Departmemt of Mathematics and Statistics

CAPSTONE PAPER REVIEW WEEK 4 (PAPER 1)

Analyzing and Replicating "A comparative machine learning study for time series oil production forecasting: ARIMA, LSTM, and Prophet"

Gianyce Michelle Gesualdo Ortiz

Supervisor: **Dr. Shushen Pu**

Contents

CHAPTER 1

Background/Motivation

In considering the background for this project, I specifically wanted a text that deals with time series analysis. Having read several papers on ARIMA, I wanted to find an article that provided access to both the code and the data, so I could practice implementing an ARIMA model in my work. To my surprise, I stumbled upon an article titled "A Comparative Machine Learning Study for Time Series Oil Production Forecasting: ARIMA, LSTM, and Prophet". This article stood out to me because, although my focus is primarily on ARIMA, it offers the opportunity to explore other models like LSTM and Prophet. Although I will not go in depth with those methods in this paper, I may also use it as a reference to my capstone project.

The article aims to utilize time series data to develop a machine learning-based forecasting method, hence why they are exploring the three model types mentioned. This approach treats the existing data as a time series, extracting important characteristics from historical data to predict future trends. Specifically, the authors address the challenge of predicting the production performance of unconventional reservoirs and apply their methodology to evaluate its effectiveness across the DJ Basin.

CHAPTER 2

Methods Used

In this paper, three methods were used: ARIMA, LSTM, and Prophet, in an attempt to present machine learning-based time series forecasting techniques to capture oil production trends in wells. Specifically, in my replication, I focused on recreating the results of the ARIMA model presented in the original work.

Before presenting my results, I will first provide an overview of the ARIMA model and explain what I aimed to capture through my implementation.

The general equation for an ARIMA model, denoted as ARIMA(p, d, q), is:

$$y_t = \phi_1 y_{t-1} + \phi_2 y_{t-2} + \cdots + \phi_p y_{t-p} + \theta_1 \varepsilon_{t-1} + \theta_2 \varepsilon_{t-2} + \cdots + \theta_q \varepsilon_{t-q} + \varepsilon_t$$

- y_t: Represents the value of the time series at time t.

- $\phi_1, \phi_2, \ldots, \phi_p$: These are the autoregressive (AR) coefficients, where ϕ_1 is the coefficient for the lagged value at time $t-1$, ϕ_2 is for $t-2$, up until p.

- $\theta_1, \theta_2, \ldots, \theta_q$: The moving average (MA) coefficients, where θ_1 is the coefficient for the error term at time $t-1$, θ_2 is for $t-2$ up until q.

- ϵ_t: This is the white nose, or error term or residual at time t.

When I am structuring an ARIMA model to find the optimal combination of (p, d, q), these are what the variables mean in the overall equation:

- p: The order of the autoregressive part, this is used to indicate how many lagged observations are considered.

- d: The degree of differencing required to make the series stationary.

- q: The order of the moving average part, indicating how many lagged error terms are considered.

As we will see, the steps on finding p, d, q in ARIMA are first, to find d, we begin my checking for stationarity using a rolling statistic test. The paper and myself used the Augmented Dickey-Fuller (ADF) test. If the data was not stationary, we would need apply differencing. Typically it would be first order differencing, where $d = 1$. If the data is then not stationary the number of differencing, or d is continued to be determined.

After differencing, we use the ACF and PACF test to find p. The ACF genereally is used to correlate the time series with its historical values. The PACF shows the correlation of the time series with the lags. If there is a need for an MA component in the values, this would be shown if there are significant spikes in the ACF plot. If there were significant spikes in the PACF plot, this would indicate that there would be a need for an AR component.

Then, once combinations are achieved, their strength can be tested with the use of an AIC (Akaike Information Criterion) or a BIC (Bayesian Information Criterion) to see how well the combinations of (p, d, q) perform. Model with the lowest AIC or BIC score is what we are looking for.

AR and MA terms:

AR Term: $\mathrm{y}_t = \epsilon_t + \sum_{i=1}^{p} \phi_i y_{t-i}$

The ARIMA model can become the AR model seen above when the condition that $d = 0$ and $q = 0$. Because when there is no differencing required (as seen in stationary data) and the MA component ($q = 0$ the ARIMA is able to be reduced to just the autoregressive part.

So, when we are looking at the equation above, we can see the summation of the p, where p is the nuumber of autregressive lags plus any white nose that the model has.

MA Term: $\mathrm{y}_t = \epsilon_t + \sum_{i=1}^{q} \theta_i \epsilon_{t-i}$

The ARIMA model can become the MA model when the condition $p = 0$ and $d = 0$ is met. This happens when there are no autoregressive terms and no differencing required (the data is stationary), reducing the ARIMA model to just the moving average component.

In the equation above, we focus on the summation of q, where q is the number of

moving average lags plus any white noise the past values can create in model.

Aside from the ARIMA model, the text also uses the Prophet model and an LSTM model for forecasting data. LSTM is unique from other recurrent neural networks as it introduces memory units/cells into its network. Neural networks in general have input layer and an output. LSTMs have a hidden layer that chains repeating cells of the network, where it can contain memory cells at time steps. Due to the memory gates within each unit, LSTM has the capacity to learn long term dependencies, then the hidden layer passes into the output layer for prediction. I did not do this within my code, but to summarize the process that was conducted, after data was prepared a LSTM model was constructed. Here, it was specified how the input layer was to be reshaped, initialize parameters and a definitive output was created. Then, this LSTM model would be trained. This is where there would be an included loss function and optimizer, plus the number of epochs that they would set would be included here. Lastly, the model would then make predictions based on the data.

Lastly, the Prophet Model, which was created by Facebook, was used as a way of modeling time series in this article as well. This specific model is effective if there is nonlinear time series, so data is effected by outliers, missing data, seasonality, and other effects. Oil production in our case, we can see in later images is generally on a decline over the years, therefore it is not directly effected by seasonality or holiday effect, so this model in our case may not be as robust.

CHAPTER 3

Coding and Programming

In my replication project, Python was used as the primary tool for data analysis and visualization. The data and code from the original paper were accessed via a GitHub repository titled GitHub Repository: Oil Rate Predict. The original goal was to replicate the ARIMA model used for forecasting oil production. However, due to the absence of a specific file, referred to as '6N,' which was critical for the predictions in the original study, I had to adapt my approach.

Instead of a direct replication, I focused on three key tasks. First, I graphed the oil production's decline curve over the years for Gobbler's wells. Second, I calculated a rolling mean and standard deviation to analyze the variability of the data over time. Lastly, I replicated the AutoCorrelation Function (ACF) of different log series for the Gobbler wells, providing insights into the time-dependent relationships in the data.

Data Cleaning/Preparation Steps: The initial step involved cleaning and preparing the dataset. Rows with string values were filtered out, and the necessary string values were converted to float. Date and time formats were fixed, and any NaN values were removed. Additionally, I grouped the data by unique API values, saving the data into separate CSV files. To automate this process, I defined the target directory and looped through each group of APIs, using the 'glob' library to facilitate file saving.

Some important libraries that I used within my project were Pandas, for data manipulation and analysis, this was also beneficial in the data cleaning and preparation as it allowed for me to handle the missing values when working with the time series data. NumPy for the log transformation and differencing in the time series. Statsmodels

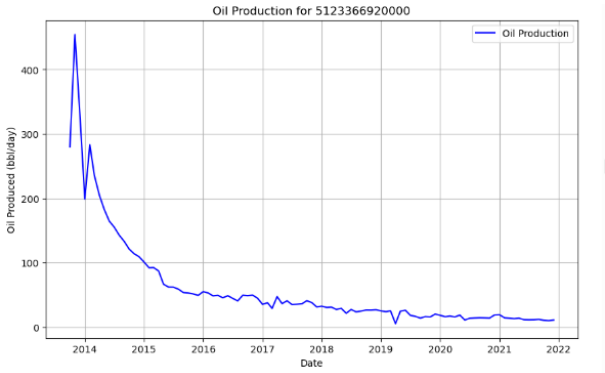

Figure 3.1: Replicated Oil Production Model of Gobbler Well

was used for the ARIMA model implementation and for the Augmented Dickey-Fuller (ADF) test. Finally, Sklearn was used for calculations such as R-squared, RMSE, and others that were important for evaluation the ARIMA model. According to the article, the procedure of building an ARIMA model for time series analysis includes three steps: 1) stationarize the series, 2) find optimal hyperparameters to build the ARIMA model and 3) evaluate and make predictions. So, as I continue my paper I will go into each of these sections to try and interpret each of these steps in the article.

Oil Production's Decline: The original paper illustrated a decline in oil production over the years, although it didnât explicitly specify which oil production data was used in their plots. In my replication, after identifying Gobbler's unique APIs, I graphed the oil production for each of these wells. The results reflected a similar trend of declining production (Figure 3.1), confirming that the data aligned with the general assumptions made in the original study (Figure 3.2). From these figures, it is important to note that since the release of the original article, additional data was added for another year, extending the oil production date range to 2022 in my work, whereas the article's data only went up to 2021. Despite this difference, the general curve remains consistent, showing a continuous decline in oil production over the years.

Testing for Stationary Data: To test whether the data is stationary or not, I applied rolling statistics and the Augmented Dickey-Fuller (ADF) test (as shown in Figure 3.3). The null hypothesis assumes the data is non-stationary. If the p-value is below 0.05, we reject the null hypothesis and accept the alternative hypothesis that the data is stationary. While my results differ slightly from the original work due to discrepancies in the data available at the time of the article's release compared to what I had access

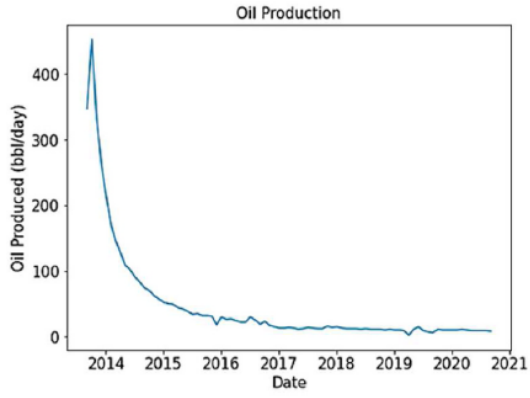

Figure 3.2: Oil Production

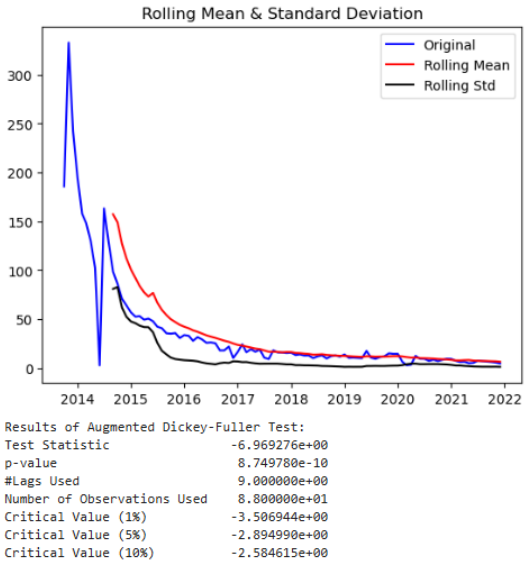

Figure 3.3: Rolling Mean and ADF Test

to, the overall outcome remains consistent. According to the ADF test, for all critical values, the p-value was significantly less than 0.05, indicating that the null hypothesis of a non-stationary time series should be rejected. Additionally, the ADF test statistic was -6.96, which is lower than the 1% critical value of -2.58, further confirming stationarity.

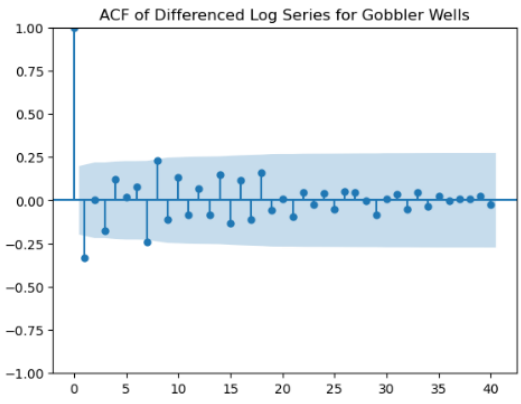

Figure 3.4: Replicated ACF

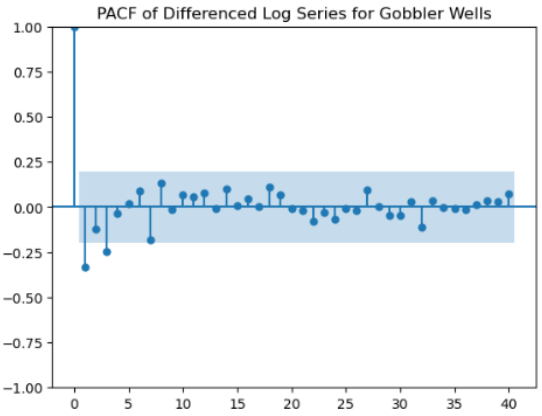

Figure 3.5: Replicated PACF

In the paper, the importance of the Autocorrelation Function (ACF) plot was emphasized to assess the correlation between a time series and its lagged values. Specifically, in ARIMA modeling, ACF helps in identifying the moving average (MA) component by showing how past values (lags) are correlated to the present.

The ACF plot of the difference log series for Gobbler wells, as seen in my results, displays a significant spike at the first lag, indicating a strong correlation with the

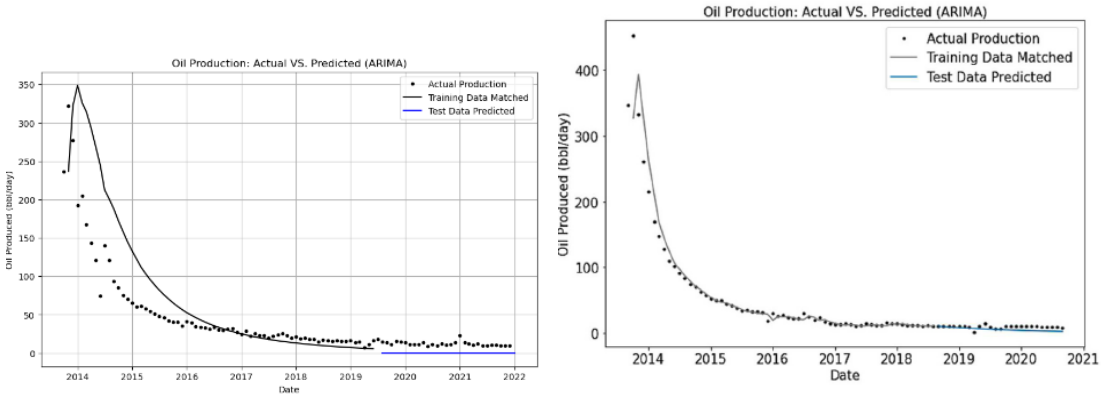

Figure 3.6: ARIMA for Gobbler Wells (left) and ARIMA from the Article (right)

immediate past value. After this, the values quickly oscillate around zero, showing that beyond the first lag, time is not a significant correlator. This behavior is ideal for ARIMA, as it suggests that after differencing, the series becomes stationary, a prerequisite for ARIMA models. This is aligned with the paper's findings where they highlight the importance of ACF plots in identifying the moving average (MA) process and determining the right configuration for the ARIMA model.

Similarly, the Partial Autocorrelation Function (PACF) plot is used to help identify the autoregressive (AR) part of the model.

In Figure 3.6, we see my replication of the ARIMA model for Gobbler wells, which shows a similar overall trend to the articleâs model shown in Figure 3.7. However, one key difference is that my data extends a year later (until 2022), while the articleâs data ends in 2021. This is why there is a slight variation between the two figures. Despite this, I attempted to stay true to the original ARIMA model from the article, using the same (0,1,1) configuration.

The strength of the ARIMA model is that it effectively captures trends in oil production decline, as seen in both replications and the article. However, one weakness is that ARIMA requires stationary data and can struggle with capturing sudden changes or irregularities in time series data. Despite the differences in data availability and time span, my results align with the patterns shown in the article, demonstrating that the (0,1,1) model is robust for predicting long-term oil production trends. However, I would personally look more into depth within my model itself, hopefully with more time access more of the code and try to get exactly what they were able to achieve. But overall it was fine.

CHAPTER 4

Strengths and Limitations

Reflecting on the strengths and limitations I encountered while using ARIMA, I would highlight several key points. First, one of ARIMA's main strengths is its simplicity and ease of implementation. The process of determining whether the data is stationary, calculating rolling statistics, and using ACF/PACF to identify the appropriate p, d, and q parameters is relatively straightforward. Additionally, ARIMA was particularly well-suited to this dataset since it involved non-seasonal time series data. Moreover, ARIMA's performance, in terms of error, proved to be competitive. As indicated in the article, the median RMSE values for both ARIMA and LSTM methods across four different pads ranged from 3 to 10, showing their capacity to predict oil production trends with reasonable accuracy.

However, ARIMA does have its limitations. A key challenge is that ARIMA requires prior transformation of the data into a stationary time series before the model can be applied. This adds an extra preprocessing step, which may not be necessary for other forecasting models. Another limitation is that while ARIMA is effective for short-term predictions, its performance diminishes when attempting long-term forecasts. The model is best suited for short-term oil production predictions, and as noted in the article, ARIMA excels in short-term forecasts where the actual production matches the predicted results more accurately.

CHAPTER 5

Conclusion and Relevance to the Capstone

As mentioned in the beginning, my primary goal was to develop and run my own ARIMA model. I have thoroughly dived into the coding process, carefully following each step to ensure I understood how to build the model from the ground up. Throughout this process, I closely examined the results I obtained during the replication, comparing and contrasting them with those presented in the article. This allowed me to identify the similarities and differences in performance between my approach and that of the original paper. Additionally, I evaluated the strengths and limitations of the ARIMA model, particularly in the context of predicting oil production data as discussed in the article. By doing so, I gained valuable insights into how the ARIMA model behaves under certain conditions and when it is most effective for time series forecasting. With this newfound understanding, I intend to apply these techniques and knowledge in my capstone project moving forward, building upon what Iâve learned here to explore more advanced time series models and their applications.

Bibliography

[1] Ning, Yanrui, Hossein Kazemi, and Pejman Tahmasebi. "A Comparative Machine Learning Study for Time Series Oil Production Forecasting: ARIMA, LSTM, and Prophet." *Computers & Geosciences*, vol. 164, 2022, 105126. Available online 6 May 2022.