Time Series Model Performance Analysis with ARIMA and SARIMA for Streamflow Forecasting in the Appalachia

Adam O. Taylor[1]

Abstract: The primary goal of this study was to identify a model that can be used to accurately forecast streamflow within the temperate humid subtropical and oceanic climates of the southern Appalachia. Data was compiled for the Pound and Russell Fork rivers from both USGS web interface and NOAA API requests at 15-minute intervals. The data was then analyzed and cleaned before proceeding with the study. Previous studies were found which made monthly streamflow predictions using the stochastic univariate autoregressive integrated moving average (ARIMA) model with acceptable results [1]; however, monthly predictions are not particularly useful for every application. The 15-minute data was aggregated into 9 time-intervals ranging from 15-minutes to 1-month, where each time-interval was generated by using a) the first datapoint, b) the datapoint with the maximum value within the range, and c) the mean of all datapoints within the range, giving 25 unique datasets. The ARIMA model produced good results with the 1-hour, 6-hour, 12-hour, and daily mean time intervals, and the seasonal autoregressive integrated moving average (SARIMA) model gave acceptable results for the 7-day, 14-day, and monthly mean time intervals. The hyper-parameters for each model were selected and optimized by using the augmented Dickey-Fuller (ADF) test for stationarity, autocorrelation (ACF), partial autocorrelation (PACF), Akaike information criterion (AIC), root mean square error (RMSE), and mean absolute percentage error (MAPE). The results were validated using time series cross validation and residual analysis for in-sample and out-of-sample forecasts. It was found that time interval and aggregation method were both highly significant in model performance.

Keywords: time series; SARIMA; ARIMA; VAR; streamflow forecasting; model optimization

I. INTRODUCTION

Time Series analysis is a way of analyzing data to understand past trends and forecast future trends by extracting useful statistical information from sets of data that are arranged chronologically. A time series dataset is a collection of data points that are recorded at specific time intervals where time itself is the independent variable. The applications are far reached and extend through many disciplines such as medicine, bioinformatics, weather, economics, and finance. Time series forecasting, like most methodologies in data science, is a multidisciplinary approach that incorporates techniques from the fields of mathematics, statistics, and computer science. Depending on the type of data, a deeper understanding in other fields may be necessary.

This study focuses on several time series forecasting models, including ARIMA, SARIMA, and vector autoregression (VAR). The ARIMA(p,d,q) model is a combination of an autoregressive model and a moving average model, coupled with an additional parameter to handle datasets where mean and/or variance change over time. It is stochastic and quantitative since it allows for random variation and the data is structured, objective, and measurable. The equations for Autoregression (AR) and linear regression (LR) are similar, but the main difference is that in AR the predictions are made based on lagged values of the dependent variable, whereas in LR the forecasts are derived from a separate independent variable. As mentioned above, the independent variable in univariate time series analysis is time. The autoregressive term is denoted by (p), and is the first parameter in the ARIMA(p,d,q) model. It is also known as the non-seasonal autoregressive term. The moving average (MA) term (q), also known as the non-seasonal moving average term, refers to the lags of the

forecasted errors, or the residuals. The third hyperparameter (d) represents the number of non-seasonal differences that will be applied to the dataset. The ARIMA model requires the data to be stationary to make accurate predictions, and a dataset is non-stationary if its mean or variance change over time. The goal is to eliminate any trend that may exist in the data. This detrending can be achieved through numerous methods, but here differencing will be the type of data transforamtion used (Figure 1). This method simply subtracts each datapoint from the previous datapoint.

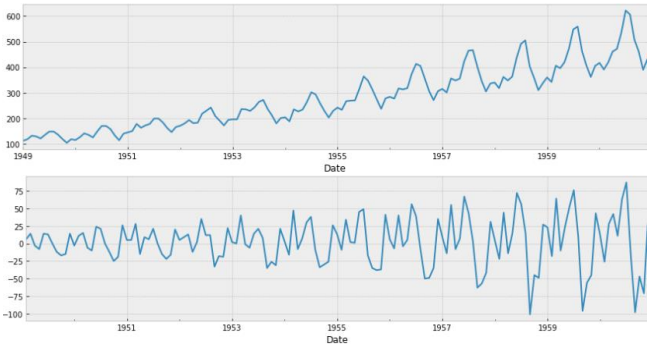


Figure 1. Line plots of the air passenger dataset before and after differencing with Pandas diff() method as an example of a change in mean over time causing the dataset to be non-stationary. The second graph which is the result of passing the dataframe to diff() is an example of a change in variance over time.

Other factors that negatively impact model performance are the presence of anomalies, irregular data, and missing data. These are generally handled in the pre-processing stage and require anomaly detection and imputation after close analysis of the data. For example, the USGS Pound and Russell Fork river data contained duplicate rows and some missing data. The duplicates were removed and it was determined the missing data could be imputed by forward filling without any significant change in the overall dataset. After preprocessing, it is common practice to begin with the Box-Jenkins methodology. This describes the process of ARIMA optimal parameter selection as an iterative approach that cycles through model identification, parameter estimation, and model checking. Further optimization and parameter tuning can be done with Auto-ARIMA and other more custom approaches, which will be described later.

ARIMA(p,d,q) Model Equation:

$$\mathbf{y'_t = c + \phi_1 y'_{t-1} + \cdots + \phi_p y'_{t-p} + \Theta_1 \varepsilon_{t-1} + \cdots + \Theta_q \varepsilon_{t-q} + \varepsilon_t} \quad (1)$$

y'_t = differenced time series (I) or (d) at time (t)

C = intercept or constant

Φ = autoregressve parameters of order p

Θ = moving average parameters of order q

y'_{t-p} = lagged values of y

ε = error term

$\phi_1 y'_{t-1} + \cdots + \phi_p y'_{t-p}$ = lagged values (AR) or (p)

$\Theta_1 \varepsilon_{t-1} + \cdots + \Theta_q \varepsilon_{t-q} + \varepsilon_t$ = lagged errors (MA) or (q)

By looking at the above equation it can be noted that if p = 2, then the predictors for y'(t) will be y'(t-1)...y'(t-2). Likewise, if q = 2 then the predictors for y'(t) will be E(t-1)...E(t-2).

SARIMA, also referred to as seasonal ARIMA, is an extension of ARIMA that adds functionality to handle seasonal trend. As discussed above, the ARIMA model requires the data to be stationary to make accurate forecasts, and for data to become stationary, trend must be reduced. Seasonal trend or seasonality can be described as patterns that exist in a time series dataset where repetitions can be observed at predictable time intervals. During the study it was noticed that the effects of seasonality on forecast outcome is determined by the time interval of the data. SARIMA deals with seasonal trend by adding additional parameters (P,D,Q,m), where (m) represents the length of the season; for example, if you have a monthly dataset and a yearly seasonal pattern is noticed, m or the seasonal period will be 12.

SARIMA(p,d,q)(P,D,Q,m) Model Equation:

$$\mathbf{y'_t = c + \sum_{n=1}^{p} \emptyset_n y'_{t-n} + \sum_{n=1}^{q} \theta_n \varepsilon_{t-n} + \sum_{n=1}^{P} \eta_n y'_{t-mn} + \sum_{n=1}^{Q} \omega_n \varepsilon_{t-mn} + \varepsilon_t}$$

y': differenced time series determined by **d** and **D**

p: number of autoregressive components

P: number of seasonal auto-regressors

ϕ: coefficients of the autoregressive components (lags)

ω: coefficients of the seasonal autoregressive components

ε: forecast error terms, the moving-average components

q: number of lagged error components

Q: number of seasonal moving-average components

θ: coefficients of the lagged forecast errors

η: coefficients of the seasonal forecast errors

m: length of season

While streamflow forecasting is not a novel concept, many applications revolve around flood, emergency, and resource management, hydropower, and agriculture implications. Also, more advanced models and hybrid models have been developed for real time forecasting. The three main goals of this study are to develop an automated processes for running ARIMA/SARIMA models, obtain deeper understanding of ARIMA/SARIMA model optimization through comparing different characteristics of the dataset, and forecasting streamflow for recreational purposes.

[RESULT SUMMARY]

II. MATERIALS AND METHOD

The Russell Fork River is located on the border of the Tennessee and Ohio River basins within the larger Mississippi-Missouri River basin. The 2-digit hydrologic unit code (HUC) for these watersheds is 06 and 05, respectively. The majority of the Russell Fork lies within Pine Mountain Management Area in the northernmost part of the Cumberland Mountains of the Appalachian, but originates in southwestern Virginia. Russell Fork is approximately 50 miles long and serves as a tributary to Levisa Fork near Pikeville, KY. The target section of this study is the Russell Fork Gorge, located in Breaks Gorge of Breaks Interstate Park. The Russell Fork Gorge is known worldwide as being a popular whitewater destination. The characteristics of this river are similar to other such destinations in the area, which are defined by 7 attributes: size, shape, topography, geology, climate, vegetation, and land use. Each of these features greatly affect the nature of streamflow, thus this river can be considered a good predictor to other such destinations in the southern Appalachia.

A. Data Collection

Streamflow and precipitation data were collected for the 10 year period between 1/1/2013 1:00 and 12/31/2022 23:45 with the datapoints every 15 minutes. Streamflow data was obtained through the USGS national water dashboard for the Russell Fork at site number 03208500 with coordinates (37.20705367, -82.2956993), and for the Pound River at site number 03209000 with coordinates (37.23705384, -82.34320179). Precipitation data was acquired through NOAA API requests for station ID GHCND:USC00440766 located in Blacksburg, VA. Precipitation data was used solely for multivariate vector autoregression (VAR) modeling.

B. Preprocessing

To get accurate streamflow observations for the Russell Fork Gorge, which lies several miles below the confluence of the Russell Fork and Pound Rivers, the readings from each river needed to be combined. Analysis of the data found duplicate and missing data for each dataset. A method was created that plotted each day that contained missing data, along with matching precipitation for the previous 5 days. It was determined that imputation of the data using forward filling, ffill() from the Pandas library, would give observations that would have no significant impact on model fitting or resulting forecasts, Figure 2.

12/24/2022 2:15	EST	89.3
12/24/2022 6:15	EST	88.6
12/24/2022 10:15	EST	88
12/24/2022 14:15	EST	87.3
12/24/2022 18:15	EST	86.6
12/24/2022 22:15	EST	85.9
12/25/2022 2:15	EST	85.3
12/25/2022 6:15	EST	84.8
12/25/2022 10:15	EST	84.4
12/25/2022 14:15	EST	83.9
12/25/2022 18:15	EST	83.7
12/25/2022 22:15	EST	83.5
12/26/2022 2:15	EST	83.2
12/26/2022 6:15	EST	83
12/26/2022 10:15	EST	82.7
12/26/2022 14:15	EST	82.4
12/26/2022 18:15	EST	82.2
12/26/2022 22:15	EST	81.8
12/27/2022 2:15	EST	81.4
12/27/2022 6:15	EST	80.9
12/27/2022 10:15	EST	80.5
12/27/2022 14:15	EST	80.4
12/27/2022 18:15	EST	80.2
12/27/2022 22:15	EST	80
12/28/2022 2:15	EST	79.8
12/28/2022 6:15	EST	79.7
12/28/2022 10:15	EST	79.6
12/28/2022 14:15	EST	79.5
12/28/2022 18:15	EST	79.4
12/28/2022 22:15	EST	79.2

Figure 2. Table giving an example readout for a 5-day period of missing streamflow(cfs) data which was imputed by forward filling. No missing data was found to occur after a significant rain event.

Before imputing the data, the dataframes were reindexed to correct for erroneous timestamp values. The Pound and Russell Fork data were then merged on the independent variable, time, to create the 15-minute Russell Fork Gorge

dataset. A function was created that performed downsampling using three aggregation methods; first, mean, and max. Dataframes were created for each aggregation method with the parameter specified frequency using the 15-minute Russell Fork Gorge dataset, then returned. An associative array or dictionary in python was used to compile the data with dataframes as values and both frequency and aggregation type as the keys. The time intervals used in this study are as follows: 30-minute, 1-hour, 6-hours, 12-hours, 1-day, 7-days, 14-days, and 1-month. As already mentioned, 25 unique datasets were used in the study, Figure 3.

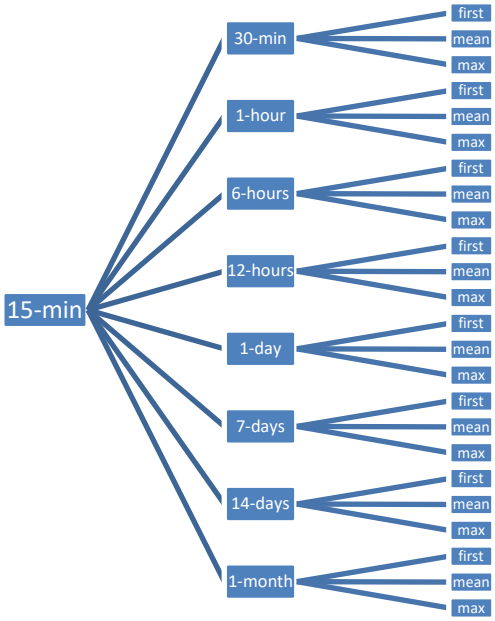
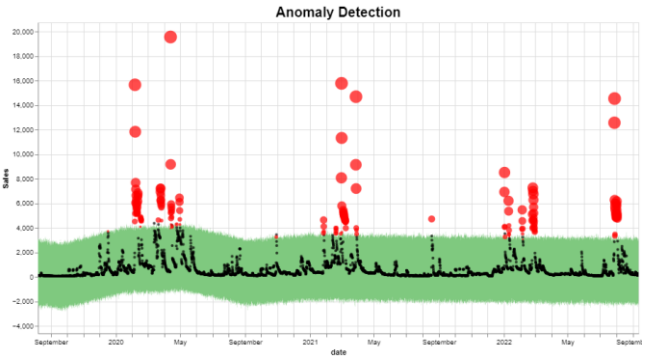


Figure 3. Visual represnetation of the datasets where the first column is the 15-minute RFG dataset, the second column shows the results of downsampling, and the third column shows the datasets obtained after both downsampling and aggregation.

The last step in the preprocessing phase was anomaly detection. One limitation of ARIMA and SARIMA models is they tend to produce inaccurate results when the data contains a large number of outliers. Overprediction for values on the extreme low end and underprediction for values on the extreme high end should be expected. This can be dealt with by normalizing the data, removing the anomalies, or replacing the values. Since the outliers can be explained by significant precipitation events it was decided the anomalies would be kept. The figures below show the outliers for the 6-hour frequency dataset that was resampled using the mean function. Facebook Prophet was used to detect the anomalies in the dataset.



	ds	trend	yhat	yhat_lower	yhat_upper	fact	anomaly	importance
59	2013-01-15 18:00:00	1416.104237	1416.104237	-1163.948804	4083.593435	4347	1	0.060595
60	2013-01-16 00:00:00	1414.991861	1414.991861	-1180.205793	4160.526621	6730	1	0.381794
61	2013-01-16 06:00:00	1413.879484	1413.879484	-1257.219702	4031.549248	7058	1	0.428797
62	2013-01-16 12:00:00	1412.767108	1412.767108	-1051.813004	3964.024929	6347	1	0.375449
66	2013-01-17 12:00:00	1408.317603	1408.317603	-1691.738418	4252.343491	5107	1	0.167350
...	...	...	...	...	...	...	...	...
14000	2022-08-02 00:00:00	500.288188	500.288188	-2100.950321	3280.240161	5463	1	0.399553
14001	2022-08-02 06:00:00	500.167601	500.167601	-2310.904080	3026.216690	5192	1	0.417139
14002	2022-08-02 12:00:00	500.047013	500.047013	-1927.518670	2983.583832	5034	1	0.407314
14003	2022-08-02 18:00:00	499.926426	499.926426	-2135.376551	3034.086100	4895	1	0.380166
14004	2022-08-03 00:00:00	499.805839	499.805839	-2216.091927	3150.061575	4760	1	0.338222

438 rows × 8 columns

Figure 4. Plot showing outliers and signifance level denoted by the size of the datapoint. Table that contains all outliers from the dataset, 438 out of 14605 datapoints or roughly 3% of the data falls outside of the 99% confidence interval.

C. ARIMA Model

As mentioned in the introduction, stationarity is an assumption that must be met to successfully model the data. The main test used in this study to check for stationarity was the Augmented Dickey Fuller (ADF) test. Before running the ADF test, preliminary exploration using seasonal decomposition revealed the data did not present a noticeable change in mean or trend. This suggests the data is stationary, however the seasonal component shows an obvious yearly seasonal trend, Figure 5.

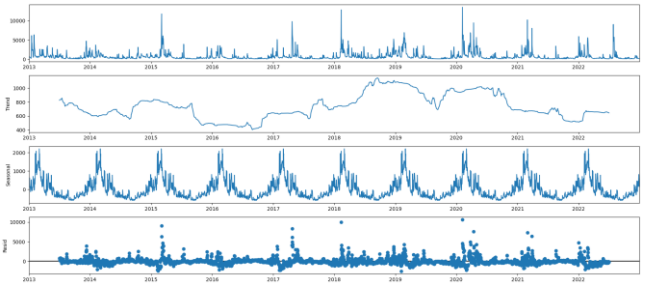


Figure 5. Seasonal decomposition using statsmodels seasonal_decompose() method for the daily-mean dataset using an additive model with period set to 365. Here the

trend seems to remain consistent but a yearly pattern can be noticed.

The ADF test was implemented using a method from the Python module statsmodels, which returns the ADF test statistic, p-value, and 1%, 5%, and 10% critical values. The null hypothesis is there is a unit root, or that the data is non-stationary. To reject the null hypothesis and accept the alternative hypothesis, the p-value must be below the 0.05 significance level. If the p-value is close to 0.05, the test statistic can be used and must be below the 5% critical value. A function was created that iterates through the dictionary of resampled datasets and passes each dataset to the adfuller() statsmodel method [1]. If the null hypothesis cannot be rejected then the dataset will be differenced until the data becomes stationary according to the ADF test. The function returns a dicitonary that contains the processed datasets. The only dataset that did not pass the ADF test was the monthly-mean with p-value:0.25, test statistic:-2.08, and 5% critical value:-2.89. The dataset became statioary after a singe differencing, Figure 6.

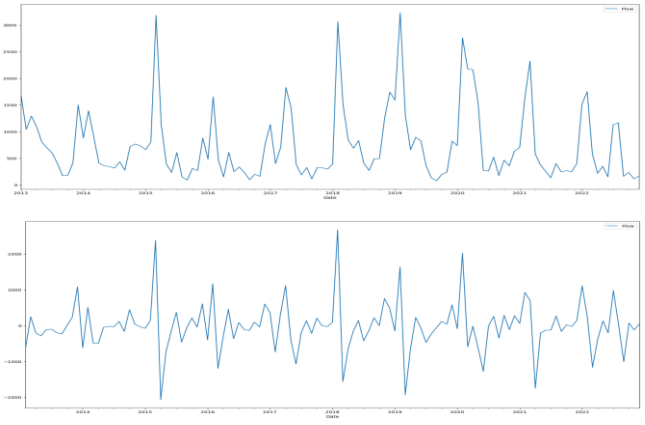
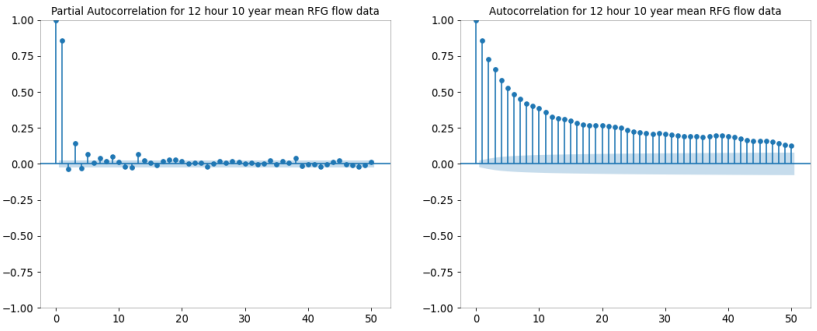


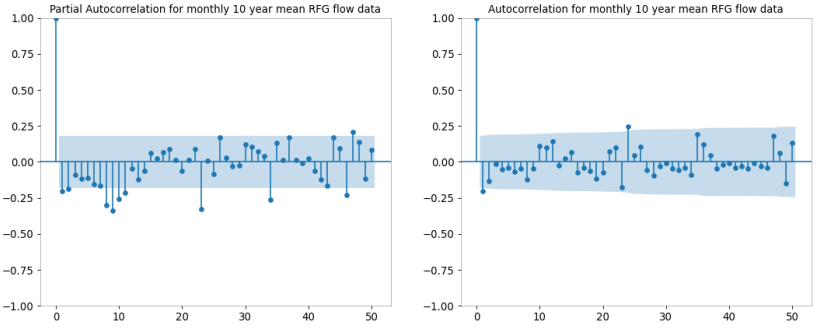
Figure 6. Plot of the monthly-mean dataset before and after differencing using Pandas diff() method.

The ADF test above was used in selecting the starting parameter d for the ARIMA(p,d,q) model. The next step was to determine the starting parameters for p and q. This was achieved by creating ACF and PACF plots for each dataset. ACF plots show the correlation coefficient between a time series and its lags, where 1 indicates perfect correlation, 0 represents no correlation, and -1 expresses negative correlation. PACF plots show the correlation coefficient of the residuals and lags. Each plot displays the 95% confidence interval (CI). The PACF plots for every dataset with a time interval less than or equal to 1 day had a sharp drop-off and the ACF plots had a gradual decay, Figure 7a. This indicates an AR process, where the value for p will be a positive integer greater than 0, and q will be 0. If the inverse were true where the ACF plot showed a sharp drop and the PACF plot expressed a gradual decay then an MA proccess would be used, which would look like the following; ARIMA(0,d,q), where the value for p would be 0. Datasets with time intervals greater than 1 day showed no gradual decay in either plot, suggesting an ARMA process be used. Gradual decay can also be referred to as geometrically declining. The autocorrelation plot in Figure 7 suggests that the data still has a non-stationary component and further data transformation may be required, as presented in the results. This is due to the fact that the autocorrelations are positive and outside of the 95% CI for a signinifact number of lags, and trends to zero very slowly, despite ADF results. This is different from an ACF or PACF plot that is considered to be geometrically declining, suggesting an AR or MA process, respectively. The main difference is the number of significantly correlated lags. Figure 7c summarizes these concepts.

a.



b.



c.

	S(m)	AR(p)	I(d)	MA(q)
ACF	Denoted by an oscillating pattern in the autocorrelations or by repeating non-zero spikes	Geometric Decay	Significant number of non-zero autocorrelations or autocorrelations that do not reach zero.	Sharp tail-off at q
PACF		Sharp tail-off at p		Geometric Decay

Figure 7. PACF and ACF plots for 12-hour-mean (a) and monthly-mean datasets (b). (c) Table summarizing the process of selecting ARIMA(p,d,q) and SARIMA(p,d,q,m) parameters through ACF and PACF plot analysis.

By looking at the plots in figure 7, the following parameters were selected for fitting the ARIMA model with the 12-hour-mean and monthly-mean datasets, respectively: ARIMA(1,0,0), ARIMA(1,1,1). When selecting the starting orders p and q, the last period that has significant correlation was selected. If that doesn't produce desired results other significant spikes can be used for selecting the AR or MA order. The following were used as starting ARIMA parameters for each of the datasets, Figure 8.

Interval	Model	pdq
1H_first	ARIMA	(2, 0, 0)(0, 0, 0, 0)
1H_mean	ARIMA	(3, 0, 0)(0, 0, 0, 0)
1H_max	ARIMA	(2, 0, 0)(0, 0, 0, 0)
6H_first	ARIMA	(1, 0, 0)(0, 0, 0, 0)
6H_mean	ARIMA	(3, 0, 0)(0, 0, 0, 0)
6H_max	ARIMA	(3, 0, 0)(0, 0, 0, 0)
12H_first	ARIMA	(2, 0, 0)(0, 0, 0, 0)
12H_mean	ARIMA	(1, 0, 0)(0, 0, 0, 0)
12H_max	ARIMA	(1, 0, 0)(0, 0, 0, 0)
1D_first	ARIMA	(2, 0, 0)(0, 0, 0, 0)
1D_mean	ARIMA	(1, 0, 0)(0, 0, 0, 0)
1D_max	ARIMA	(1, 0, 0)(0, 0, 0, 0)
7D_first	SARIMA	(3, 0, 0)(1, 1, 1, 52)
7D_mean	SARIMA	(1, 0, 0)(0, 1, 1, 52)
7D_max	SARIMA	(2, 0, 3)(0, 1, 1, 52)
14D_first	SARIMA	(2, 0, 2)(1, 1, 1, 26)
14D_mean	SARIMA	(1, 0, 3)(0, 1, 1, 26)
14D_max	SARIMA	(1, 0, 1)(0, 1, 1, 26)
1M_first	SARIMA	(0, 0, 0)(1, 1, 0, 12)
1M_mean	SARIMA	(1, 1, 1)(1, 1, 1, 12)
1M_max	SARIMA	(0, 0, 0)(1, 1, 0, 12)

Figure 8. The ARIMA(p,d,q) parameters that were selected from ACF and PACF plot analysis.

Once the starting parameters were determined the ARIMA(p,d,q) model was then fitted with the streamflow time series data. Due to the number of datasets, functions were created to assist with model fitting, making predicitons, and plotting the results [2]. In-sample non-rolling, in-sample 1-step ahead rolling, and out-of-sample 1-step ahead rolling forecasts were performed on each set of data. The SARIMAX predict() and forecast() functions were both used for the rolling predictions to compare results. The datasets for the rolling forecasts were also sequentially split into training and testing sets where the training sets were used to fit the ARIMA model, and the testing sets were used for forecasting. AIC, RMSE, and MAPE were used to determine how well the models performed for streamflow forecasting and were used to tune the parameters. After several iterations the top three models were chosen for further analysis, which included the 1-hour-mean, 6-hour-mean, 12-hour-mean, and daily-mean datasets.

Residual analysis was performed on the top three models. The residuals (r) are what the model was unable to explain and can be found by taking the difference of the fitted values ($\hat{y}$) from the actual values (y).

$$r_i = y_i - \hat{y}_i$$

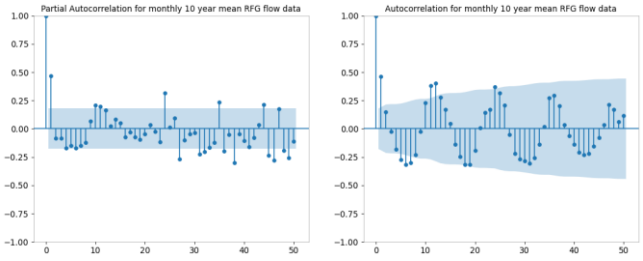
The main characteristics to look for in residual analysis are that the residuals are not correlated, are normally distributed, have constant variance, and have a mean centered at or close to zero. These attributes were found using ACF and PACF plots of the residuals, histograms, quantile-quantile (Q-Q)

plots, and residual plots, respectively. The Ljung-Box test was also used to analyze residual correlation. After validating the model, rolling cross-validation was used in evaluating performance on differenct sets of data.

D. SARIMA Model

Since the streamflow data used here shows a clear seasonal pattern that repeats yearly, which can be seen in both Figure 5 and Figure 9, the next step was to fit the data to a SARIMA model.

a.



b.

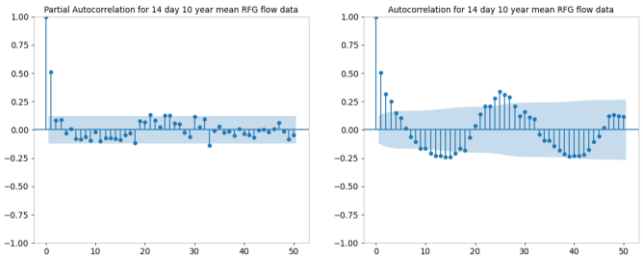


Figure 9. PACF and ACF plots for monthly(a) and 14-day(b) mean datasets showing seasonal trend. Correlation peaks at lags of 12, 24, and 36, where 12 lags represent 1 year (a). Correlation peaks at lags 24 and 48, where 24 lags represent 1 year (b).

The 7-day, 14-day, and monthly datasets were used with the seasonal parameter, m, set to 52, 26, and 12 based on the significantly coorelated lags in the PACF and ACF plots. If the seasonal pattern is unclear by simply plotting the data or by analyzing autocorrelation plots, the Canova-Hansen (CH) test can be used to determine a good starting parameter value for (D), which is the seasonal difference parameter in the SARIMA model. Since the seasonal trend was apparent in the Russel Fork streamflow data, this was an unnecessary step. The values for the parameters (p,d,q) are the same as in the ARIMA model, and the P and Q parameters can also be selected by looking at autocorrelation plots. In figure 9a, the AR order could be either 1 or 2 since the lags at 10 and 11 are only slightly significant, but the lag at 23 is highly significant. The MA order is 1 since the lag at order 12 is a peak and is also highly significant, giving a seasonal order of (1,1,1,12) or (2,1,1,12). By analyzing the plots in figure 9b, the seasonal order for the 14-day dataset is (0,1,1,26). Since each of these datasets have definitive seasonal patters, D will be set to 1 for each. The seasonal differencing should never be greater than 2, and the total differencing for a SARIMA(p,d,q)(p,d,q,m) model should never exceed 3. The following are the starting parameters for the 7-day, 14-day, and monthly datasets:

```
order_list = [[5,0,0],#'15min 1 year'
              [3,0,0],#'30min 2 year'
              [3,0,0],#'30 min 2 year mean'
              [3,0,0],#'30 min 2 year max'
              [2,0,0],#'hourly 5 year'
              [3,0,0],#'hourly 5 year mean'
              [2,0,0],#'hourly 5 year max'
              [3,0,0],#'6 hour 10 year'
              [3,1,0],#'6 hour 10 year mean'
              [3,0,0],#'6 hour 10 year max'
              [2,0,0],#'12 hour 10 year'
              [3,1,0],#'12 hour 10 year mean'
              [2,0,0],#'12 hour 10 year max'
              [1,0,0],#'daily 10 year'
              [1,1,0],#'daily 10 year mean'
              [3,0,0],#'daily 10 year max'
              [1,0,1],#'7 day 10 year'
              [1,1,3],#'7 day 10 year mean'
              [1,0,2],#'7 day 10 year max'
              [1,0,1],#'14 day 10 year'
              [1,1,3],#'14 day 10 year mean'
              [1,0,1],#'14 day 10 year max'
              [1,0,1],#'monthly 10 year'
              [2,2,1],#'monthly 10 year mean'
              [1,0,1]]#'monthly 10 year max'
```

Figure 10. The SAIMA(p,d,q)(P,D,Q,m) parameters that were selected by analyzing PACF and ACF plots.

Once the parameters were selected, the same process described above was used for fitting the data to the SARIMA model, making forecasts, and analyzing the residuals for model validaiton. After several cycles through the iterative Box-Jenkins methodology it was determined the 1-hour, 6-hour, 12-hour, and 1-day datasets were a good fit for ARIMA modeling, and the 7-day, 14-day, and 1-month datasets were best fitted to a SARIMA model. For simplification and time complexity reduction these combinations were used in the results section below, along with eliminating the 15-minute and 30-minute datasets. A method was also created to reduce the amount of data contained in each dataset by setting a maximum limit, which was set to 10,000 rows.

E. Time Series Module

A module was created in python that automates much of this process. The module contains classes for preprocessing the raw data, creating the dataframes, generating plots and

tables for parameter estimation, fitting the data to the models, producing diagnostic outputs for model analysis, creating forecast and cross validation results plots, running auto-ARIMA and auto-SARIMA, and file handling due to a complex directory structure. The output from the module is the following: ACF and PACF plots, ADF results table, auto-ARIMA and auto-SARIMA results, cross validation plots and results tables, decomposition plots, diagnostic plots including residual plots, histograms, Q-Q plots, and correlogram plots, and forecasting results plots and tables. Where applicable the resulting files are organized based on modeling method, model type, and forecasting method. Since one of the goals of this study was to compare time series models, aggregation methods, forecasting types, and selecting optimal parameters based on varying metrics, a more automated process was needed to decrease the complexity and time involved.

F. Model Comparison

The process used to determine optimal hyper-parameters can be summarized by the following: parameter selection using ACF, PACF, and ADF results, parameter refinement using Box-Jenkins methodology, and an automated process that iterates through a range of parameters configurated by user input then determines model order based on metric results. The metrics used in the auto-ARIMA and auto-SARIMA class were MAPE, AIC, and RMSE. The top models for each metric were determined and then analyzed for performance. As mentioned previously, time interval had a significant impact on which model performed better, ARIMA or SARIMA, for each dataset. This is directly impacted by the season of the data, which was a yearly pattern for the streamflow data. It was also noticed that parameter selection based on MAPE values was the best predictor for overall model performance. However, this metric gives insight to model performance based on a comparison between the forecast results and observations, unlike AIC which only needs the data to be fitted to the model. Since rolling forecasting can take a significantly greater amount of time than non-rolling forecasting, MAPE and RMSE values from the in-sample non-rolling forecast results were used for parameter selection.

III. RESULTS

The primary goal of this study was to discover a model that could accurately forecast streamflow values in the range of 1 to 5 days for streams located in the southern Appalachian Mountain range with specific attributes described below. The variables that most accurately define a stretch of river used for whitewater recreation are gradient, topography, and size. The Russell Fork Gorge has a gradient of roughly 140 feet per mile(fpm), and the steep mountainous terrain provides a large surface area where rainfall is quickly deposited into narrow streams. Many rivers and creeks with these specific attribues can be found in the mountains of Kentucky, West Virginia, Tenessee, and North Carolina. Other useful applications for streamflow forecasting in these areas are flood control and emergency management. For this purpose, the models with the best performance are highlighted in the results.

Another focus in this study was to analyze model performance and determine which factors outside of hyperparameter values were the most important to take into consideration. The model attributes examined were time interval, data aggregation method, forecasting method, and which metric was most useful for parameter tuning. The results cooresponding to these were collected and are presented in the discussion below.

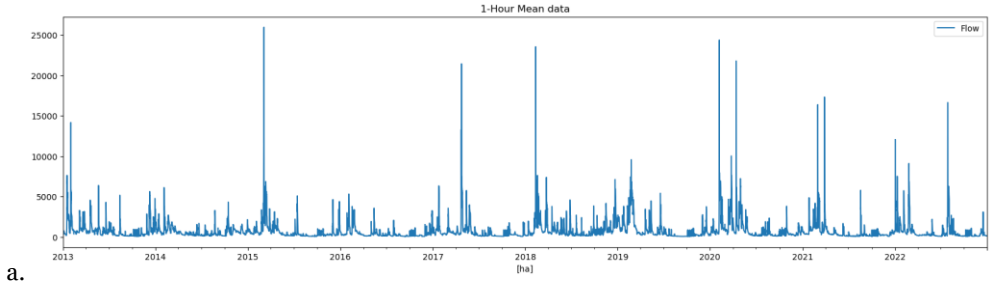
A. Data Collection

Important characteristics to mention about the streamflow data collected from the U.S. Geological Survey (USGS) between January 1, 2013 – December 31, 2022 can be summarized as follows. The streamflow measured in cubic feet per second (cfs) is noticed to have a range from 40cfs – 26,326 cfs, with the mean being 736cfs. With an upper quartile value of 800cfs, and high values for standard deviation and variance, 1150cfs and 1,313,883cfs, it is expected to have outliers that negatively affect model performance, Figure 4. Since these peaks can be explained, and are necessary to include in the model, no outliers were removed. A seasonal trend that repeats yearly can be noticed in the line plots of the 10-year streamflow data,

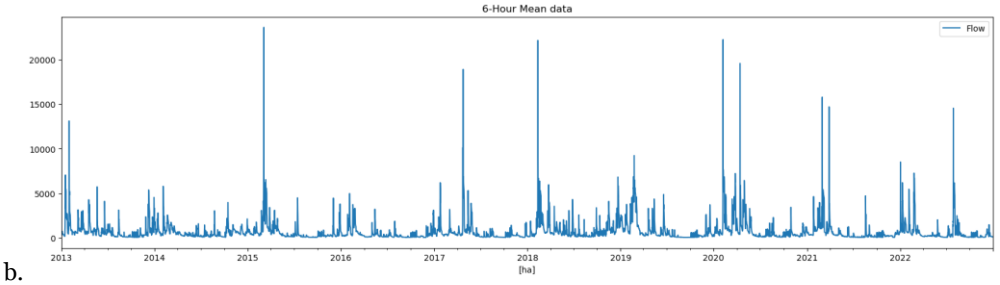
Figure 11, with most streamflow occuring during the months of January-May, August, and October. By cross examining dam release from the John W. Flannagan dam and flow values from the Russell Fork at the Haysi, Virginia observation site, it was noticed that much of the observed flow recorded in the months of August and October can be explained by dam release, and a majority of the recorded flow from January-May was due to natural precipitation. This suggests multiple models may be necessary for more accurate forecasts.

B. Determining Stationarity

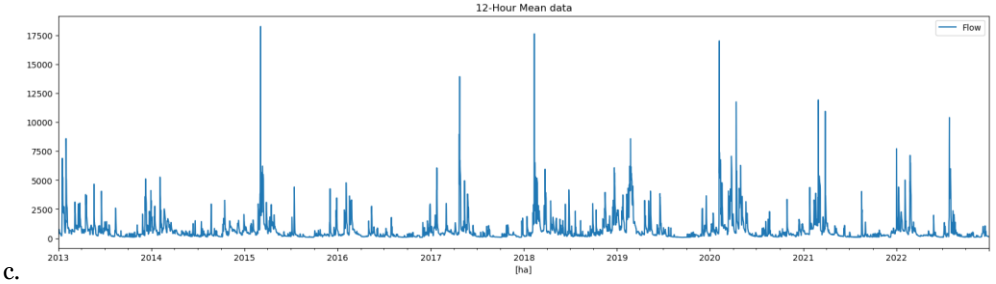
The plots in Figure 11. suggest the mean and variance are consistent throughout the dataset, with the exception of the years ranging from 2018 – 2020. This implies the data could be stationary, but the cyclical nature most apparent in the datasets with a greater interval difference could point towards non-stationarity. The predictability of these cycles are difficult to determine through visual examination of a line plot and other tests were performed.



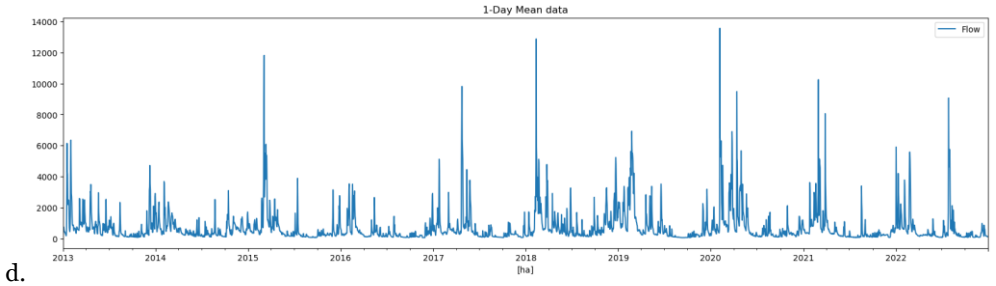
a.



b.

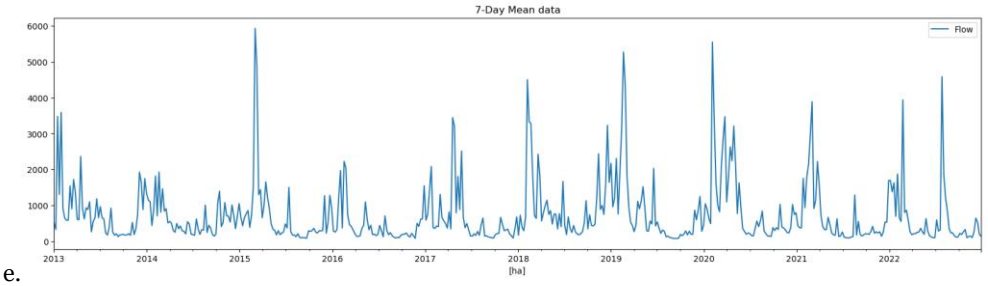


c.

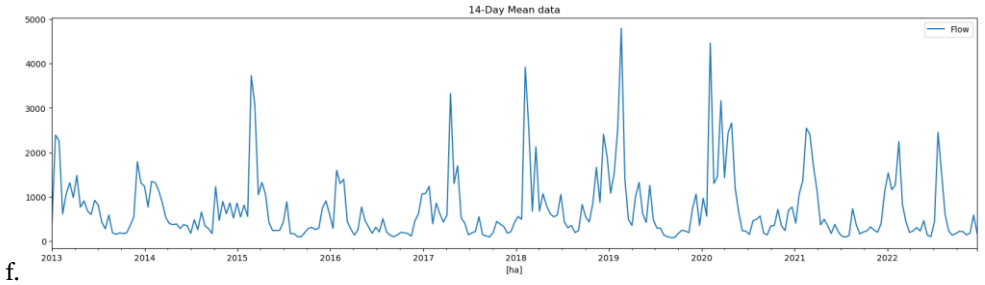


d.

e.



f.



g.

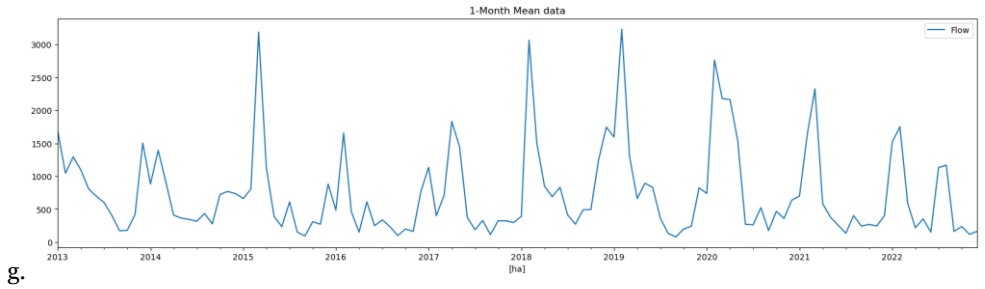
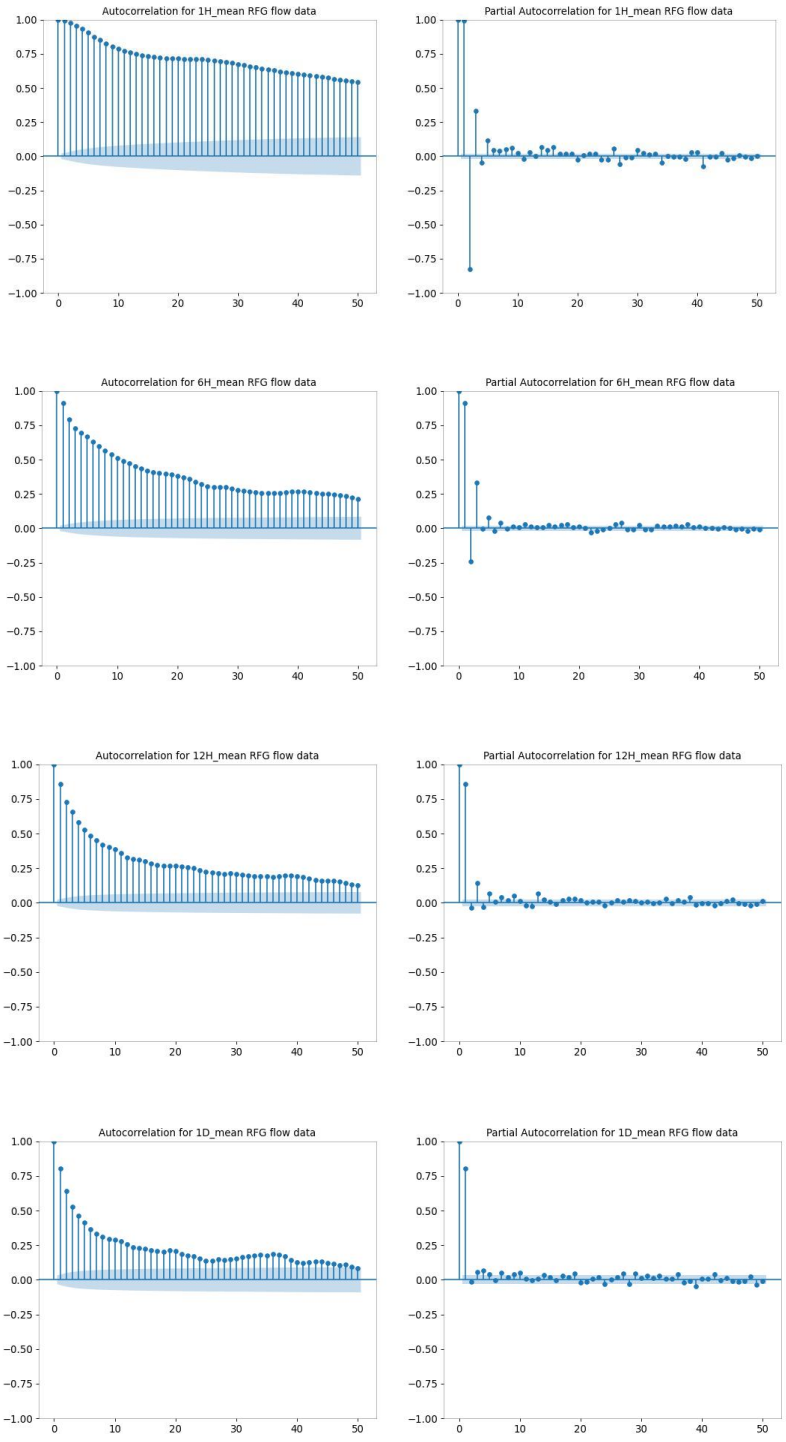


Figure 11. Line plots representing the data at differing time intervals: (a) 1-hour, (b) 6-hour, (c) 12-hour, (d) 1-day, (e) 7-day, (f) 14-day, and (g) 1-month for the period between (01/01/2013 – 12/31/2022).

The ADF unit root test was performed on each dataset with results summarized in Figure 12. The only dataset where the presence of a unit root could be observed, and the null hypothesis could not be rejected, was the 1-month-mean dataset. Autocorrelation plots were then created for each time interval based on the ADF test outcome, and the results from the mean datasets can be seen in Figure 13. The autocorrelation plots for each time series, except for the 1-month dataset, suggest the data is indeed non-stationary. It also gives insight into the seasonality of the data by showing periodic fluctuations with a repetition interval of 1 year for each dataset. This suggests further transformation is needed. The seasonality can also be seen in the decomposition plots of the monthly-mean data with a period set to 12.

Time Interval	p-value	ADF Value	Critical Value	Times Differenced
1H_first	5.18E-26	-13.93	-2.86	0
1H_mean	3.61E-26	-14.02	-2.86	0
1H_max	2.06E-25	-13.59	-2.86	0
6H_first	1.21E-20	-11.32	-2.86	0
6H_mean	1.77E-20	-11.24	-2.86	0
6H_max	7.40E-21	-11.41	-2.86	0
12H_first	1.43E-20	-11.28	-2.86	0
12H_mean	5.85E-20	-11.03	-2.86	0
12H_max	7.37E-23	-12.3	-2.86	0
1D_first	1.04E-20	-11.34	-2.86	0
1D_mean	1.16E-10	-7.33	-2.86	0
1D_max	9.66E-16	-9.33	-2.86	0
7D_first	1.66E-06	-5.55	-2.87	0
7D_mean	1.48E-08	-6.46	-2.87	0
7D_max	8.46E-18	-10.14	-2.87	0
14D_first	6.98E-14	-8.6	-2.87	0
14D_mean	2.62E-15	-9.16	-2.87	0
14D_max	7.56E-09	-6.58	-2.87	0
1M_first	4.19E-18	-10.26	-2.89	0
1M_mean	2.18E-11	-7.62	-2.89	1
1M_max	9.25E-16	-9.33	-2.89	0

Figure 12. Augmented Dickey-Fuller results for each dataset used.



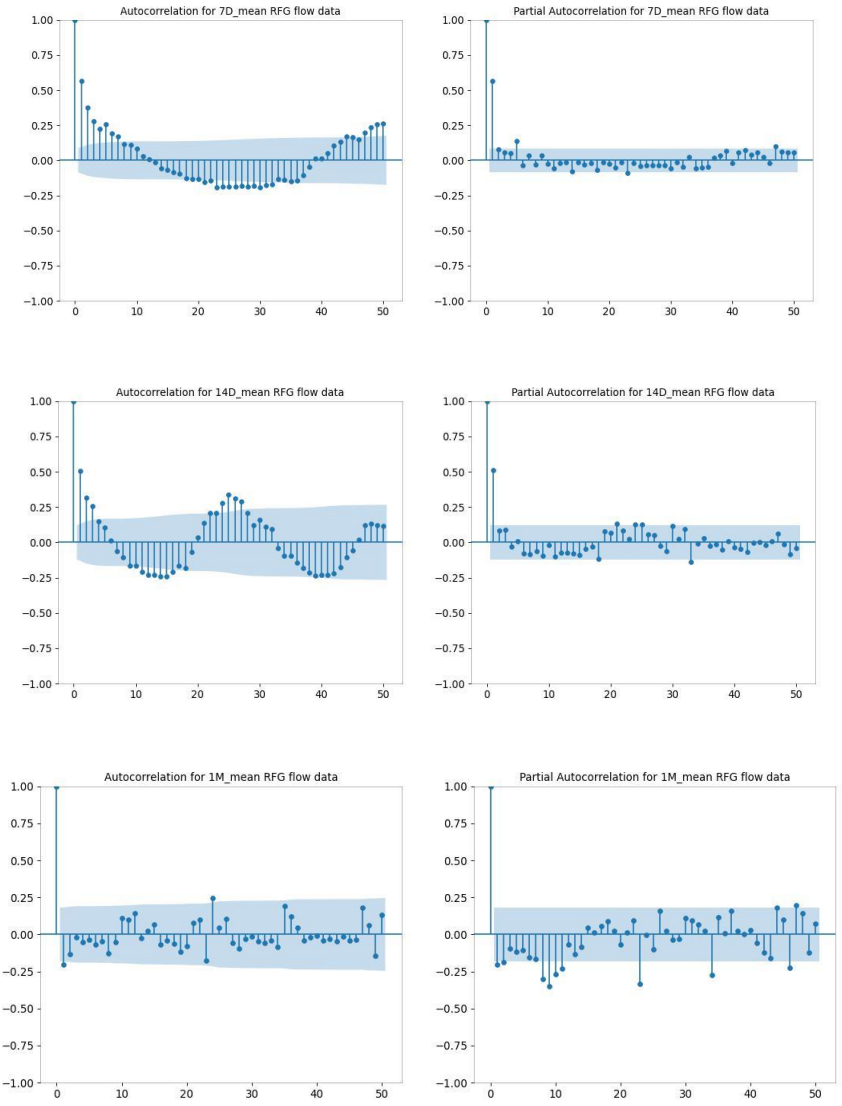


Figure 13. ACF and PACF plots for each time interval aggregated using mean resampling.

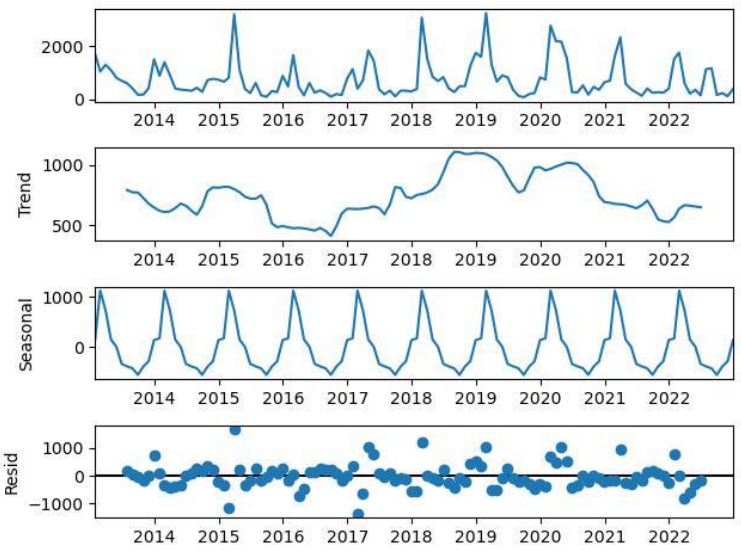


Figure 14. Decomposition plots for the monthly data using mean resampling, period = 12.

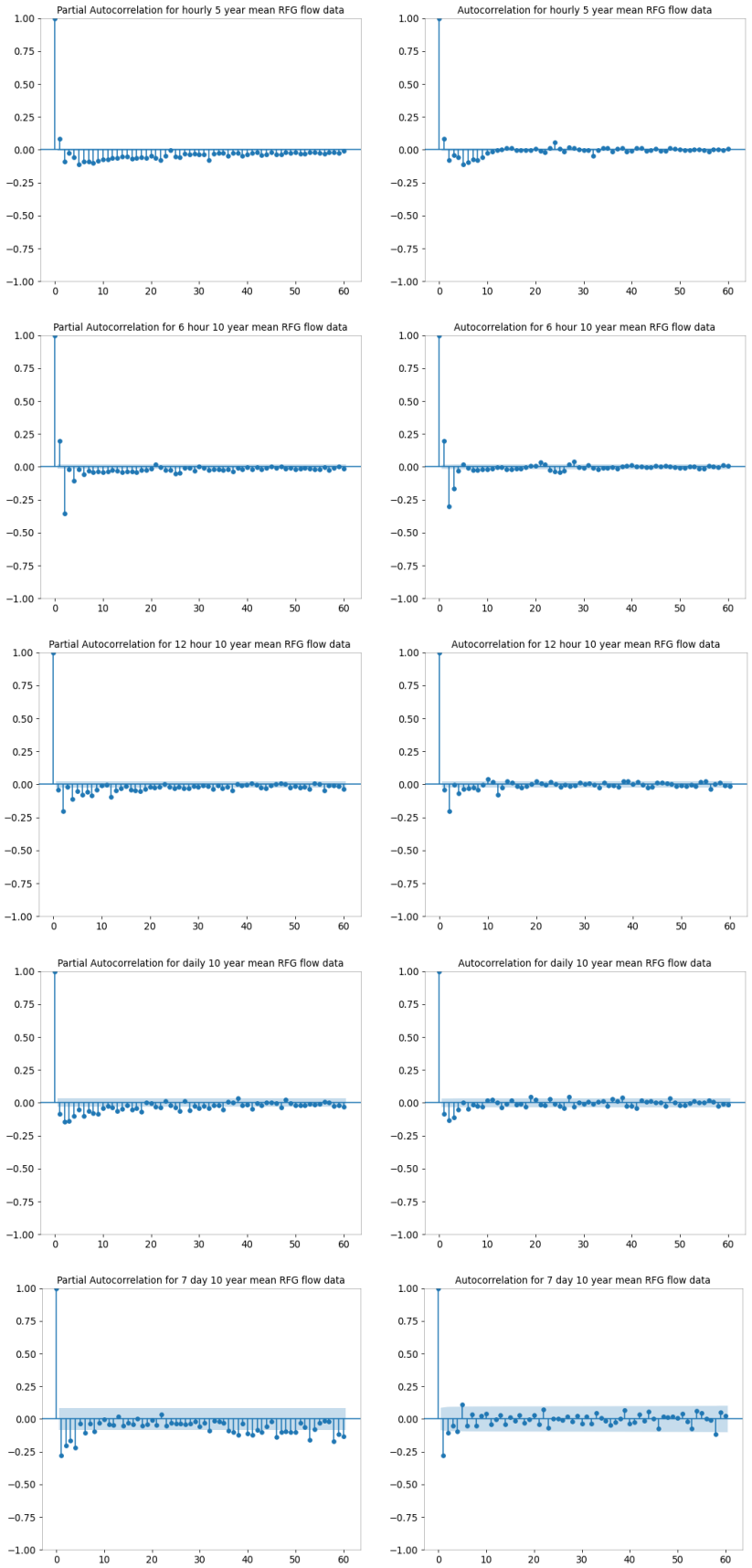


Figure 15. ACF and PACF plots for the 1-hour-mean, 6-hour-mean, 12-hour-mean, 1-day-mean, and 7-day-mean datasets with an order of differencing of 2, 1, 1, 1, 1 respectively.

C. Model Selection

Figure 15. shows ACF and PACF plots with the updated order of differencing. Parameters were selected using both the Box-Jenkins methodology and auto-ARIMA methods, and the best performing models for each dataset are listed in Figure 16. Results from the following forecasting methods are displayed in the tables below; in-sample non-rolling, in-sample rolling, and out-of-sample rolling. It appears from the results that the best performing models are the ones with the smallest interval difference. In order to validate these results, further evaluation was performed on the models listed in Figure 16.

a.

Interval	Model	pdq	AIC	RMSE	MAPE
1H_mean	ARIMA	(3, 1, 2)(0, 0, 0, 0)	111713	64	2
6H_mean	ARIMA	(0, 1, 1)(0, 0, 0, 0)	151654	476	10
12H_mean	ARIMA	(1, 0, 3)(0, 0, 0, 0)	113137	561	15
1D_mean	ARIMA	(1, 1, 0)(0, 0, 0, 0)	57798	663	26
7D_mean	SARIMA	(2, 0, 0)(1, 0, 0, 52)	7512	737	49
14D_mean	SARIMA	(1, 0, 0)(1, 0, 0, 26)	3744	704	48
1M_mean	SARIMA	(1, 0, 0)(1, 0, 3, 12)	1306	606	52

b.

Interval	Model	pdq	Test Set Size	AIC	RMSE	MAPE
1H_mean	ARIMA	(3, 1, 2)(0, 0, 0, 0)	50	111703	0	0
6H_mean	ARIMA	(0, 1, 1)(0, 0, 0, 0)	50	151639	36	2
12H_mean	ARIMA	(1, 0, 3)(0, 0, 0, 0)	50	113122	138	14
1D_mean	ARIMA	(1, 1, 0)(0, 0, 0, 0)	50	57784	143	21
7D_mean	SARIMA	(2, 0, 0)(1, 0, 0, 52)	10	8367	192	61
14D_mean	SARIMA	(1, 0, 0)(1, 0, 0, 26)	10	4154	258	61
1M_mean	SARIMA	(1, 0, 0)(1, 0, 3, 12)	10	1861	504	112

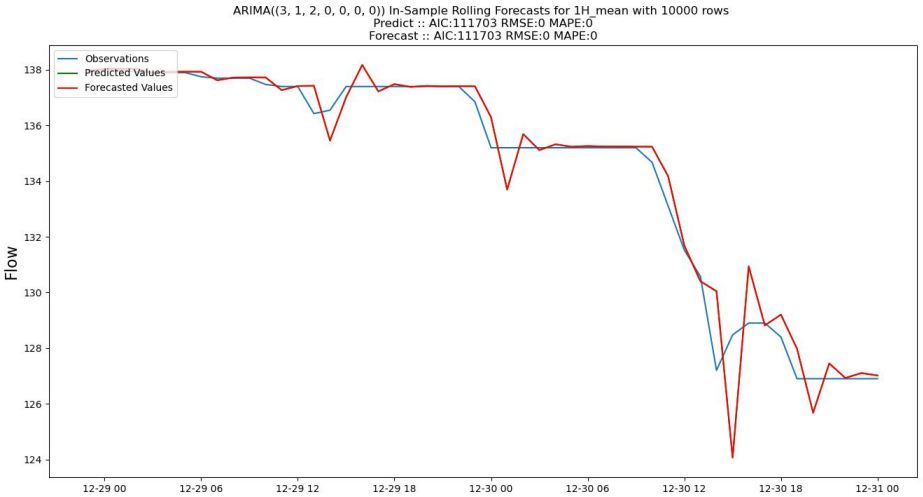
c.

Interval	Model	pdq	Test Set Size	AIC	RMSE	MAPE
1H_mean	ARIMA	(3, 1, 2)(0, 0, 0, 0)	50	111703	5	3
6H_mean	ARIMA	(0, 1, 1)(0, 0, 0, 0)	50	151639	414	244
12H_mean	ARIMA	(1, 0, 3)(0, 0, 0, 0)	50	113119	350	69
1D_mean	ARIMA	(1, 1, 0)(0, 0, 0, 0)	50	57781	289	42
7D_mean	SARIMA	(2, 0, 0)(1, 0, 0, 52)	10	8367	249	47
14D_mean	SARIMA	(1, 0, 0)(1, 0, 0, 26)	10	4153	332	97
1M_mean	SARIMA	(1, 0, 0)(1, 0, 3, 12)	10	1852	589	135

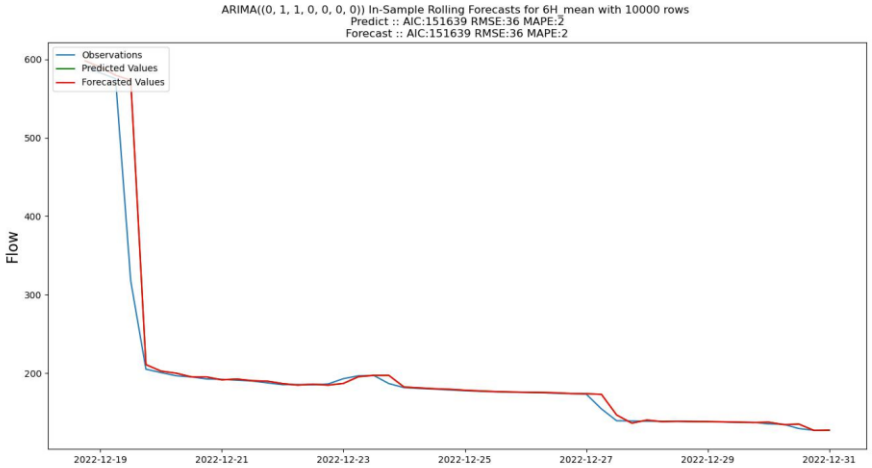
Figure 16. (a) In-Sample Non-Rolling, (b) In-Sample Rolling, and (c) Out-of-Sample Rolling results for the best performing models. The parameters for the 1-hour-mean and 1-day-mean datasets were selected from Box-Jenkins methodology, and parameters for 6-hour-mean, 12-hour-mean, 7-day-mean, 14-day-mean, and 1-month-mean were selected from the auto-ARIMA and auto-SARIMA MAPE methods.

D. Model Evaluation

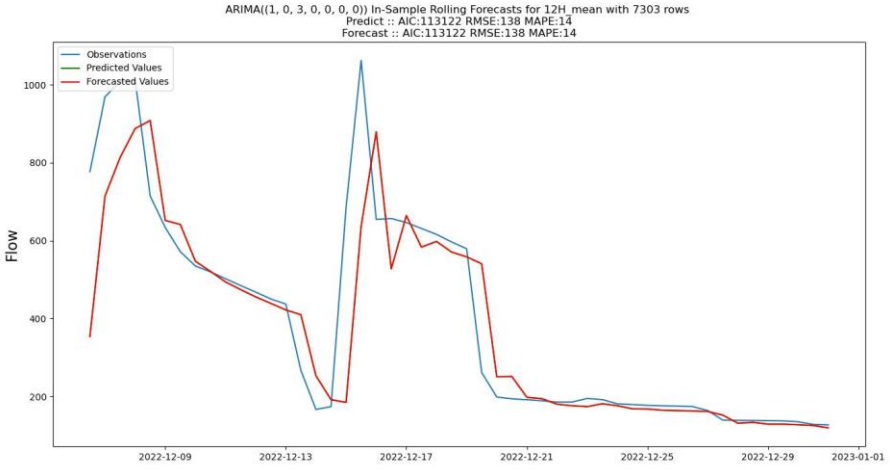
Line plots showing the in-sample 1-step-ahead rolling forecast results for each of the top models are displayed in Figure 17. and Figure 18. As seen in Figure 16., the ARIMA model outperformed SARIMA for the 1-hour, 6-hour, 12-hour, and 1-day datasets, but SARIMA outperformed ARIMA for the 7-day, 14-day, and 1-month datasets. The ARIMA results ranged from good to sufficient according to RMSE and MAPE values, but the 7-day, 14-day, and 1-month SARIMA results were less than ideal for making accurate streamflow predictions. This is reflected in the line plots displayed in Figure 17 and Figure 18. The ARIMA top models were able to capture significant rise and fall, and had good reactiveness to rapid increases and decreases in streamflow. For the 1-hour and 6-hour datasets, the data appears to have a normal distribution. The model appears to react well when there is a sudden rise or fall in streamflow, however when there is a sudden change in direction the model tends to overpredict or underpredict. Also, the 12-hour and 1-day forecasts seem to be skewed right. Since the SARIMA results fail the MAPE evaluation metric, nothing more will be mentioned here.


ARIMA((3, 1, 2, 0, 0, 0, 0)) In-Sample Rolling Forecasts for 1H_mean with 10000 rows
Predict :: AIC:111703 RMSE:0 MAPE:0
Forecast :: AIC:111703 RMSE:0 MAPE:0
Observations
Predicted Values
Forecasted Values
Flow
138
136
134
132
130
128
126
124
12-29 00
12-29 06
12-29 12
12-29 18
12-30 00
12-30 06
12-30 12
12-30 18
12-31 00

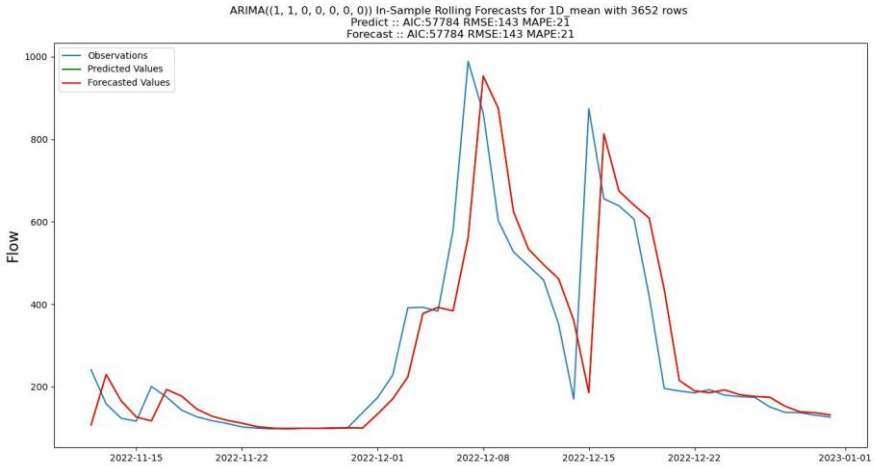
a.


ARIMA((0, 1, 1, 0, 0, 0, 0)) In-Sample Rolling Forecasts for 6H_mean with 10000 rows
Predict :: AIC:151639 RMSE:36 MAPE:2
Forecast :: AIC:151639 RMSE:36 MAPE:2
Observations
Predicted Values
Forecasted Values
Flow
600
500
400
300
200
2022-12-19
2022-12-21
2022-12-23
2022-12-25
2022-12-27
2022-12-29
2022-12-31

b.

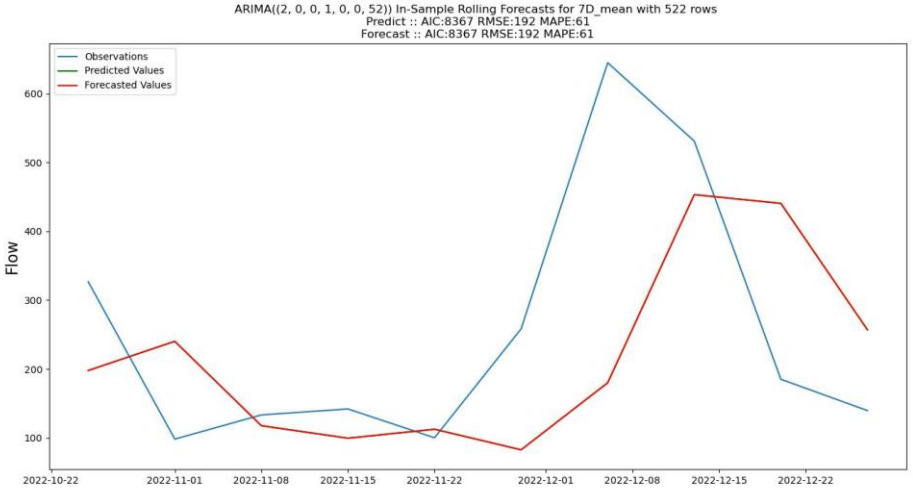

ARIMA((1, 0, 3, 0, 0, 0, 0)) In-Sample Rolling Forecasts for 12H_mean with 7303 rows
Predict :: AIC:113122 RMSE:138 MAPE:14
Forecast :: AIC:113122 RMSE:138 MAPE:14
Observations
Predicted Values
Forecasted Values
Flow
1000
800
600
400
200
2022-12-09
2022-12-13
2022-12-17
2022-12-21
2022-12-25
2022-12-29
2023-01-01

c.

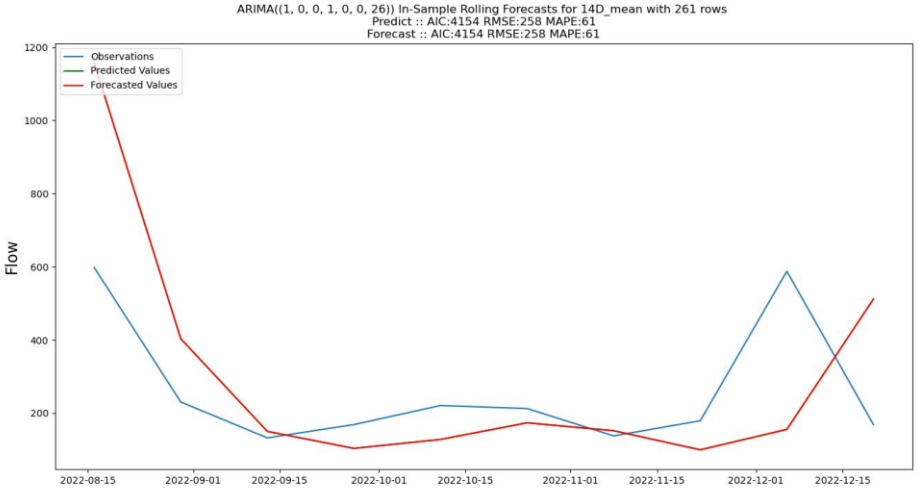


d.

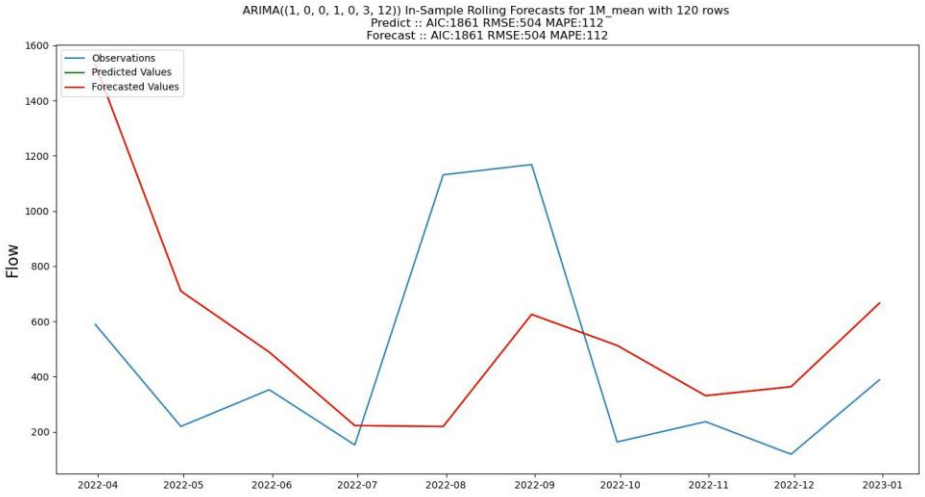
Figure 17. In-Sample 1-step ahead rolling forecasts for (a) 1-hour-mean ARIMA(3,1,2), (b) 6-hour-mean ARIMA(0,1,1), (c) 12-hour-mean ARIMA(1,0,3), and (d) 1-day-mean ARIMA(1,1,0) with a testing set size of 50.



a.



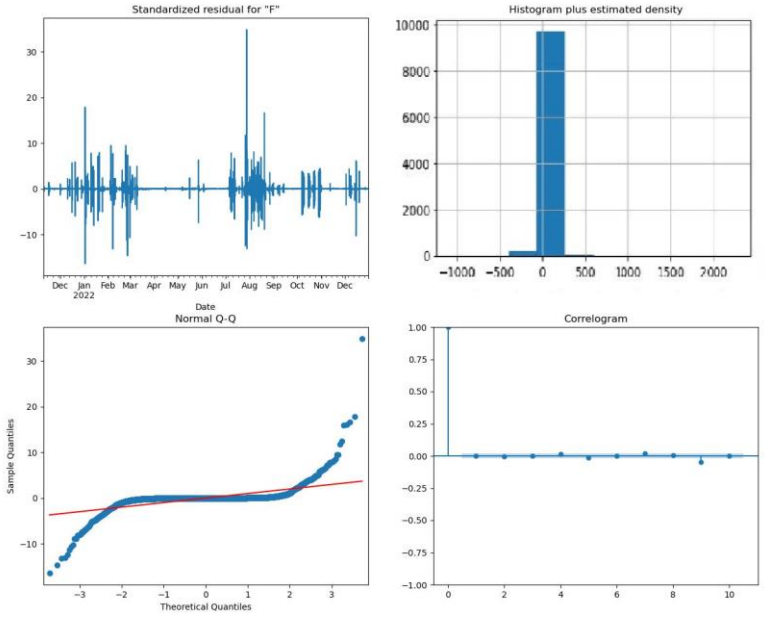
b.



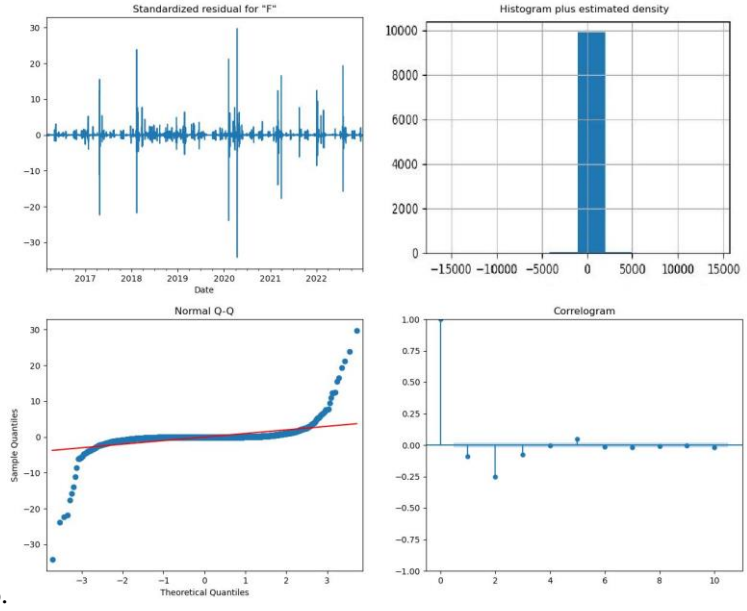
c.

Figure 18. In-Sample 1-step ahead rolling forecasts for (a) 7-day-mean SARIMA(2,0,0)(1,0,0,52), (b) 14-day-mean SARIMA(1,0,0)(1,0,0,26), and (c) 1-month-mean SARIMA(1,0,0)(1,0,3,12) with a testing set size of 10.

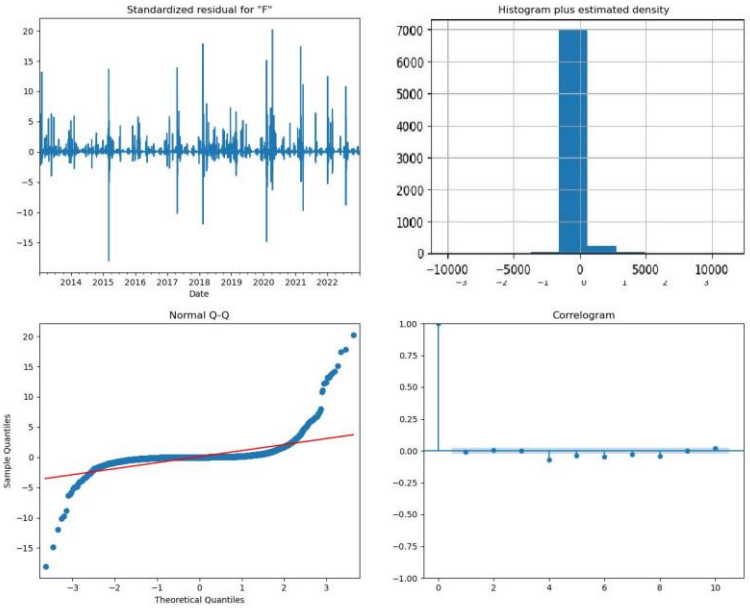
Diagnostic plots for each time series were created using the function plot_diagnostics(), which is included in the Python module statsmodels and compliments the SciPy library (Figure 19),(Figure 20). The resulting output contains 4 plots: standardized residuals, histogram, normal Q-Q, and an ACF plot of residuals. From the standardized plot of residuals it can be seen that the resuduals fluctuate around zero, with a mean very close to zero. This indicates that the forecast results do not appear to be biased. The histogram of the residuals show the data has a normal distribution, although the right tail for the 1-month-mean forecast is long and is trending towards an abnormal distribution. The quantile plots for every dataset are heavy tailed, which means more data was obseved at the extremes than can be found in a normal distribution. This suggests the presence of outliers and further transformation is possibly needed. Outside of the heavy tails, the quantile plot looks to be normal. Lastly, by examining the correlogram, it appears that the residuals have little correlation, which means there wasn't much informaiton that could not be explained by the forecasts.



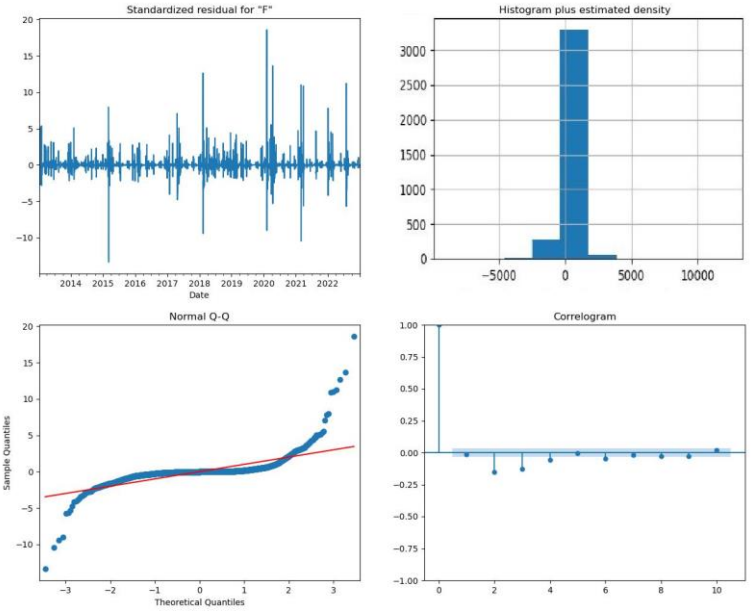
a.


Standardized residual for "F"
Histogram plus estimated density
Normal Q-Q
Correlogram
Date
Sample Quantiles
Theoretical Quantiles

b.

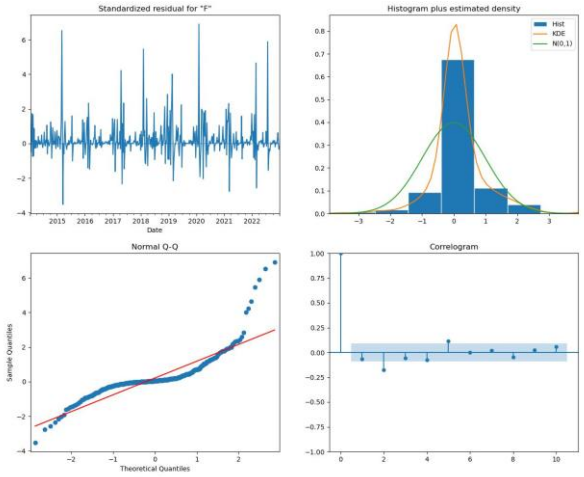

Standardized residual for "F"
Histogram plus estimated density
Normal Q-Q
Correlogram
Date
Sample Quantiles
Theoretical Quantiles

c.

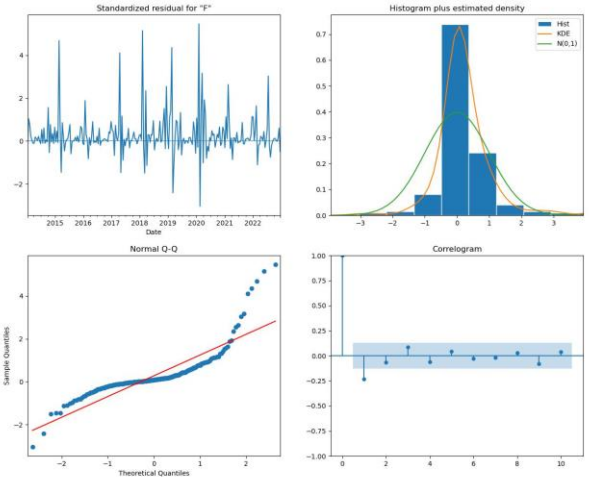

Standardized residual for "F"
Histogram plus estimated density
Normal Q-Q
Correlogram
Date
Sample Quantiles
Theoretical Quantiles

d.

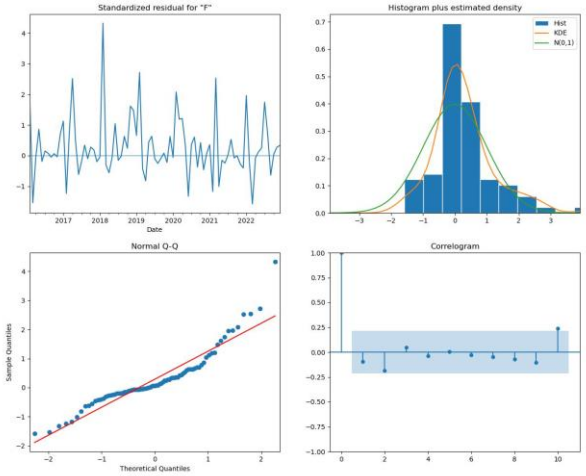
Figure 19. Plot of the residuals, histrogram, Q-Q plot, and correlogram for (a) 1-hour-mean ARIMA(3,1,2), (b) 6-hour-mean ARIMA(0,1,1), (c) 12-hour-mean ARIMA(1,0,3), and (d) 1-day-mean ARIMA(1,1,0) datasets.



a.



b.



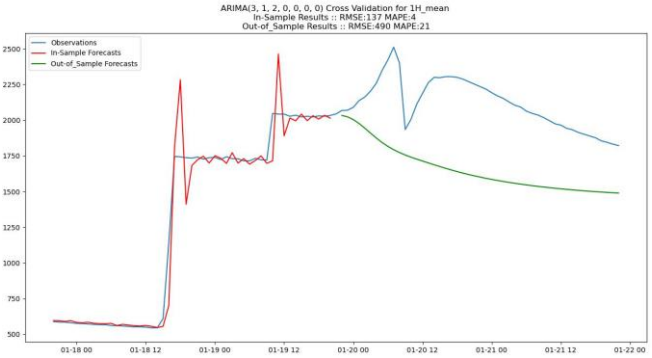
c.

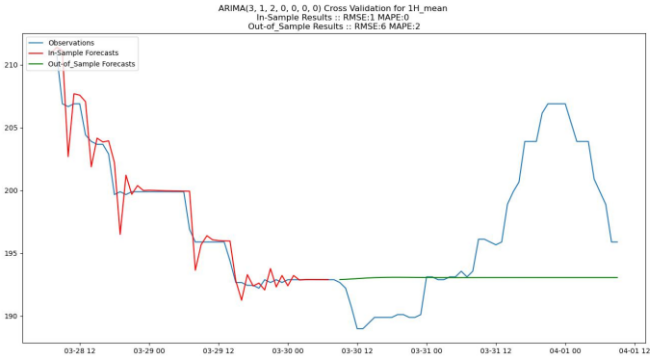
Figure 20. Plot of the residuals, histrogram, Q-Q plot, and correlogram for (a) 7-day-mean SARIMA(2,0,0)(1,0,0,52), (b) 14-day-mean SARIMA(1,0,0)(1,0,0,26), and (c) 1-month-mean SARIMA(1,0,0)(1,0,3,12) datasets.

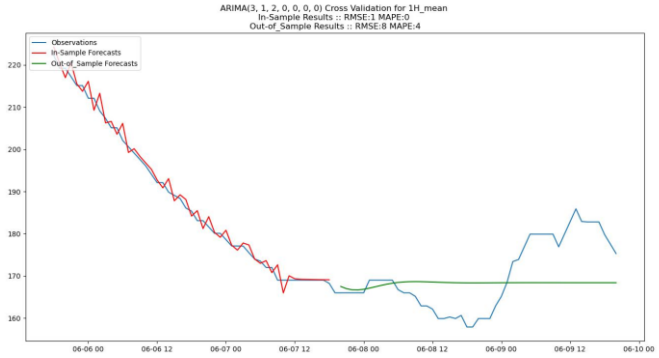
Both in-sample and out-of-sample rolling cross validation were performed on the data, and a summary of the results are listed in Figure 21. The expanding window validation method was used for the in-sample forecasts, and an expanding window with a gap method was used for the out-of-sample forecasts. The testing size for in-sample 1-step ahead was 50, and for out-of-sample the testing size was 10. Again, the 1-hour-mean, 6-hour-mean, 12-hour-mean, and 1-day-mean datasets performed well with the ARIMA model, and the 7-day, 14-day, and 1-month mean datasets had some acceptable results but overall failed the evaluation metrics. The plots from the 1-hour-mean ARIMA(3,1,2) model can be seen in Figure 22. The number of cross validaiton groups were set to 5 in the configuration, and each of the 5 groups are displayed here. The average MAPE score for the 5 1-hour-mean groups was 4%, indicating good prediction performance.

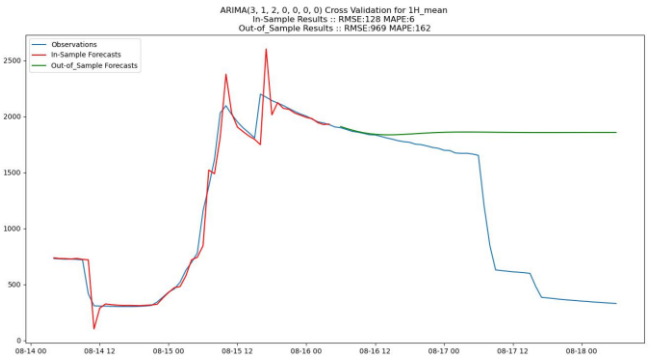
Interval	Group	pdq	Test Set Size	RMSE In-Sample	MAPE In-Sample	RMSE Out-of-Sample	MAPE Out-of-Sample
1H_mean	1	(3, 1, 2, 0, 0, 0, 0)	50	137	4	490	21
1H_mean	2	(3, 1, 2, 0, 0, 0, 0)	50	1	0	6	2
1H_mean	3	(3, 1, 2, 0, 0, 0, 0)	50	1	0	8	4
1H_mean	4	(3, 1, 2, 0, 0, 0, 0)	50	128	6	969	162
1H_mean	5	(3, 1, 2, 0, 0, 0, 0)	50	83	10	21	8
6H_mean	1	(0, 1, 1, 0, 0, 0, 0)	50	2248	18	234	23
6H_mean	2	(0, 1, 1, 0, 0, 0, 0)	50	292	20	1435	53
6H_mean	3	(0, 1, 1, 0, 0, 0, 0)	50	114	19	51	12
6H_mean	4	(0, 1, 1, 0, 0, 0, 0)	50	34	5	315	169
6H_mean	5	(0, 1, 1, 0, 0, 0, 0)	50	16	3	61	31
12H_mean	1	(1, 0, 3, 0, 0, 0, 0)	50	106	13	1060	79
12H_mean	2	(1, 0, 3, 0, 0, 0, 0)	50	290	16	262	55
12H_mean	3	(1, 0, 3, 0, 0, 0, 0)	50	184	14	3893	83
12H_mean	4	(1, 0, 3, 0, 0, 0, 0)	50	5	6	183	72
12H_mean	5	(1, 0, 3, 0, 0, 0, 0)	50	99	9	364	67
1D_mean	1	(1, 1, 0, 0, 0, 0, 0)	50	380	29	382	38
1D_mean	2	(1, 1, 0, 0, 0, 0, 0)	50	348	27	336	40
1D_mean	3	(1, 1, 0, 0, 0, 0, 0)	50	1808	32	1279	96
1D_mean	4	(1, 1, 0, 0, 0, 0, 0)	50	75	17	453	103
1D_mean	5	(1, 1, 0, 0, 0, 0, 0)	50	177	23	83	24
7D_mean	1	(2, 0, 0, 1, 0, 0, 52)	10	551	69	311	35
7D_mean	2	(2, 0, 0, 1, 0, 0, 52)	10	327	42	146	43
7D_mean	3	(2, 0, 0, 1, 0, 0, 52)	10	1477	54	847	61
7D_mean	4	(2, 0, 0, 1, 0, 0, 52)	10	70	25	607	60
7D_mean	5	(2, 0, 0, 1, 0, 0, 52)	10	219	64	446	56
14D_mean	1	(1, 0, 0, 1, 0, 0, 26)	10	496	61	1503	55
14D_mean	2	(1, 0, 0, 1, 0, 0, 26)	10	157	35	616	70
14D_mean	3	(1, 0, 0, 1, 0, 0, 26)	10	1408	73	446	68
14D_mean	4	(1, 0, 0, 1, 0, 0, 26)	10	416	58	1953	74
14D_mean	5	(1, 0, 0, 1, 0, 0, 26)	10	235	33	708	61
1M_mean	1	(1, 0, 0, 1, 0, 3, 12)	10	1094	55	1111	270
1M_mean	2	(1, 0, 0, 1, 0, 3, 12)	10	302	71	472	66
1M_mean	3	(1, 0, 0, 1, 0, 3, 12)	10	961	37	1239	54
1M_mean	4	(1, 0, 0, 1, 0, 3, 12)	10	800	61	613	61
1M_mean	5	(1, 0, 0, 1, 0, 3, 12)	10	356	55	876	240

Figure 21. In-Sample 1-step ahead and out-of-sample rolling cross validaiton results for the top performing models for each dataset.


ARIMA(3, 1, 2, 0, 0, 0) Cross Validation for 1H_mean
In-Sample Results :: RMSE:137 MAPE:4
Out-of_Sample Results :: RMSE:490 MAPE:21
Observations
In-Sample Forecasts
Out-of_Sample Forecasts


ARIMA(3, 1, 2, 0, 0, 0) Cross Validation for 1H_mean
In-Sample Results :: RMSE:1 MAPE:0
Out-of_Sample Results :: RMSE:6 MAPE:2
Observations
In-Sample Forecasts
Out-of_Sample Forecasts


ARIMA(3, 1, 2, 0, 0, 0) Cross Validation for 1H_mean
In-Sample Results :: RMSE:1 MAPE:0
Out-of_Sample Results :: RMSE:8 MAPE:4
Observations
In-Sample Forecasts
Out-of_Sample Forecasts


ARIMA(3, 1, 2, 0, 0, 0) Cross Validation for 1H_mean
In-Sample Results :: RMSE:128 MAPE:6
Out-of_Sample Results :: RMSE:969 MAPE:162
Observations
In-Sample Forecasts
Out-of_Sample Forecasts

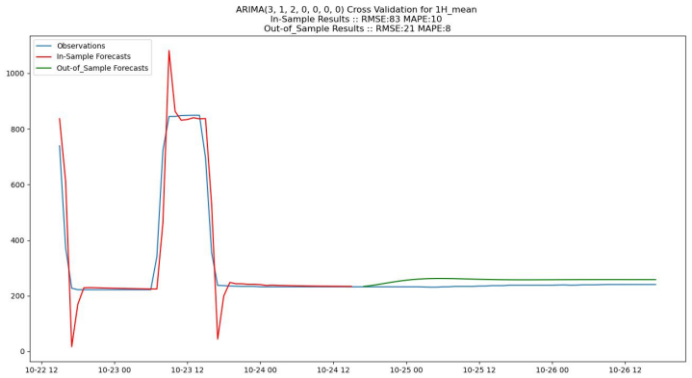


Figure 22. In-Sample 1-step ahead and out-of-sample rolling cross validaiton plots for each of the 5 groups for the 1-hour-mean ARIMA(3,1,2) model.

IV. DISCUSSION

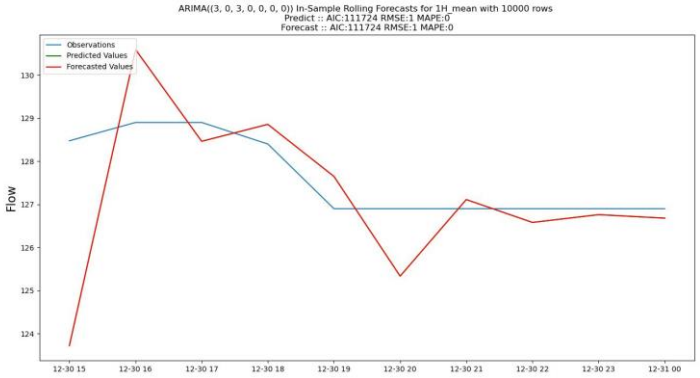
As presented in the results above, the Autoregressive Integrated Moving Average (ARIMA) model performs well when forecasting future streamflow values for the Russell Fork Gorge; however, SARIMA should not be used to predict streamflow for the Russell Fork without further analysis. As previously mentioned, removing outliers or transforming the data with scaling or log transformation before fitting the data to the SARIMA model could be beneficial and will be analyzed in future studies. With that said, the time complexity of running an ARIMA model is significantly less than using the SARIMA model. If both models were to have comparable forecasts, then the obvious choice would be to use the ARIMA model. Another thing to note is the primary metrics used were AIC, RMSE, and MAPE, where MAPE proved to be the best metric for determining order parameters. AIC and RMSE were useful in comparing one model from another but were not great for determining overall model performance. This could be caused by the presence of extreme values since RMSE penalizes outliers more heavily than MAPE.

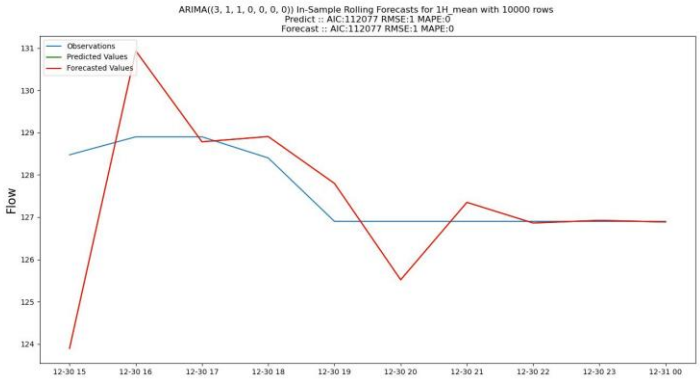
The secondary question presented in this study was which attributes had the greatest impact on model performance and which of those attributes gave the best results. The attributes included were: time interval (1-hour, 6-hour, 12-hour, 1-day, 7-day, 14-day, and 1-month), model comparison using (MAPE, AIC, and RMSE), and aggregation method utilizing the first(), mean(), and max() functions from the Pandas library. The effects of time interval on forecast results can be clearly seen in the results section, where increasing time interval decreased performance with both the ARIMA and SAIMA models.

One way to determine model parameters is to run a program that iterates through each parameter combination in a predetermined range. There are various ways to estimate how well the models will perform. The evaluation metrics used for parameter selection were AIC, RMSE, and MAPE from the in-sample non-rolling forecast results. Once the values were obtained, they were then sorted by AIC, RMSE, and MAPE, with the results stored into files. Figure 23 presents the results from the 1-hour-mean and 1-day-mean datasets after running in-sample rolling, out-of-sample rolling, and cross validation using the parameters selected by each sorting method. The forecasts using AIC, RMSE, and MAPE for the 1-hour-mean dataset have a very similar performance, but by analyzing the plots for ARIMA(3,0,3) and ARIMA(3,0,1), it appears the parameters selected from MAPE values performed slightly better than with AIC or RMSE. Parameter selection using MAPE had a more noticeable advantage with the 1-day-mean dataset. MAPE values determined by averaging the cross-validation results were significantly better and close examination of the in-sample and out-of-sample rolling forecast plots reveals

slightly better results. Since the results were similar it may be beneficial to analyze model performance with each metric, but knowing that MAPE appears to give the best overall results is useful knowledge.

1-hour-mean ARIMA			In-Sample Non-Rolling			In-Sample Rolling		CV-Average	
Result	Auto-Method	pdq	AIC	RMSE	MAPE	Test Set Size	AIC2	RMSE3	MAPE4
1	AIC	(3, 0, 3)(0, 0, 0, 0)	111688	64	2	10	111724	48	8
2	AIC	(2, 1, 3)(0, 0, 0, 0)	111701	64	2				
3	AIC	(3, 0, 2)(0, 0, 0, 0)	111705	64	2				
1	MAPE	(3, 1, 1)(0, 0, 0, 0)	112056	65	1	10	112077	49	8
2	MAPE	(1, 1, 3)(0, 0, 0, 0)	112058	65	1				
3	MAPE	(3, 1, 0)(0, 0, 0, 0)	112064	65	1				
1	RMSE	(3, 0, 3)(0, 0, 0, 0)	111688	64	2	10	111724	48	8
2	RMSE	(2, 1, 3)(0, 0, 0, 0)	111701	64	2				
3	RMSE	(3, 0, 2)(0, 0, 0, 0)	111705	64	2				





1-day-mean ARIMA			In-Sample Non-Rolling			In-Sample Rolling		CV-Average	
Result	Auto-Method	pdq	AIC	RMSE	MAPE	Test Set Size	AIC2	RMSE3	MAPE4
1	AIC	(1, 2, 3)(0, 0, 0, 0)	57381	632	35	10	57428	114	28
2	AIC	(3, 1, 3)(0, 0, 0, 0)	57382	629	38				
3	AIC	(2, 1, 3)(0, 0, 0, 0	57384	630	38				
1	MAPE	(1, 0, 2)(0, 0, 0, 0)	57518	639	23	10	57550	115	19
2	MAPE	(3, 0, 0)(0, 0, 0, 0)	57533	641	23				
3	MAPE	(1, 0, 1)(0, 0, 0, 0	57574	643	23				
1	RMSE	(2, 0, 3)(0, 0, 0, 0)	57390	629	32	10	57437	108	23
2	RMSE	(3, 0, 3)(0, 0, 0, 0)	57392	629	32				
3	RMSE	(2, 0, 2)(0, 0, 0, 0	57404	629	32				

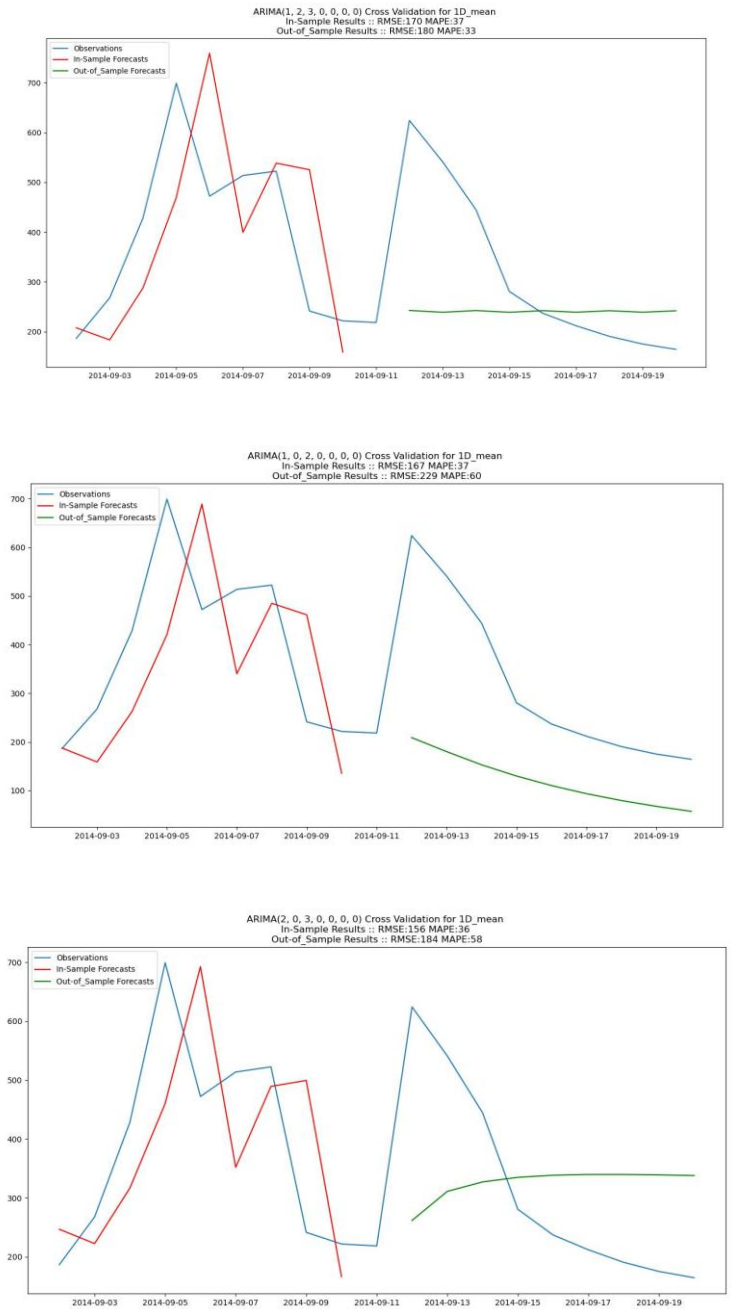


Figure 23. Tables comparing in-sample non-rolling and in-sample rolling forecast results using the top parameters selected by running auto-ARIMA then sorting based on evaluation metric. The top 3 auto-ARIMA results for AIC, MAPE, and RMSE sorting methods are displayed along with in-sample non-rolling results. In-sample rolling results are displayed for the top model in each metric group where RMSE and MAPE values are averaged from the 5 cross validation groups. In-sample rolling forecast result plots are diplayed for the 1-hour-mean top models, AIC/RMSE: ARIMA(3,0,3) and MAPE: ARIMA(3,1,1), and the 1-day-mean top models, AIC: ARIMA(1,2,3), MAPE: ARIMA(1,0,2), and RMSE: ARIMA(2,0,3).

By examining the results listed in Figure 24, it appears ARIMA performs better when the raw data used to fit the model is aggregated by taking the mean of every datapoint in the interval range. Also, the first and maximum aggregation methods produced usable, but inferior results. Another attribute that was analyzed was the use of the statsmodel SARIMAResults predict() function compared to the forecast() function. The predict() function is designed for in-sample and out-of-sample

predictions and forecasting, but the forecast() function is designed solely for out-of-sample forecasting. Both functions were used in the in-sample rolling and out-of-sample rolling forecasts and there were no noticable differences in results.

Interval	Model	pdq	Test Set Size	AIC	RMSE	MAPE
1H_first	ARIMA	(3,0,3)	50	596638	1	0
1H_mean	ARIMA	(3,1,1)	50	572661	0	0
1H_max	ARIMA	(3,0,3)	50	592589	1	0
6H_first	ARIMA	(0,2,2)	50	225402	51	4
6H_mean	ARIMA	(0,1,1)	50	219445	36	2
6H_max	ARIMA	(0,2,1)	50	225926	51	4
12H_first	ARIMA	(0,1,0)	50	119044	626	22
12H_mean	ARIMA	(1,0,3)	50	113122	138	14
12H_max	ARIMA	(0,1,1)	50	120117	626	22
1D_first	ARIMA	(0,1,0)	50	60557	143	20
1D_mean	ARIMA	(1,0,2)	50	57504	141	19
1D_max	ARIMA	(3,0,0)	50	62375	559	31

Figure 24. Table comparing aggregation method. The parameters used were obtained from running auto-ARIMA and sorting the results based on MAPE values for each dataset.

One last comparison to make is the difference in forecasts from the Box-Jenkins method and the auto-ARIMA parameter estimation method. The results can be viewed in Figure 25. for each time interval aggragated using the mean() function. The top performing results are highlighted in bold and include order values determined by each methodology. This shows the importance of being able to determine ARIMA and SAIMA parameter order by analyzing ACF, PACF, and ADF results and then using evaluation metrics to refine the parameters. Once this step is completed, the results from Box-Jenkins can be used to narrow the range of parameter value combinations used by the iterative auto-ARIMA and auto-SARIMA methods. Due to the time complexity of running ARIMA and SARIMA, simply iterating through every possible order combination can be impractical or impossible depending on certain data characteristics, thus making this process advantageous for any time series analysis using these models. It may be advantageous to introduce parallelism through multiprocessing to increase efficiency of training and fitting the models, especially for SARIMA.

Interval	Model	pdq	Methodology	AIC-ISNR	RMSE-ISNR	MAPE-ISNR	Test Set Size-ISR	AIC-ISR	RMSE Forecast-ISR	RMSE Predict-ISR	MAPE Forecast-ISR	MAPE Predict-ISR
1H_mean	ARIMA	(3, 0, 0)(0, 0, 0, 0)	ACF/PACF/ADF	111957	65	2	10	111984	1	1	0	0
1H_mean	**ARIMA**	**(3, 1, 2)(0, 0, 0, 0)**	**Box-Jenkins**	**111713**	**64**	**2**	**10**	**111733**	**1**	**1**	**0**	**0**
1H_mean	ARIMA	(3, 1, 1)(0, 0, 0, 0)	Auto-MAPE	112056	65	1	10	112077	1	1	0	0
1H_mean	ARIMA	(3, 0, 3, 0, 0, 0, 0)	Auto-AIC	111688	64	2	10	111724	1	1	0	0
1H_mean	ARIMA	(3, 0, 3, 0, 0, 0, 0)	Auto-RMSE	111688	64	2	10	111724	1	1	0	0
6H_mean	ARIMA	(3, 0, 0)(0, 0, 0, 0)	ACF/PACF/ADF	150440	448	11	10	150474	6	6	4	4
6H_mean	ARIMA	(3, 1, 0)(0, 0, 0, 0)	Box-Jenkins	150649	453	11	10	150679	2	2	0	0
6H_mean	**ARIMA**	**(0, 1, 1)(0, 0, 0, 0)**	**Auto-MAPE**	**151654**	**476**	**10**	**10**	**151670**	**1**	**1**	**0**	**0**
6H_mean	ARIMA	(3, 1, 3, 0, 0, 0, 0)	Auto-AIC	150237	444	16	10	150277	14	14	10	10
6H_mean	ARIMA	(3, 1, 3, 0, 0, 0, 0)	Auto-RMSE	150237	444	16	10	150277	14	14	10	10
12H_mean	ARIMA	(1, 0, 0)(0, 0, 0, 0)	ACF/PACF/ADF	113416	570	16	10	113418	11	11	7	7
12H_mean	ARIMA	(3, 1, 0)(0, 0, 0, 0)	Box-Jenkins	113418	572	16	10	113448	9	9	4	4
12H_mean	**ARIMA**	**(1, 0, 3)(0, 0, 0, 0)**	**Auto-MAPE**	**113137**	**561**	**15**	**10**	**113183**	**8**	**8**	**5**	**5**
12H_mean	ARIMA	(2, 1, 3, 0, 0, 0, 0)	Auto-AIC	112982	555	25	10	113022	31	31	21	21
12H_mean	ARIMA	(3, 0, 3, 0, 0, 0, 0)	Auto-RMSE	112986	555	21	10	113032	26	26	17	17
1D_mean	ARIMA	(1, 0, 0)(0, 0, 0, 0)	ACF/PACF/ADF	57589	643	23	10	57590	17	17	9	9
1D_mean	**ARIMA**	**(1, 1, 0)(0, 0, 0, 0)**	**Box-Jenkins**	**57798**	**663**	**26**	**10**	**57799**	**10**	**10**	**5**	**5**
1D_mean	ARIMA	(1, 0, 2)(0, 0, 0, 0)	Auto-MAPE	57518	639	23	10	57550	11	11	5	5
1D_mean	ARIMA	(1, 2, 3, 0, 0, 0, 0)	Auto-AIC	57381	632	35	10	57428	42	42	27	27
1D_mean	ARIMA	(2, 0, 3, 0, 0, 0, 0)	Auto-RMSE	57390	629	32	10	57437	36	36	23	23
7D_mean	SARIMA	(1, 0, 0)(0, 1, 1, 52)	ACF/PACF/ADF	6711	757	76	10	7541	351	351	177	177
7D_mean	SARIMA	(1, 1, 3)(1, 0, 1, 52)	Box-Jenkins	7445	714	77	10	8323	229	229	122	122
7D_mean	**SARIMA**	**(2, 0, 0)(1, 0, 0, 52)**	**Auto-MAPE**	**7512**	**737**	**49**	**10**	**8367**	**192**	**192**	**61**	**61**
7D_mean	SARIMA	(1, 1, 2, 1, 1, 2, 52)	Auto-AIC	5818	763	88	10	5803	364	364	192	192
7D_mean	SARIMA	(2, 1, 2, 1, 0, 2, 52)	Auto-RMSE	6630	698	91	10	6615	210	210	110	110
14D_mean	SARIMA	(1, 0, 3)(0, 1, 1, 26)	ACF/PACF/ADF	3262	669	72	10	3721	745778	745779	131638	131638
14D_mean	SARIMA	(1, 1, 3)(1, 0, 2, 26)	Box-Jenkins	3243	660	91	10	4104	269	269	130	130
14D_mean	**SARIMA**	**(1, 0, 0)(1, 0, 0, 26)**	**Auto-MAPE**	**3744**	**704**	**48**	**10**	**4154**	**258**	**258**	**61**	**61**
14D_mean	SARIMA	(0, 1, 2, 1, 2, 3, 26)	Auto-AIC	2076	815	98	10	2063	283	283	107	107
14D_mean	SARIMA	(1, 0, 1, 3, 0, 3, 26)	Auto-RMSE	2880	632	63	10	2865	215	215	95	95
1M_mean	SARIMA	(1, 1, 1)(1, 1, 1, 12)	ACF/PACF/ADF	1446	605	75	10	1658	578	578	135	135
1M_mean	SARIMA	(2, 2, 1)(1, 0, 1, 12)	Box-Jenkins	1653	680	83	10	1851	545	545	155	155
1M_mean	**SARIMA**	**(1, 0, 0)(1, 0, 3, 12)**	**Auto-MAPE**	**1306**	**606**	**52**	**10**	**1861**	**504**	**504**	**112**	**112**
1M_mean	SARIMA	(0, 1, 2, 1, 2, 3, 12)	Auto-AIC	892	831	101	10	877	655	655	84	84
1M_mean	SARIMA	(2, 0, 2, 3, 0, 3, 12)	Auto-RMSE	1272	567	57	10	1257	499	499	104	104

Figure 25. Table comparing results from using Box-Jenkins methodology and the different methods of running the auto-ARIMA/SARIMA program. Top results are highlighted in bold. ISNR refers to In-Sample Non-Rolling and ISR refers to In-Sample Rolling.

V. CONCLUSION

The findings in this study showed that the Autoregressive Integrated Moving Average model was able to accurately forecast streamflow in the range of 1 hour to several days in advance simply by using historical observations of streamflow. It was proven to be a robust model through residual analysis and cross validation, and thus could be used in a real world scenerio to predict river levels for the Russell Fork Gorge using the order values presented in the results. This could be used for recreational purposes and, with more testing, possibly for flood control or an advanced warning for emergency management. The later is purely speculation and more advanced and accurate systems may already be utilized for those purposes.

It was also found that the datasets with shorter time intervals performed the best, including 1-hour, 6-hour, 12-hour, and 1-day. For the purpose of this study this finding was ideal since the realistic and ideal prediction range for recreational purposes would be from 1-7 days in advance. As for aggregating real world raw data into more usable time intervals, it was concluded that using the mean aggregation method performed the best in nearly all scenarios. Parameter tuning is a very important step when using the Box-Jenkins method to determining optimal order for both ARIMA and SARIMA modeling. Many evaluation metrics are used in this process and one goal of this study was to determine if any of the metrics produced higher performing models. It was determined that MAPE appeared to extract order values that made more accurate forecasts with the datasets used in this study. The other metrics used were AIC and RMSE. If a tie occurred while ranking model performance based on MAPE, AIC, and RMSE values, then a nested sort was used as a tiebreaker with the following sorting order for each respective metric: (MAPE, AIC, RMSE), (AIC, MAPE, RMSE), and (RMSE, MAPE, AIC). For the last comparison between the Box-Jenkins and auto-ARIMA/SARIMA methods it was determined that a majority of the top results were found by iterating through a range of parameters, but only after narrowing the order ranges using results from the Box-Jenkins methodology. In two scenarios further improvement on the model was not achieved by using auto-ARIMA or auto-SARIMA methods. Further studies may reveal that removing outliers could improve SARIMA modeling results.

The Python module and configuration files used in this study can be viewed at the links provided in the references section. Vector Autoregrassion was also used for modeling streamflow for the Russell Fork Gorge but with insufficient results. A link to these results is also provided. The limiting factor for running the VAR model was obtaining reliable and accurate precipitation data. Rain gauges in the region were found to be missing massive amounts of data and were highly inconsistent. USGS has recently added rain gauges at locations adjacent to streamflow monitoring stations but data only begins in November of 2022, which proved to be an insufficient amount of data.