ACT 2 - Anomalous Detection


```
Eric Gómez - A01378838
Felipe Osornio - A01377154
Rafael Moreno - A01378916
Uriel Pineda - A01379633
Hector Hernandez - A01374009
```


Introduction

Networks and distributed processing systems have become an essential technology in any Enterprise Environment. The rapid growth of the amount of data that those environments have to deal has given rise to a depletion in expertise of human operators to manage them. A lot of efforts has gone into developing systems and protocols for collecting network traffic statistics. One of the approaches researchers found was the ARIMA model.

An ARIMA model is a class of statistical model for analyzing and forecasting time series data. ARIMA is an acronym that stands for AutoRegressive Integrated Moving Average. It is a generalization of the simpler AutoRegressive Moving Average and adds the notion of integration.

In this practice, the dataset analysis was made that includes the following attributes:

`ts` , `uid` , `id_orig_h` , `id_orig_p` , `id_resp_h` , `id_resp_p` , `trans_depth` , `method` , `host` , `uri` , `referrer` , `user_agent` , `request_body_len` , `response_body_len` , `status_code` , `status_msg` , `info_code` , `info_msg` , `filename` , `tags` , `username` , `password` , `proxied` , `orig_fuids` , `orig_mime_types` , `resp_fuids` , `resp_mime_types` .

Our main goal is to run the ARIMA algorithm to this dataset in order to determine the performance and forecast differences between bayesian and classical inference models.

Methodology

For this analysis, the data set from `http.log` was taken. All records contained here are HTTP transaction. As a first step, it was decided to format the information and use only four columns out of all those available. The reason it was decided to take `request_body_len` was because most of the columns only represented text strings, whereas for the case of `request_body_len` and `response_body_len` the representation was an integer. Finally, it was chosen to take `request_body_len` since its values presented anomalies in the size of the request.

```
**********************************************************************
 http-1.log INFO
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 807537 entries, 0 to 807536
Data columns (total 4 columns):
 #   Column            Non-Null Count   Dtype
---  ------            --------------   -----
 0   ts                807537 non-null  datetime64[ns]
 1   id_orig_h         807537 non-null  object
 2   id_resp_p         807537 non-null  int64
 3   request_body_len  807537 non-null  int64
dtypes: datetime64[ns](1), int64(2), object(1)
```

```
memory usage: 24.6+ MB
None
```

As shown, the timestamp, source IP, the response port and the request body length were selected from the entire data set. For training, 80% of the entire data set was used, while for the tests only 20% of the original size was taken. The following shows how the data set was formatted after training.

Training

```
**********************************************************************
 train_DF
                             ts      id_orig_h  id_resp_p  request_body_len
0      2012-05-30 19:09:27.177343  192.168.88.10         80                 0
1      2012-05-30 19:09:28.343725  192.168.88.10         80                 0
2      2012-05-30 19:09:29.124170  192.168.88.10         80                 0
3      2012-05-30 19:09:29.142869  192.168.88.10         80                 0
4      2012-05-30 19:09:29.602005  192.168.88.10         80                 0
...                           ...            ...        ...               ...
646024 2014-03-14 22:17:08.208324  192.168.54.10         80                 0
646025 2014-03-14 22:17:10.428163  192.168.54.10         80                 0
646026 2014-03-14 22:17:10.893226  192.168.54.10         80                 0
646027 2014-03-14 22:17:11.695421  192.168.54.10         80                 0
646028 2014-03-14 22:17:11.693397  192.168.54.10         80                 0

[646029 rows x 4 columns]
```

Test

```
**********************************************************************
 test_DF
                             ts      id_orig_h  id_resp_p  request_body_len
646029 2014-03-14 22:17:12.630713  192.168.54.10         80                 0
646030 2014-03-14 22:17:13.736534  192.168.54.10         80                 0
646031 2014-03-14 22:17:15.287393  192.168.54.10         80                 0
646032 2014-03-14 22:17:15.924981  192.168.54.10         80                 0
646033 2014-03-14 22:17:15.927022  192.168.54.10         80                 0
...                           ...            ...        ...               ...
807532 2013-03-13 14:01:49.374728  192.168.42.10         80                 0
807533 2013-03-13 14:01:49.368503  192.168.42.10         80                 0
807534 2013-03-13 14:01:49.514270  192.168.42.10         80                 0
807535 2013-03-13 14:02:48.369699  192.168.42.10         80                 0
807536 2013-03-13 14:02:48.355899  192.168.42.10         80                 0

[161508 rows x 4 columns]
```

In the console printing of training there are around 646000 records taken for training and for printing test; about 161000 were taken. Due to the nature of the ARIMA model as a time series, it is not possible for the data to be generated randomly. That is, the intention of the model is to find an anomaly or even a forecast of how the data could be behaving, therefore the order of each record is important, if this order is lost, one of the most important factors within of the models.

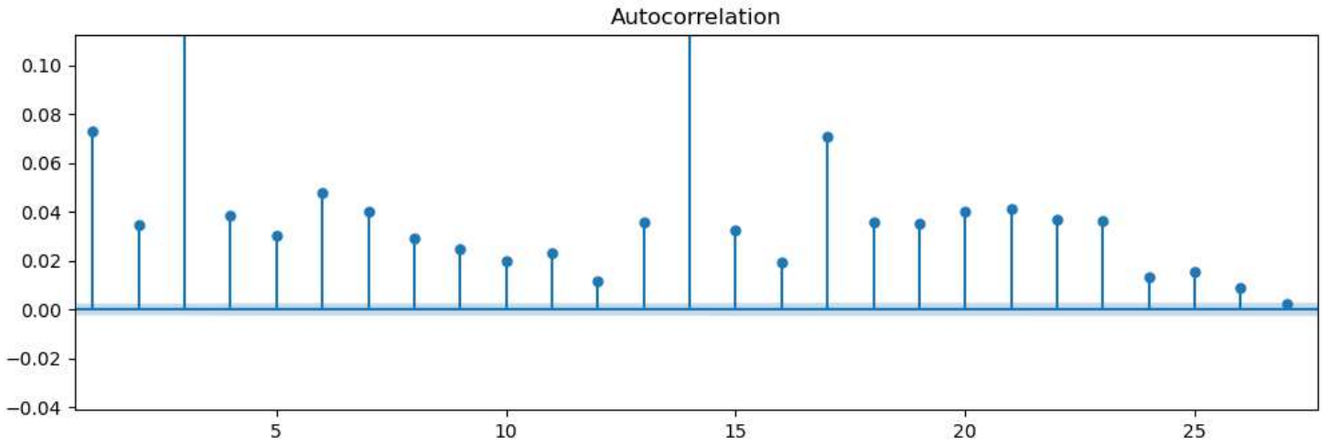
After the Ad Fuller Test, all the analysis was performed and the graphs of the original series, of the [autocorrelation] (# autocorrelation) and the partial autocorrelation.

Ad Fuller Test

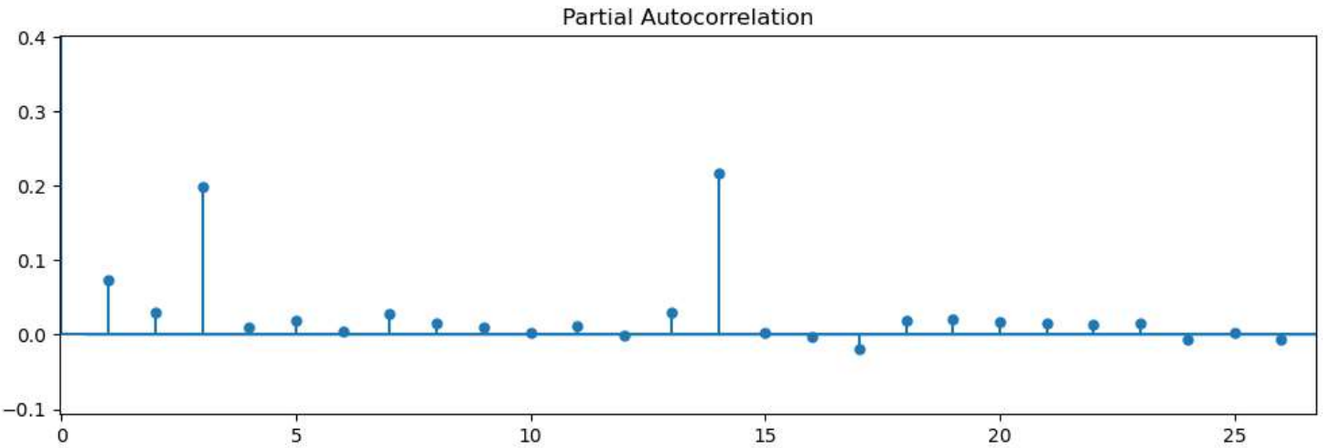
```
**********************************************************************
 Training...
```

```
**********************************************************************
 ADF Test Statistic
ADF Test Statistic : -98.85731665469174
p-value : 0.0
#Lags Used : 52
Number of Observations : 645976
strong evidence against the null hypothesis(Ho), reject the null hypothesis. Data is stationary
```

Autocorrelation



Partial autocorrelation



Finally, a comparison was made between both inference models: classical and Bayesian. Within this analysis, the difference between MLE and PML was compared by the classical models, on the other hand, in the Bayesian models the Laplace model and the Metropolis-Hastings model were analyzed.

Finds

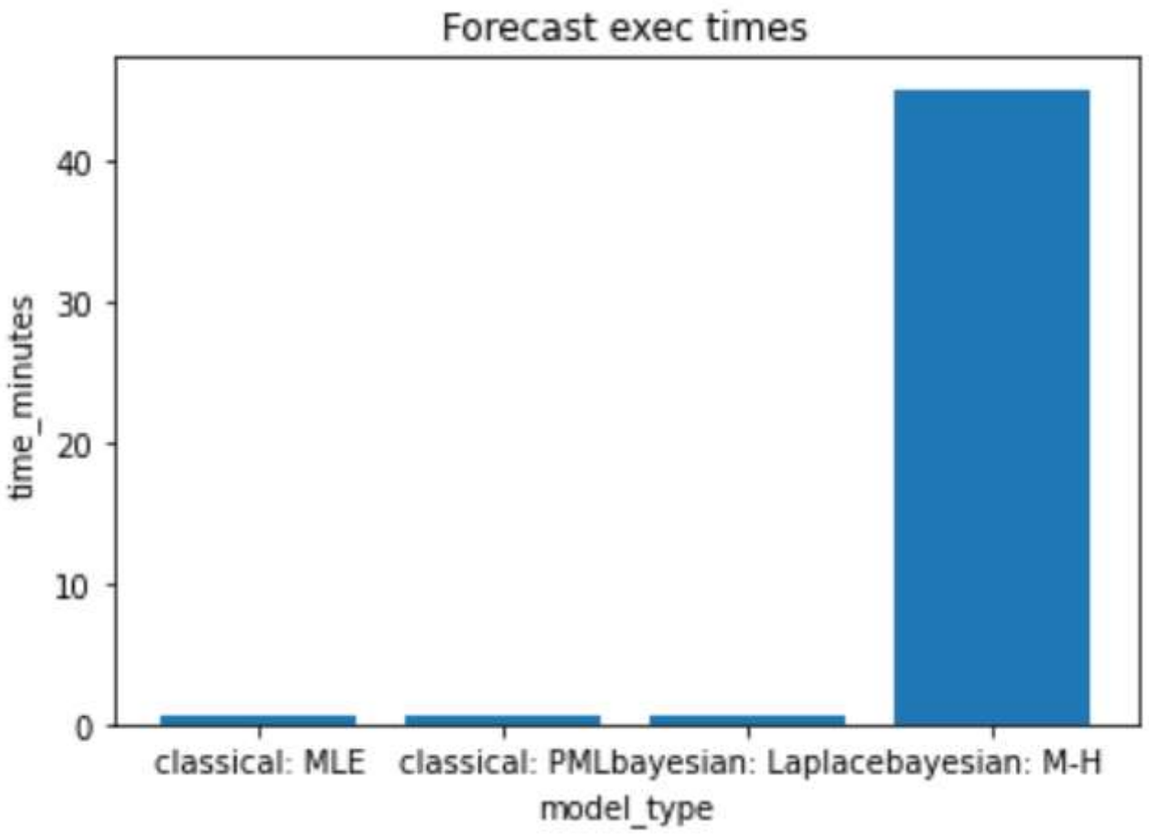
During this analysis, different important details were identified. The first of these was the size of the initial dataset. When processing it, there came a time when the use of RAM memory reached 100%, in a memory of 32GB, so trying to do the analysis on computers with little storage in RAM would be somewhat complex to conclude the analysis.

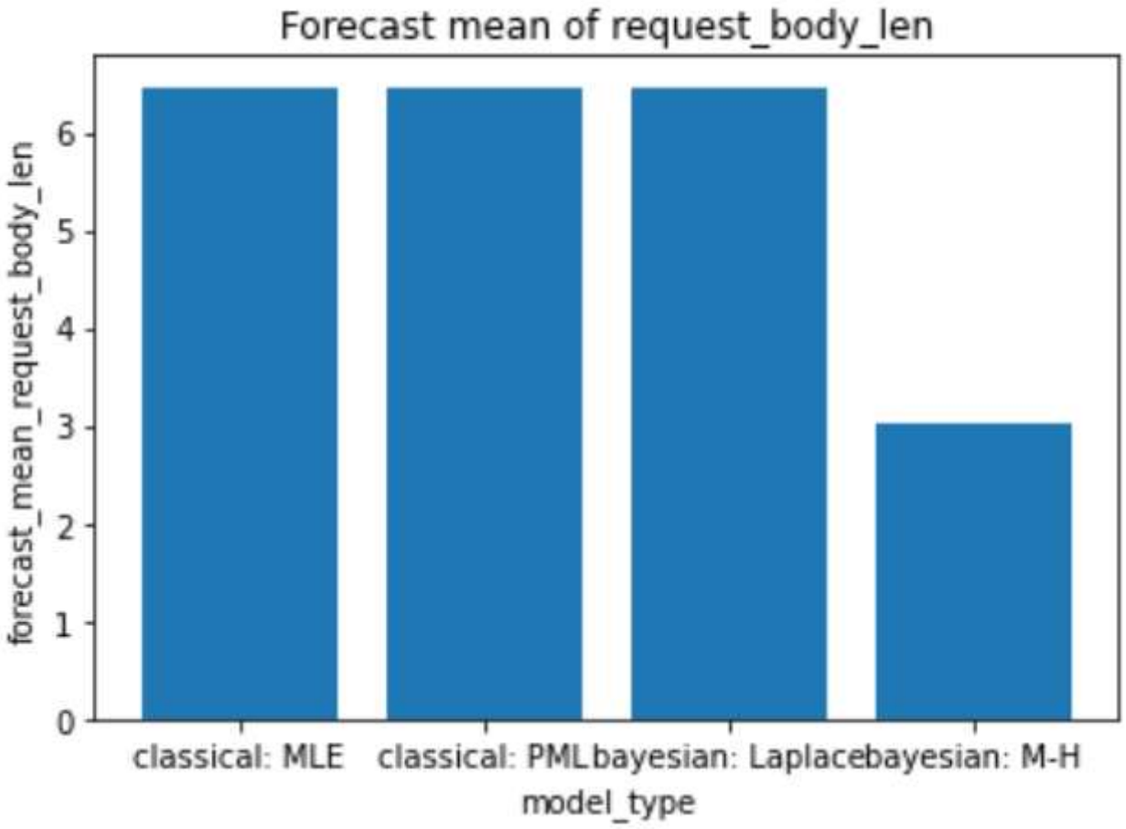
Second, when inspecting the information contained in the correlation and partial correlation graphs, patterns were found between the time segments. The analysis was only developed in a range from 1 to 6. The reason why the entire segment was not analyzed was for the following reasons:

1. The sample size turned out to be too heavy to do the analysis in a range of 1 to 26. For example, the following printout details the estimated times for each of the models run. The first three models (MLE, PML and Laplace) result in a fairly short time, considering the last one, which took 45 minutes.

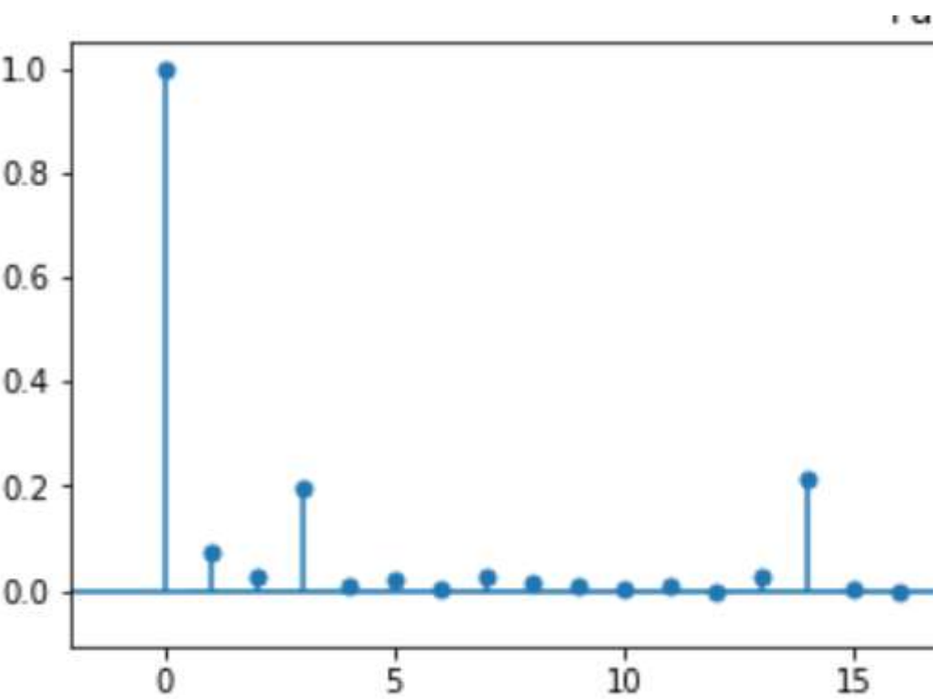
```
**********************************************************************
 Forecast exec times
          model_type     time  forecast_mean_request_body_len
0     classical: MLE   0.5450                        6.460586
1     classical: PML   0.5503                        6.460572
2  bayesian: Laplace   0.6027                        6.460572
3      bayesian: M-H  45.0088                        3.037051
```

The comparison of time and the mean of the predicted values between the models is graphically displayed here. We can se that the algoritm of Metropolis-Hastings isn't the optimal for this tipe of data forecast, justified by the huge diferences in execution time and the avarage of predicted results.





2. It was identified that between the correlation and partial correlation graphs, there are patterns in the range from 2 to 5 and and from 13 to 16. For this same reason, it was also decided to discard the analysis of the entire range from 1 to 26.



As a last point, for the comparison of both inference models: classical and Bayesian, there were complications when obtaining two packages: OLS and BBVI. Both packages marked an internal error in `pyflux`, so the decision was made to only do the analysis using MLE, PML, Laplace and Metropolis-Hastings.

Conclusion

Concluding, we can determine that the classical models are much faster than the Bayesian models. Likewise, the difference between each of these models is quite significant. Classical statistics has foundations under the concept of experience. the analysis is taken through historical data in order to

give an estimate. In contrast, Bayesian models focus mainly on the uncertainty factors under the concept of limited knowledge, which is why the mean and median are favorable tools for these models.