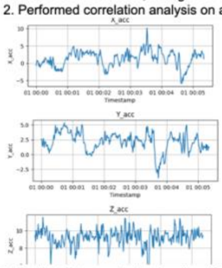
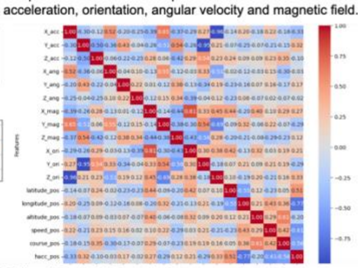
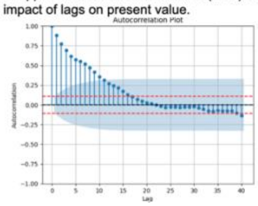
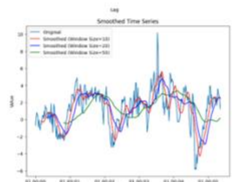
1. Tried a 3D simulation, histograms and plots of time series to explore data
2. Performed correlation analysis on acceleration, orientation, angular velocity and magnetic field. Plotted and analyzed the correlation matrix to figure out essential relations such as between Y orientation and Y acceleration and similar variables



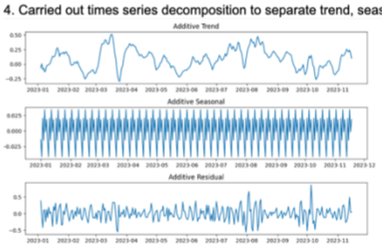




3. Applied Autocorrelation function (ACF) and Partial ACF on the features and tried Moving averages method to gain understanding of trends and relation with time, noticed the importance of moving averages to reconstruct the time series and impact of lags on present value.

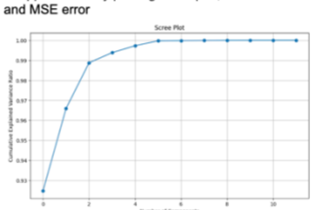



4. Carried out times series decomposition to separate trend, seasonality and residual, to further deep dive into residuals, observed that the seasonality in most cases is of small magnitude whereas residuals can be as high as the trend magnitude.




5. Applied PCA by plotting scree plot, determined variance ratio by number of components, studied reconstruction error across num_components to compress data. Observed the best number of components to have is 5 or 6 as per variance ratio and MSE error




```
Component of PCA :  4
Shape of transformed data :  (322, 4)
Variance Ratio :  [0.92488301 0.04107403 0.02280375 0.00505894]
Length of Inversed transformed data :  322
Mean Squared Error (MSE) :  4.687291194344113
Root Mean Squared Error (RMSE) :  2.1650152873234205

 ------------------------------------------

Component of PCA :  5
Shape of transformed data :  (322, 5)
Variance Ratio :  [0.92488301 0.04107403 0.02280375 0.00505894 0.00344936]
Length of Inversed transformed data :  322
Mean Squared Error (MSE) :  2.071202426204015
Root Mean Squared Error (RMSE) :  1.4391672683201266

 ------------------------------------------

Component of PCA :  6
Shape of transformed data :  (322, 6)
Variance Ratio :  [0.92488301 0.04107403 0.02280375 0.00505894 0.00344936 0.00244955]
Length of Inversed transformed data :  322
Mean Squared Error (MSE) :  0.21339756773252216
Root Mean Squared Error (RMSE) :  0.4619497458950726
```

6. Attempted auto-encoders to take another step towards compression and also, implement findings from PCA such as limiting the number of layers and layers dimensions

Next Steps:
1. Perform Residual analysis to extract information from noise and determine its nature - kurtosis, histogram, ACF , PACF
2. Try fitting AR(1) and AR(2) models and see if "eps" are independent
3. Try out singular spectrum analysis, SVD, FFT to extract trends (may be sinusoidal)
4. Conduct residual analysis on PCA to see if synthetic data creating is feasible

1. Performed Vector Autoregression (VAR) to find correlation among variables in time, each variable in the system is regressed on its own lagged values as well as the lagged values of all other variables in the system. The results were synchronous with correlation matrix and the lagged values gave insights into timewise contributions of variables.

```
Correlation matrix of residuals
          X_acc     X_ang     X_mag     X_ori
X_acc  1.000000 -0.089139 -0.014436 -0.128012
X_ang -0.089139  1.000000  0.007767  0.000139
X_mag -0.014436  0.007767  1.000000 -0.097609
X_ori -0.128012  0.000139 -0.097609  1.000000


Mean Squared Error (MSE): 0.011634351824959418
Root Mean Squared Error (RMSE): 0.10786265259560149
AIC: -23.212397386092743
Eigenvalues: [115.74693534  64.18124466  27.21897732   8.43054386]
Critical Values (90%): [[44.4929 47.8545 54.6815]
 [27.0669 29.7961 35.4628]
 [13.4294 15.4943 19.9349]
 [ 2.7055  3.8415  6.6349]]
```

```
Results for equation X_acc
==========================================================================
              coefficient       std. error           t-stat            prob
--------------------------------------------------------------------------
const            0.014273         0.030231            0.472           0.637
L1.X_acc         0.805548         0.061501           13.098           0.000
L1.X_ang        -0.048390         0.032410           -1.493           0.135
L1.X_mag        -1.237286         0.307351           -4.026           0.000
L1.X_ori        -0.006966         0.053983           -0.129           0.897
L2.X_acc        -0.056627         0.079735           -0.710           0.478
L2.X_ang         0.008194         0.037323            0.220           0.826
L2.X_mag         1.726427         0.666928            2.589           0.010
L2.X_ori        -0.086498         0.068220           -1.268           0.205
L3.X_acc         0.130728         0.079914            1.636           0.102
L3.X_ang         0.013685         0.037785            0.362           0.717
L3.X_mag        -0.786100         0.718663           -1.094           0.274
L3.X_ori         0.048527         0.067661            0.717           0.473
```

2. Tried out Canonical Correlation Analysis (CCA) to double check results of correlation matrix and VAR. Highly negative values suggest strong negative correlation and highly positive suggest positive strong correlation, e.g. X_ori and X_mag have strong positive correlation.

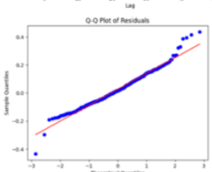
```
X=df_combined[['X_ori']]
Y=df_combined[['X_mag']]
click to expand
# Fit CCA model
cca.fit(X, Y)

# Transform data to canonical space
X_c, Y_c = cca.transform(X, Y)

# Canonical correlation coefficients
canonical_corr = cca.coef_
print("Canonical correlation coefficients:", canonical_corr)

Canonical correlation coefficients: [[23.57597103]]
```

3. After implementation of PCA and determining ideal components as 6, I implemented auto-encoders for compression by having 1 hidden layer of dimension = 6, obtaining a low reconstruction error of approx 0.25. I reconstructed the time series data, subtracted from original data to find residuals and analyzed them.

```
Epoch 50/50
13/13 [==============================] - 0s 5ms/step - loss: 0.0244 - val_loss: 0.0214
3/3 [==============================] - 0s 4ms/step
3/3 [==============================] - 0s 3ms/step
Reconstruction error: X_acc    0.013924
Y_acc    0.013466
Z_acc    0.023136
X_ang    0.022585
Y_ang    0.013126
Z_ang    0.011021
X_mag    0.076161
Y_mag    0.011542
Z_mag    0.022590
X_ori    0.035614
Y_ori    0.011969
Z_ori    0.018126
```

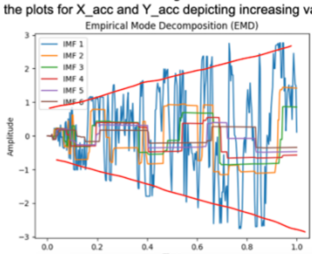
4. In process of conducting residual analysis using ACF, PACF (observe spikes beyond confidence level), kurtosis of histogram (expecting 3), Q-Q plot (check deviation from diagonal), Kolmogorov-Smirnov Test, Shapiro-Wilk Test, and Anderson-Darling Test (Hypothesis testing). These tests will help us understand if the residuals are normally distributed or are just noise/fluctuations, e.g. residuals of X_acc are not normally distributed by majority vote.

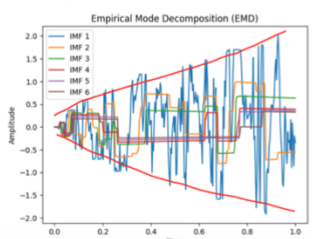

Screenshot 2024-02-24 at 2.12.37 AM.png

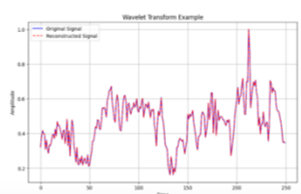



```
Shapiro-Wilk Test:
Statistic: 0.9799064993858337
p-value: 0.00017648971697781235
Residuals are not normally distributed (reject null hypothesis)
--------------------------------------------------------------------
Anderson-Darling Test Statistic: 1.2149597622179158
Critical Values: [0.569 0.648 0.778 0.907 1.079]
Significance Levels: [15.  10.   5.   2.5  1. ]
Significance Levels: 5.0 Critical Values: 0.778
Residuals are not normally distributed (reject null hypothesis)
--------------------------------------------------------------------

Jarque-Bera Test Statistic: 72.61766839242163
P-Value: 1.7032314364173505e-16
Residuals are not normally distributed (reject null hypothesis)
--------------------------------------------------------------------

Kolmogorov-Smirnov Test Statistic: 0.052134604643712806
P-Value: 0.3340181564456235
Residuals are not normally distributed (reject null hypothesis)
```

5. Performed Empirical Mode Decomposition (EMD) which is used for decomposing a time series signal into a finite number of oscillatory components known as Intrinsic Mode Functions (IMFs) and a residual. At each iteration, the next IMF is extracted from the residual signal. I observed that IMF has heteroscedasticity suggesting non-normality in residuals. This led me to interesting techniques like Non-Negative Matrix Factorization (NMF), Wavelet Transformation etc., e.g. below are the plots for X_acc and Y_acc depicting increasing variance in magnitude of residuals with time.

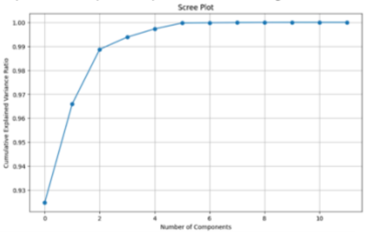



6. I implemented FFT and in the process of implementing Singular Spectrum Analysis (SSA) to extract high frequency components.

1. As per the scree plot, components from the range of 4-8 can be considered for latent space dimensions of VAE, which means the data can be shrunk to 4-8 dimensions

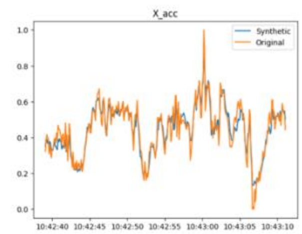


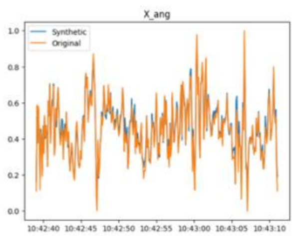

2. Plotted the synthetic (reconstructed) data from the VAE, along with the original data to visually check the variance and accuracy.

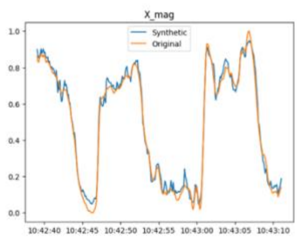
3. Tweaked the principal components in the PCA technique to see if there was any improvement in step 1. However, experimentally, neither of the PCA component values i.e. 4-8 gave overlapping/accurate reconstructed time series compared to the original.

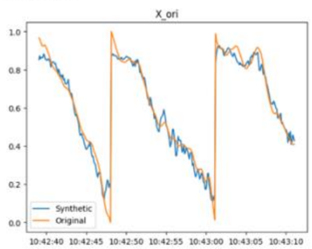
4. Hence, upsampling the data into higher dimension latent space seemed like a solution, after trying out 8/12/16/24/32/64 dimensions in 1 layered VAE, the synthetic data only improved (and reconstruction error dropped)

```
Epoch 49/50
13/13 [==============================] - 0s 7ms/step - loss: 0.0018 - val_loss: 0.0019
Epoch 50/50
13/13 [==============================] - 0s 7ms/step - loss: 0.0017 - val_loss: 0.0019
3/3 [==============================] - 0s 5ms/step
3/3 [==============================] - 0s 5ms/step
Reconstruction error: X_acc    0.001809
Y_acc    0.001460
Z_acc    0.002138
X_ang    0.001371
Y_ang    0.001247
Z_ang    0.001219
X_mag    0.001062
Y_mag    0.002169
Z_mag    0.001759
X_ori    0.002047
Y_ori    0.004106
Z_ori    0.002244
```



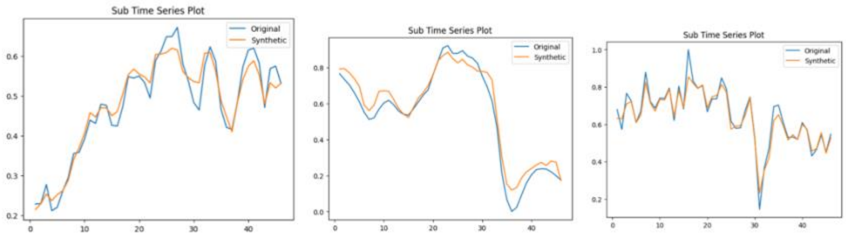

5. Extracted the synthetic data and original data into 2 separate dataframe with the same index, 322 rows (to represent 322 timestamps) and shape 322x12 . After a transpose, add a new column as 'target' to both the dataframes with value = 0 for synthetic data and 1 for original data. Now, both data frames were concatenated, shape = 24x323

6. Attempted to train and run logistic regression (In the context of neural networks, a logistic classifier may refer to a single-layer neural network with a logistic (sigmoid) activation function in the output layer, used for binary classification tasks), due to less sample data points the model did not perform well.

7. Planned to divide the time series dataframe, from shape 12x322 to 84x46, essentially, slicing each time series into 7 sub time series to increase the data, assuming that the characteristics of each time series is restored. This way the synthetic and original dataframe now became 84x46 each, after concatenating, became 168x47 (1 column for 'target')

| | | | |
|---|---|---|---|
| **167** | 0.807139 | 0.867438 | 0.898225 |

168 rows × 47 columns




8. Now, the logistic regression on this dataset after performing a 80/20 split with equal distribution of classes in train/test data, the model is giving close to 45% accuracy (closer to random guess), hence, deriving a conclusion that the synthetic data and original data do not have inherent pattern/sequence/characteristics to be picked by the model in order to classify them as different. This is our goal with the synthetic data as it should be so close to the original that the ML model fails to differentiate between the two.

```
X_train shape:  (134, 46)
X_test shape:  (34, 46)
y_train shape:  (134,)
y_test shape:  (34,)
Item 0: 67 occurrences
Item 1: 67 occurrences
Item 0: 17 occurrences
Item 1: 17 occurrences
Accuracy: 0.4411764705882353
```

9. Took the experiment to the next level, implemented and trained a simple neural network with 4 dense layers (32,64,32,8). Trained the model over 100 epochs with learning rate decay to figure out whether MLP learns any hidden patterns, clues, statistics, characteristic, features or signals from the synthetic and original time series to classify them, however, with larger training data the MLP could not perform well, even a shallower MLP could not perform. This certifies our objective of reconstructing the original time series in such a way that all the underlying patterns are reconstructed in the time series.

Graphs for loss, MAE, accuracy for training and validation both are fluctuating and not converging which means the model is unstable because of the data.

```
- loss: 0.2390 - mae: 0.4793 - accuracy: 0.5597 - val_loss: 0.3143 - val_mae: 0.5523 - val_accuracy: 0.2647
- loss: 0.2462 - mae: 0.4763 - accuracy: 0.5448 - val_loss: 0.3434 - val_mae: 0.5643 - val_accuracy: 0.3529
- loss: 0.2450 - mae: 0.4824 - accuracy: 0.5448 - val_loss: 0.2710 - val_mae: 0.5180 - val_accuracy: 0.3824
- loss: 0.2426 - mae: 0.4873 - accuracy: 0.5746 - val_loss: 0.2770 - val_mae: 0.5245 - val_accuracy: 0.3235
- loss: 0.2392 - mae: 0.4850 - accuracy: 0.5821 - val_loss: 0.2957 - val_mae: 0.5388 - val_accuracy: 0.2647
- loss: 0.2373 - mae: 0.4802 - accuracy: 0.5597 - val_loss: 0.2902 - val_mae: 0.5356 - val_accuracy: 0.2941
- loss: 0.2380 - mae: 0.4829 - accuracy: 0.5746 - val_loss: 0.2832 - val_mae: 0.5298 - val_accuracy: 0.3235
- loss: 0.2377 - mae: 0.4830 - accuracy: 0.5821 - val_loss: 0.2867 - val_mae: 0.5330 - val_accuracy: 0.3235
- loss: 0.2354 - mae: 0.4806 - accuracy: 0.6343 - val_loss: 0.2977 - val_mae: 0.5416 - val_accuracy: 0.3235
- loss: 0.2351 - mae: 0.4775 - accuracy: 0.5970 - val_loss: 0.3178 - val_mae: 0.5548 - val_accuracy: 0.2647
- loss: 0.2359 - mae: 0.4746 - accuracy: 0.5746 - val_loss: 0.3227 - val_mae: 0.5583 - val_accuracy: 0.2647
- loss: 0.2338 - mae: 0.4721 - accuracy: 0.5746 - val_loss: 0.3185 - val_mae: 0.5557 - val_accuracy: 0.3235
- loss: 0.2333 - mae: 0.4734 - accuracy: 0.6045 - val_loss: 0.3114 - val_mae: 0.5473 - val_accuracy: 0.3529
- loss: 0.2378 - mae: 0.4759 - accuracy: 0.5672 - val_loss: 0.3270 - val_mae: 0.5563 - val_accuracy: 0.3824
```

Last 14 epochs from training