Assessment 1.2: Time Series ARIMA Model parameter selection for Sunspot activity using Genetic Algorithm

Hamza Waheed, ID: 21171402 [* 1]
Word Count: 3210

Abstract

Time series prediction using ARIMA model is an important area of space weather study. The problem with ARIMA model arises when selecting its p, d, q parameters. Different techniques have been developed like Partial Autocorrelation Function (PACF), Autocorrelation Function (ACF), Bayesian Information Criterion (BIC) etc. to calculate these parameters.

This report uses Genetic Algorithm (GA) with three different crossovers: Single Point crossover, Uniform Crossover and Crossover Order to calculate ARIMA parameters using Root Mean Square Error (RMSE) as fitness function.

The fitness results of the model are compared with the ”auto.arima” function of the ”Forecast” package to assess the specification search of GA. For the given problem, GA tuned ARIMA model had higher RMSE values and had introduced more complexity by selecting higher values of (p, d, q) as opposed to the parameters selected by ”auto.arima”.

The results highlight that GA's specification search in parameter tuning of ARIMA may not be suitable for small datasets. The findings of the report require further experiments with different fitness functions & datasets to investigate the results of GA tuned parameters for ARIMA model.

1. Background Literature

Solar activity is classified as low order chaotic event (Spiegel, 1994) that makes it hard to predict (Orfila, A. et al., 2002).

[1]M.Sc. Big Data Analytics, School of Computing and Digital Technology, Birmingham City University, UK. Correspondence to: Hamza Waheed <Hamza.Waheed2@mail.bcu.ac.uk>.

Modern Optimisation Report, CMP 7213 - Modern Optimisation.

In recent years, space weather is an active area of research concerned with studying solar activity & outer space emissions that pose threat to Earth.

Solar activity involves sunspots that are the marks that appear on the sun's surface due to disturbances in the sun's magnetic field. Magnetic fields in these sunspots causes solar flares.

Earth's magnetic field protects against these flares. However, strong ejections can cause disruption to world's electronics i.e. global satellite systems, radars, astronauts, grid systems. (Suparta, 2014) showed the effects of solar flares on electronic components. Therefore, predicting sunspot and finding pattern is an actively sought after area.

Predicting sunspot using past data is widely used throughout history. (Pala & Atici, 2019) used sunspot data from 1749 to 2018 with 12-fold cross validation for forecasting sunspots using deep learning algorithms. (Siami-Namini & Namin, 2018) used Long Short-Term Memory (LSTM), a neural network algorithm, highlighting its superiority over ARIMA. However, the problem with these studies is, before 1818, data was collected on a yearly basis and had discrepancies, as highlighted by (Orfila, A. et al., 2002).

Studies like (Aguirre et al., 2008) used Monte Carlo simulation on the sunspot data constructed by (Wolf, 1852). The transformations were applied to fix the issues with uniformity of data mentioned previously. However, transformations often result in loss of underlying trends.

2. Candidate Optimization Method

This section addresses the approach to optimization method employed for parameter selection of Time Series for sunspot prediction. It starts with brief introduction to Time Series Analysis, rationale behind using ARIMA method and using Genetic Algorithm (GA) for optimization. The section concludes with how they are used together.

2.1. Time Series (TS)

Time Series Analysis is a widely used technique for regression analysis where future forecast is based on the past observations of the same variable. Some widely used time-series models are:

2.1.1. AUTOREGRESSIVE (AR)

In AR, the predictor variable is estimated using the historical observations of the variable itself. The term auto-regression shows regression against itself.

AR model is mathematically represented by (Fuller, 1985) as;

$$y_t^{'} == g(t) + \sum_{i-1}^{p} \alpha_i Y_{t-1} + e_p \tag{1}$$

2.1.2. MOVING AVERAGE (MA)

Moving average (MA) uses past errors with fixed average amount for regression. MA is represented mathematically as equation 2 by (Messias et al., 2016);

$$y_t^{'} = c + e_t + \theta_1 e_{t1} + \theta_2 e_{t2} + ... + \theta_q e_{t-q} \tag{2}$$

y' is the current predicted value, θ is a fixed value, c is the average value of the predictor and e_t is the past error in our estimate with respect to time lag t.

2.1.3. AUTOREGRESSIVE MOVING AVERAGE (ARMA)

ARMA model referred to as ARMA(p, q) combines both AR, MA model such that p is the order of AR(p) and q is the order of MA(q). The models can handle noise, but is susceptible to seasonality. ARMA model assume stationarity (mean = 0, S.D = 1, No Seasonality) which is difficult to achieve in real data.

2.1.4. ARIMA

ARIMA is an extension of ARMA to address the seasonality & non-linearity issue within the data. It does so by introducing small differencing steps called ”d” that slices the time series into smaller parts so that the slices become stationary. The model is defined using (p, d, q) parameters. Mathematically, ARIMA is represented as follows (Ariyo et al., 2014);

$$y_t^{'} = c + \theta_1 y_{t-1} + ... + \theta_p y_{t-p}^{'} + \theta_1 \epsilon_{t-1} + ... + \theta_q \epsilon_{t-q} + \epsilon_t \tag{3}$$

where c is intercept, Y_{t-1} is the previously observed value and ϵ_p is the error with the previous observed value. p and q are the order of AR and MA respectively. ”d” is the number of times required for differencing or slicing the past series to make it stationary. ”d” shows differencing steps over every previous iteration of output. These variables are also called lag variables, as these variables are the previously calculated predictions used for the current iteration.

2.2. Problem Instance

The challenge is to choose the right values for (p, d, q). The weakness of ARIMA model is the difficulty of selecting these parameters (Messias et al., 2016). Each value of p, d, q represents previous range of observations, which are to be taken into account for the current prediction interval. If large values of p, d, q are chosen, then the lag observation to be taken into account would be large for each prediction. This poses serious computational challenge for large dataset. If the parameter values are high, the model becomes overfit and black box. Similarly, simpler models may lose the underlying trends. The objective is to find the tradeoff between simplicity and accuracy. It's also worth mentioning that the ”auto.arima” function of ”forecast” package in R chooses p, d, q values based on AIC, Bayesian Information Criterion (BIC) and it's internal optimization methods. In this study, GA is used to tune the parameters of p, d, q using Root Mean Square Error RMSE and the aim is to compare results with ”auto-arima” function to check the specification search of GA.

The following includes the rationale behind using Genetic Algorithm and its variations.

2.3. Genetic Algorithm (GA)

This report applies Genetic Algorithm (GA) (Holland, 1992) for optimizing (p, d, q) operators for sunspot dataset to reach optimal solution. GA is based on the Darwin's theory of ”Survival of the fittest”, where better suited individuals (solutions) are retained. In GA, the algorithm evolves itself to ensure that better solutions are retained to reach the optimal solution. In contrast to traditional methods like Gradient Descent, Simulated Annealing, etc. GA is useful as what (Leamer, 1978) suggested as ”Specification Search”.

(Beenstock & Szpiro, 2002) highlighted GA as an aid for specification search in time series. The experimental results of (Beenstock & Szpiro, 2002) suggests GA's success in specification estimation. Other studies like (Leung et al., 2003) used GA for parameter tuning of model. GA's global search of candidate solution is ideal for non-linear Time Series data (Gallagher & Sambridge, 1994) (Orfila, A. et al., 2002).

However, (Beenstock & Szpiro, 2002) intimated that specification search of GA may not be much different from traditional methods due to stochastic nature of GA. This claim will be verified by comparing GA results with auto.arima function of ”forecast” package in R.

2.3.1. CANDIDATE SOLUTION

The experiment uses 10-bit chromosome representation of real number variables p, d, q. 4-bits are chosen to represent the maximum lag value of 15 for p and q part. (Bits required to represent a real number; $2^n - 1$ where n = number of bits). The first 4-bits in the chromosome represent the AR(p), middle 2-bits represent differencing (d) to have a maximum differencing of 4 and last 4-bits represent MA(q). The three parts are binary version of real numbers. The bits are selected by GA and are converted into real numbers to feed ARIMA model. The quality of result is calculated using the fitness function, more on that in the next section 2.3.2. The experiment is repeated for 30 iterations with 30 generations. The best way to decide GA parameters is through trial and error. As, the number of generations required to run GA is problem specific. (Lukoseviciute & Ragulskis, 2010).

2.3.2. Fitness Function

The fitness function in GA tells which candidate solution is better suited for the next generation. For this experiment, The fitness function is calculated using the root-mean-square error (RMSE). RMSE is considered as de facto standard for regression analysis. RMSE is useful when lower residual values are preferred and dealing with large error values. RMSE is sensitive to model parameters, hence, minimizing the error function by adjusting the parameters is important (Chai & Draxler, 2014). Other experiments have also used RMSE for result evaluation of Sunspot using time series, e.g. (Gkana & Zachilas, 2015), and (SILSO World Data Center) where the data for this study comes from. (Lukoseviciute & Ragulskis, 2010) used RMSE for non-uniform time lag calculation of fuzzy time series.

RMSE is given as follows (Pala & Atici, 2019);

$$RMSE = \sqrt{\frac{1}{n}\sum_{i}^{n}(e_i)^2} \tag{4}$$

where

$$e = P_i – O_i \tag{5}$$

P_i is the predicted value for the ith observation in the dataset. O_i is the observed value for the ith observation in the dataset, n is the sample size

RMSE measures the residuals or difference between the actual and predicted values. The objective is to minimize the RMSE to reach the optimal solution. The RMSE is calculated by the residuals of the ARIMA model.

2.4. Crossover Operators

The report assesses the performance of following three crossover methods.

2.4.1. Uniform Crossover

In Uniform Crossover, a new offspring of the same length as the parents is created. Each gene of the child is selected randomly from the corresponding parent gene. Uniform crossover is shown as follows;

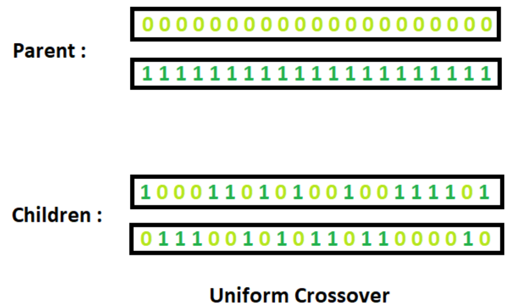

Figure 1. Uniform Crossover

(Picek & Golub, 2010) experiment compared performance of different crossover operators on binary coded GA with different fitness functions. (Picek & Golub, 2010) experimental results showed uniform crossover obtained the best fitness values across various fitness criterion. Uniform crossover is helpful to avoid the illegal offspring problem faced by other crossover methods (geeksforgeeks).

2.4.2. One Point Crossover

In this method, a crossover point is randomly selected in two parents. Genes in the tails of the parents are swapped to generate two offspring, as illustrated in figure (Tutorials Point);

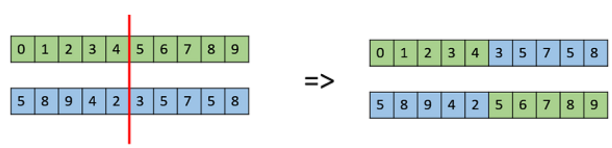

Figure 2. One Point Crossover

One point crossover is the most widely used crossover method (Kora & Yadlapalli, 2017). Many advance crossover techniques incorporate it for its simplicity.

2.4.3. Crossover Order

Crossover Order is a probabilistic crossover method. In this method, a swath of alleles from Parent1 is passed to the child, and the remaining values are passed in the same order as they appear in parent2. The method is illustrated as follows (Rubicite);

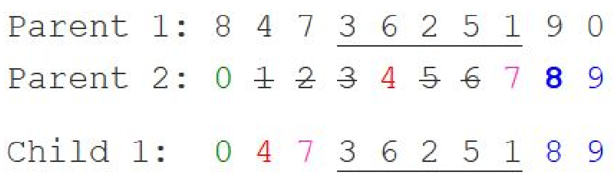

Figure 3. Crossover Order

3. Methodology

The Sunspot data used in the experiment for parameter tuning of ARIMA is obtained from the World Data Center SILSO, Royal Observatory of Belgium, from January 1992 to February 2022 (SILSO World Data Center). The dataset contains mean sunspots, Mean sunspots of Sun's Northern and Southern Hemisphere. Only recent data is taken into account to address the criticism discussed in section 1.

To get a univariate Time Series, the pre-processing was performed to only get monthly mean sunspot numbers. Additionally, month & year columns were merged into "date" column to get a monthly interval for time series. The dataset was split into 80% training set and 20% test set to verify the accuracy of the model. A snapshot of the data used is given in the table 1. The sunspot time series is shown in fig 4.

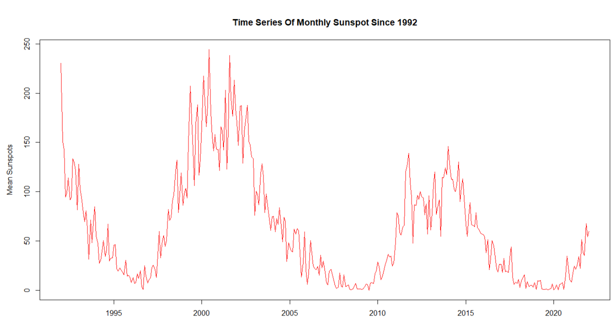

Figure 4. Sunspot Time Series From 1992 to 2022

First, the model is trained on values until from 1992 to 01/2016. The obtained model is then used to forecast the monthly mean sunspot numbers for the next 60 months. The results of the model are represented in the fig 6 and table 2. Fig 5 shows the fitted or predicted values by the "auto.arima" model. Meanwhile, table 3 shows the Mean error & RMSE

Table 1. Sunspot Dataset

Year	Month	Monthly Sunspots
1992	02	230.7
1992	03	151.0
1992	04	142.2

for training and testing data. The lower RMSE of 5.67 on testing data highlight that the model is performing good on test data.

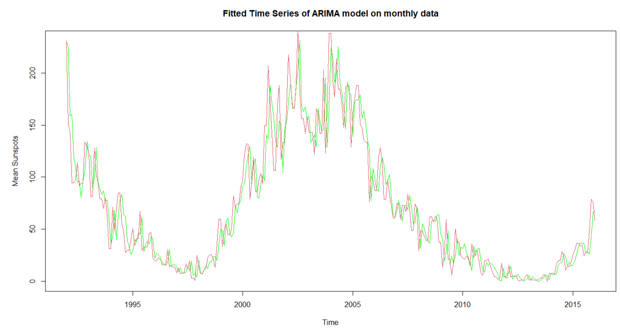

Figure 5. Auto Arima Fitted Model from 1992 to 2016

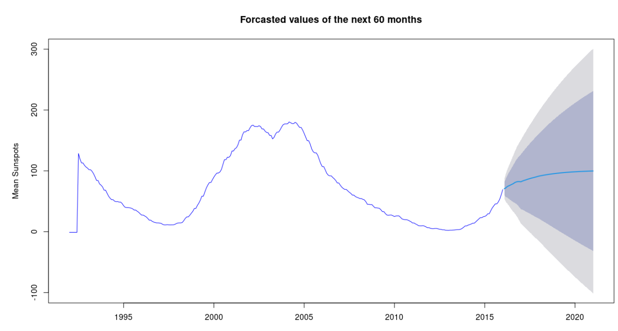

Figure 6. 60 Months predicted Time series from 2016 to 2021 using Auto Arima

Table 2. Snapshot of Auto Arima Forecast for next 60 Months from 02/2016

Time	Point Forecast	Lower 80%	Upper 80%	Lower 95%	Upper 95%
Feb 2016	70.65700	60.1859835	81.12801	54.6429636	86.67103
Mar 2016	71.96903	57.1648371	86.77323	49.3279696	94.61009
Apr 2016	74.29334	56.1616921	92.42499	46.5633760	102.02331
May 2017	85.77037	32.3497307	139.19100	4.0705552	167.47018
Jun 2017	86.58051	31.0339002	142.12712	1.6293016	171.53172
Jul 2018	94.04246	12.4674545	175.61747	-30.7157466	218.80067
Aug 2018	94.41489	10.9423289	177.88745	-33.2453765	222.07516
Sep 2018	94.77179	9.4198661	180.12372	-35.7627154	225.30630
Nov 2019	98.16129	-11.6565997	207.97918	-69.7906800	266.11326
Dec 2019	98.31683	-13.1256833	209.75934	-72.1197846	268.75344
Feb 2020	98.60740	-16.0378920	213.25270	-76.7274459	273.94225
Jan 2021	99.78363	-31.4722029	231.03946	-100.9548423	300.52210

Table 3. Accuracy of the Auto. Arima Model

Accuracy Measure	Training Data	Test Data.
Mean Error	0.1649231	-0.7691479
RMSE	8.056694	5.679984

The highlighted region in Fig 6 represents the values predicted by the model. The darker region represents the 80% confidence interval showed in table 2. The lighter region in the fig 6 highlights the 95% confidence interval for predicted values to fall in that region. It's worth noting that the values for 80% & 95% confidence interval grow larger as the prediction time increases.

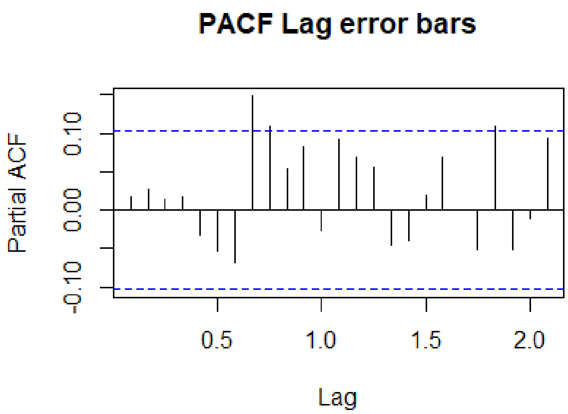

Figure 7. PACF LAG

Fig 7 shows the Partial auto-correlation graph (PACF). PACF graph highlights the direct correlation of lag value at certain lag point in time over the current predicted value. For the given sunspot dataset, lag value of 0.7 and 1.7 are above the error bar lines in the plot. This showcases that the lag values of 0.7 month and 1.7 month have direct correlation on the current predicted data point of the time series.

As mentioned in section 2.2, "auto-arima" decides the p, d, q part of the time series under the hood. Summary statistic of the trained model shows that the "auto.arima" used the ARIMA (1,1,3) p, d, q values for the given sunspot data. A full snapshot of all the AR, MA coefficients and parameter values can be found at Appendix A

4. Experiment

The Genetic algorithm is used to find the p, d, q values of the ARIMA model. The experiment is repeated for 30 iterations, 30 generations and population size of 10. The rationale behind 30 iterations is to statistically compare the results of three variation of GA by using the z-test, that requires a minimum sample size (n) of 30 or more. The z-test is used to calculate the error bars of the three GA variations.

The GA library used in this experiment maximizes the fitness. To combat that, the inverse of RMSE was used, so that the values fall between (0, 1). The inverse of RMSE was maximized, meaning higher values produce the least residuals. This turns the problem into a minimization problem, making the objective the same as before.

As mentioned by (Lukoseviciute & Ragulskis, 2010), the best way to decide GA parameters is through trial and error, table 4 summarizes the GA's experimental setup used for the experiment after trying different variations.

Table 4. Configuration parameters for Genetic Algorithm.

Population Size	10.
Maximum Generations	30
Iterations	30
Crossover Probability	0.4
Crossover Variations	Single Point Crossover; Uniform Crossover; Crossover Order.
Normalized Fitness	$\frac{1}{1+RMSE.}$ RMSE = Root Mean Squared Error

Error bars were used to statistically compare the three GA variations. The error bars are calculated using the following equation (Azad & Ryan, 2013);

$$\bar{X} \pm 1.96\frac{\sigma}{\sqrt{n}} \tag{6}$$

where $\bar{X}$ and σ are the mean and standard deviation of n observations; n = 30 represents the number of runs. "1.96" shows 95% confidence in obtained results. Therefore, Error bars are the equivalent of 95% confidence interval of z-test score to statistically compare the fitness results of three variation.

5. Results

The aim of this study was to tune the AR(p), MA(q), Integration(d) parameters of the ARIMA model using Genetic algorithm (GA). The goal is to investigate the performance of three different crossover methods for selecting p, d, q parameters and compare the performance with the auto.arima's selected p, d, q values. Fig 8 demonstrates the fitness error bars of the single point, uniform & blend crossovers of GA. The table 5 highlights the best solutions or bit values GA came up with.

Fig 8 shows the overlapping error bars. This suggests that the corresponding solutions at each generation of the three crossovers aren't different from each other.

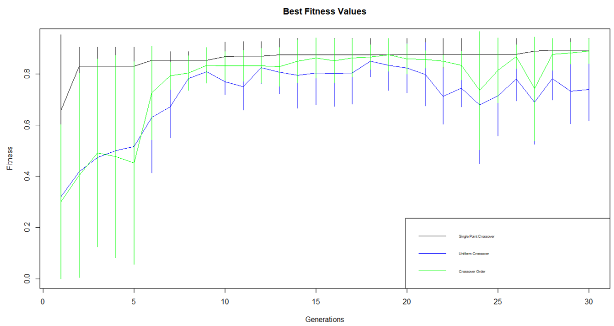

Figure 8. Fitness Comparison of Single Point, Unifrom & Crossover Order

Table 5. Best Solutions Of Different Crossovers

Crossover Type	Best Solution	Real Value
Single Point	1 1 0 1 1 1 1 1 1 0	13, 3, 14
Uniform	1 1 0 1 1 1 1 1 1 0	13, 3, 14
Crossover Order	1 1 0 1 1 1 1 1 1 0	13, 3, 14

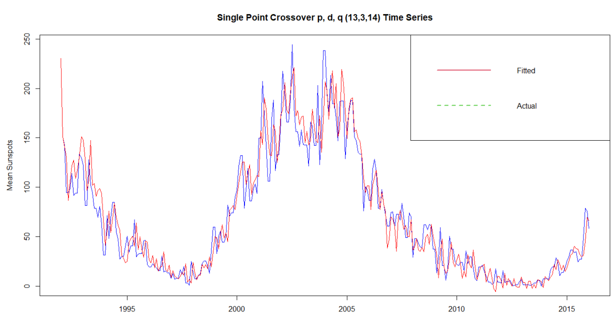

Figure 9. Single Point Crossover ARIMA(13,3,14) model predicted Time Series

Table 6. Snapshot of ARIMA (13,3,14) Forecast for next 60 Months

Time	Point Forecast	Lower 80%	Upper 80%	Lower 95%	Upper 95%
Feb 2016	62.22567	38.24414	86.20720	25.549079	98.90226
Mar 2016	67.50208	37.58036	97.42381	21.740751	113.26342
Apr 2016	74.72373	42.29632	107.15114	25.130287	124.31718
May 2017	122.05927	53.42535	190.69319	17.092743	227.02579
Jun 2017	126.47491	53.14853	199.80129	14.331884	238.61793
Jul 2017	126.58261	48.75227	204.41295	7.551378	245.61385
Nov 2019	286.87554	50.11685	523.63424	-75.215637	648.96672
Dec 2019	294.83462	50.06121	539.60803	-79.514007	669.18325
Jan 2021	388.87973	29.46106	748.29839	-160.803688	938.56314

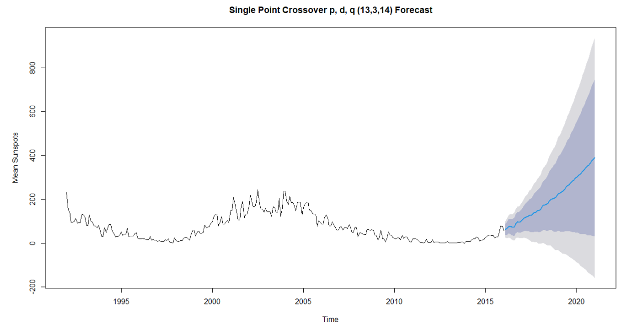

Figure 10. Single Point Crossover Arima(13,3,14) model next 60 months Predictions

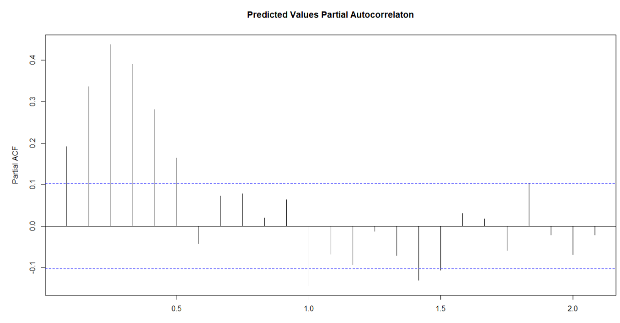

Figure 11. Single Point Crossover ARIMA (13,3, 14) model fitted PACF

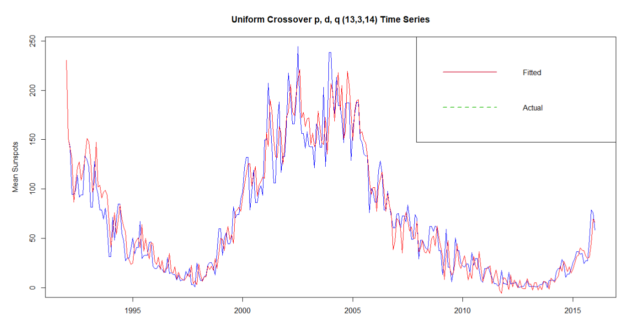

Figure 12. Uniform Crossover ARIMA (13,3,14) model predicted Time Series

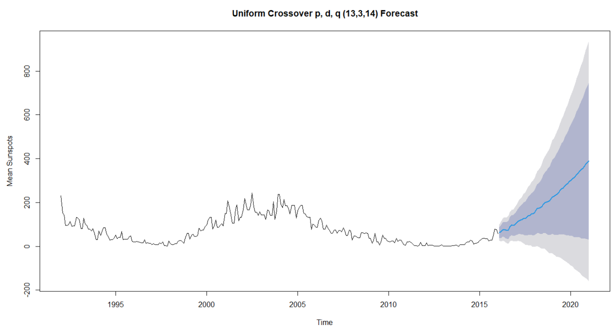

Figure 13. Uniform Crossover ARIMA (13,3,14) model next 60 months Predictions

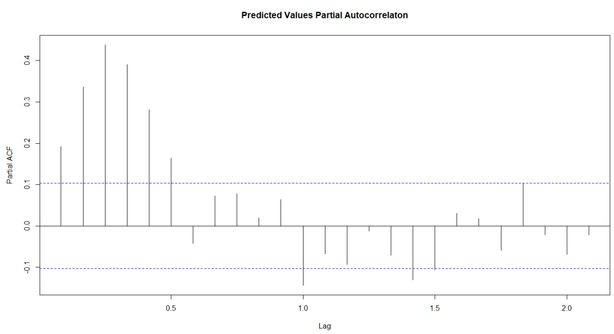

Figure 14. Uniform Crossover ARIMA (13,3,14) model fitted PACF

Table 7. Snapshot of uniform crossover ARIMA (13,3,14) Forecast for next 60 Months

Time	Point Forecast	Lower 80%	Upper 80%	Lower 95%	Upper 95%
Feb 2016	62.22567	38.24414	86.20720	25.549079	98.90226
Mar 2016	67.50208	37.58036	97.42381	21.740751	113.26342
Apr 2016	74.72373	42.29632	107.15114	25.130287	124.31718
May 2016	77.34911	43.08955	111.60866	24.953642	129.74458
Jun 2016	74.15595	37.52527	110.78663	18.134155	130.17774
Jul 2016	73.01268	33.87093	112.15443	13.150542	132.87482
Aug 2017	131.82357	49.71607	213.93106	6.250990	257.39614
Sep 2017	138.60209	51.95766	225.24652	6.090874	271.11331
Oct 2017	141.97350	51.69804	232.24896	3.909095	280.03790
Nov 2018	207.47762	51.39070	363.56455	-31.236725	446.19197
Dec 2018	216.56173	54.20733	378.91613	-31.737896	464.86136
Jan 2021	388.87973	29.46106	748.29839	-160.803688	938.56314

Table 8. Snapshot of crossover order ARIMA (13,3,14) Forecast for next 60 Months

Time	Point Forecast	Lower 80%	Upper 80%	Lower 95%	Upper 95%
Feb 2016	62.22567	38.24414	86.20720	25.549079	98.90226
Mar 2016	67.50208	37.58036	97.42381	21.740751	113.26342
Apr 2016	74.72373	42.29632	107.15114	25.130287	124.31718
May 2017	122.05927	53.42535	190.69319	17.092743	227.02579
Jun 2017	126.47491	53.14853	199.80129	14.331884	238.61793
Jul 2017	126.58261	48.75227	204.41295	7.551378	245.61385
Aug 2018	200.38037	62.38469	338.37606	-10.665813	411.42656
Sep 2018	201.07076	57.77061	344.37091	-18.087909	420.22942
Oct 2018	203.78680	54.34336	353.23023	-24.767213	432.34081
Nov 2019	286.87554	50.11685	523.63424	-75.215637	648.96672
Dec 2019	294.83462	50.06121	539.60803	-79.514007	669.18325
Jan 2021	388.87973	29.46106	748.29839	-160.803688	938.56314

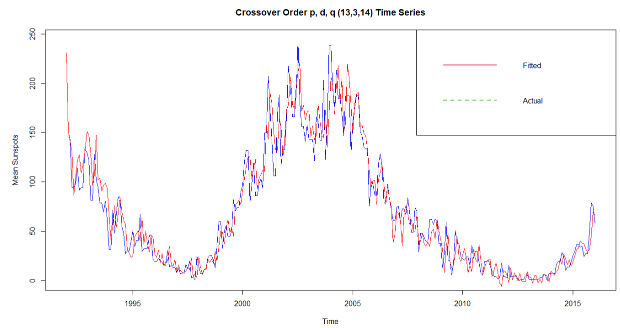

Figure 15. Crossover order ARIMA (13,3,14) model predicted Time Series

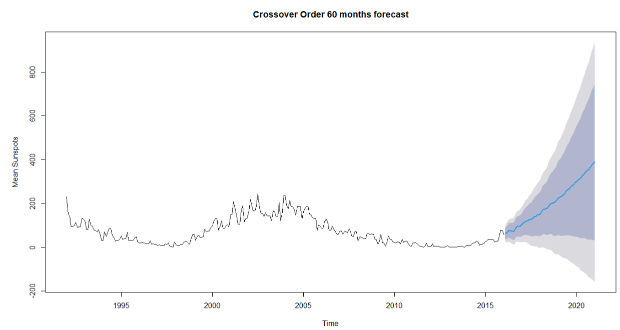

Figure 16. Crossover order Arima (13,3,14) model next 60 months Predictions

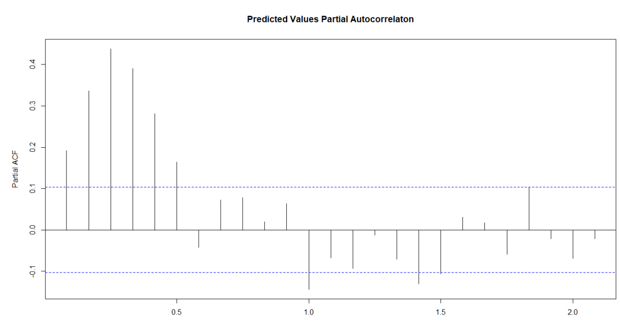

Figure 17. Crossover order ARIMA (13,3,14) model fitted PACF

Table 9. Accuracy of Arima Model Tuned by GA

Crossovers	RMSE	Mean Error (ME).
Single Point	17.7148	-1.803432
Uniform	17.7148	-1.803432
Crossover Order	17.7148	-1.803432

6. Discussion

In this experiment, ARIMA (p, d, q) parameters were selected by three Genetic crossover operators. All three variations of GA came up with the same bit sequence, as represented in table 5. The ”auto.arima” results of (p = 1, d = 1, q = 3) order are highlighted in fig 6, 7, 2 and accuracy measure in table 3.

Fig 8 represents the error bars and the value of fitness function (RMSE) of three crossovers over 30 generations. The best fitness value achieved by all three crossover operators was approx.”0.96”. Single point crossover in fig 8 achieves maximum fitness in early generations. Crossover order reached the maximum fitness value at generation 16. Uniform crossover turns out to be the last to reach the maximum fitness value at generation 20. Furthermore, fluctuations can be observed in the fitness values of uniform crossover & crossover order, highlighting the probabilistic nature of these crossovers.

Table 5 shows, the optimal solution of three crossovers. Unlike ”auto.arima”'s solution of (1, 1, 3), all three GA solutions were (13, 3, 14). It's interesting to note that GA came up with the value of d = 3, that is the differencing part, to standardize a non-stationary time series highlighting the non-stationarity in the data as can be seen in fig 4.

Table 6, 7, 8 all represent the monthly sunspot forecast of three GA variations from 2016 to 2021. The table 7, 6 and 8 all represent similar predictions, starting with ”62” sunspots in February 2016. However, a general increase in values for 80% and 95% confidence interval range can be seen as the future time difference increases. The high values are indicative of the model getting less accurate with increase in time range.

As mentioned in section 2.3.2, RMSE penalizes the large residuals or errors. This can be verified from figure 8, where the error bar of each crossover gets smaller in future generations. This highlights that the candidates (bit solutions) in the population that had high RMSE were not retained for the future generations.

Single point crossover reaches the optimal solution during the initial generations. Meanwhile, crossover order & uniform crossover take few generations to reach the optimal solution. The overlapping error bars of three variations suggest that the optimal solution reached by the three variations in not statistically different. In fact, table 5 shows that all the three variations of GA reach the similar solution.

Table 9 & 3 show the RMSE of ARIMA model produced by both GA ARIMA & ”auto.arima”. For the given dataset, ”auto.arima” performed better with RMSE of ”8” over ”17” in GA's case. The results of the finding suggest that ”auto.arima” p, d, q had less RMSE score and more predictive power on both training & test data. Also, the complexity of the model is less with (p=1, d=1, q=3) compared to (p=13, d=3, q=14). This leads to the issue raised by (Beenstock & Szpiro, 2002) that the GA specification search might yield results that may not be much different or better to the original method. A complete description of the coefficients of AR (p) and MA (q) for three GA variations is available at appendix A.

7. Conclusion & Future work

This report implements Genetic Algorithm for ARIMA model's parameter tuning (p, d, q) with three different crossover operators for monthly sunspot prediction. Experimental results showed that ”auto.arima” performed better both in accuracy & simplicity over GA. The results also highlight that all the crossover methods reach the same optimal solution. However, single point crossover reached the optimal solution the fastest, in comparison to the other two crossover techniques.

The results of the predictions also highlight that the values in confidence interval gradually start increasing as the prediction period increase. The PACF graph of the three crossover shows that lag of less than 0.5 has the direct influence on the current predicted value.

The objective of the report was to investigate GA specification search ability when tuning ARIMA model (p, d, q) irrespective of the data used. However, It's worth mentioning that the results of the report are limited by the number of observations in the data. The report only took into consideration the monthly sunspot numbers since 1992.

Limited amount of observations may affect the GA's specification search for Time Series modelling, as we are calculating the optimal lag values. Large values of ”d” also introduced infinities in the solution, as the slicing of the series becomes too big to calculate. So in such case, large data may further improve the performance of the GA in calculating the parameters.

Additionally, the report doesn't tune the seasonality parameters of the ARIMA model that are indicated using big (P, D, Q). Exploring large values of both (p, d, q) & (P, D, Q) require immense computational resources. As, these values grow large, the computational time to calculate a single generation of ARIMA model using GA grows exponentially. Furthermore, more work is needed to check the impact of other fitness functions like AIC, BIC & MSE. Here, RMSE is used as fitness function, so candidate solutions with lower fitness score are penalized.

Lastly, this study can further benefit from employing different selection methods that may help in improving the specification search of ARIMA by GA.

8. Appendix

A. Auto Arima & GA Summary statistics

```
Model Information:
Series: solar_data_ts_data_train
ARIMA(1,1,3)(2,0,1)[12]

Coefficients:
         ar1      ma1     ma2     ma3     sar1     sar2    sma1
      0.9487  -0.9492  0.0008  0.0677  -0.3320  -0.0406  0.1218
s.e.  0.0421   0.0715  0.0815  0.0664   4.2237   0.9404  4.2056

sigma^2 = 66.76:  log likelihood = -1010.38
AIC=2036.76   AICc=2037.28   BIC=2066.06

Error measures:
                    ME     RMSE      MAE       MPE     MAPE      MASE        ACF1
Training set 0.1649231 8.056694 2.480933 0.8871242 4.961733 0.1051726 0.003668905
```

Figure 18. Auto Arima model Summary Statisitcs

```
Series: solar_data_ts_data_train
ARIMA(13,3,14)

Coefficients:
          ar1      ar2      ar3      ar4      ar5      ar6      ar7      ar8
      -1.3288  -1.4723  -1.5652  -1.3286  -0.4343  -0.1834  -0.3483  -0.6812
s.e.   0.1798   0.2385   0.2479   0.2900   0.3145   0.2851   0.2762   0.2540
          ar9     ar10     ar11     ar12     ar13      ma1      ma2     ma3
      -0.9343  -1.3361  -1.0828  -0.8018  -0.3901  -0.9251  -0.2398  0.0870
s.e.   0.2467   0.2031   0.1999   0.1584   0.0705   0.1900   0.2880  0.3028
          ma4      ma5     ma6     ma7      ma8      ma9    ma10     ma11    ma12
      -0.1547  -0.5705  0.8817  0.3658  -0.1320  -0.3573  0.4923  -0.7188  0.0649
s.e.   0.2383   0.1492  0.2087  0.2517   0.2563   0.2024  0.1403   0.2209  0.2896
        ma13    ma14
      0.0619  0.1558
s.e.  0.3083  0.1617

sigma^2 = 350.2:  log likelihood = -1238.57
AIC=2533.15   AICc=2539.47   BIC=2635.52

Training set error measures:
                    ME    RMSE      MAE  MPE MAPE      MASE       ACF1
Training set -1.803432 17.7148 12.66728 -Inf  Inf 0.4493667 0.03785735
```

Figure 19. Uniform Crossover ARIMA Model Summary Statistics

```
Series: solar_data_ts_data_train
ARIMA(13,3,14)

Coefficients:
          ar1      ar2      ar3      ar4      ar5      ar6      ar7      ar8
      -1.3288  -1.4723  -1.5652  -1.3286  -0.4343  -0.1834  -0.3483  -0.6812
s.e.   0.1798   0.2385   0.2479   0.2900   0.3145   0.2851   0.2762   0.2540
          ar9     ar10     ar11     ar12     ar13      ma1      ma2     ma3
      -0.9343  -1.3361  -1.0828  -0.8018  -0.3901  -0.9251  -0.2398  0.0870
s.e.   0.2467   0.2031   0.1999   0.1584   0.0705   0.1900   0.2880  0.3028
          ma4      ma5     ma6     ma7      ma8      ma9    ma10     ma11    ma12
      -0.1547  -0.5705  0.8817  0.3658  -0.1320  -0.3573  0.4923  -0.7188  0.0649
s.e.   0.2383   0.1492  0.2087  0.2517   0.2563   0.2024  0.1403   0.2209  0.2896
        ma13    ma14
      0.0619  0.1558
s.e.  0.3083  0.1617

sigma^2 = 350.2:  log likelihood = -1238.57
AIC=2533.15   AICc=2539.47   BIC=2635.52

Training set error measures:
                    ME    RMSE      MAE  MPE MAPE      MASE       ACF1
Training set -1.803432 17.7148 12.66728 -Inf  Inf 0.4493667 0.03785735
```

Figure 20. Single Point Crossover ARIMA Model Summary Statistics

```
Series: solar_data_ts_data_train
ARIMA(13,3,14)

Coefficients:
          ar1      ar2      ar3      ar4      ar5      ar6      ar7      ar8
      -1.3288  -1.4723  -1.5652  -1.3286  -0.4343  -0.1834  -0.3483  -0.6812
s.e.   0.1798   0.2385   0.2479   0.2900   0.3145   0.2851   0.2762   0.2540
          ar9     ar10     ar11     ar12     ar13      ma1      ma2     ma3
      -0.9343  -1.3361  -1.0828  -0.8018  -0.3901  -0.9251  -0.2398  0.0870
s.e.   0.2467   0.2031   0.1999   0.1584   0.0705   0.1900   0.2880  0.3028
          ma4      ma5     ma6     ma7      ma8      ma9    ma10     ma11    ma12
      -0.1547  -0.5705  0.8817  0.3658  -0.1320  -0.3573  0.4923  -0.7188  0.0649
s.e.   0.2383   0.1492  0.2087  0.2517   0.2563   0.2024  0.1403   0.2209  0.2896
        ma13    ma14
      0.0619  0.1558
s.e.  0.3083  0.1617

sigma^2 = 350.2:  log likelihood = -1238.57
AIC=2533.15   AICc=2539.47   BIC=2635.52

Training set error measures:
                    ME    RMSE      MAE  MPE MAPE      MASE       ACF1
Training set -1.803432 17.7148 12.66728 -Inf  Inf 0.4493667 0.03785735
```

Figure 21. Crossover Order ARIMA Model Summary Statistics

B. Code Available at

https://github.com/xahram/timeseries-parameter-GA

References

Aguirre, L. A., Letellier, C., and Maquet, J. Forecasting the time series of sunspot numbers. *Solar Physics*, 249(1): 103–120, May 2008. ISSN 1573-093X. doi: 10.1007/s11207-008-9160-5. URL https://doi.org/10.1007/s11207-008-9160-5.

Ariyo, A. A., Adewumi, A. O., and Ayo, C. K. Stock price prediction using the arima model. In *2014 UKSim-AMSS 16th International Conference on Computer Modelling and Simulation*, pp. 106–112, 2014. doi: 10.1109/UKSim.2014.67.

Azad, R. and Ryan, C. A simple approach to lifetime learning in genetic programming-based symbolic regression. *Evolutionary computation*, 22, 09 2013. doi: 10.1162/EVCO_a_00111.

Beenstock, M. and Szpiro, G. Specification search in nonlinear time-series models using the genetic algorithm. *Journal of Economic Dynamics and Control*, 26(5):811–835, 2002. ISSN 0165-1889. doi: https://doi.org/10.1016/S0165-1889(00)00083-X. URL https://www.sciencedirect.com/science/article/pii/S016518890000083X.

Chai, T. and Draxler, R. R. Root mean square error (RMSE) or mean absolute error (MAE)? – Arguments against avoiding RMSE in the literature. *Geoscientific Model Development*, 7(3):1247–1250, June 2014. ISSN 1991-959X. doi: 10.5194/gmd-7-1247-2014. URL https://gmd.copernicus.org/articles/7/1247/2014/. Publisher: Copernicus GmbH.

Fuller, W. A. Nonstationary autoregressive time series. In *Handbook of Statistics*, volume 5 of *Time Series in the Time Domain*, pp. 1–23. Elsevier, January 1985. doi: 10.1016/S0169-7161(85)05003-9. URL https://www.sciencedirect.com/science/article/pii/S0169716185050039.

Gallagher, K. and Sambridge, M. Genetic algorithms: A powerful tool for large-scale nonlinear optimization problems. *Computers & Geosciences*, 20(7):1229–1236, August 1994. ISSN 0098-3004. doi: 10.1016/0098-3004(94)90072-8. URL https://www.sciencedirect.com/science/article/pii/0098300494900728.

geeksforgeeks. Crossover in genetic algorithm — Geeksforgeeks. URL https://www.geeksforgeeks.org/crossover-in-genetic-algorithm/.

Gkana, A. and Zachilas, L. Sunspot numbers: Data analysis, predictions and economic impacts. *Journal of Engineering Science and Technology Review*, 8:79–85, 02 2015. doi: 10.25103/jestr.081.14.

Holland, J. H. *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control and Artificial Intelligence*. MIT Press, Cambridge, MA, USA, 1992. ISBN 0262082136.

Kora, P. and Yadlapalli, P. Crossover Operators in Genetic Algorithms: A Review. *International Journal of Computer Applications*, 162 (10):34–36, March 2017. ISSN 09758887. doi: 10.5120/ijca2017913370. URL http://www.ijcaonline.org/archives/volume162/number10/kora-2017-ijca-913370.pdf.

Leamer, E. E. *Specification searches: ad hoc inference with nonexperimental data.* John Wiley Sons Incorporated, New York, 1978.

Leung, F., Lam, H., Ling, S., and Tam, P. Tuning of the structure and parameters of a neural network using an improved genetic algorithm. *IEEE Transactions on Neural Networks*, 14(1):79–88, January 2003. ISSN 1941-0093. doi: 10.1109/TNN.2002.804317. Conference Name: IEEE Transactions on Neural Networks.

Lukoseviciute, K. and Ragulskis, M. Evolutionary algorithms for the selection of time lags for time series forecasting by fuzzy inference systems. *Neurocomputing*, 73(10):2077–2088, 2010. ISSN 0925-2312. doi: https://doi.org/10.1016/j.neucom.2010.02.014. URL https://www.sciencedirect.com/science/article/pii/S0925231210001554. Subspace Learning / Selected papers from the European Symposium on Time Series Prediction.

Messias, V. R., Estrella, J. C., Ehlers, R., Santana, M. J., Santana, R. C., and Reiff-Marganiec, S. Combining time series prediction models using genetic algorithm to autoscaling Web applications hosted in the cloud infrastructure. *Neural Computing and Applications*, 27(8):2383–2406, November 2016. ISSN 0941-0643, 1433-3058. doi: 10.1007/s00521-015-2133-3. URL http://link.springer.com/10.1007/s00521-015-2133-3.

Orfila, A., Ballester, J. L., Oliver, R., Alvarez, A., and Tintoré, J. Forecasting the solar cycle with genetic algorithms. *A&A*, 386(1):313–318, 2002. doi: 10.1051/0004-6361:20020246. URL https://doi.org/10.1051/0004-6361:20020246.

Pala, Z. and Atici, R. Forecasting sunspot time series using deep learning methods. *Solar Physics*, 294(5): 50, May 2019. ISSN 1573-093X. doi: 10.1007/s11207-019-1434-6. URL https://doi.org/10.1007/s11207-019-1434-6.

Picek, S. and Golub, M. Comparison of a crossover operator in binary-coded genetic algorithms. *WSEAS Transactions on Computers*, 9, 09 2010.

Rubicite. Order 1 crossover — Rubicite. URL https://www.rubicite.com/Tutorials/GeneticAlgorithms/CrossoverOperators/Order1CrossoverOperator.aspx.

Siami-Namini, S. and Namin, A. S. Forecasting Economics and Financial Time Series: ARIMA vs. LSTM. *arXiv:1803.06386 [cs, q-fin, stat]*, March 2018. URL http://arxiv.org/abs/1803.06386. arXiv: 1803.06386.

SILSO World Data Center.

Spiegel, E. A. Lectures on solar and planetary dynamos. pp. 245, 1994. ISSN 0521461421.

Suparta, W. Space weather effects on microelectronics devices around the leo spacecraft environments. *Journal of Physics: Conference Series*, 539:012025, 10 2014. doi: 10.1088/1742-6596/539/1/012025.

Tutorials Point. Genetic algorithms - crossover — Tutorialspoint. URL https://www.tutorialspoint.com/genetic_algorithms/genetic_algorithms_crossover.htm.

Wolf, R. Sunspot epochs since a.d. 1610: The periodic return of sunspot minima. *Acad. Sci*, 35:704 – 705, 1852.