Earthquake Predictions using the Time Series and Machine Learning methods.

Max Kenworthy, Anuj Verma, and Nisarg Patel

Abstract: Predicting the timing and magnitude of an earthquake is a fundamental goal of geoscientists. In a laboratory setting, we show we can predict “labquakes” by applying new developments in machine learning (ML), which exploits computer programs that expand and revise themselves based on new data. We use time-series techniques to identify telltale sounds—much like a squeaky door— that predict when a quake will occur. The experiment closely mimics faulting in the Earth's crust, so the same approach may work in predicting timing, but not size, of an earthquake.

The original dataset was a very long series of these labquakes recorded from the experiment. We reduced the data to a more manageable size for analysis by averaging chunks at a time. To better understand our data and prepare for analysis, we created visualizations to explore the overall features and characteristics of the data. After gaining some insight, we implemented different machine learning techniques that could effectively predict the timing of an earthquake. Many techniques were trained and evaluated on our data, however, the error in our final results were too high to definitively say the model could be used to predict the timing of an earthquake.

---

1. Introduction

A large focus of Earth science research belongs to forecasting the timing and severity of earthquakes. Earthquakes are notoriously difficult to predict and can have devastating impact on those unfortunate enough to experience one. One of the few predictors of earthquakes known to scientists is overall seismic activity measured from vibrations caused by movement in tectonic plates in the earth's crust. Los Alamos National Laboratory (LANL) is an organization studying the predictive effects of seismic waves leading up to an earthquake. LANL is currently hosting a Kaggle competition that tasks participants to predict when an earthquake will occur given time series seismic data. Forecasting an earthquake has obvious health and financial incentives for society as a whole and should be addressed as a priority. Our goal is to forecast when the next laboratory earthquake will take place based on seismic data using time series analysis and machine learning techniques.

In this study, stochastic models known as autoregressive integrated moving average (ARIMA) models in combination with a linear regression were used to predict the timing of lab generated earthquakes. We construct a reduced time series from the original dataset and perform exploratory analysis for a basis to start modeling. An appropriate ARIMA model is fit to the time series and then used to provide predictions of seismic data. Finally, those predictions are fed to a linear regression to predict time_to_failure.

2. Related Work

The previous studies in this geological field have used a more ideological approach that study an laboratory experiment with ideal scenarios of an earthquake. In those scenarios, a naïve model based on periodicity of events (average interventions of time) can be used to

predict the time to next earthquake. However, the performance of this naïve model has been reported very low as compared to other sophisticated machine learning algorithms such as Random Forest [2]. More research papers show the use of radial bias function (RBF) neural network (NN) models to further improve the performance of models to predict the time for next earthquake [6,7]. This study tries to address a more realistic scenario of earthquake and the data generated from the laboratory experiment shows more similarity to the actual earthquakes. We try to tackle current problem using a mix of previously proven techniques with timeseries approach.

3. Algorithms

3.1 Linear Regression

Linear regression is a statistical model that measures the linear relationship between a response (dependent) variable and explanatory (independent) variables. The relationships are modeled using a linear function whose model parameters are estimated from the data [3]. Linear regression is often used for forecasting by fitting a model to training data to establish parameter values then inputting values of the explanatory variable to the model equation (below) to make a prediction of the response variable.

$$y_i = \beta_0 1 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip} + \varepsilon_i = \mathbf{x}_i^{\mathsf{T}} \boldsymbol{\beta} + \varepsilon_i, \qquad i = 1, \ldots, n, \quad [3]$$

3.2 Support Vector Machine

Support vector machines (SVM) are supervised machine learning algorithms that model data as points in multidimensional space. These points are mapped to maximize their difference so there is a wide gap or margin between dissimilar points. For prediction, new points are plotted into that space and typically projected to belong to one side of the margin. A version of SVM's can be used for regression (SVR) that predicts a value within a margin based on equation:

$$y = \sum_{i=1}^{N} (\alpha_i - \alpha_i^*) \cdot \langle x_i, x \rangle + b \quad [4]$$

3.3 Random Forest

A random forest is an ensemble technique that aggregates multiple decision trees that have been trained on different samples of the data. A prediction from a random forest is essentially an average of the outputs of the decision trees. If being used for regression, the random forest algorithm trains multiple decision trees that predict a numerical value then those outputs are bagged together and averaged. Equation below depicts how predictions are calculated for unseen samples (*x'*) using *B* number of trees:

$$\hat{f} = \frac{1}{B} \sum_{b=1}^{B} f_b(x') \quad [5]$$

3.4 ARIMA

An autoregressive integrated moving average model, the future value of a variable is assumed to be a linear function of several past observations and random errors. That is, the underlying process that generate the time series has the form

$$y_t = \theta_0 + \phi_1 y_{t-1} + \phi_2 y_{t-2} + \cdots + \phi_p y_{t-p}$$
$$+ \varepsilon_t - \theta_1 \varepsilon_{t-1} - \theta_2 \varepsilon_{t-2} - \cdots - \theta_q \varepsilon_{t-q},$$

where yt and t are the actual value and random error at time period t, respectively; [6]

4. Methodology

4.1 Data Overview

The dataset was provided by Los Alamos National Laboratory (LANL) on Kaggle platform as part of a competition to predict the earthquake using the real-time seismic/acoustic data. The data was generated artificially during an experiment by LANL using a device known as a “classic lab earthquake model”. The classic lab earthquake model simulates the cycle of loading and failure on tectonic faults.

The dataset contains one huge wave recording with more than 600 million data points. The seismometer used to detect seismic waves can record the signal continuously for 0.0375 seconds before giving a small lag. The two variables used in this experiment are:

acoustic_data: [input variable] The acoustic data is also called seismic wave data detected by seismometer. The amplitude of the waves is recorded at 4 MHz. One chunk of data is received after 0.0375 seconds with 150000 records.

time_to_failure: [output variable] represents the time in seconds until the next laboratory earthquake

4.2 Data Processing

A new feature was created using the original dataset. This feature takes the chunks of 50,000 records and average over them. We take average of 50,000 records mainly for two reasons: first, to reduce the size of data substantially and second, to retain at least 3 records per seismic signal received before the lag-time. This newly created feature was used in our further experimental procedures explained in section 5.1

Final dataset contains 10,485 records.

4.3 Metrics

Below metrics were used to compare different models and results.

4.3.1 Akaike Information Criteria (AIC)

The Akaike information criterion (AIC) is an estimator of the relative quality of statistical models for a given set of data. Given a collection of models for the data, AIC estimates the quality of each model, relative to each of the other models. Thus, AIC provides a means for model selection**.** AIC is founded on information theory. AIC estimates the relative amount of information lost by a given model: the less information a model loses, the higher the quality of that model. In estimating the amount of information lost by a model, AIC deals with the trade-off

between the goodness of fit of the model and the simplicity of the model. In other words, AIC deals with both the risk of overfitting and the risk of underfitting. The formula for AIC is given by:

$AIC = 2k - 2ln(L)$, where L is the maximum value of likelihood function.

4.3.2 Mean Absolute Error (MAE)

The mean absolute error is a measure of difference between two continuous variables. The mae is one of the well-established ways to compare the forecasts with their eventual outcomes. The mean absolute error is given by:

$$MAE = \frac{1}{n}\sum_{i=1}^{n} |y_i - x_i| = \frac{1}{n}\sum_{i=1}^{n} |e_i|$$

4.4 Method 1

This method uses a blend of time series and machine learning algorithms explained in section 3. The strategy used in this method is explained below.

```
Strategy:
Step 1: Compute average on 50,000 acoustic_data value chunks and store in
new dataframe
Step 2: Split the data by 95-5 to keep the 5% data for validation of final
model.
Step 3: Train ARIMA models on 95% data created in above step.
Step 4: Backtest ARIMA model using 5% validation data and calculate the MAE
for model.
Step 5: Repeat steps 3 and 4 to find best ARIMA model for acoustic_data.
Step 6: Use 95% data from step 2 and split further by 90-10 split.
Step 7: Train machine learning algorithms (LR, SVM, RF) on 90% train data
from above step.
Step 8: Test the ml algorithm performance using MAE on 10% test data.
Step 9: Repeat steps 7 and 8 to find best machine learning algorithm for
time_to_failure prediction.
Step 10: Use the ARIMA model in step 5 and ml model in step 9 to predict
time_to_failure.
Step 11: Compute performance of final model using validation data from step
2.
```

4.5 Method 2: Time Series regression

The method uses linear regression algorithm to explore the linear relationship between time_to_failure and seismic data. A linear model is constructed for time_to_failure using the seismic data as independent variable. We then construct a timeseries model using ARIMA on the residuals of the linear model and analyze the model and its residuals for a good fit.

4.6 Method 3: Time Series regression with SVM

The method uses support vector machine algorithm to explore the relationship of time_to_failure with seismic data. We use seismic data as independent variable to create a non-linear model on dependent variable time_to_failure. We then construct a timeseries model using ARIMA on the residuals of the non-linear model and analyze the model and its residuals for good fit.

5. Experimental Procedures

5.1 Exploratory Analysis

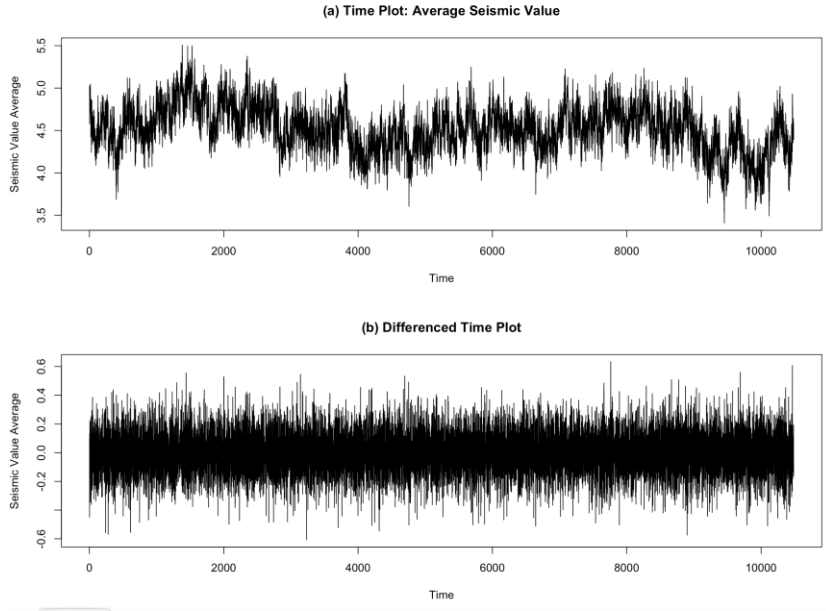


Fig. 5.1: Time Plot of the seismic dataset and it's difference.

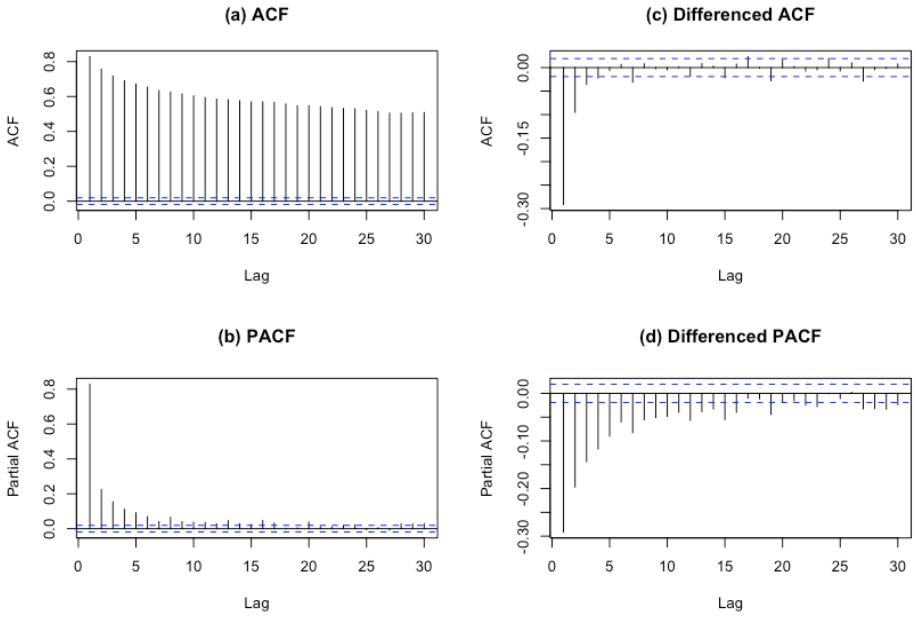


Fig 5.2: ACF/PACF plot for the seismic dataset and it's difference

Initial exploratory analysis focuses on seismic data as it will be used to build a model to predict time_to_failure. After reducing the dataset by averaging large chunks of seismic data we have a time series (fig. 5.1a) that looks volatile, mostly non-stationary, with some hints of seasonality. First, the distribution of the time series is examined by a histogram and Jarque-Bera test, both of which indicate normality. Next, correlation plots of the data indicate the series

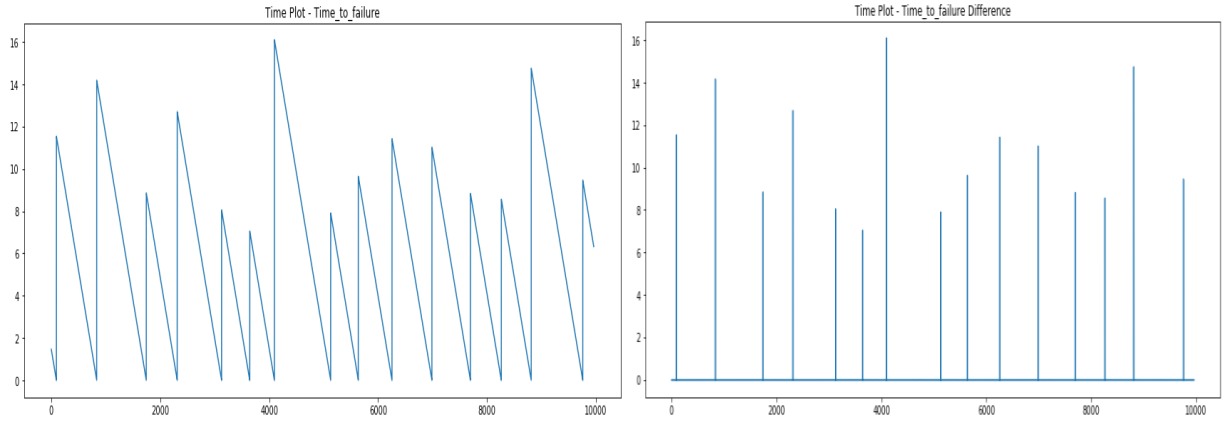


Fig 5.3: Time plot for the time_to_failure and it's difference

is likely non-stationary as the ACF (fig. 5.2a) decays very slowly over many lags. Additionally, the PACF (fig. 5.2b) shows an AR signature. To test serial correlation, an augmented Dickey-Fuller (ADF) test was calculated on the data and did not reject the null hypothesis of stationarity with a non-significant p-value = 0.17.

In order to model non-stationary data it is best practice to examine the series in terms of its first difference or returns (fig. 5.1b). The differenced series looks more stationary with a constant mean equal to zero. ADF test on differenced data has significant p-value = 0.01 meaning returns of the series are stationary. It is important to analyze autocorrelation of the returns to see what else can be gleaned about the overall behavior of the series. ACF/PACF plots of the returns (fig. 5.2c & 5.2d) reveal two important aspects of the data: 1) the exponentially decaying ACF means the series also has an MA signature and 2) there is subtle seasonality every 6 to 8 lags. Finally, an extended autocorrelation (EACF) plot visualizes possible order for an ARIMA model and which lag should be targeted for seasonality.

Secondly, we tried to build the model using time_to_failure as a time series to predict the next time_to_failure. For that we did some exploratory analysis on the dataset. First, we plot the time plot for the time_to_failure data and for the difference of time_to_failure as shown in fig.3 below.

We plotted the ACF & PACF plot for the time_to_failure and it's difference. First two plot of fig. 4 showed that time_to_failure is a random walk series. Now the last two plot of fig. 4 ACF/PACF plot for the time_to_failure difference show that the now series is close to white noise as it shows that there is no seasonality and after 1st lag it is 0.

To test serial correlation, an augmented Dickey-Fuller (ADF) test was calculated on the time_to_failure difference and it rejected the null hypothesis of stationarity with a significant p-value = 0.000. So, the series doesn't have unit root and is stationary.

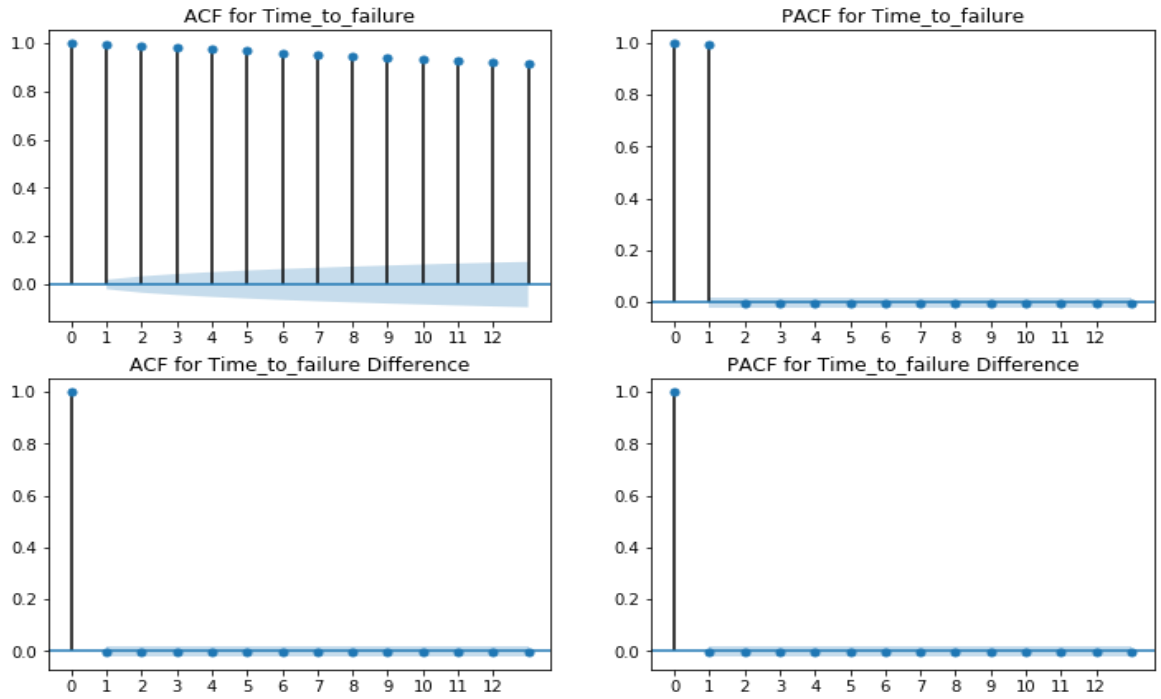


Fig 5.4: ACF/PACF plot for time_to_failure and it's difference

6. Model Evaluation and Residual Analysis

6.1 ARIMA

Model	Backtest MAE	Residual Analysis	
		Box-Ljung Test	JB Normality Test
ARIMA(1,1,1)	0.113	p=0.01	p=0.28
ARIMA(4,1,4)	0.112	p=0.89	p=0.25
ARIMA(2,1,1)	0.113	p=0.72	p=0.27
ARIMA(1,1,2)	0.113	p=0.56	p=0.25
ARIMA(2,1,2)	0.113	p=0.97	p=0.23
SARIMA(1,1,1)x(1,1,1)8	0.114	p=0.01	p=0.27
SARIMA(1,1,1)x(1,1,1)6	0.113	p=.005	p=0.28
ARMA-GARCH(1,1)	0.115	p=0.01	p=0.23

Table 6.1 : Seismic Data Model Building and results

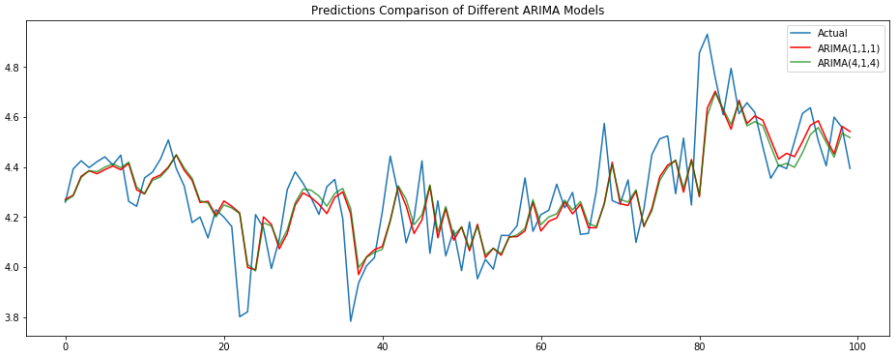


Fig 6.1: Predictions of ARIMA(1,1,1) and (4,1,4) model.

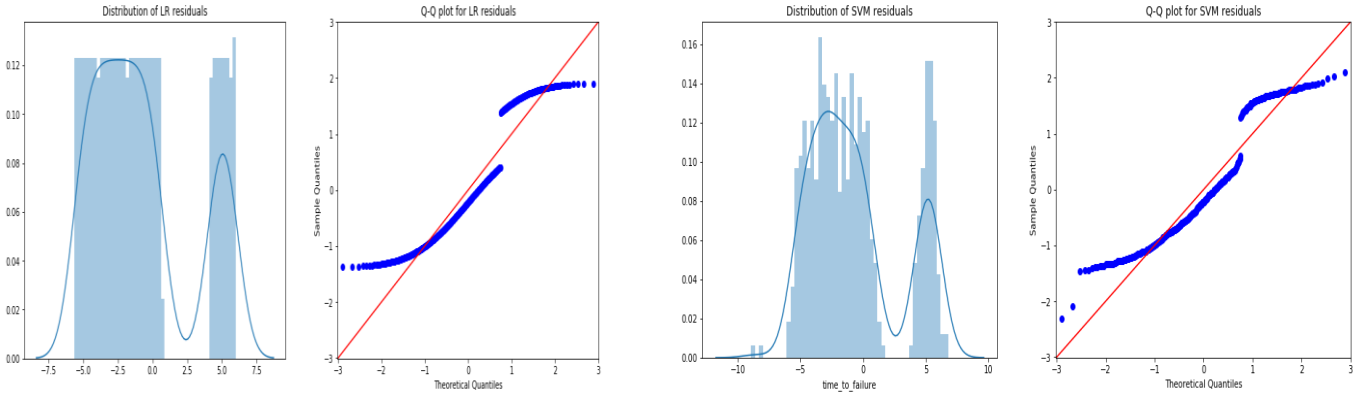
The first model built in this analysis focused only on the time series of seismic waves with the objective to build a model that can accurately predict wave amplitude. In the exploratory analysis, correlation plots were presented to visualize features of the time series. In terms of building an ARIMA model, the correlation plots are instructive of which parameters to use that would best fit the model to the data. The PACF (fig. 5.2b) suggests that AR terms should be added to a model. While, the differenced ACF (fig. 5.2c) shows correlation exponentially decaying to zero with 3-4 significant lags indicating a few MA terms could be useful in a ARIMA model as well. EACF plot (see appendix) suggest 1 AR and 2 MA term would be adequate parameters to start modelling. Additionally, the EACF reports nearly all significant values at column 8, reinforcing the suspicion of seasonality around lag 8.

Given the conclusions of the exploratory analysis an ARIMA(1,1,1) model was fit first followed by more complex models with more terms. There was enough evidence in the ACF/PACF plots of seasonality in the series to justify fitting a few SARIMA models with periodicity ranging from 6 to 8. Finally, an ARMA-GARCH model was fit in attempt to account for volatility in the series.

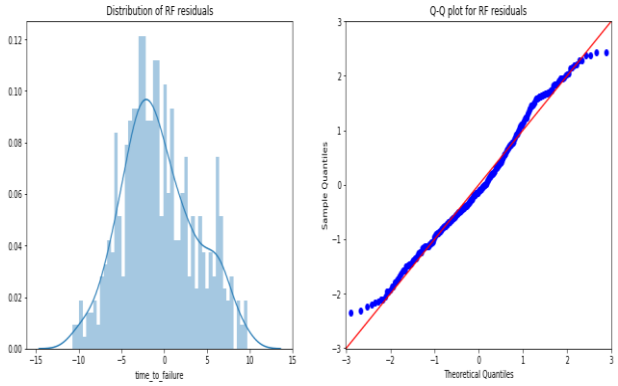
The models listed in table 1 (and time_to_failure model under *additional experiments*) were trained on 95% of the data and backtested on the remaining 5% to produce mean absolute error (MAE) scores. Residual analysis was performed by examining the autocorrelation and distribution of residuals. A model is a good fit if the residuals are serially uncorrelated and normally distributed. A Box-Ljung test determined correlation and Jarque-Bera test of normality indicated distribution of residuals. The results in table 1 were largely the same except for a few models with correlated residuals noted by the red boxes under Box-Ljung. Further evaluation by a coefficient test (see appendix) revealed many models had coefficients that were insignificant or had absolute values greater than 1. An ARMA process with a coefficient greater than 1 will not be stationary, therefore cannot accurately model a time series. Considering the results of residual analysis and a coefficient test there are 2 model parameters appropriate for this data: ARIMA(1,1,1) and ARIMA(2,1,1). Given the new MAE scores from backtesting of both models, it would be prudent to select the more parsimonious model ARIMA(1,1,1) to make predictions of seismic data.

6.2 Linear Regression

A linear regression model was trained on the data as per step 7 of method1 in section 4.6. The model is expected to capture any linear relationships between the averaged seismic



(a) (b)



(c)

Fig 6.2: Histogram and Q-Q plot of (a) LR , (b) SVM and (c) RF

data and time_to_failure variables. The correlation between the two variables is very low i.e. -0.02, which indicates almost negligible correlation between the two variables. The residuals of linear regression does not show a normal distribution as shown in figure 6.2(a).

6.3 Support Vector Machine

After training an SVR model with gaussian kernel and strict constraints for margins and support vector, we get the residuals shown in figure 6.2(b). The residuals do not have a normal distribution in histogram or qqplot.

6.4 Random Forest

The residual plot for Random Forest shows some new possible directions with this dataset. The residuals were close to a normal distribution as can be seen from histogram and qqplots (figure 6.2(c)). Also, the residual plot shows the time series behaviour which should be investigated. We can also see the intervention after 400th observation. It will be interesting to use random-forest with time-series models and see the results we get.

6.5 Timeseries Regression

We used time series regression analysis to find the relation between the averaged seismic data with time_to_failure. The correlation of averaged seismic data and time_to_failure is very low i.e. -0.02. This suggests the independence of both features from each other. Assuming the time_to_failure as a timeseries we want to see how averaged seismic data affects the outcome as an external influencer.

The model that we build was using the time_to_failure as a time series directly to predict time_to_failure. In this we build the model using time series as regression model. From the exploratory analysis ACF/PACF plots and using the auto_arima() function when we trace the model we got the ARIMA(0,1,0) but after trying some other model we got some better AIC value from ARMA(1,0) model so we considered this as our best model. So after checking the model summary we saw that the 'xreg' variable is not significant and the constant and AR(1) variable is significant. After that we performed Box-Ljung test on model residual. The test result shows that there is no correlation between the residuals.

6.6 Timeseries Regression with SVM

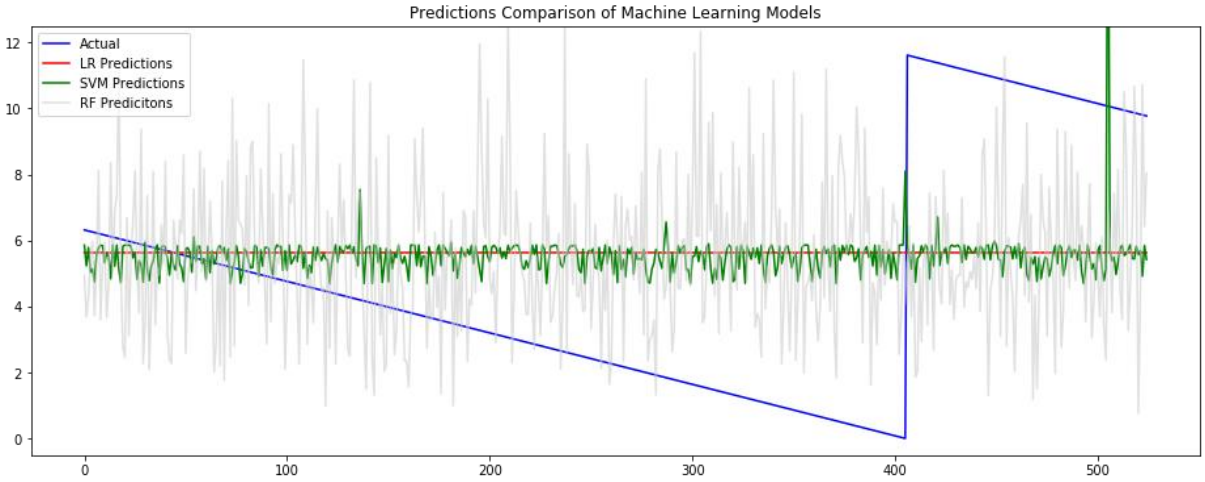
The support vector machine is a popular algorithm known to work with the non-linear data. In this attempt, we try to find the relation between the averaged seismic data with time_to_failure. We use same assumptions as in above method that time_to_failure is a timeseries and use averaged seismic data as an external influencer. The SVM residuals fails to meet the normality and autocorrelation assumptions of jarque-bera and Ljungbox tests respectively. An ARMA(0,0) model was fitted to further explore the GARCH effects in residuals, however there were no volatility issues found.

7. Conclusion

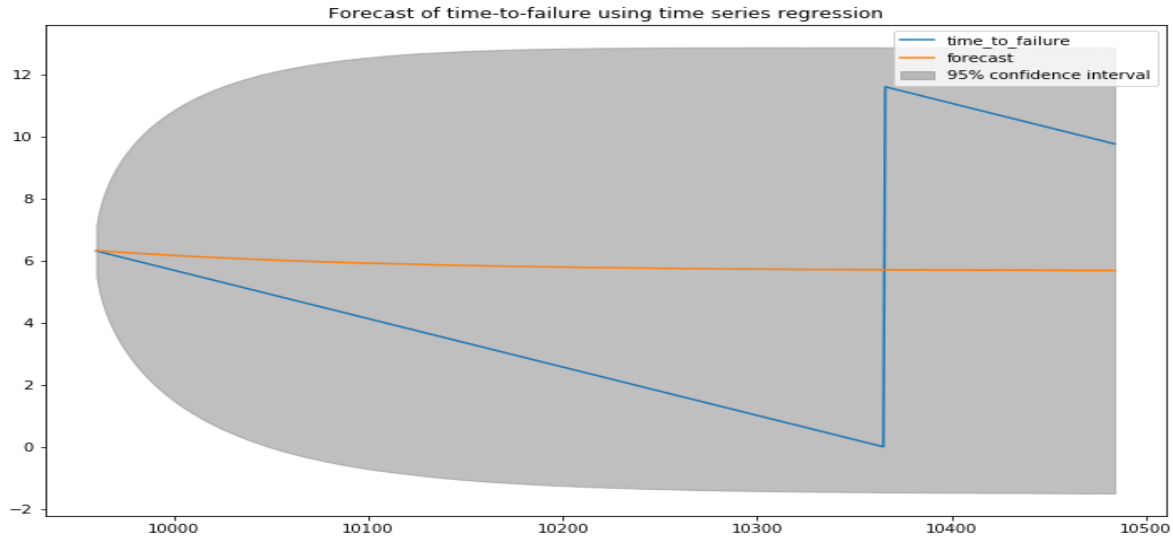
Model	MAE	Box-Ljung at Lag-1	JB Normality test
Timeseries Regression ARMA(1,0)	3.44	p=0.84	p=0.00
Linear Regression Model	3.11	p<<0	p<<0
Support Vector Regressor	3.08	p<<0	p<<0
Random Forest Regressor	3.56	p<<0	p<0

Table 7.1: MAE value result for predicted Time_to_failure

A number of different methods were tried on this dataset in order to predict the time for next earthquake using the seismic data. We reduced the dataset by averaging the chunks of records and tried to predict the time_to_failure using this reduced data. A combination of both timeseries algorithms and machine learning algorithms were applied in order to get some effective predictions out of our models. However, none of the methods were close enough of producing acceptable results. The models were unable to predict the time to next earthquake and resulted in relatively high mean absolute error in forecasts and residuals of the models were not normal and in some cases the autocorrelation was still present in residuals at lag-1. The table 7.1 shows the MAE of predicted time_to_failure on validation set along with the residual test statistics.



(a)



(b)

Fig 7.1 : Result for (a) SVM, LR and RF forecasted values (b) Time Series regression forecasted values.

The figures 7(a) and 7(b) show the comparison of forecasted time_to_failure vs actual time_to_failure. We can see clearly that none of the models were close enough in predicting the actual value satisfactorily enough. We conclude that it might not be possible to find a good model to predict time_to_failure using current data but it might be useful to explore this objective further using some other influencers or with more feature engineering.

8. Additional Experiments

Model	MAE	Residual Analysis	
		Box-Ljung Test	JB Normality Test (p-value)
ARIMA(0,1,0)	3.44	0.887	0.00

Table 8.1 : Time_to_failure Data Model Building and results

Another approach we tried was building the model directly using the time_to_failure as a time series to explore this time series further. From the exploratory ACF/PACF plots (show in Fig 5.4) and using the auto_arima() function when we trace the model we got the ARIMA(0,1,0) as our best model from the AIC criteria using maximum-likelihood principle. Now to check the model residual we performed Box-Ljung test. After doing the test we saw that there is no autocorrelation between the residuals. The result that we got from this test did improve the previous result that we got.

References

[1] Yongyang Zheng, A Novel Seismic Wave Analysis and Prediction Model using SVM Kernel and Time Series Prediction, ISSN: 1473-804x.

[2] Rouet-Leduc, B., Hulbert, C., Lubbers, N., Barros, K., Humphreys, C. J., & Johnson, P. A. (2017). Machine learning predicts laboratory earthquakes. Geophysical Research Letters, 44. https://doi.org/10.1002/2017GL074677.

[3] Hilary L. Seal (1967). "The historical development of the Gauss linear model". Biometrika. 54 (1/2): 1–24. doi:10.1093/biomet/54.1-2.1. JSTOR 2333849.

[4] Drucker, Harris; Burges, Christopher J. C.; Kaufman, Linda; Smola, Alexander J.; and Vapnik, Vladimir N. (1997); "Support Vector Regression Machines", in Advances in Neural Information Processing Systems 9, NIPS 1996, 155–161, MIT Press.

[5] Ho, Tin Kam (1995). Random Decision Forests (PDF). Proceedings of the 3rd International Conference on Document Analysis and Recognition, Montreal, QC, 14–16 August 1995. pp. 278–282.

[6] G. Peter Zhang, Time series forecasting using a hybrid ARIMA and neural network model, Neurocomputing 50 (2003) 159 – 175.

[7] Alexandridis, A., Chondrodima, E., Efthimiou, E., Papadakis, G., Vallianatos, F., & Triantis, D. (2014). Large earthquake occurrence estimation based on radial basis function neural networks. IEEE Transactions on Geoscience and Remote Sensing, 52(9), 5443–5453.

Appendices

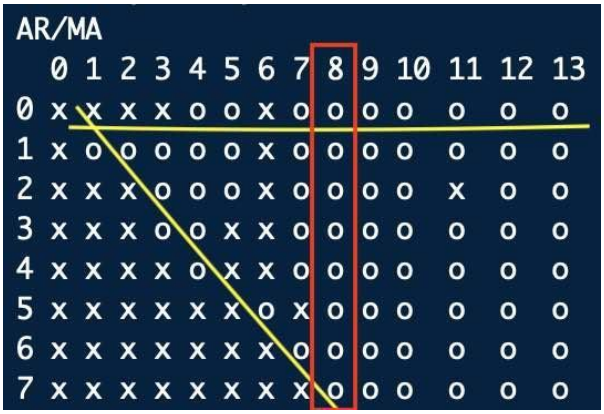


Fig: EACF of returns:

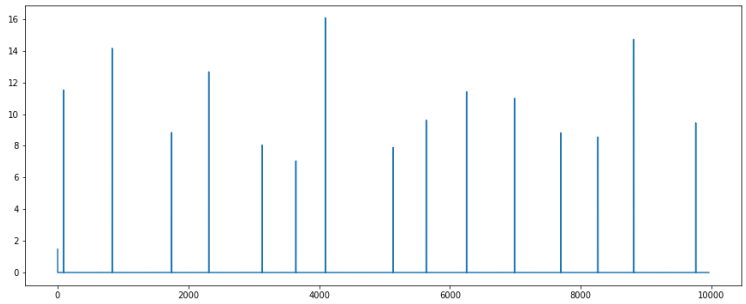


Fig: Residual Plot for ARIMA(0,1,0) on time_to_failure

	Coefficient Test									
	ar1	**ar2**	**ar3**	**ar4**	**ma1**	**ma2**	**ma3**	**ma4**	**sar1**	**sma1**
ARIMA(1,1,1)	0.47				-0.89					
ARIMA(4,1,4)	0.25	0.16	0.72	-0.29	-0.69	-0.2	-0.68	0.59		
ARIMA(2,1,1)	0.49	0.08			-0.93					
ARIMA(1,1,2)	0.69				-1.13	0.19				
ARIMA(2,1,2)	1.19	-0.28			-1.64	0.65				
SARIMA(1,1,1)x(1,1,1) [8]	0.48				-0.88				0	-0.99
SARIMA(1,1,1)x(1,1,1) [6]	0.48				-0.89				0.02	-1

Table: ARIMA Coefficient Analysis

Individual Reports

Max Kenworthy:

Overall, our team worked cohesively and equally contributed to the analysis in the final project. My teammates, Anuj and Nisarg, orchestrated the strategy we would implement to build a model that would analyze seismic wave data and then predict the timing of an earthquake. Specifically, I focused on the analysis of the seismic wave series in our data set. I began with exploratory analysis including correlation and time plots to examine any trends and overall behavior of the data. ACF and PACF plots were analyzed for seasonality and AR / MA signatures that would be helpful in model building. I utilized statistical tests Augmented Dickey-Fuller and the Jarque-Bera to determine normality and stationarity of the series. After testing stationarity, I determined differencing would be necessary to build a model. I built and tested many models (5 ARIMA, 2 SARIMA, 1 GARCH) to find the best fit for our data. I evaluated models with backtesting and residual analysis. MAE was used for backtesting scores and residuals were statistically tested for normality (JB) and correlation (Box-Ljung). Additionally, I compared models with a coefficient analysis to see if models had insignificant terms or indicated a non-stationary process (terms >= 1). Once I decided on the best fitting model, I passed on the parameters to my teammates to start forecasting. For clarification, I contributed all the content related to this in the final project including introduction/overview, exploratory analysis, model fitting, and model evaluation/residual analysis.

Learnings: I learned a lot from this analysis, specifically, the methodology of model fitting and evaluation. Intelligently choosing models based on ACF/PACF reinforced the concepts of autocorrelation in a time series. Evaluating correlation plots for AR and MA signatures, as well as seasonality, was something I wasn't really comfortable with at the beginning of this course but the final project helped solidified it. An unexpected takeaway was a better understanding of regression as a whole, not only how it applies to time series analysis. The application of linear regression in a time series made me think about how a model is really fit to data and the influence of residuals.

Nisarg Patel:

Firstly we all did the exploratory analysis together to of the original dataset from the kaggle competition. After doing the exploratory analysis me and anuj build the strategy to implement our new objective which was to predict the seismic wave and then predicting the time to failure from that output.

Then I tried the ARIMA(1,1,1) model with Anuj and Max, then Anuj and Max tried other different model and their residual GARCH effects. In that timespan I tried the time series regression approach in which the target variable was time_to_failure and 'xreg' was seismic data. I did the model evaluation using the MAE for that model and test the residuals for normality (JB) and correlation (Box-Ljung).

Then I tried that same approach on the kaggle test data which had window of 150000 in forecast. I model ran successfully on the original kaggle test dataset but the result were not good as window was too large. At last for additional experiment I tried the ARIMA model by just

considering time_to_failure as time series and then predicting the time_to_failure but it gave same result as time time series regression.

Learnings: I learned a lot from this course and understand the concepts of different time series approaches and what it can be used. I learned how to check and see seasonality in the series what are the effect of seasonality on the model. I also learned the effect of residuals on the model and how to understand it using Garch effects. I also learned how time series can be combined with other machine learning models.

Anuj Verma

I contributed in initial exploratory analysis of original dataset of earthquake prediction Kaggle competition. I researched about the domain, data and the problem statement to understand the issue clearly and explained this to teammates to enhance their understanding of the problem. The Nisarg and I brainstormed to find the approach to reach our objective with a time-series related model. I created an overall strategy to approach the problem with slightly modified objective. I contributed in building an ARIMA model to predict the seismic data timeseries. I am responsible for putting efforts toward the extra-credit work to build the machine learning models Linear Regression, SVM and Random Forest to predict the time_to_failure using the seismic data predictions generated from final ARIMA model.
I took the initiative to lead the team towards our final objective. This includes but not limited to, set up meetings and be a moderator during the meetings to stay on the topic and get best out of our time and efforts. I acted as the group liaison to contact professor via emails or in-class regarding project related work/issues/advice. I worked on project presentation and presented my part with group. I also provided the structure to write final project report in a clear and understandable format and worked on report along with the group. During the project progress, I supported for any python related issues during the project work.

Learnings: This project has been extremely challenging right from getting to know the dataset, to coming up with a strategic approach, till the very end of submitting the report of project. We all knew the challenges of working with this dataset and we accepted the challenge hoping to learn much more by our efforts. I feel very happy to share that we learned a lot more than what we expected. Geophysics was a new domain to all of us and getting to know the terminologies used, reading the research papers in this field to get an idea of what is happening in this field currently was challenging and a learning experience as well. Another challenge was working with the data itself since it contains more than 600 million records. To read and come up with strategy of averaging the chunks of data took some time. Then the next challenge was to come up with a model that can predict the time_to_failure using just seismic data. This required a lot of brainstorming and reasoning power to come-up with an effective strategy. Then I also got to apply advanced ml algorithms along with timeseries to see if anything works good for us. In the end, although we were not able to produce any predictive power with our models, we learned a lot.