Anomaly Detection in Time Series Data with Multiple Seasonal Trends

Student: Jaclyn Taylor

Student Email: taylor.918@wright.edu

Faculty: Dr. Soon Chung

Faculty Email: soon.chung@wright.edu

AFRL Sponsor: Dr. Vincent Schmidt

AFRL Directorate: RH

PA #: AFRL-2024-5723

Introduction and Background Information

- A time series is a sequence of data points indexed in time order. This is usually used to track changes in value over time.
- Anomalies are datapoints that significantly deviate from their expected/predicted value.
- Early and accurate detection of anomalies allow businesses to mitigate harmful effects
    - For example: when your bank notifies you of strange purchases on your account
- Seasonalities are cycles that repeat regularly over a period (hourly, monthly, etc.)
- Most of today’s big data are time series that contain both anomalies and multiple seasonalities
- However, most common models today are only suited for single seasonality

Our Approach

- Previous students created a multi-SARIMA model to handle multi-seasonal data. However, their method for finding parameters was a time costly grid search.
  - o This model is an extension of the SARIMA model (Seasonal Autoregressive, Integrated Moving Algorithm)
- We have chosen to implement the irace package from R to find the optimal parameters of the models.
  - o We used both the original package, and our own recreation of the package in Python
- We are also working to implement the analysis done by the multi-SARIMA model in a Deep Neural Network. For the sake of this presentation however, we will only cover the irace implementation.

What is Irace?

- The irace package in R implements the iterated racing procedure and is an extension of the Iterated F-race.
- It is primarily used to automatically configure and optimize different algorithms.
  - The package does this by iterating through multiple experiments in parallel to quickly find the optimal settings
- Irace is primarily used in R, but can be applied to Python as well (where our models are located)

The Models

- For the sake of testing the irace package in R, as well as our own implementation of irace in Python, we used three of the models created by previous students:
- Moving Average (MA):
  - Predicts future values as a weighted sum of lagged residuals
- Seasonal Autoregressive Integrated Moving Average (SARIMA):
  - Extension of the autoregressive integrated moving average (ARIMA) model
  - Incorporates one seasonal component into its forecasts
- Multiple Seasonal Autoregressive Integrated Moving Average (multi-SARIMA) (INCOMPLETE)
  - Extension of the SARIMA model with multiple seasonal components

Dataset Overview

- We chose to use all three datasets that were previously created/used for this project.
  - NYC Taxi
  - A synthetic dataset created by previous students
  - HotGym
- All 3 datasets are univariate time series datasets with two meaningful seasonal trends and hand- labeled anomalies
- For the parameter values, we made the range for both p and q to be (1, 5). For multi-Sarima, we set the range to be (0, 4). These ranges were chosen to hopefully cover all possible best parameter values.

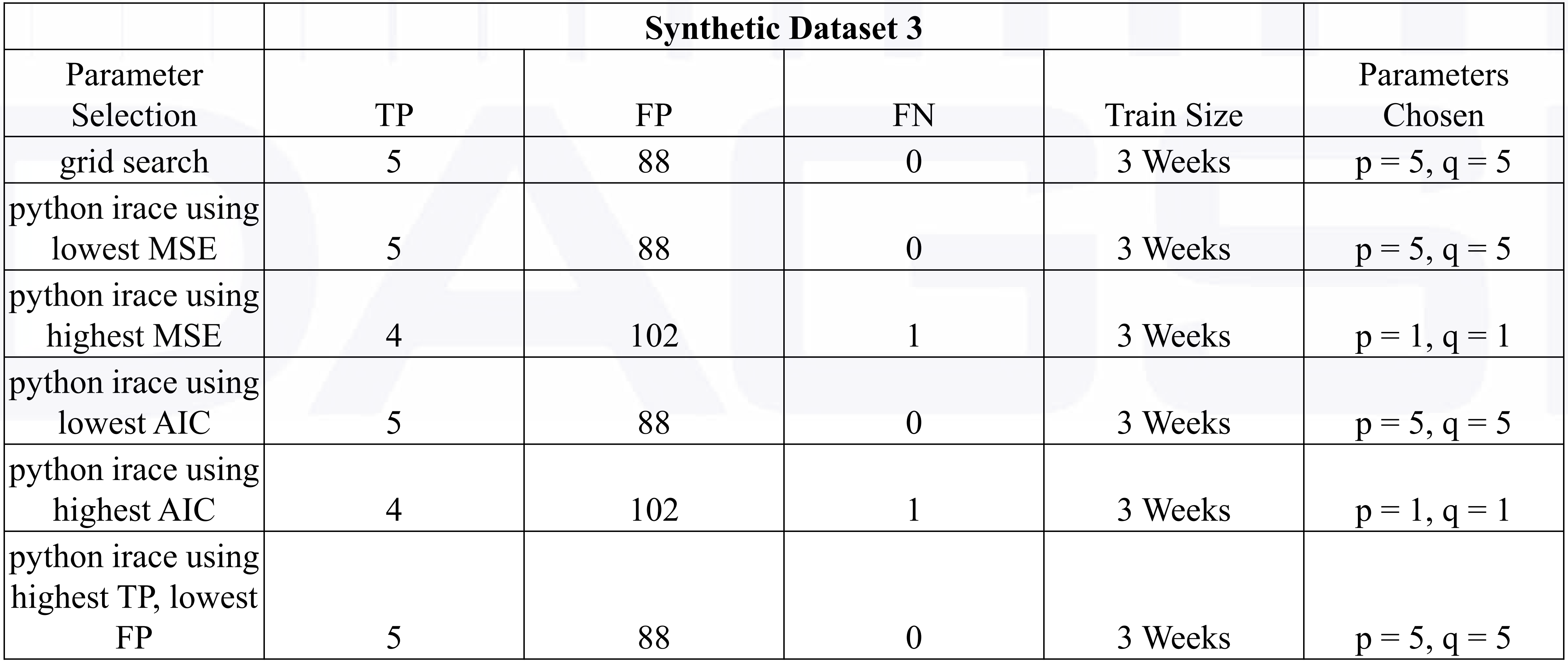
Results

- The next few slides will contain the tables of our results for the MA and SARIMA models. After testing both our python implementation of irace and the irace package itself, we found that both packages produced the same results.
    - We did not include the multi-SARIMA model. Since the model is so time consuming to run, we were unable to effectively test the irace methods due to hardware constraints
- We also decided to use various statistical methods to attempt to determine the most optimal parameters, specifically the aic (Akaike information criterion) order and the mse (mean squared error) for each trial.
    - However, we found that simply comparing the true positive and false positive values for each trial was more accurate.
- For the sake of this presentation, we have only included our python irace results to minimize table size

Results - MA Table NYCTaxi Data

	NYC_Taxi Dataset				
Parameter Selection	TP	FP	FN	Train Size	Parameters Chosen
grid search	3	3451	2	3 Weeks	q = 1
python irace using lowest MSE	0	73	5	3 Weeks	q = 5
python irace using highest MSE	3	3451	2	3 Weeks	q = 1
python irace using lowest AIC	0	73	5	3 Weeks	q = 5
python irace using highest AIC	3	3451	2	3 Weeks	q = 1
python irace using highest TP, lowest FP	3	3451	2	3 Weeks	q = 1

Results – MA Table Synthetic Data



	Synthetic Dataset 3				
Parameter Selection	TP	FP	FN	Train Size	Parameters Chosen
grid search	5	88	0	3 Weeks	p = 5, q = 5
python irace using lowest MSE	5	88	0	3 Weeks	p = 5, q = 5
python irace using highest MSE	4	102	1	3 Weeks	p = 1, q = 1
python irace using lowest AIC	5	88	0	3 Weeks	p = 5, q = 5
python irace using highest AIC	4	102	1	3 Weeks	p = 1, q = 1
python irace using highest TP, lowest FP	5	88	0	3 Weeks	p = 5, q = 5

Conclusion

- The R irace package (as well as our implementation of it in Python) is just as accurate at finding the optimal parameters for the MA and SARIMA models as the grid search algorithm.
    - Based on this data, the multi-SARIMA model would most likely work as well, however more testing needs to be completed to determine this.
- Since we do not have any run time data from the original grid search for parameters, we cannot comment on whether this irace method is more time cost efficient.
- This did show however that either irace method would be the optimal methods for finding the model parameters