Identifying key signals from data using Anomaly Detection

Archish Ramesh Babu(ARAMESHB)
Gopal Seshadri(GSESHAD)
Agastya Teja Anumanchi(AANUMANC)

Content

- Why are doing this?
- A look at the data
- Evaluation metric
- Algorithms
- Results
- Real world application
- Expansion to Multidimensional data
- Next steps
- Questions

Anomaly Detection using unsupervised methods saves a lot of time and cost

Why are we doing this?

- Anomaly Detection is a popular machine learning concept that helps identify data points that don't follow conventional behavior
- This concept has been widely used in fields like Fraud Detection, Medical Diagnosis to name a few
- Due to the disruption caused by IoT, Social media new time series data(eg : sensor data) is bring generated at a rapid pace
- Companies need to quickly analyze new data to stay competitive and implementing traditional supervised Anomaly Detection methods will take a lot of time and effort to set up

What are we doing?

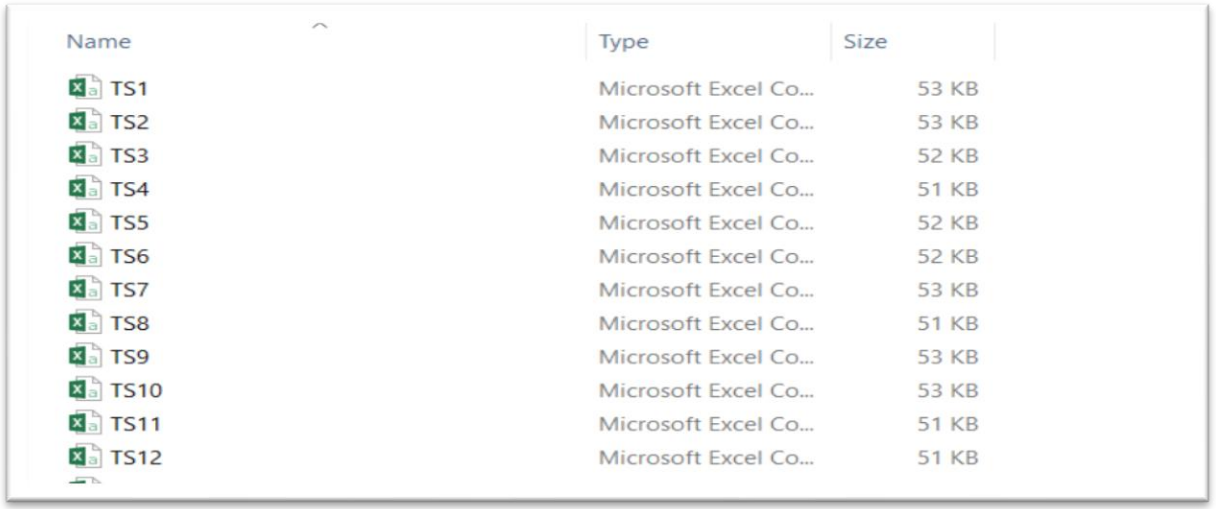
- We are building an unsupervised learning algorithms framework to help identify the anomalous points without any labelled data on time series data
- This will reduce the reaction time to anomalies
- The anomalies identified using unsupervised Machine Learning could later be used to built supervised models

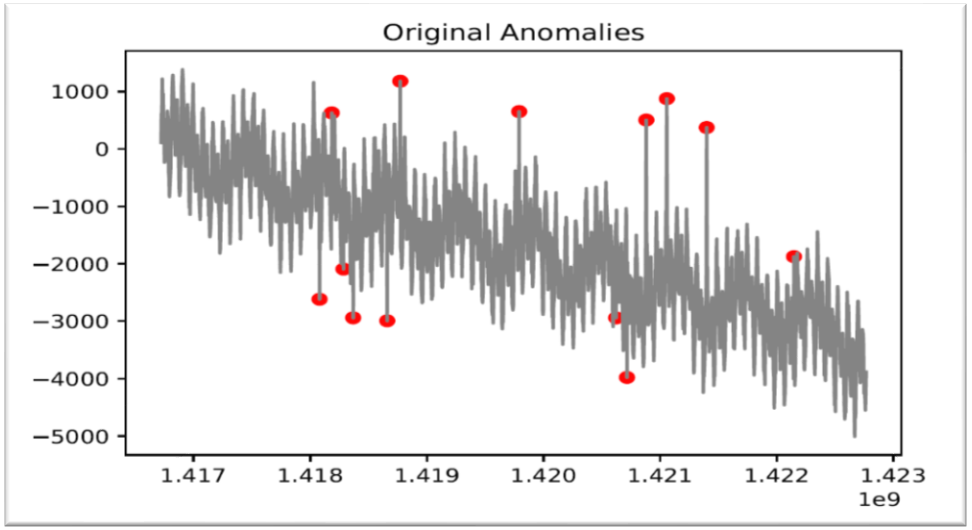
How are we doing it?

- We are using popular unsupervised algorithms like Isolation Forest, One-class SVM, K-means to identify anomalous points
- The algorithms were chosen so that they could be scaled to multidimensional data as well
- We are also validating these algorithms by testing their performance against 67 different time series data provided by Yahoo

Yahoo has provided multiple time series datasets that could be used to benchmark Anomaly Detection algorithms

Time stamp	Value	Anomaly
1416722400	-46.3943564	0
1416726000	311.3462336	0
1416729600	543.279051	0
1416733200	603.4419825	0
1416736800	652.8072434	0
1416740400	429.8209026	0

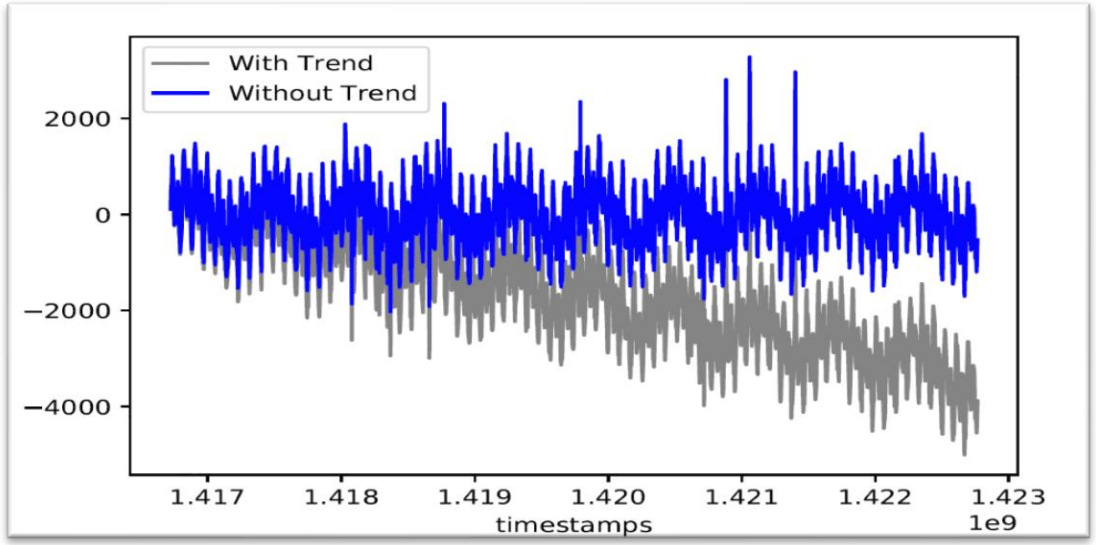


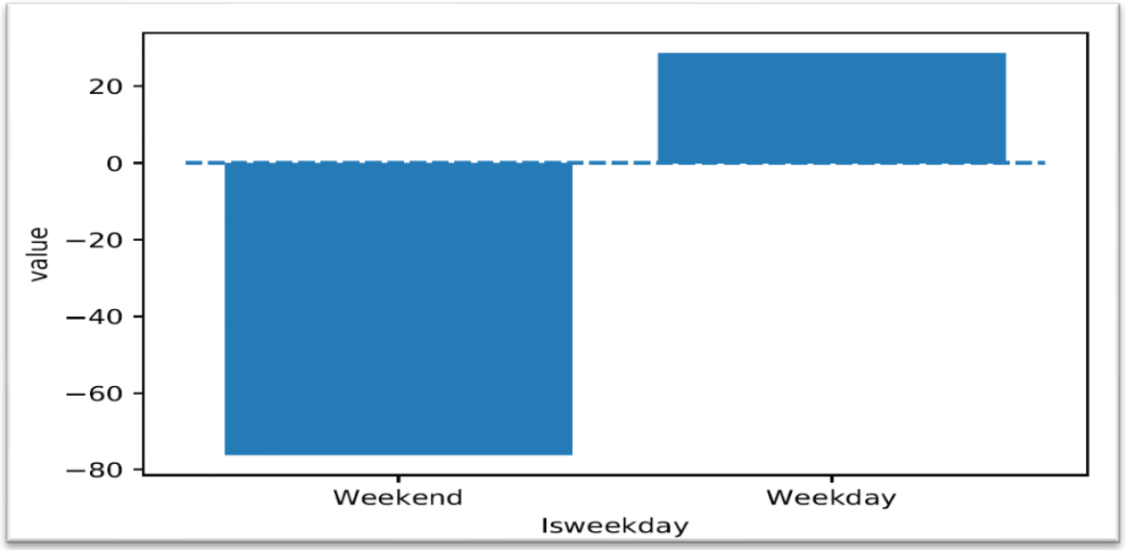


Data

- We had written a request to Yahoo and obtained 67 times series anomaly detection benchmarking files
- The data is partly from yahoos real data along with some syntactic data and the anomalies have been tagged manually by yahoo engineers
- We are using these files to benchmark the performance of our algorithms

Data has both trend and seasonality which needs to be accounted before tagging anomalies





Trend

- Time series data has a lot of trend, the sales of a new product is expected to increase over time
- Now the increase should not be tagged as an anomaly but the sudden spike or dip in the sales should be correctly identified as soon as possible for the business teams to take necessary action
- The trend was removed using the Beta coefficient simple linear Regression model, between the position of the value and the value

Seasonality

- In addition to trend the time series data also has a lot of seasonality, the traffic during the weekends is generally higher on websites due to the additional time or it could be vice verse in other cases
- Above is an image of how the time series varies by the day of a week after removing trend for a particular file

Recall and F1 score important metrics to be considered while evaluating Anomaly Detection Algorithms

Recall

- Recall metric measures the number of Anomalous points that the model is able to correctly tag to the total number of anomalous points in the data

$$Recall = \frac{True\ Positive}{True\ Positive + False\ Negative}$$

Precision

- The precision is the ratio of the number of correctly tagged anomalous points to the total number of points tagged as anomaly

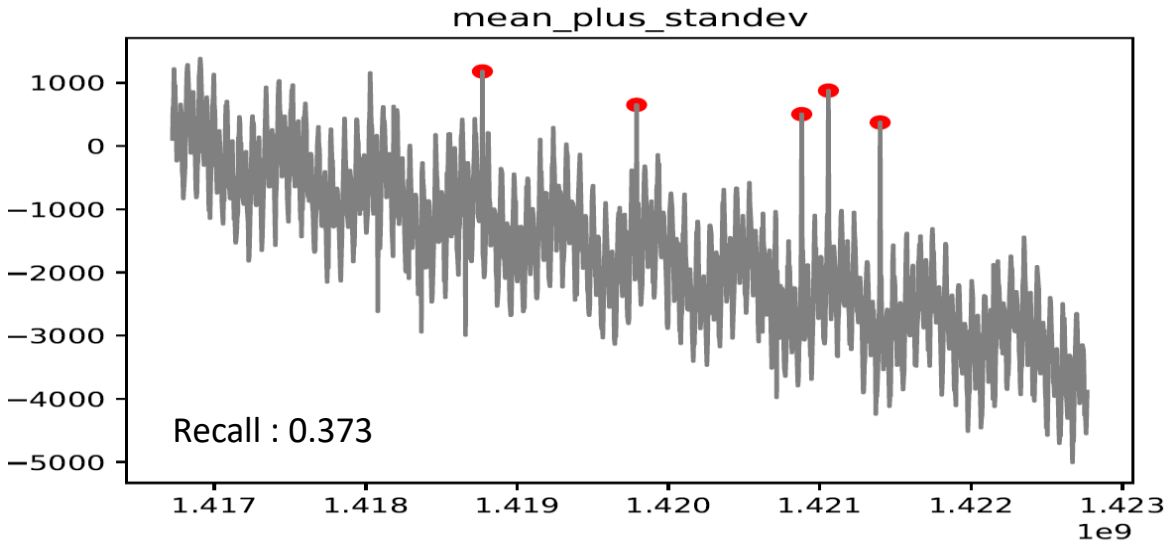
$$Precision = \frac{True\ Positive}{True\ Positive + False\ Positive}$$

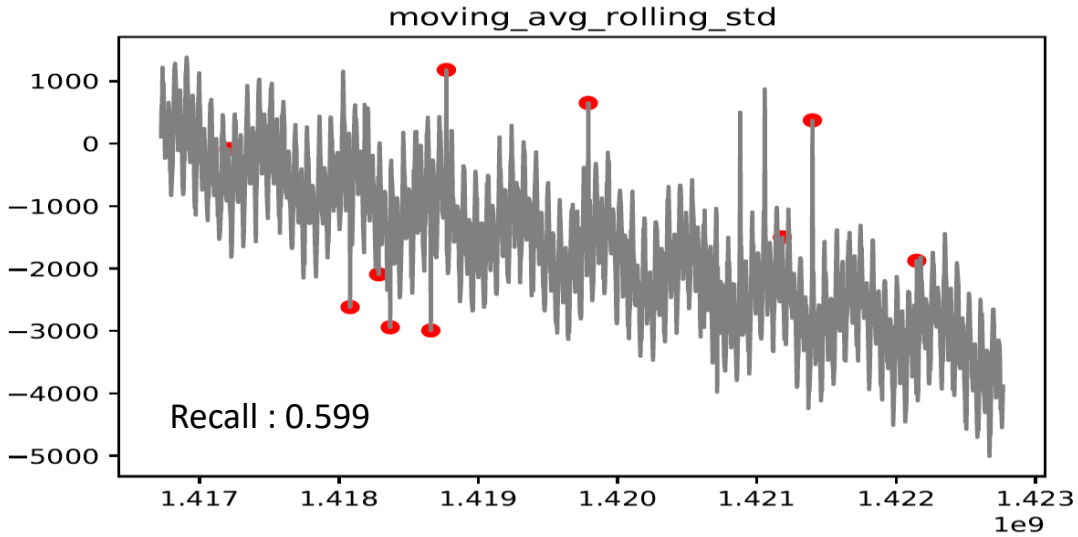
F1 Score

- F1 Score is the harmonic mean of recall and precision.
- This metric shows the balance between Recall and Precision, the ideal value for F1 score is 1

$$F1\ Score = \frac{2\ * Recall\ \ * Precision}{Recall + Precision}$$

After treating for trend, the models using mean plus standard deviation are able to tag some anomalies

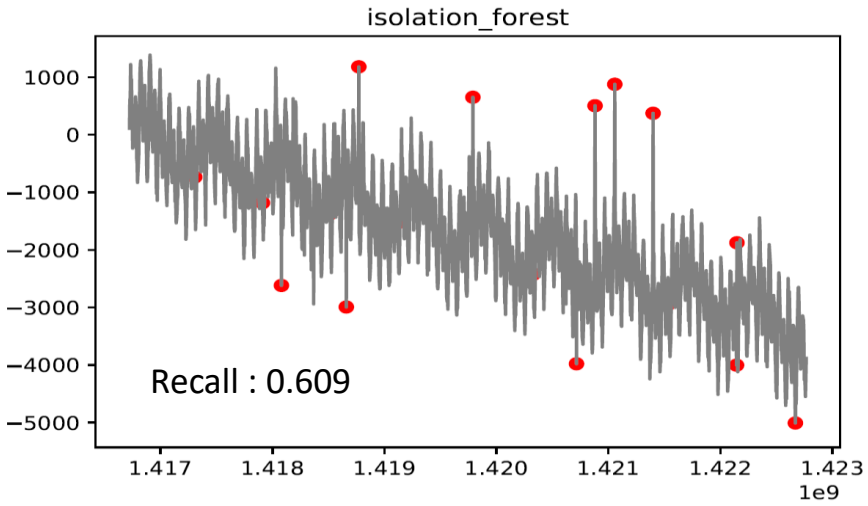


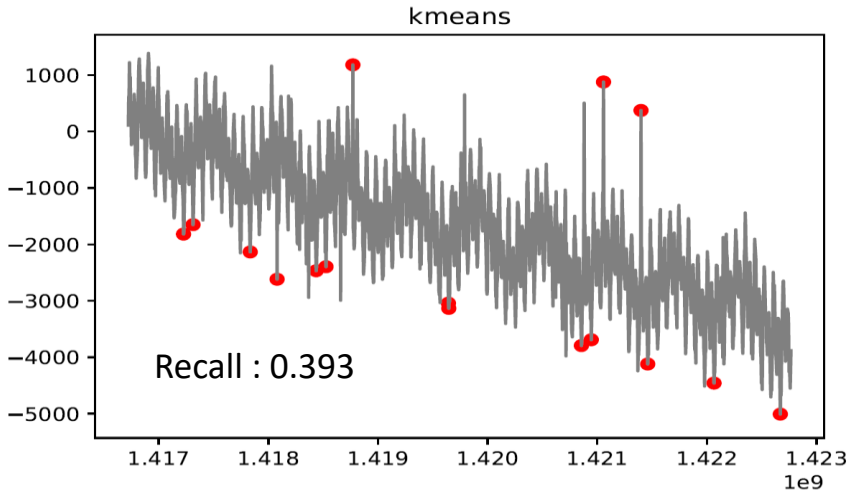


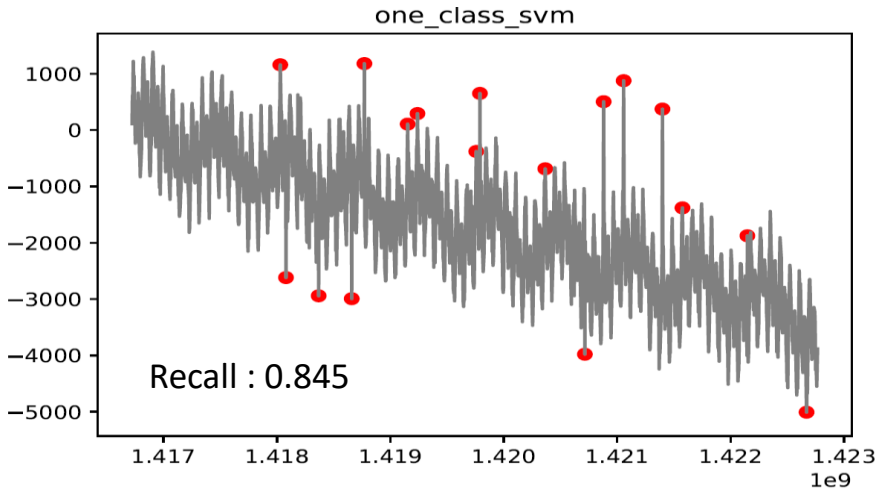
Model performance

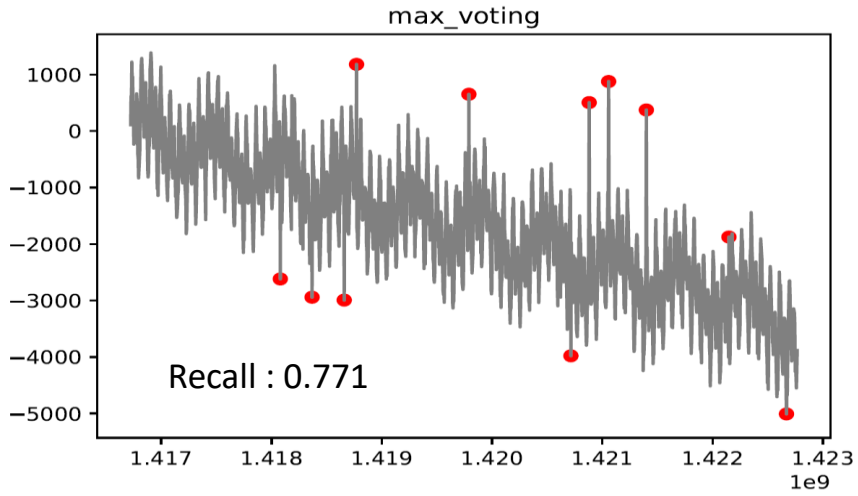
- The most intuitive way to detect anomalous points for univariate data like time series data is to use its statistical properties such as mean and standard deviation to find out extreme points in the data.
- In this method we implement this by finding the average value of the entire data and find out points which fall outside certain standard deviation(for the purpose of this exercises we are using 3 standard deviations) .
- These models are able to identify some of the anomalies, next lets see how some of the popular ML algorithms perform in comparison

Isolation Forest and One-class SVM are able to identify majority of the Anomalous points









Result

- One-class SVM is performing quite well in identifying the anomalous points followed by Isolation Forest and then K-means
- Isolation forest however runs faster compared to other algorithms
- Max voting is helpful in reducing the number of False Positives.
- Lets see how these results generalize when we run them on 67 files

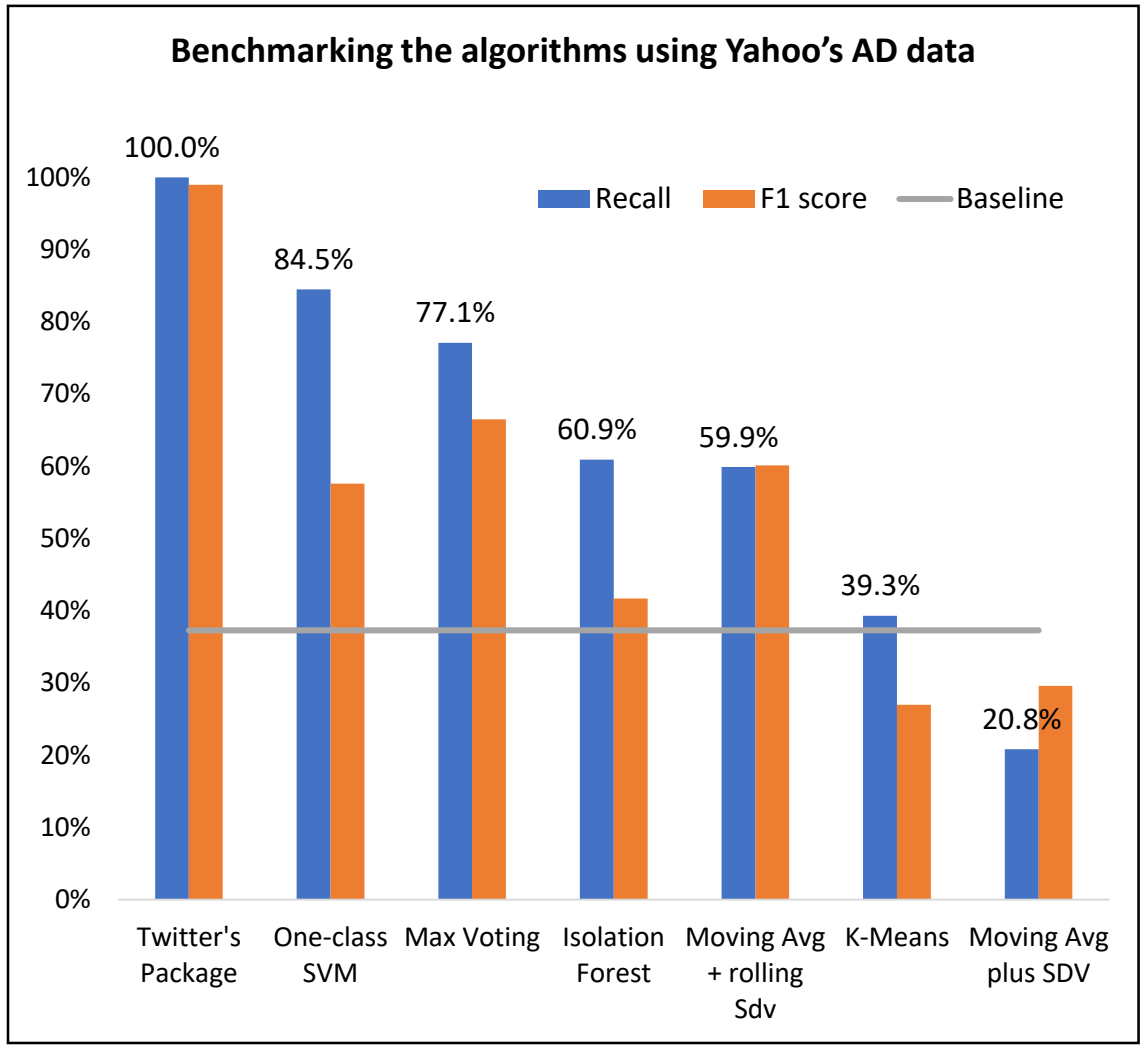
How do these Algorithms work?(1/2)

Algorithm	How they work?
Moving Average with Rolling Standard Deviation	• Mean + Standard Deviation model is that the mean is heavily influenced by fluctuations • To avoid this a rolling window is chosen and average is taken over those points in the window • This rolling window precedes the point which we are trying to detect anomaly or not
One Class SVM	• One class SVM – a variant of SVM - can only used for binary classification, one class belonging to the normal data and the other class belonging to the anomalous data • The model is trained on data with no anomalies but the model is not affected much even if it is trained with some anomalous points because the model has an inbuilt parameter that creates a soft boundary which minimized the effect of a few data points on the model prediction
Isolation Forest	• Isolation forest is built on the concept of the Decision Trees • The model tries to anomaly points by randomly splitting each variable by looking the depth of the tree, the points identified with less depth are considered as anomalies since they would be away from the major chuck of the data • The isolation forest also runs with linear complexity and is performs parallel processing

How do these Algorithms work?(2/2)

Algorithm	How they work?
K Means	• This is a distance-based anomaly detection method, in this method we first specify the number of clusters to be created • Once these clusters are formed based on the distance between the point and the cluster. The points which have a very large distance from the clusters are tagged as anomalous points. We have set the cluster size to 4 for all the data
Max Voting	• Each of the classifier algorithms have their own advantages and disadvantages • In order to boost the true positive and minimize the false positive, we have created a voting algorithm where all the above algorithms vote on a points to be an anomaly or not • If a point is tagged as an anomaly by 2 or more algorithms among the Moving Average with Rolling Standard Deviation, SVM, Isolation Forest and K means we tag that point as an anomaly
Twitter package	• This package was developed by Twitter and made opensource on the R programming environment, this package uses the Seasonal Hybrid ESD method for detecting anomalies. The back end of this method is a combination of time series decomposition and a Generalized EST test. • This model failed when the data had trend but when the trend was removed this method was very effective and was able to accurately identify majority of the anomalies on all the data set with very low False Positive.

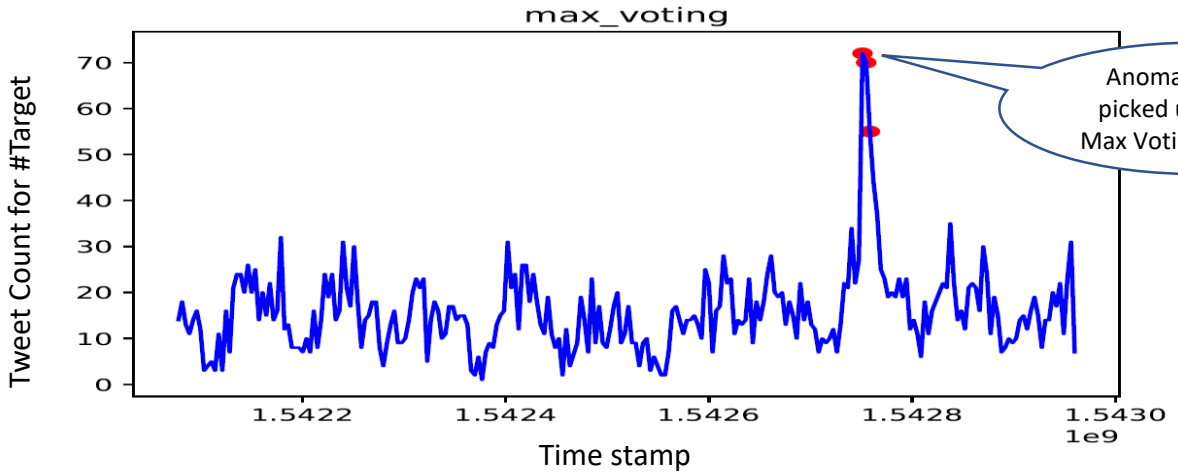
On average, all the algorithms except Moving Avg plus Sdv are able to beat the Baseline model based on Recall



Result

- All the algorithms have been run on 67 different Time Series Anomaly Detection datasets provided by Yahoo and the results are tabulated
- One-class SVM performs better the other methods for time series data except for the twitters package
- The Max voting is close to the SVM in terms of Recall Metric but it has a much higher F1 score, this is because the other algorithms reduce the number of False Positives from the data.
- Isolation Forest is performing a bit poorly than expected probably because taking a sample from an already small population might lead to unexpected behavior.
- The performance of the K Means algorithm is a lot lower than expected because the number of clusters to be used for each of the data varies a lot and it extremely hard to tune it without labelled data.

Deploying our framework on Real Twitter data could help companies in spotting trends quickly



Word Cloud for the spike

Twitter Result

- We extended our models to a real life application. We have used the Tweepy package in Python to scrape Twitter data after creating a Twitter Developer account. We can run our Anomaly Detection Algorithms on popular topics to check if there is any anomalies in it.
- For the purpose of this demo, we have considered the hashtag “Target”, for the shopping chain, to check if there is any change to its trend, upon running the Tweet Count vs time data for the Tweets regarding “#Target” during the thanksgiving week, our model have identified the above points as anomalous.
- The twitter package is also listing the same points as anomalous, hence the model is scaling well for twitter count data as well.
- To understand the drivers for this spike we have also created a word cloud using the Bag of words approach.

The performance of the models have also been benchmarked against the credit card dataset available on Kaggle

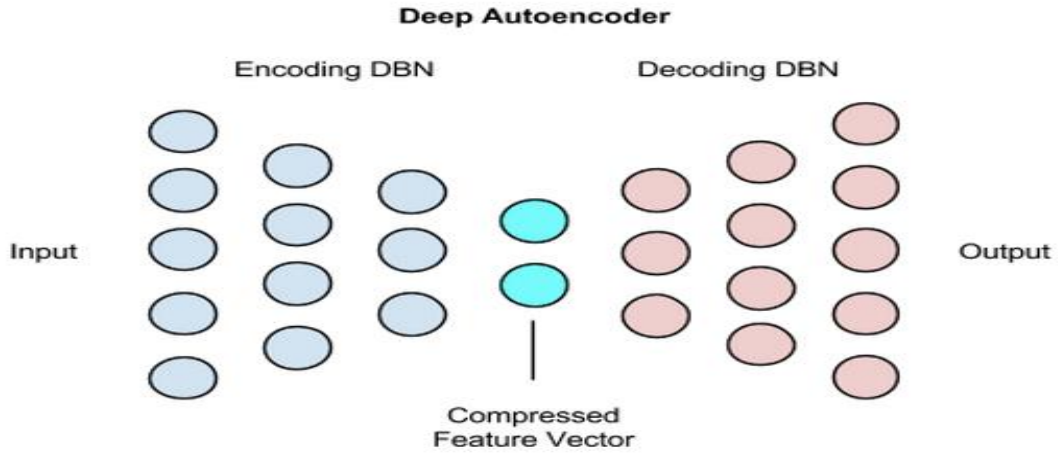
Method	Recall	AUC
One-class SVM	0.876	0.953
Isolation Forest	0.829	0.945
K-means	0.843	0.897

Result

- The algorithms could be extended to multi dimensional datasets with minimal modifications(Credit Card fraud Detection Data available on Kaggle)
- Of all the algorithms One class SVM has the highest performance followed by K-means and Isolation Forest. Isolation Forest however runs much faster compared to the other two algorithms
- All three of our models shows a high recall metric score of greater than 0.8. This implies that the models are able to tag most of the anomalies.

Right most image was obtained from : https://www.powerdms.com/resources/go-green/

Autoencoder, an unsupervised Neural Network Approach for Anomaly Detection(Needs to be explored)



Autoencoder

- One algorithm that has come up recently is the Autoencoder and needs to be explored further
- Autoencoder is an unsupervised neural network algorithm which takes a data and compresses it and then reconstructs the same data
- If we train our Autoencoder with sufficient non anomalous data it will be able to reconstruct the input with minimal change and now if anomalous data passes through the trained Autoencoder the reconstructing will be poor which will result in poor accuracy between the original data and the reconstructed
- We can identify anomalous points based on the error between the original and reconstructed data.

Image obtained from https://skymind.ai/wiki/deep-autoencoder

Any Questions?