UNIVERSIDADE FEDERAL FLUMINENSE

RODRIGO BARROSO RODRIGUES

TRATAMENTO DE INCERTEZAS 2024/1

Projeto de Avaliação de Dados de Medição em Redes

Niterói

2024

RODRIGO BARROSO RODRIGUES

TRATAMENTO DE INCERTEZAS 2024/1

Projeto de Avaliação de Dados de Medição em Redes

Trabalho de Medição e Análise de Dados em redes apresentado ao curso de Avaliação de Desempenho com o objetivo de apresentar um contato prático com resultados e análise de dados de técnicas de medição.

Professor: Antônio Augusto de A. Rocha

Niterói

2024

1. INTRODUÇÃO

Para a realização desse estudo, foram analisados dados obtidos através da ferramenta perfSONAR. Esta ferramenta é mantida pela Rede Nacional de Ensino e Pesquisa (RNP), a entidade responsável por manter o acesso à Internet para todas as instituições públicas federais de ensino e pesquisa no Brasil. Essa conectividade é feita através de Pontos de Presença (POPs) situados em cada uma das 27 unidades federativas do país, sendo identificados como POP-XX, sendo "XX" a representação da sigla da unidade federativa correspondente (por exemplo, MG para Minas Gerais, SC para Santa Catarina, etc).

Os dados extraídos e analisados são relativos aos eventos de Packet Retransmission (em português, Retransmissão de Pacotes, ou seja, a quantidade de pacotes retransmitidos) e Throughput (em português, Taxa de Transmissão, que se refere a quantidade de dados enviados num período), partindo do POP-SP (São Paulo) até o POP-RS (Rio Grande do Sul), no período de **26/03/2024 às 13:40:37 até 23/06/2024 às 11:14:52.**

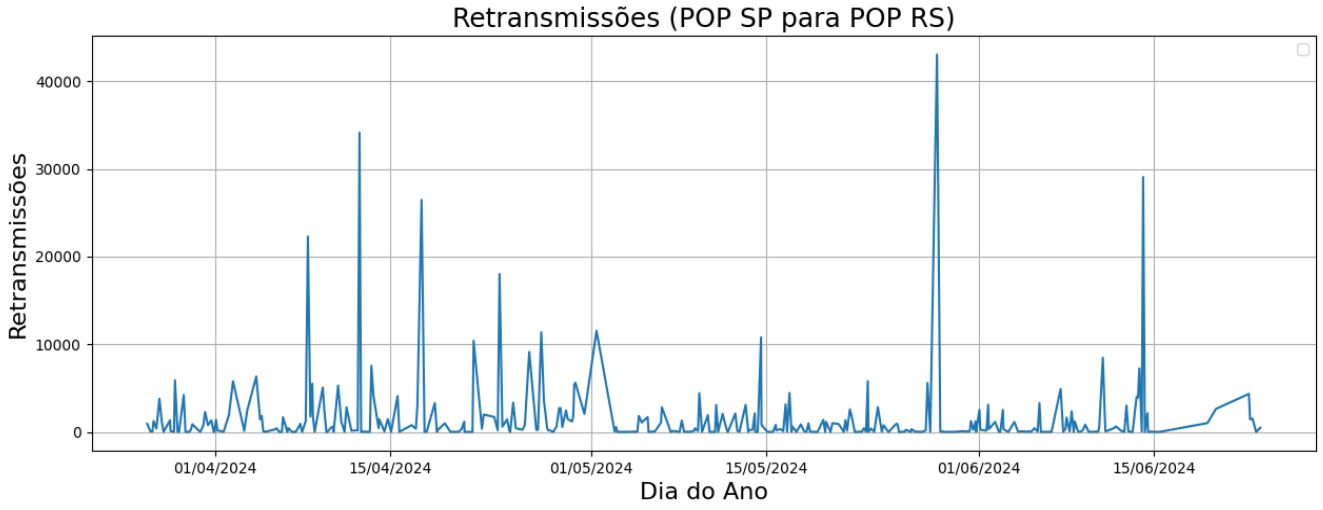
2. DADOS

Os dados utilizados na análise desse estudo, devidamente filtrados pelo período utilizado podem ser consultados através do seguinte link: Resultados de Teste perfSONAR - SP para RS. Além disso, os dados extraídos para o estudo utilizam a banda CUBIC (9999999999).

2.1. Dados de Retransmissão de Pacotes

Os dados de Retransmissão de Pacotes foram obtidos da rede RNP entre o POP-SP e o POP RS, no intervalo de **26/03/2024 às 13:40:37 até 23/06/2024 às 11:14:52***, pelo link da ferramenta Esmond do perfSonar: Dados Packet Retransmits*

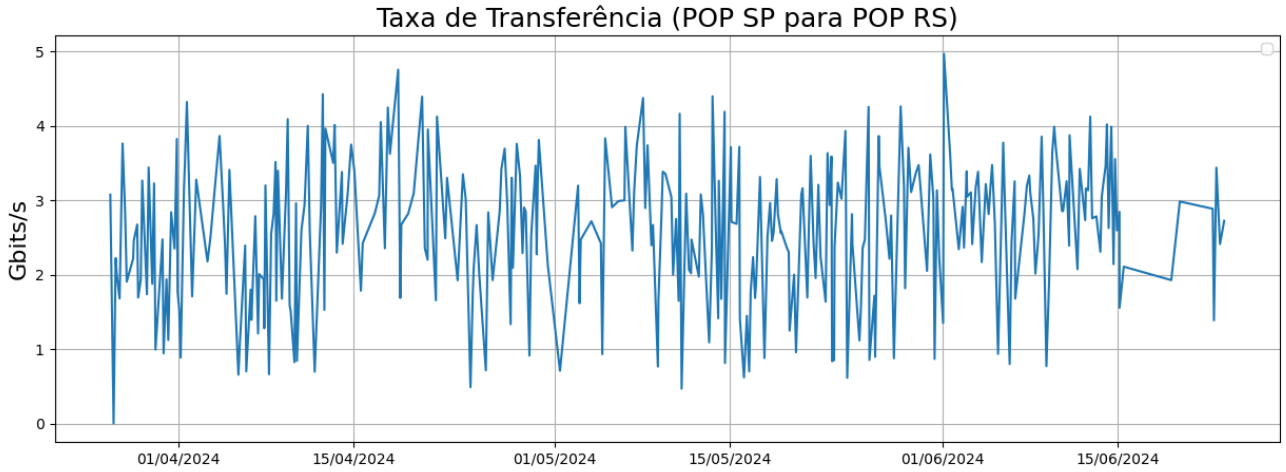
A partir dos dados extraídos, foi possível montar o seguinte gráfico, tendo o eixo x representando os dias do ano de 2024, e o eixo y representando a quantidade de pacotes retransmitidos:



2.2. Dados de Throughput

Os dados de Throughput foram também obtidos da rede RNP entre os mesmos POPs anteriores e no mesmo período, pelo link da ferramenta Esmond do perfSonar: Dados Throughput

A partir dos dados extraídos, foi possível montar o seguinte gráfico, tendo o eixo x representando os dias do ano de 2024, e o eixo y representando a taxa de transferência em Gigabits por segundo:



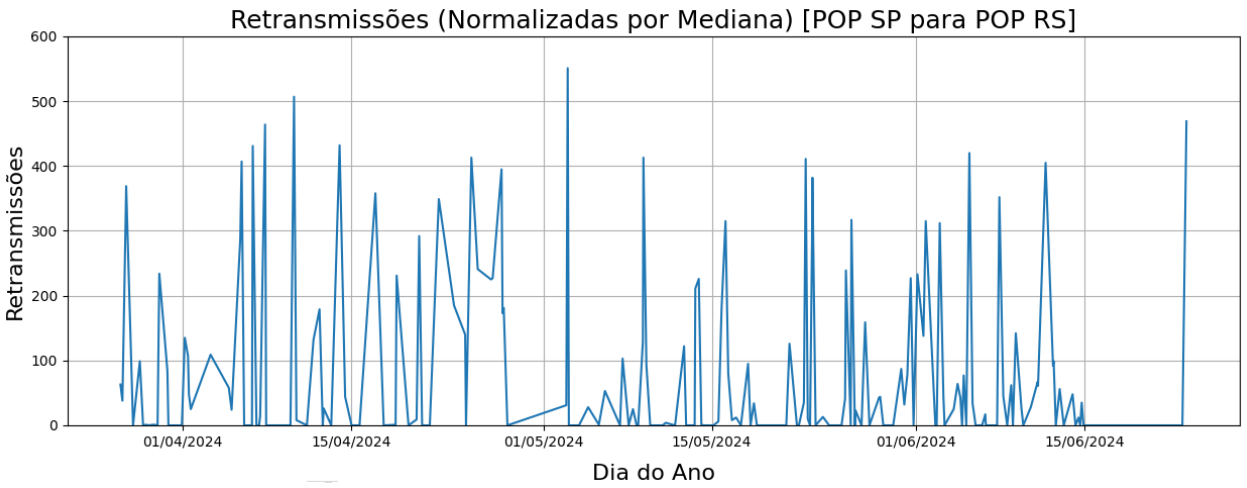
A partir dos dois gráficos apresentados, vemos que os mesmos apresentam muitos outliers, em outras palavras, valores que fogem bastante do padrão, sendo necessário removê-los através de um tratamento antes de começar a analisar os mesmos.

3. TRATAMENTO DOS DADOS

Como comentado anteriormente, é necessário tratar os dados antes de realizar as análises futuras. O principal motivo para esse tratamento é a grande quantidade de outliers, que podem acabar sendo levados em consideração na hora de plotar a linha prevista para o gráfico de previsão de comportamento desses eventos no futuro. Portanto, para normalizar o gráfico, com o objetivo de garantir que os intervalos de tempo sejam constantes, foi utilizado a mediana, e a partir dela, os outliers também foram removidos.

3.1. Tratamento dos Dados de Retransmissão de Pacotes

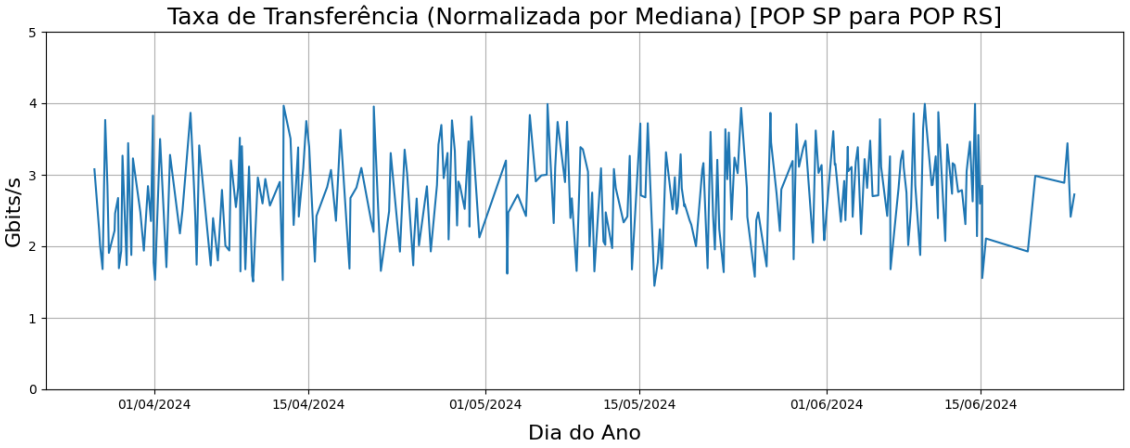
Após o tratamento, foi obtido o seguinte gráfico:



A partir da normalização proposta no gráfico acima, podemos ver que a grande maioria dos dias apresentava poucas retransmissões, porém antes da normalização o que mais chamava atenção eram os picos, que chegavam a milhares de retransmissões, que agora removidos, é possível trabalhar numa previsão.

3.2. Tratamento dos Dados de Throughput (Taxa de Transferência)

Após o tratamento, foi obtido o seguinte gráfico:



A normalização proposta para o Throughput deixou o gráfico mais padronizado, além de remover alguns outliers, podendo ver agora que não existe mais valores que sejam menores que 1Gbits/s e maiores que 4Gbits/s. A partir disso, também é possível trabalhar numa previsão.

4. ANÁLISE DOS DADOS

Para fazer a análise dos dados, esses que estão dispostos em forma de série temporal, escolhi por usar um modelo matemático para previsão de dados, o modelo ARIMA (AutoRegressive Integrated Moving Average, em português, modelo autorregressivo integrado de médias móveis). Para aplicar esse modelo, é necessário testar a estacionariedade ou não-estacionariedade das séries temporais dos eventos propostos acima. Esse teste é necessário pois quando é provado que uma série temporal é estacionária, isso significa que alguns valores como a média, a variância se mantém constante ao longo do tempo, e essas propriedades são necessárias para a aplicação do modelo ARIMA.

4.1. Teste de Estacionariedade via Dickey-Fuller

Para provar se as séries temporais dos eventos de retransmissão de pacotes e throughput são estacionárias, apliquei o teste de Dickey-Fuller Aumentado (ADF) através da biblioteca statsmodels da linguagem de programação Python.

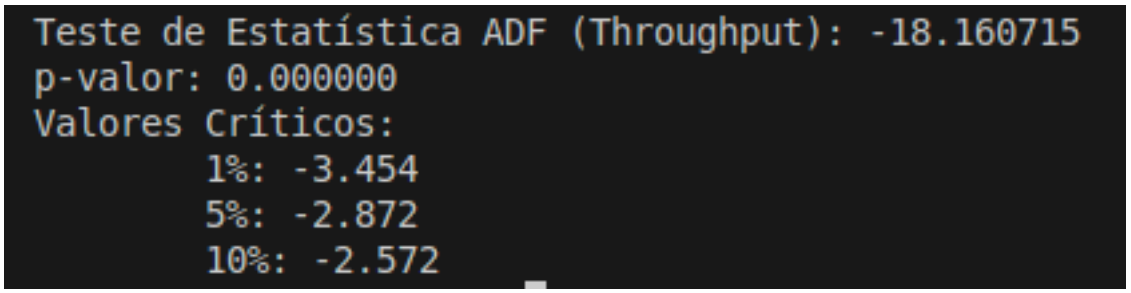
O teste ADF basicamente prova se uma série temporal é estacionária de 2 formas: A partir dos valores críticos de 1%, 5% e 10% calculados pelo teste sobre a série temporal, caso o valor estatístico de Dickey-Fuller seja menor que um de desses valores críticos calculados, e se o p-valor calculado for significativamente menor que a mesma percentagem do valor crítico.

4.1.1 Estacionariedade da série temporal de Retransmissão de Pacotes

```
Teste de Estatística ADF (Retransmissão de Pacotes): -5.852468
p-valor: 0.000000
Valores Críticos:
        1%: -3.463
        5%: -2.876
        10%: -2.574
```

A partir do teste ADF acima para o evento de Retransmissão de Pacotes, podemos ver que o valor de –5.85 é menor que o valor crítico de 5%, que é -2.87. Além disso, o p-valor sendo 0.00, bem menor que 0.05, portanto a série temporal em questão é estacionária, sendo possível aplicar o modelo ARIMA.

4.1.2. Estacionariedade da série temporal de Throughput



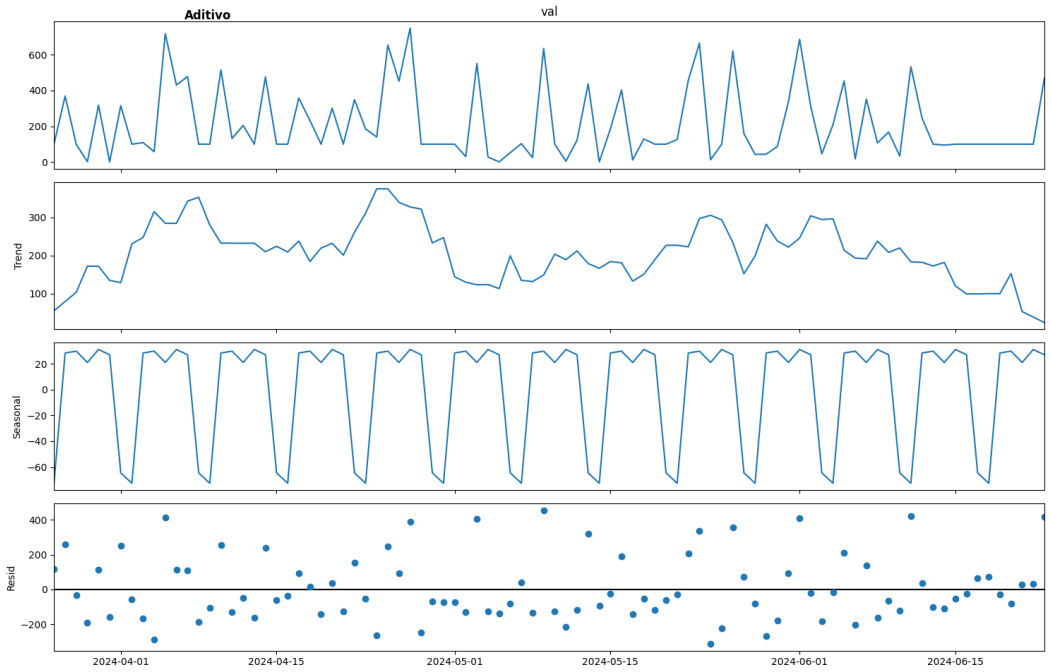
Da mesma forma do teste anterior, o teste ADF para o evento de Throughput também é estacionário pois o seu valor é bem menor que o valor crítico de 5% e além disso, o p-valor também é 0.00, logo, essa série temporal também pode ter o modelo ARIMA aplicado por ser estacionária.

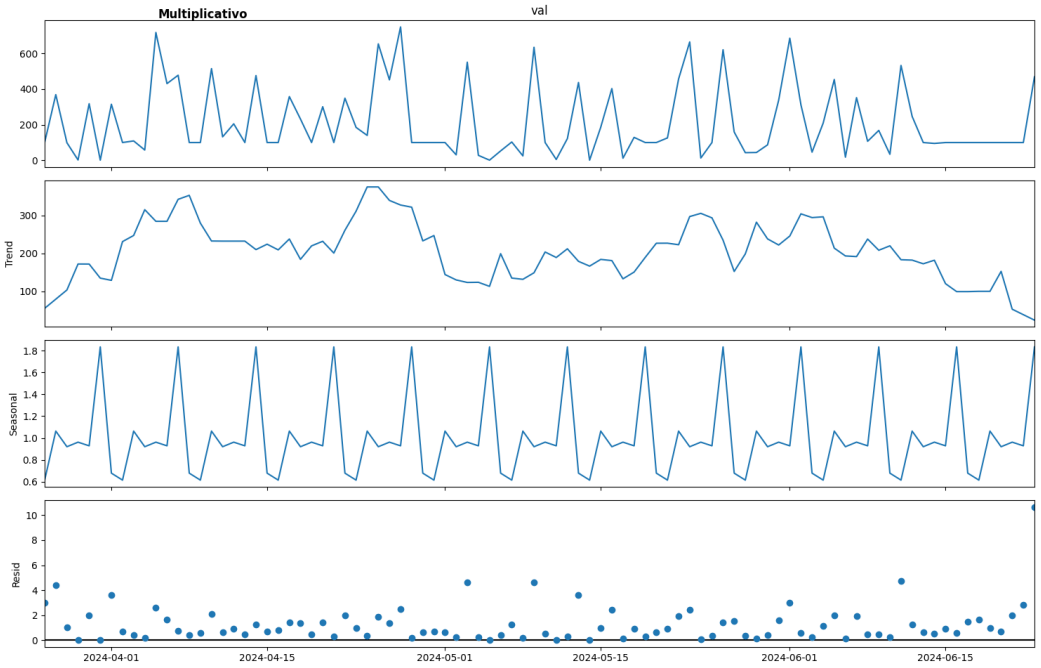
5. DECOMPOSIÇÃO DOS DADOS

Após os 2 testes ADF demonstrarem que as séries são estacionárias, realizei a decomposição das 2 séries com o objetivo de entender mais sobre elas, e a partir disso foi possível observar a sazonalidade dos dados, mostrando como eles se repetem ao longo do tempo. Para isso, foi utilizado o módulo seasonal_decompose da biblioteca statsmodels da linguagem de programação Python.

5.1. Decomposição da série temporal de Retransmissão de Pacotes

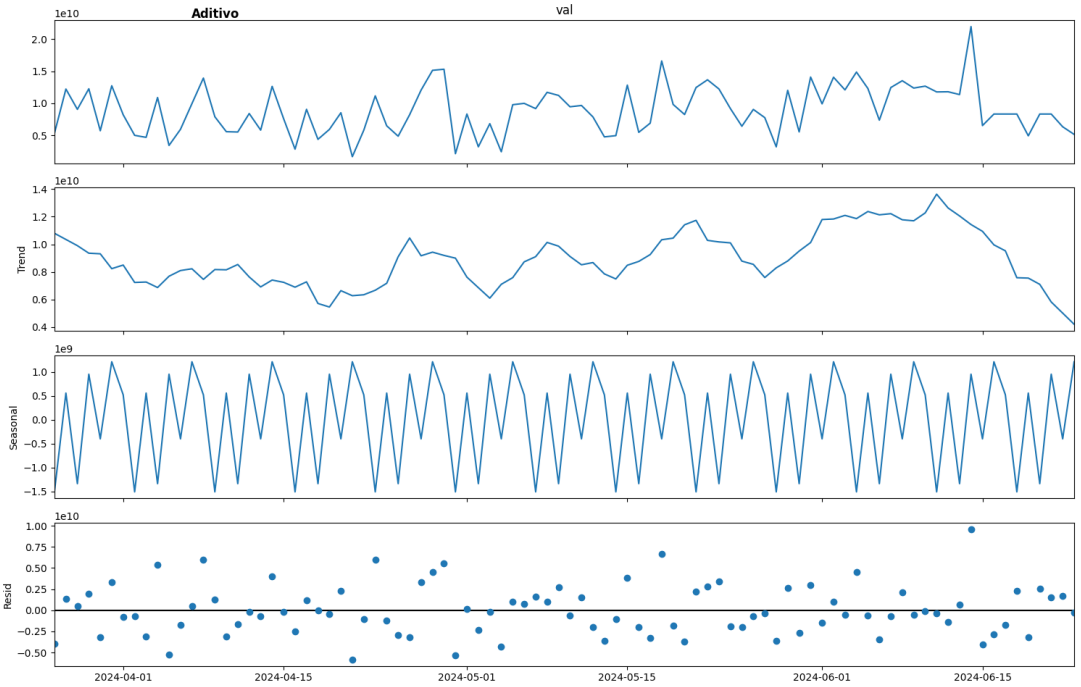
Na decomposição da série de retransmissão de pacotes, foi aplicado 2 métodos, aditivo e multiplicativo, com o objetivo de entender se haveria muitas divergências, porém ambos os métodos apresentam resultados bastante semelhantes. Abaixo, os gráficos, sendo que para cada um, o primeiro representa os dados em si, o segundo a tendência, o terceiro a sazonalidade e o quarto os resíduos.

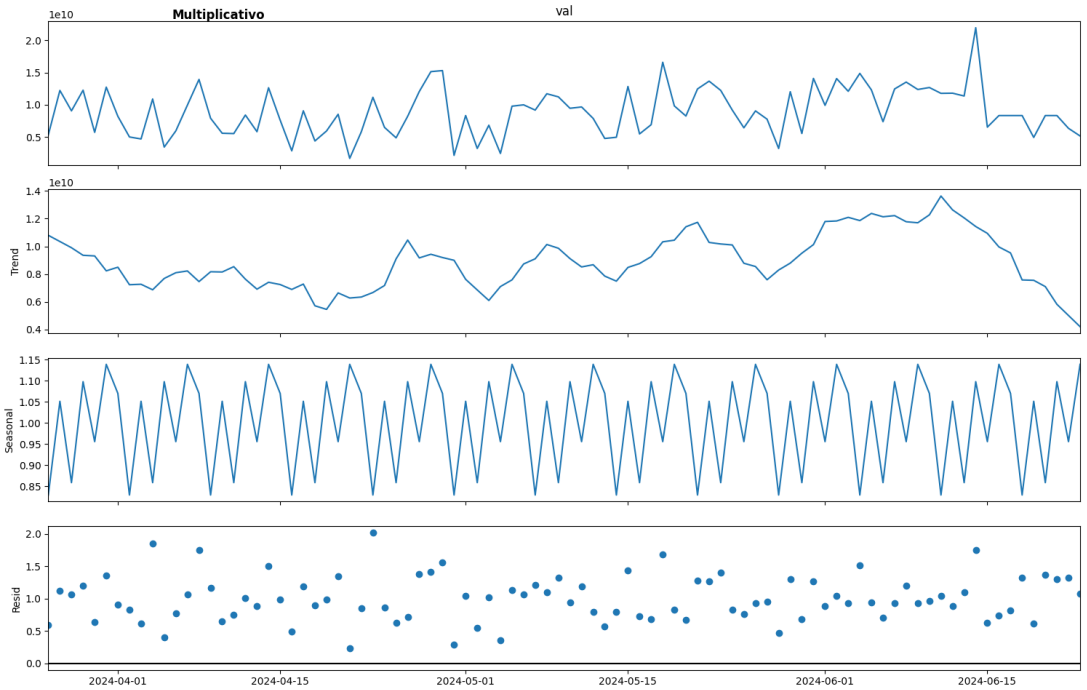




5.2. Decomposição da série temporal de Throughput

Similar a decomposição da série anterior, foram aplicados ambos os métodos, além disso, os resultados foram ainda mais semelhantes. Além disso, ambos a série tem a tendência de queda.



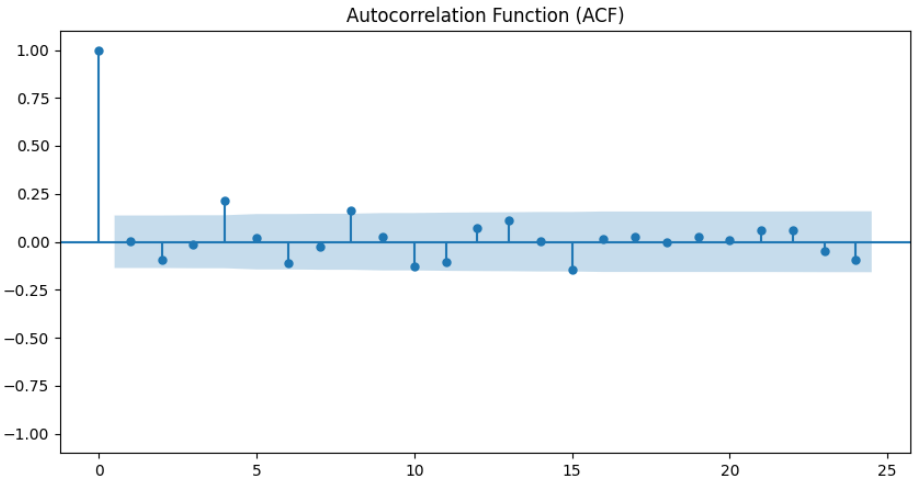

Multiplicativo
val
1e10
2.0
1.5
1.0
0.5
1e10
1.4
1.2
1.0
0.8
0.6
0.4
Trend
1.15
1.10
1.05
1.00
0.95
0.90
0.85
Seasonal
2.0
1.5
1.0
0.5
0.0
Resid
2024-04-01
2024-04-15
2024-05-01
2024-05-15
2024-06-01
2024-06-15

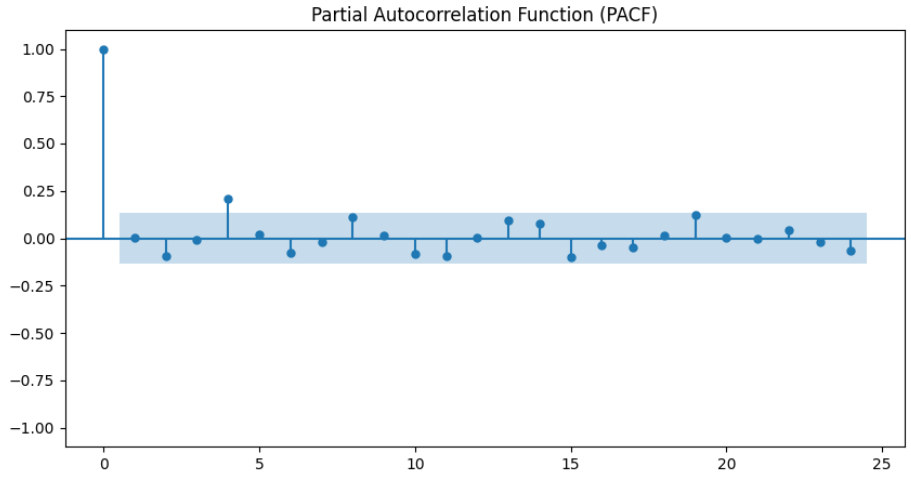
6. APLICAÇÃO DO MODELO ARIMA

Para a aplicação do modelo ARIMA, antes é necessário calcular 2 parâmetros necessários, os parâmetros p e q, que são os valores específicos para cada série de entrada.

A obtenção desses valores é feita a partir da análise dos gráficos das funções Autocorrelação (ACF) e da Autocorrelação Parcial (PACF). A partir desses gráficos, podemos encontrar p, que é o valor aproximado para o teto, sendo o primeiro valor que sai do intervalo de confiança da PACF, e q, que é o primeiro valor que sai do intervalo de confiança da ACF.

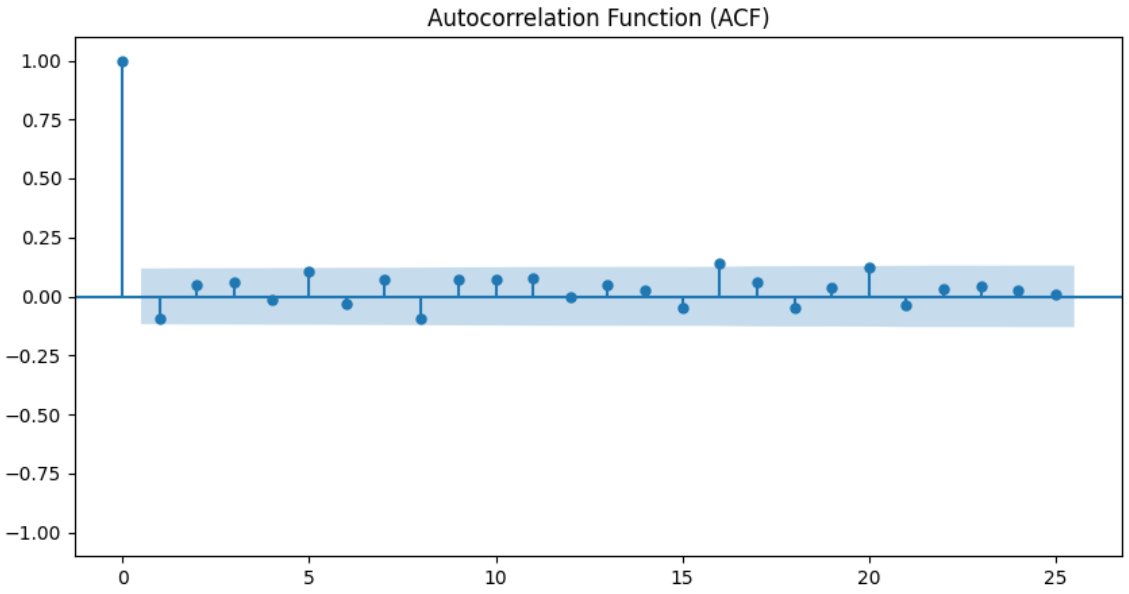
6.1. Encontrando P e Q da série de Retransmissão de Pacotes

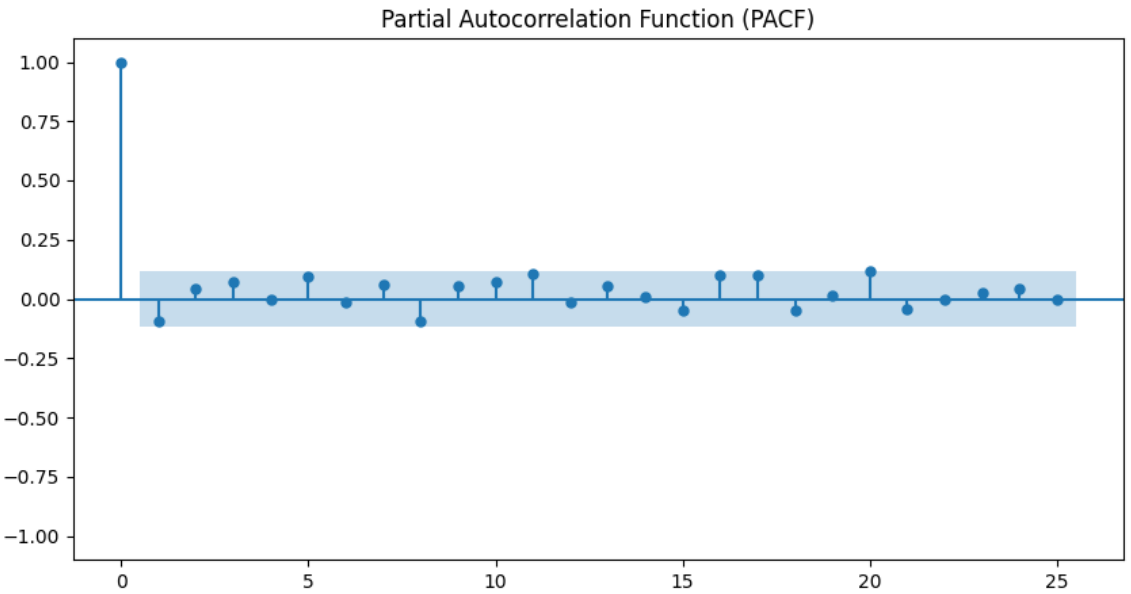




É possível perceber a partir dos gráficos de ACF e PACF que o valor P e Q para a série de retransmissão de pacotes será igual para ambos, e será no caso 1.

6.2. Encontrando P e Q da série de Throughput

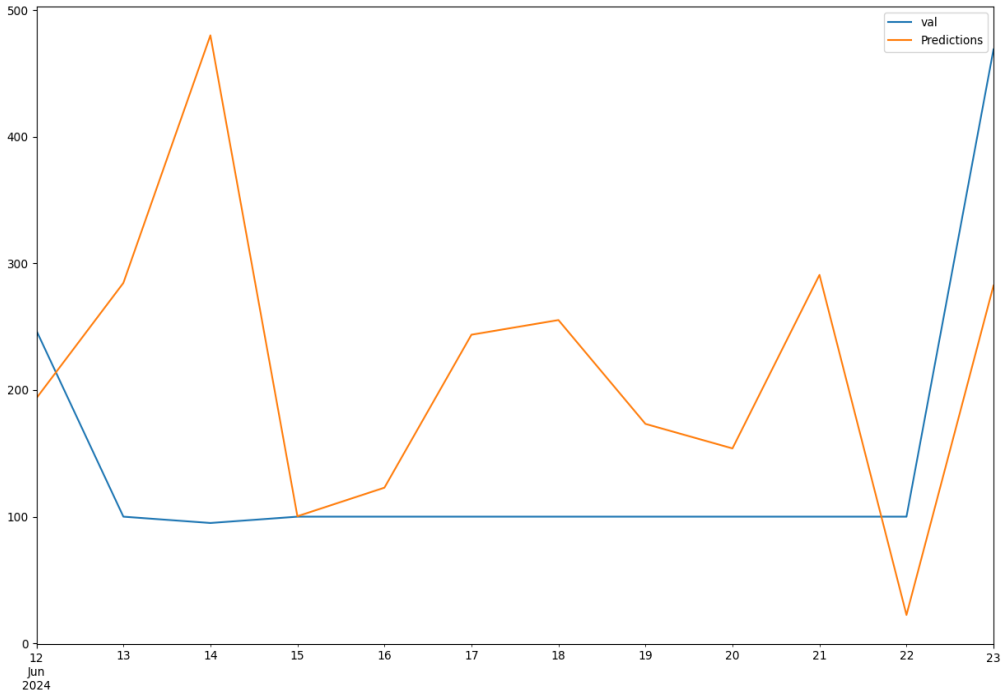




Assim como na série de retransmissão de pacotes, a série de throughput também tem seus valores p e q iguais, e também são iguais a 1.

6.3. Aplicação do ARIMA à série de Retransmissão de Pacotes

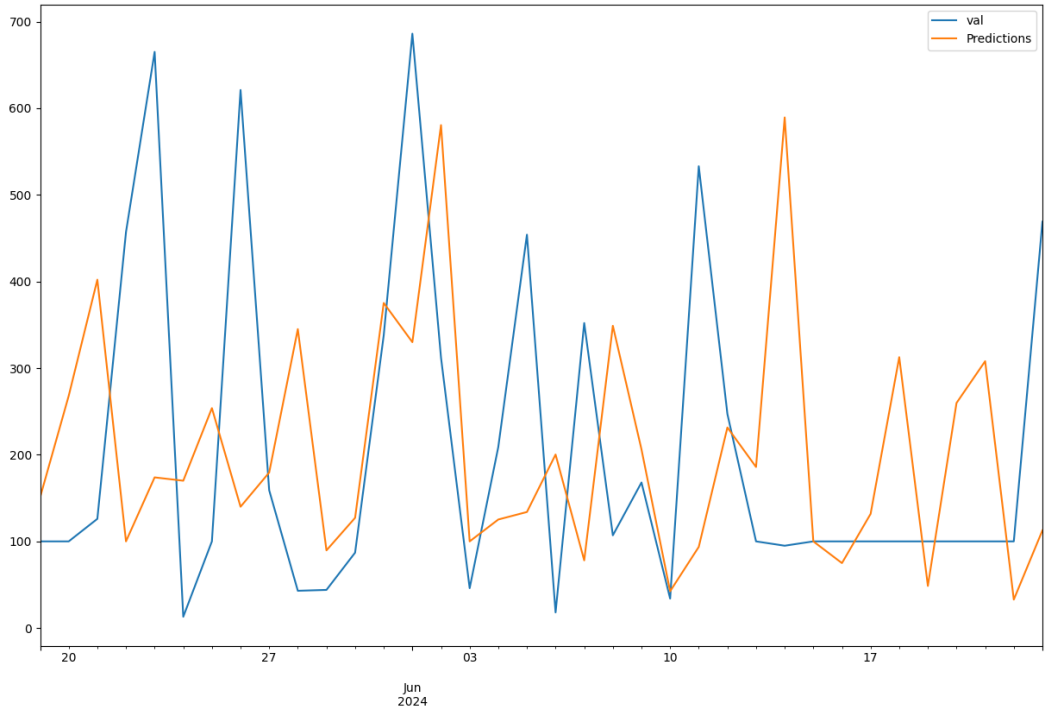
Para verificar se a aproximação dos dados é boa o suficiente, testei o modelo em uma parte específica do gráfico de retransmissão de pacotes. O gráfico abaixo mostra em azul o valor real extraído do perfSONAR e em laranja a previsão de como a série temporal se comportaria no período:



Apesar disso, o teste mostrou uma grande falha ao longo da série temporal de dias, porém, ao chegar no penúltimo, ele conseguiu prever o crescimento da curva.

Além disso, testando outras opções de valores para o treino não apresentaram gráficos melhores.

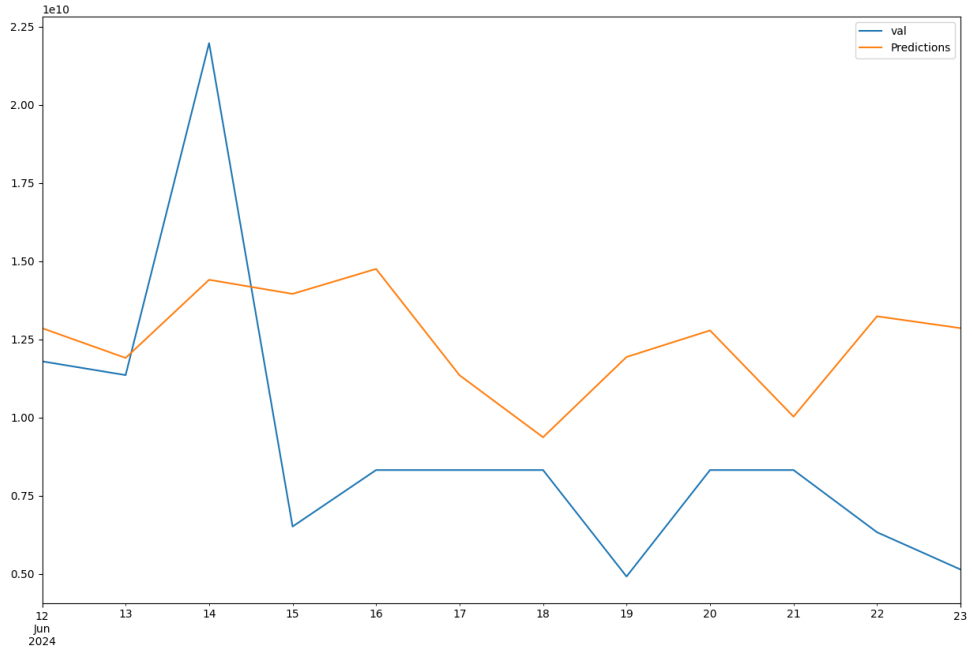
No entanto, ao olhar para outros períodos, como do dia 20 de maio até 22 de junho, o gráfico parece seguir a tendência:



Logo, o período de 13 a 22 de junho pode ser tratado como uma exceção, pois apresentou um grande vale no meio de um período com muitos picos e pequenos vales.

6.4. Aplicação do ARIMA à série de Throughput

De forma semelhante a série de retransmissão de dados, vamos agora testar no mesmo período o modelo ARIMA para a série de throughput:

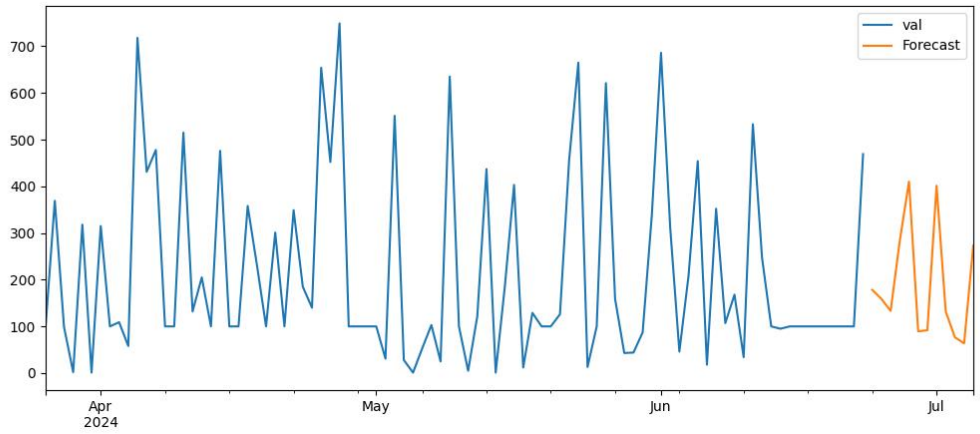


Por sua vez, a curva laranja de previsão do modelo conseguiu prever de forma relativamente razoável a curva azul, porém não levando em consideração o grande crescimento no dia 14 de junho de 2024 e também apresentando a mesma tendência da curva azul, mas sendo um pouco atrasada, tendo por exemplo o vale no dia 19 de junho acontecendo no dia 21 de junho na previsão.

7. PREVISÕES

7.1. Previsão para a série temporal de Retransmissão de Pacotes

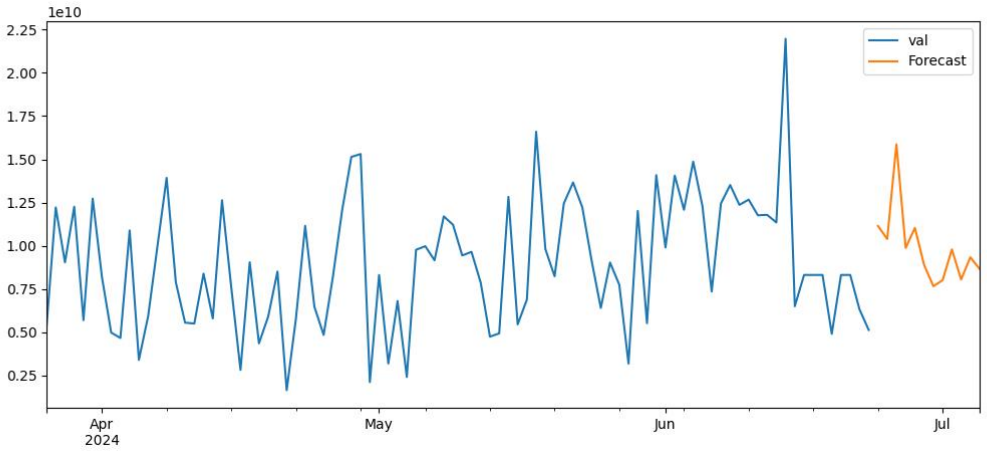
No gráfico abaixo, a linha azul representa a série temporal de Retransmissão de Pacotes, e a linha laranja representa a previsão proposta pelo modelo ARIMA.



Podemos ver que apesar do teste feito no primeiro período ter sido de certa forma ruim, o segundo mostrou que o modelo tinha o potencial de prever uma tendência futura, e de fato, gerou curvas possíveis de acontecer.

7.2. Previsão para a série temporal de Throughput

Similar ao gráfico anterior, a linha azul representa a série temporal de Throughput, e a linha laranja representa a previsão proposta pelo modelo ARIMA.



Em relação a previsão da série temporal de Throughput, essa mostra um cenário inverso a curva da série de retransmissão, pois a primeira mostra uma tendência inicial de queda, enquanto essa mostra uma tendência inicial de crescimento. Mais uma vez, olhando para o gráfico ao longo do tempo, a previsão do modelo mostra uma curva bem factível, com picos e vales rápidos.

8. CONCLUSÃO

Por fim, podemos concluir que para a série temporal de Retransmissão de Pacotes, pode não ter tido uma boa previsão feita pelo modelo ARIMA, devido aos dados não terem sido suficientes para sua análise ou pelo próprio modelo não ter conseguido descrever o dado, apesar disso, olhando um período 36 dias antes do último dia coletado, apresentou curvas bem mais condizentes, o que pode ter levado a previsão que foi relativamente razoável.

A previsão da série temporal de Throughput por sua vez, descreveu bem melhor o comportamento da curva dos dados, assim, conseguiu prever com uma maior qualidade.

Documentações:

1. *Documentação Statsmodels: statsmodels.org*
2. *Documentação PerfSONAR: docs.perfsonar.net*
3. *Documentação Pandas: pandas.pydata.org*
4. *Documentação MatplotLib: matplotlib.org*
5. *Documentação Numpy: numpy.org*

Código desenvolvido durante esse estudo disponível no GitHub.