(p) models, but any nonlinear time series models.

1.3. **Monte Carlo Simulation and Results.** The authors demonstrate through R that $K - fold$ CV and $LOOCV$ outperform other approaches, using both general AR(p) and non-parametric models. The simulation contains 1000 Monte Carlo trials for three experiments: data-generating processes from AR(3), invertible MA(1), and

seasonal AR(12) (this is a counter-example where CV fails). In each Monte Carlo trial, the time series data has length 200, with 70% being used as in-set (for $\hat{PE}$) and 30% hidden as out-set (for PE).

The methods used and compared are $5-fold$ CV, $LOOCV$, $nonDepCV$, and OOS evaluation. For model fitting, the authors use linear AR models with up to 5 lags and a neural network (MLP) model with 5 hidden units. To see how $\hat{PE}$ approximates PE, the authors use **mean absolute prediction accuracy error** and **mean prediction accuracy error**, i.e., $MAPAE = \frac{1}{k}\sum_j^k |\hat{PE}_j - PE_j|, MPAE = \frac{1}{k}\sum_j^k(\hat{PE}_j - PE_j)$. **Root mean squared error** and **mean absolute error** are used to evaluate for the out-set error, i.e., $RMSE = \frac{1}{N-T}\sum_{t=T}^N \sqrt{(y_t - g(x_t, \hat{\theta})^2}, MAE = \frac{1}{N-T}\sum_{t=T}^N |y_t - g(x_t, \hat{\theta}|$. Similarly, for consistency, the in-set error measures also use **RMSE** and **MAE**.

The results show that $5-fold$ CV and $LOOCV$ outperform OOS and $nonDepCV$ in the first two experiments. For the last experiment with misspecified models, none of the methods performed well. More details can be found in the original paper.

1.4. **Example on R.** The paper also includes a real-world example to evaluate the performance of cross-validation techniques and OOS evaluation using the yearly sunspot series (289 observations from 1700 to 1988). The dataset can be downloaded on R. The authors made some transformation to the original data. They applied both $5-fold$ cross-validation and OOS evaluation. The results showed that the model selected using $5-fold$ CV had a more reasonable lag structure than OOS and outperformed the OOS model slightly in prediction error. Both methods performed similarly, but the CV model had a more balanced configuration. The findings demonstrate that cross-validation is effective in real-world scenarios for model selection and error estimation. More details can be found in the original paper.

1.5. **Assumption and limitation.** The main contribution of this paper is showing that cross-validation method works for auto-regressive time series models both theoretically and practically, as long as certain conditions are met (Bergmeir et al., 2018). In the following paragraph, we address some limitations of the paper despite its advancements.

- **Assumptions of Uncorrelated Errors**: The validity of using K-fold CV as proposed in the paper replies on the assumption that the errors in the auto-regressive model are uncorrelated. If this assumption is violated, for example, in the presence of model misspecification, cross-validation procedure would

fail. This limitation is acknowledged by the authors. This suggest to always check for the residuals for the serial correlation.

- **Applicability to Complex Models**: While the CV methods work well with simple models in the simulation results, the authors do not extend the experiment incorporating complex time series models, such as hybrid models. The practical validity for complex models remains unknown.
- **Computational Constraints**: Although the paper demonstrates that CV can be applied to auto-regressive models, it does not address the potential computational cost, especially when using CV with large K, on large datasets or with complex models. This remains a practical consideration for researchers and practitioners.
- **Applicability in real-world datasets**: CV performance in datasets with missing values or high noise are not being addressed here. This would help to determine which method is more stable and reliable under challenging conditions that are common in real-world datasets.

In conclusion, the paper makes contribution by challenging the conventional view that normal CV methods are not appropriate to use for time series data. It opens the door for broader use of general $K - fold$ CV in time series forecasting, while also highlighting the importance of carefully checking model assumptions.

2. Mini-proposals

2.1. **Proposal 1: Comparing normal-CV vs blocked-CV.** The first proposal is to compare the performance of normal CV with blocked CV (Bergmeir et al., 2014), especially when performed on time series data with seasonal trends. Normal CV distributes data points into different folds randomly, which potentially breaks the serial dependencies within the series. Blocked CV retains the sequential order of data within each block, which could preserve "some" patterns. For seasonal data, where cycles and temporal dependencies are key, blocked CV might be a more realistic method.

A good model to test the performance of blocked-CV versus normal-CV on seasonal time series data would be a seasonal auto-regressive (SAR) model. This is a simple extension of the basic auto-regressive (AR) model that accounts for seasonality and we can also assure stationarity and other assumptions needed. To evaluate whether blocked CV performs better, we can proceed with a similar experimental set-up as the paper discussed. And we can compare for different error measures as mentioned in the paper.

Moreover, we can investigate the effect of varying the block size in blocked CV. Smaller blocks may capture finer seasonal details, while larger blocks may better preserve long-term trends and dependencies. We can also explore the potential connection between the optimal block size and the series lag order. This would help optimizing the blocked CV method for different types of seasonal data. There are some other potential topics of using blocked-CV that can be investigated.

- Reduced Data Leakage: In time series analysis, training on data that is temporally after the testing data (as can happen in normal-CV) can lead to data leakage (Shao et al., 2019), where the model "sees the future." Blocked-CV, by keeping the temporal sequence in place within the blocks, might can reduce the future information leak into the training set than normal CV, and therefore provide a more realistic evaluation of how the model performs on unseen future data.
- More Realistic Testing for Practical Forecasting: In real-world forecasting, predictions are usually made sequentially (e.g., forecasting next month's value based on past months). Blocked-CV simulates this real-world scenario by evaluating the model on blocks of data that reflect continuous time sequences.

2.2. **Proposal 2: CV techniques for multi-step-ahead predictions.** The second proposal is to extend the time series cross-validation techniques to different forecast horizons. We can compare the performance between various forecast lengths in the model (e.g., one-step-ahead, multi-step-ahead predictions). This would provide a more flexible evaluation framework. Exploring the use of cross-validation (CV) for multi-step-ahead predictions in time series analysis might contribute in many ways:

- Evaluating Generalization Performance: Multi-step-ahead predictions involve forecasting future with several steps ahead (Cheng et al., 2006), in which the forecasting process could significantly increase uncertainty and error accumulation over time. Exploring CV techniques might provide a way to systematically assess the model's ability. With proper cross-validation, we may help to prevent the model from overfitting, performing poorly on real-world multi-step forecasts.
- Error Propagation in Multi-Step Forecasting: In multi-step forecasting, errors from earlier steps can propagate into the next steps (Cheng et al., 2006). By using CV, we might be able to monitor how quickly these errors accumulate, and investigate whether certain model selections or configurations (such as tuning hyperparameters) can help to mitigate the error growth.

- Handling Temporal Dependencies: We can combine with the idea of the first proposal to see if maintaining the temporal structure in blocks like using blocked-CV can out-perform other methods.

3. Project report

The extension project involves adding regularization techniques to the CV procedure. We introduce the **Lasso Regression** to AR models, and evaluate the performances when using different CV and other methods. Adapting Lasso regression in time series modeling is not new. Wang et al. (Wang et al., 2007) studies the linear regression with auto-regressive errors adapting Lasso procedure with a fixed order. In our project, adding the use of Lasso regression, would help to assess how regularization affects the bias-variance trade-off in cross-validation strategies. Moreover, we compare the performances of different CV methods with OOS evaluation, including the blocked-CV. The simulations are done in Python.

3.1. **Theoretical Aspect.** Adding a Lasso term, which imposes an $L1$ regularization penalty, to the objective function seems likely to affect the proof of **Theorem 1** (since it changes the objective function), but not necessarily in a negative way. Let's consider the same sets of observation data from stationary process as in equation (1.1) and equation (1.3). Now the objective function in equation (1.2) becomes

$$Q(\theta) = \sum_{t=1}^{n} (y_t - g(x_t, \theta))^2 + \alpha \sum |\theta_i|$$

This added term shrinks some of the coefficients toward zero, which has the effect of performing an automatic variable selection and to prevent over-fitting. In common cases, people also use λ to represent the penalty weight. The reason why we use α for notation is to be consistent with the notation used in *Sktlearn* package in Python, which we shall use later for simulation.

The Lasso penalty appears in the model-fitting process (i.e., during the estimation of θ^{α}):

$$\theta^{\alpha} = argmin_{\theta} \left(\sum_{t=1}^{n} (y_t - g(x_t, \theta))^2 + \alpha \sum |\theta_i| \right)$$

The prediction error on the out-set data in equation (1.4) now becomes:

$$PE = E\left[(\tilde{y} - g(\tilde{x}, \hat{\theta^{\alpha}}))^2 \right] \tag{3.1}$$

where $g(\tilde{x}, \hat{\theta^{\alpha}})$ now is based on the Lasso-regularized coefficients $\hat{\theta^{\alpha}}$. The Lasso penalty term affects the values of $\hat{\theta^{\alpha}}$ whereas PE focuses on the difference between

the predictions $g(\tilde{x}, \hat{\theta_\alpha})$ and true value $\tilde{y}$. In a similar vein, the *LOOCV* prediction error on the in-set data in equation (1.5) becomes:

$$\hat{PE} = \frac{1}{n-p} \sum_{t=p+1}^{n} \left(y_t - g(x_t, \theta^{\hat{\alpha}}_{-t}) \right)^2 \tag{3.2}$$

Theorem 1 essentially proves that the cross-validation prediction error (*LOOCV* used for the proof) is a consistent estimator of the true prediction error for stationary auto-regressive models. The proof relies on asymptotic properties of least squares estimators in time series models, such as stationarity and weak dependence. When adding the Lasso term to the objective function, the Lasso regularization changes the coefficient values in the estimator. However, Lasso estimators have well-known asymptotic properties. They are consistent and converge to the true parameters under certain conditions, especially when applied to stationary time series models (Tibshirani, 1996). And the observation data sets are generated from stationary and ergotic auto-regressive models as before and hence satisfy the assumptions as before.

Thus, while the proof of **Theorem 1** would need to be modified to account for the Lasso penalty, the core result (that cross-validation provides consistent prediction error estimates) should still hold, provided that: the data remains stationary and satisfies the necessary assumptions. And the regularization parameter α is chosen appropriately (not too large, which could over-penalize the coefficients).

3.2. **Monte Carlo Simulation.** We follow a similar experimental design to the original paper and (Bergmeir et al., 2014). The key steps are:

- **Data generation**:
  - We generate **1000 Monte Carlo trials** for three model experiments: one from a stable **AR(3)** process, one from a stable **AR(5)** process, and one from a stable **AR(8)** process.
  - Coefficients are randomly generated for each trial to explore a broader parameter space with characteristic roots lying outside of the unit circle.
  - The data are scaled and adjusted to be positive.
- **Data partitioning**: Each trial consists of a **200-length time series**. The first **70%** is used as "in-set" data for CV, and the last **30%** is the "out-set" data mimicking future points.
- **Methods compared**: **Normal 5-fold CV**, **Blocked 5-fold CV**, **LOOCV**, and **OOS**.
- **Model fitting and error measures**:

- For model fitting, we use AR(P) model with and without Lasso regression term, with different α values, i.e., $\alpha = \{0.1, 0.6, 1\}$.
- Error measures used between PE and $\hat{PE}$ are **mean absolute predictive accuracy error (MAPAE)** and **mean predictive accuracy error (MPAE)**, which are the same as in the original paper. You can find the formulas in section 1.3 of this report.
- Error measures used for calculating the prediction error on the out-set data and the CV error on the in-set data are **root mean squared error (RMSE)** and **mean absolute error (MAE)**, which are the same as in the original paper. You can find the formulas in section 1.3 of this report.

3.3. **Simulation Results.** In all three experiment, we use AR models with the same lag order as in the data generation process. We compare the **RMSE** and **MAE** between PE and $\hat{PE}$ using **MAPAE** and **MPAE**. In these cases, the models are not misspecified. We observed that for all three experiments, Blocked CV and Normal CV outperform $LOOCV$ and OOS for **RMSE** error comparison, and all CV methods outperform OOS for **MAE** error comparison. When fitting the data without the Lasso term in the regression model, normal CV provides slightly lower error rates. When fitting the data with the added Lasso term in the regression model, blocked CV tends to perform better.

Based on the simulation results from all three experiments, our findings align with the theoretical expectation that adding a Lasso regularization term to the objective function does not cause the CV methods to break down. In fact, across nearly all the methods, we observe different levels of improvement in accuracy for most fitting scenarios with the added Lasso term. Typically, in Lasso regression, too small alpha values tend to show limited improvements on prediction accuracy. However, too large alpha values can lead to over-regularization, which causes under-fitting. The optimal α value here is not being investigated in this study.

The results are presented in tables from Figure 1 to Figure 3. In the result tables, the data is fitted using a regular AR(p) model from rows 1 to 4, an AR(p) Lasso regression model with $\alpha = 0.1$ from row 5 to row 8, an AR(p) Lasso regression model with $\alpha = 0.6$ from row 9 to row 12, and an AR(p) Lasso regression model with $\alpha = 1$ from row 10 to row 16. The minimum error values across each row are marked in blue. It is important to note that both **MAE** and **RMSE** are scale-dependent metrics. If the data has large values, both **MAE** and **RMSE** will also tend to be large, and vice versa. Also, be aware that **MPAE** doesn't accurately reflect the magnitude of the

errors, since positive and negative errors can cancel out each other, but it provides a directional bias of the predictions, i.e., whether the model tends to over-predict or under-predict on average.

FIGURE 1. Experiment 1

		Data generated by stationary AR(3) process			
		5-folds blocked CV	5-folds normal CV	LOOCV	OOS
AR(3)	MAPAE_RMSE	7.5734e-02	7.4817e-02	1.0559e-01	8.3549e-02
	MAPAE_MAE	6.9030e-02	6.7081e-02	6.6624e-02	7.4770e-02
	MPAE_RMSE	-4.5669e-03	-2.5766e-02	-9.4554e-02	-2.5358e-02
	MPAE_MAE	-4.5669e-03	-2.6699e-02	-2.3719e-02	-2.6578e-02
with Lasso, alpha = 0.1	MAPAE_RMSE	7.3002e-02	7.4732e-02	1.0817e-01	8.3433e-02
	MAPAE_MAE	6.5641e-02	6.6927e-02	6.7015e-02	7.4871e-02
	MPAE_RMSE	-2.4137e-02	-2.6185e-02	-9.8052e-02	-2.5689e-02
	MPAE_MAE	-2.4137e-02	-2.6935e-02	-2.7226e-02	-2.6846e-02
with Lasso, alpha = 0.6	MAPAE_RMSE	7.3496e-02	7.4738e-02	1.0817e-01	8.3453e-02
	MAPAE_MAE	6.6025e-02	6.7003e-02	6.7017e-02	7.4890e-02
	MPAE_RMSE	-2.5697e-02	-2.5965e-02	-9.8051e-02	-2.5690e-02
	MPAE_MAE	-2.5697e-02	-2.6874e-02	-2.7226e-02	-2.6848e-02
with Lasso, alpha = 1	MAPAE_RMSE	7.3496e-02	7.4685e-02	1.0817e-01	8.3453e-02
	MAPAE_MAE	6.6025e-02	6.6935e-02	6.7017e-02	7.4890e-02
	MPAE_RMSE	-2.5697e-02	-2.6049e-02	-9.8051e-02	-2.5690e-02
	MPAE_MAE	-2.5697e-02	-2.6930e-02	-2.7226e-02	-2.6848e-02

FIGURE 2. Experiment 2

		Data generated by stationary AR(5) process			
		5-folds blocked CV	5-folds normal CV	LOOCV	OOS
AR(5)	MAPAE_RMSE	6.6004e-02	6.4355e-02	9.7680e-02	7.3868e-02
	MAPAE_MAE	5.8257e-02	5.6288e-02	5.5559e-02	6.4793e-02
	MPAE_RMSE	-4.4331e-03	-2.2135e-02	-9.0186e-02	-2.4382e-02
	MPAE_MAE	-4.4331e-03	-2.3349e-02	-1.8647e-02	-2.5492e-02
with Lasso, alpha = 0.1	MAPAE_RMSE	6.3399e-02	6.4092e-02	1.0189e-01	7.3608e-02
	MAPAE_MAE	5.5364e-02	5.6134e-02	5.6129e-02	6.4511e-02
	MPAE_RMSE	-2.5842e-02	-2.2336e-02	-9.5728e-02	-2.4821e-02
	MPAE_MAE	-2.5842e-02	-2.3747e-02	-2.4278e-02	-2.5930e-02
with Lasso, alpha = 0.6	MAPAE_RMSE	6.3619e-02	6.3888e-02	1.0189e-01	7.3603e-02
	MAPAE_MAE	5.5527e-02	5.5985e-02	5.6128e-02	6.4506e-02
	MPAE_RMSE	-2.6522e-02	-2.2517e-02	-9.5730e-02	-2.4832e-02
	MPAE_MAE	-2.6522e-02	-2.3909e-02	-2.4280e-02	-2.5940e-02
with Lasso, alpha = 1	MAPAE_RMSE	6.3619e-02	6.4099e-02	1.0189e-01	7.3603e-02
	MAPAE_MAE	5.5528e-02	5.6077e-02	5.6128e-02	6.4506e-02
	MPAE_RMSE	-2.6522e-02	-2.2525e-02	-9.5730e-02	-2.4832e-02
	MPAE_MAE	-2.6522e-02	-2.3875e-02	-2.4280e-02	-2.5940e-02

FIGURE 3. Experiment 3

		Data generated by stationary AR(8) process			
		5-folds blocked CV	5-folds normal CV	LOOCV	OOS
AR(8)	MAPAE_RMSE	5.8728e-02	5.5140e-02	9.1391e-02	6.6226e-02
	MAPAE_MAE	5.1462e-02	4.7940e-02	4.7786e-02	5.7992e-02
	MPAE_RMSE	2.8669e-03	-1.9281e-02	-8.6856e-02	-1.8983e-02
	MPAE_MAE	2.8669e-03	-2.1864e-02	-1.3607e-02	-2.1479e-02
with Lasso, alpha = 0.1	MAPAE_RMSE	5.5274e-02	5.5005e-02	9.9709e-02	6.6413e-02
	MAPAE_MAE	4.7890e-02	4.8045e-02	4.8126e-02	5.8267e-02
	MPAE_RMSE	-2.4575e-02	-2.1131e-02	-9.7366e-02	-2.0786e-02
	MPAE_MAE	-2.4575e-02	-2.3867e-02	-2.4296e-02	-2.3332e-02
with Lasso, alpha = 0.6	MAPAE_RMSE	5.5353e-02	5.5006e-02	9.9719e-02	6.6408e-02
	MAPAE_MAE	4.7963e-02	4.8108e-02	4.8135e-02	5.8269e-02
	MPAE_RMSE	-2.5732e-02	-2.1085e-02	-9.7372e-02	-2.0798e-02
	MPAE_MAE	-2.5732e-02	-2.3906e-02	-2.4302e-02	-2.3348e-02
with Lasso, alpha = 1	MAPAE_RMSE	5.5353e-02	5.4928e-02	9.9719e-02	6.6408e-02
	MAPAE_MAE	4.7963e-02	4.8011e-02	4.8135e-02	5.8269e-02
	MPAE_RMSE	-2.5732e-02	-2.1198e-02	-9.7372e-02	-2.0798e-02
	MPAE_MAE	-2.5732e-02	-2.3839e-02	-2.4302e-02	-2.3348e-02

In addition to the error comparison table, we created violin plots to illustrate the distribution of absolute errors of 1000 trials; the absolute error here means the absolute difference between in-set errors and out-set errors. Each "violin" represents the absolute error distribution between PE and $\hat{PE}$ for a specific method (e.g., $LOOCV$), 2 specific models (e.g., AR(3) and AR(3) Lasso with $\alpha = 0.6$) and a specific error measure (e.g., **RMSE**). The width reflects density, with wider sections indicating more frequent errors, and the boxplot inside summarizes the error statistics. The blue diamond marks the mean for the AR model, while the red diamond represents the AR Lasso model. The results, shown in Figures 4 to 9, indicate that Blocked CV and Normal CV have a higher frequency of errors near zero compared to $LOOCV$ and OOS, especially for **RMSE** over **MAE**. Additionally, the Lasso term appears most effective in Blocked CV, as seen by the asymmetry in its violin plot.

FIGURE 4. Experiment 1: RMSE between PE and $\hat{PE}$

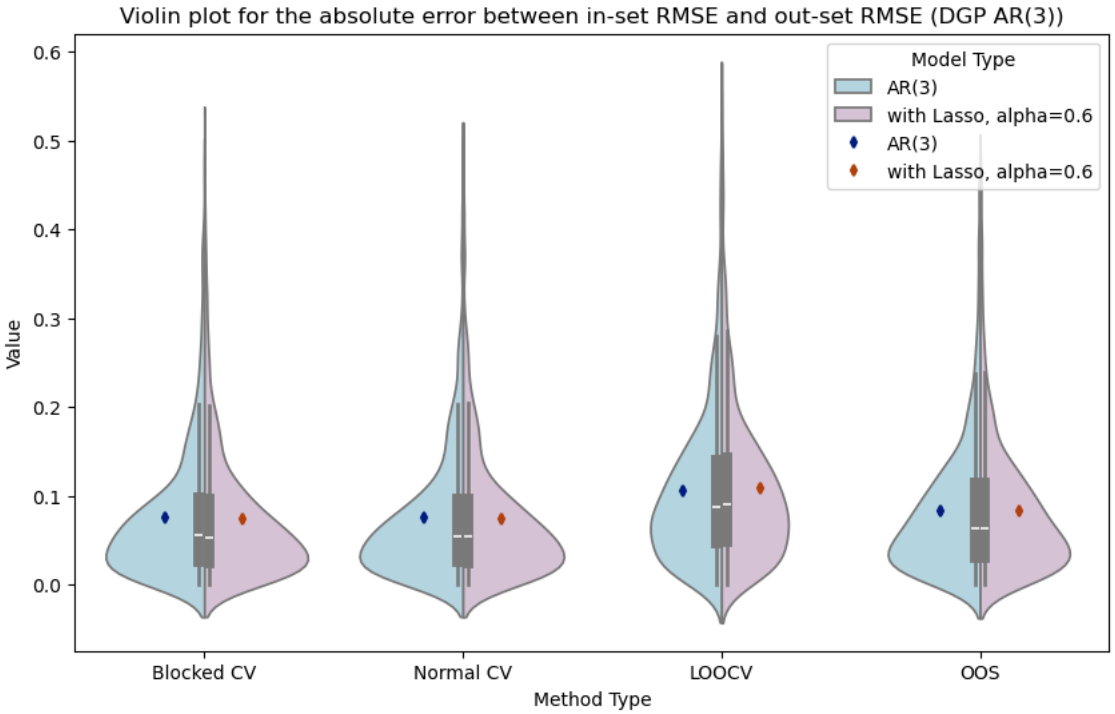


FIGURE 5. Experiment 1: MAE between PE and $\hat{PE}$

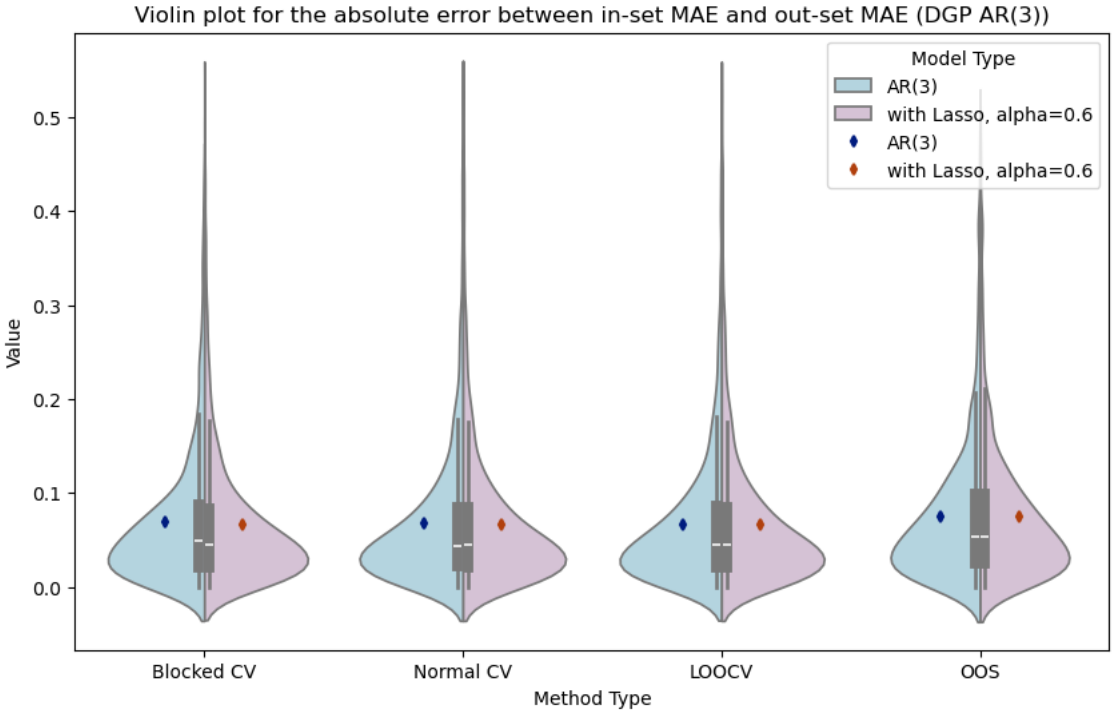


FIGURE 6. Experiment 2: RMSE

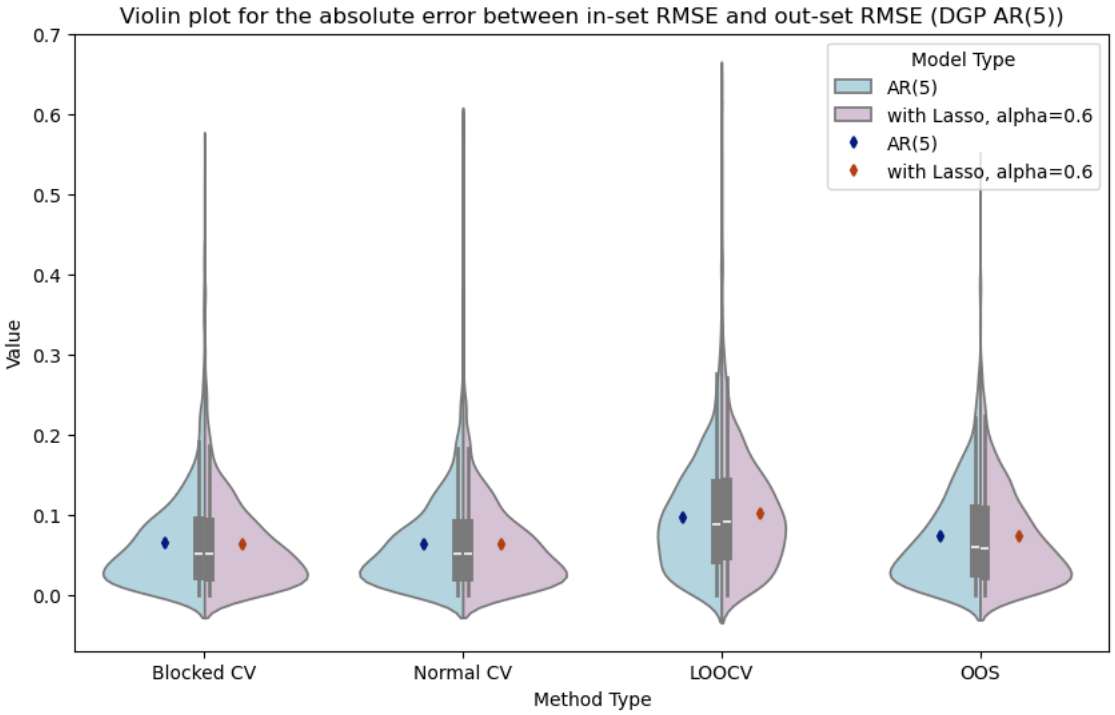


FIGURE 7. Experiment 2: MAE

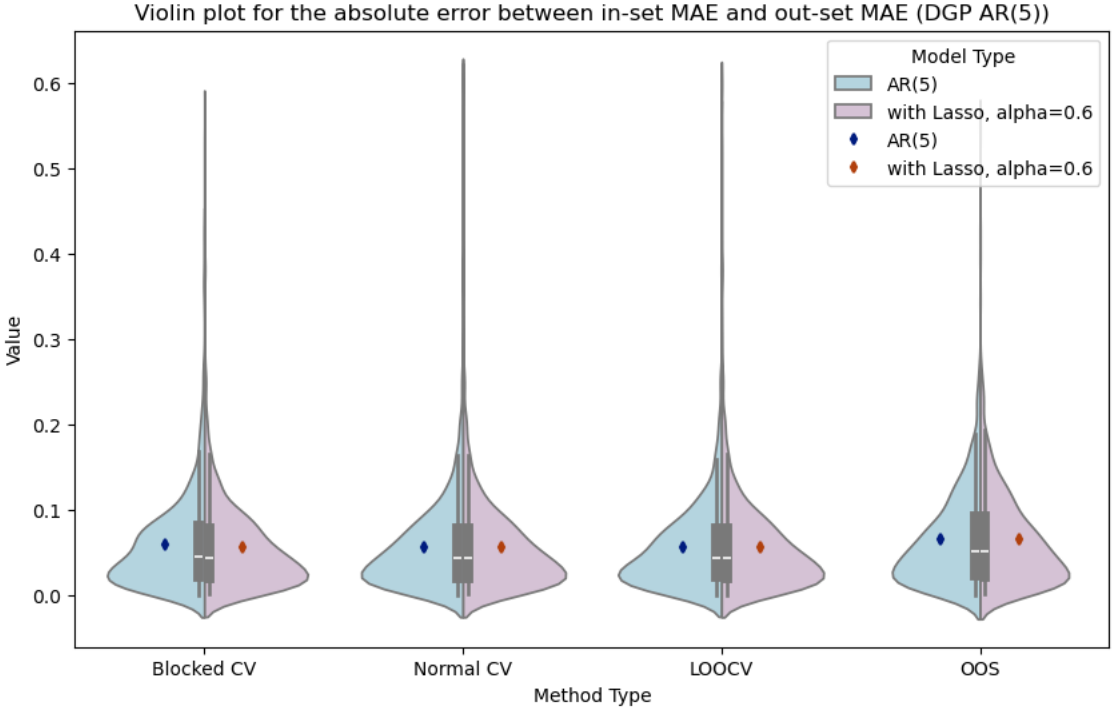


FIGURE 8. Experiment 3: RMSE

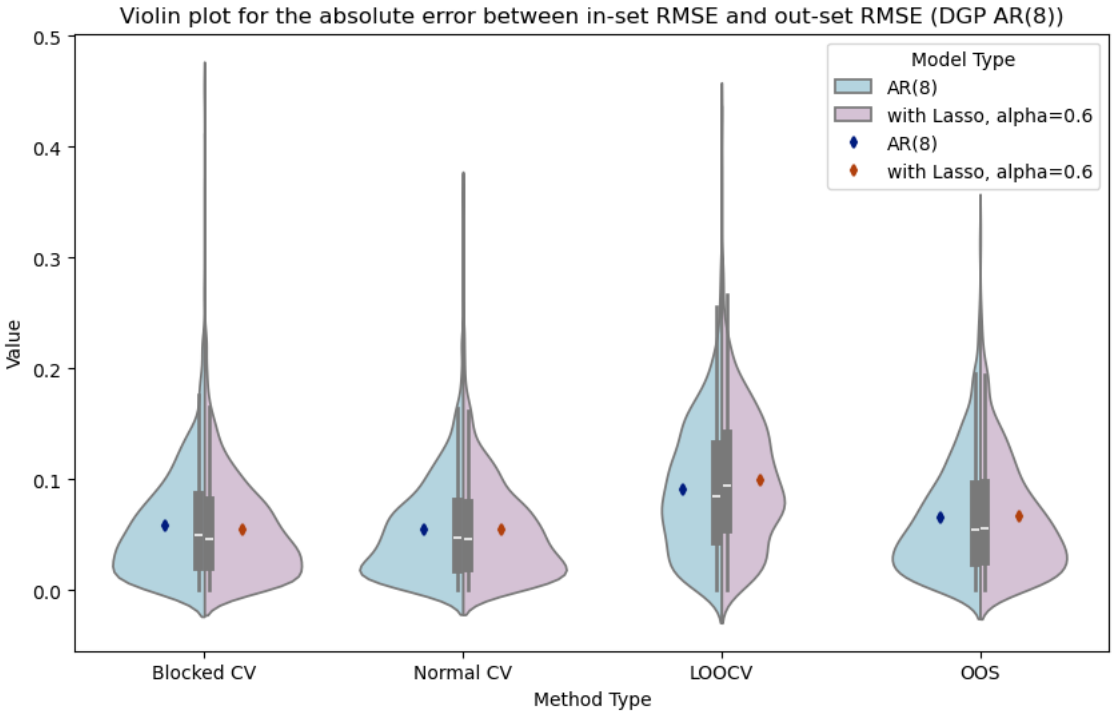
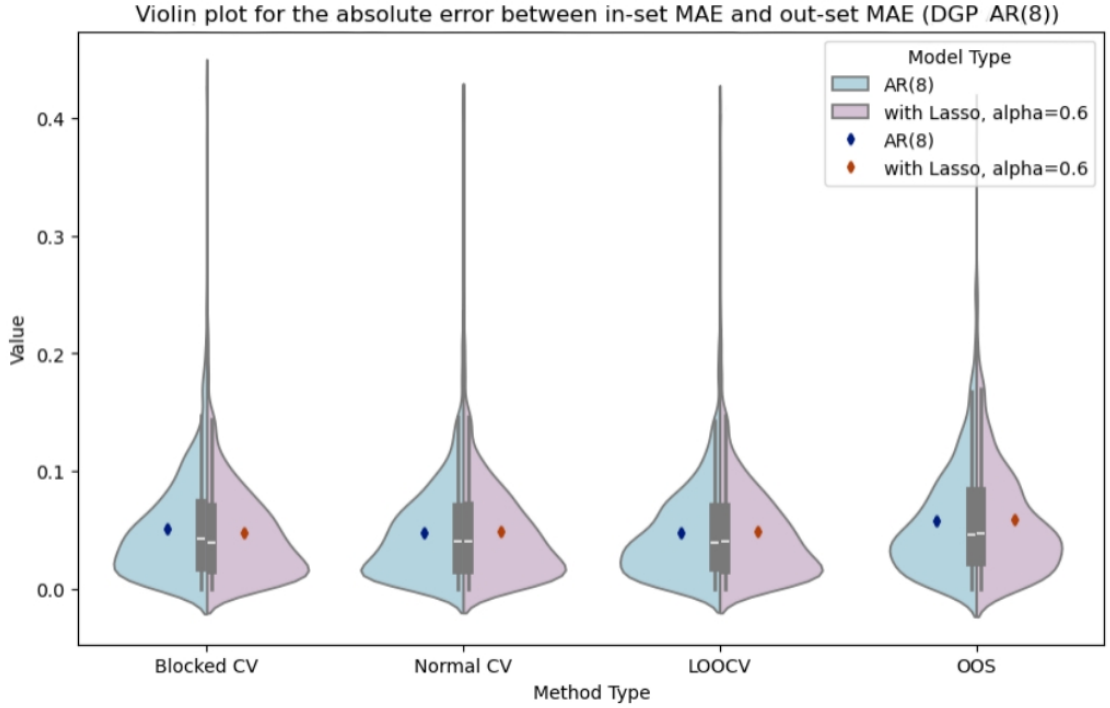


FIGURE 9. Experiment 3: MAE



3.4. **Future Improvements.** We can extend our study for future improvements in several different aspects and directions. In this section, we discuss five potential improvements to our current experimental design and methodology. These improvements aim for better generalization and error estimation for applications on time series models.

Choice of Lasso Parameters: Besides testing with a limited and fixed set of α values as we did in our experiments, we can also use cross-validation within the in-set data to automatically select the optimal α value. And we can simulate the process for each Monte Carlo trial to see if a general range of α values out-performs for the chosen fitting model.

Extension to Ridge Regression: Ridge regression is another regularization technique like Lasso, but instead of penalizing the sum of the absolute values of the coefficients (Lasso), Ridge regression penalizes the sum of the squared values of the coefficients. The key difference is that Ridge regression introduces L2 regularization, which shrinks coefficients but does not necessarily set them to zero (as Lasso might do). We can use similar experiments set-up to compare with the results of Lasso regression.

Extension to Other Time Series Models: We can extend our simulation to different time series models for data generation process and fitting procedure. However, when using complex models, we should always check if they meet the required assumptions.

Other Error Metrics: We can explore using different error measures on in-set cross validation and out-set data. During the experiment, we observed a vast range of values for **RMSE** and **MAE** when using different scaling factors for the time series data. Using error measures like the percentage error might provide a more standard performance evaluation, since it normalizes the error value according to the data's value range. We can also explore additional measures like **BIC** or **AIC** to see how the model complexity affects the generalization error across different fitting methods. We can investigate on how much regularization or model selection improves the model's ability to generalize.

Model Interpretability: As Lasso can drive some coefficients to zero, it would be interesting to track on how many coefficients are reduced to zero across trials. This would provide some insights on how effective Lasso is and to identify any irrelevant lags.

References

Bergmeir, C., & Benitez, J. M. (2012). On the use of cross-validation for time series predictor evaluation. *Information Sciences*, *191*, 192–213.

Bergmeir, C., Costantini, M., & Benitez, J. M. (2014). On the usefulness of cross-validation for directional forecast evaluation. *Computational Statistics and Data Analysis*, *76*, 132–143.

Bergmeir, C., Hyndmanb, R. J., & Koob, B. (2018). A note on the validity of cross-validation for evaluating autoregressive time series prediction. *Computational Statistics and Data Analysis*, *120*, 70–83.

Bradley, R. C. (2005). Basic properties of strong mixing conditions: A survey and some open questions. *Probability Surveys*, *2*, 107–144.

Brockwell, P. J., & Davis, R. A. (1991). *Time series: Theory and methods.* Springer.

Burman, P., Chow, E., & Nolan, D. (1994). A cross-validatory method for dependent data. *Biometrika*, *81*(2), 351–358.

Burman, P., & Nolan, D. (1992). Data-dependent estimation of prediction functions. *Journal of Time Series Analysis*, *13*(3), 189–207.

Cheng, H., Tan, P., Gao, J., & Scripps, J. (2006). Multistep-ahead time series prediction. *Lecture Notes in Computer Science()*, *3918*.

Doukhan, P. (1994). *Mixing: Properties and examples*. Springer.

Gyorfi, L., Hardle, W., Sarda, P., & Vieu, P. (1989). *Nonparametric curve estimation from time series*. Springer Verlag, Berlin.

Hastie, T., Tibshirani, R., & Friedman, J. (2009). *Elements of statistical learning*. Springer.

Shao, Y., Li, X., Zhang, T., Chu, S., & Liu, X. (2019). Time-series-based leakage detection using multiple pressure sensors in water distribution systems. *Sensors*, *19*(3070).

Takemura, A. (2016). Exponential decay rate of partial autocorrelation coefficients of arma and short-memory processes. *Statistics & Probability Letters*, *110*, 207–210. https://doi.org/https://doi.org/10.1016/j.spl.2015.12.023

Tashman, L. J. (2000). Out-of-sample tests of forecasting accuracy: An analysis and review. *International Journal of Forecasting*, *16*(4), 437–450.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society*, *58*(1), 267–288.

Wang, H., Li, G., & Tsai, C. (2007). Regression coefficient and autoregressive order shrinkage and selection via the lasso. *Journal of the Royal Statistical Society*, *69*(1), 63–78.