**Anomaly Detection**

**Definition**

Finding patterns in data that lead to unexpected behaviour.

**Related topics**

* Noise removal - Not of interest but acts as hindrance to analysis
* Novelty detection - Detect previously unobserved patterns
  + Incorporated into the normal model after being detected, in contrast to anomaly detection

**Challenges**

1. Anomalous observation near the boundary can be normal, and vice versa.
2. Anomalies by malicious adversaries appear normal, hence more difficult to define normal behaviour
3. Current notion of normal behaviour may not be representative in the future
4. Exact notion of anomaly is different across application domains
5. Labeled data is hard to get
6. Data often contains noise similar to actual anomalies, which are hard to distinguish and remove

**Applications**

1. Cyber-intrusion detection
2. Fraud detection
3. Medical anomaly detection
4. Industrial damage detection
5. Image processing
6. Textual anomaly detection
7. Sensor networks

**Techniques: Analysis of computation complexity in training and testing**

1. Classification based
2. Clustering based
3. Nearest Neighbor based
4. Statistical
5. Information theory
6. Spectral theory

**Anomaly types**

* Point anomalies: Individual data instance
* Contextual anomalies: Anomalous only in specific contexts (or conditional anomaly)
  + Time-series and spatial data
* Collective anomalies: Anomalous collectively but non-anomalous as individual data instances
  + E.g. Low electrocardiogram output for abnormally long time

**Detection types**

* Supervised: Predict normal vs anomaly classes
  + Imbalanced class
  + Accurate and representative labels for anomaly classes is challenging
* Semi-supervised: Labeled only for normal class
* Unsupervised: Do not require training data
  + Assume normal instances far more frequent than anomalies, or else suffer from high false alarm rate

**Application Domain: Industrial Damage Detection**

* Techniques: Fault detection in mechanical units
  + Parametric statistical modeling [7.1]
  + Non-parametric statistical modeling [7.2.2]
  + Neural networks [4.1]
  + Spectral [9]
  + Rule based systems [4.4]
* Techniques: Structural damage detection
  + Statistical profiling using histograms [7.2.1]
  + Parametric statistical modeling [7.1]
  + Mixture of models [7.1.3]
  + Neural networks [4.1]

**Application Domain: Sensor Networks**

* Techniques: Anomaly detection in sensor networks (sensor fault or intrusion)
  + Bayesian networks [4.2]
  + Rule-based systems [4.4]
  + Parametric statistical modeling [7.1]
  + Nearest neighbor-based techniques [5]
  + Spectral [9]
* Challenges
  + Lightweight due to severe resource constraints
  + Distributed data mining approach due to distributed data collection
  + Presence of noise

**Technique: Rule-based** (2 steps)

* Approach

1. Rule learning algorithm: Learn rules from training data

* RIPPER, decision trees
* Each rule has confidence value proportional to prediction accuracy

1. Find rule that best captures the test instances

* Confidence with best rule = Inverse of anomaly score of test instance
* Algorithms
  + Association rule mining (unsupervised): 1-class anomaly detection

**Technique: Nearest neighbor-based** (unsupervised)

* Assumption: Normal data instances occur in densely, anomalies sparsely
* Distance/similarity measures
  + Euclidean distance (continuous data)
  + Matching coefficient (categorical data)
* Anomaly score
  + Distance of data instance to k nearest neighbors
    - Connectivity of a hypergraph that models categorical values
  + Inverse of relative density of each data instance = 1 / [n / (pi \* d\*\*2)] (or 1/n, 1/d)
* Disadvantages
  + Computation complexity: O(n^2)
  + Normal instances with low density may become false positives, or vice versa
  + Performance relies on distance measure (challenging when data is complex)

**Technique: Statistical detection**

* Parametric
  + Assumption: Normal data generated by parametric distribution
  + Anomaly score: Inverse of probability density function
  + Alternative: Hypothesis testing
  + Examples
    - Gaussian model-based
      * Parameters estimated using Maximum Likelihood Estimates (MLE)
      * Anomaly score: Distance to estimated mean (3σ away, 99.7% of data)
    - Regression model-based
      * E.g. Autoregressive Moving Average (ARMA) model
        + Transform multivariate to univariate time-series using projection pursuit to maximise Kurtosis coefficient, followed by univariate test statistics
* Nonparametric
  + Histogram-based
    - Bin training data; If test instance falls outside all bins, it is anomalous
    - Anomaly score: Frequency of bin it falls into
    - Challenge: Size of bins (high false alarm rate vs high false negative rate)
* Disadvantages
  + Assume data is generated from a distribution, which often does not hold true, especially from high dimensional real data sets
  + Histograms cannot capture interactions between different attributes in multivariate data. An anomaly might have attribute values that are individually frequent but rare in combination.

**Technique: Spectral detection**

* Assumption: Data embedded into lower dimensional subspace can cause normal instances and anomalies to appear significantly different
* Principal Component Analysis (PCA)

**Contextual Anomalies**

* Attributes: Spatial, graphs, sequential, profile (attributes)
* Approaches
  + Reduce to Point Anomaly Detection problem
  + Use structure in the data (e.g. regression)
* Disadvantage: Only used when context can be defined

**Time Series Outlier Detection**

* Algorithms
  + Autoregressive moving average (ARMA)
  + Autoregressive integrated moving average (ARIMA)
  + Seasonal autoregressive moving average (SARIMA)
    - Trend elements
      * p: Trend autoregression order
      * d: Trend difference order
      * q: Trend moving average order
    - Seasonal elements
      * P: Seasonal autoregressive order
      * D: Seasonal difference order
      * Q: Seasonal moving average order
      * m: No. of timesteps for single seasonal period
  + Holt-Winters
    - Univariate forecasting method involving only one explanatory variable.
    - Automatic procedure but the user can intervene if required.
    - Uses simple exponential smoothing and forecasted values depend on the level, slope and seasonal components of the series.
  + Vector autoregression (VARMA)
  + Cumulative sum statistics (CUSUM)
  + Exponentially weighted moving average, etc
* Unsupervised Discriminative Approaches

(Relies on definition of similarity function that measures similarity between sequences)

* + Similarity measures
    - Match count based sequence similarity (efficient)
    - Normalized length of longest common subsequence (adjustable to noise but expensive because dynamic programming)
  + Clustering methods

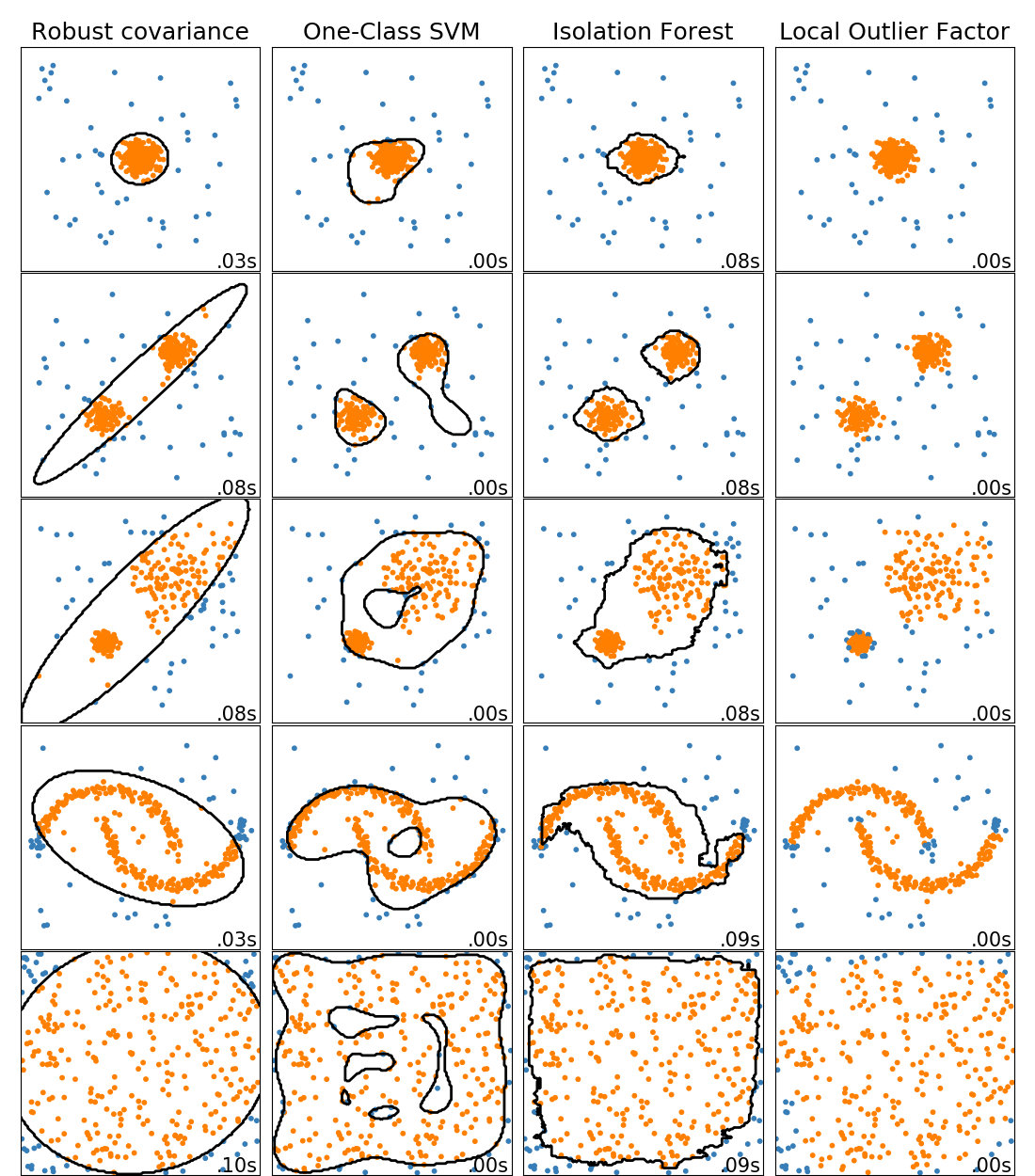
(Application specific as different complexity & varying adaptability to clusters of different numbers, shapes and sizes)

* + - K-means
    - EM
    - Phased k-means
    - Dynamic clustering
    - K-medoids
    - Single-linkage clustering
    - Clustering of multivariate time series in the principal components space
    - One-class SVM
    - Self-organizing maps
* Unsupervised Parametric Approaches
  + Models
    - Finite state automata (FSA)
    - Markov models
    - Hidden markov models (HMM)

**Outlier Detection for Stream Data**

* Evolving Prediction Model
  + Online sequential discounting
    - E.g. Sequentially Discounting Auto-Regressive (SDAR) model learning
  + Dynamic cluster maintenance
  + Dynamic bayesian networks
* Distance based Outliers for Sliding Windows
  + Distance based Global Outliers
  + Distance based Local Outliers
* Outliers in High-dimensional Data Streams

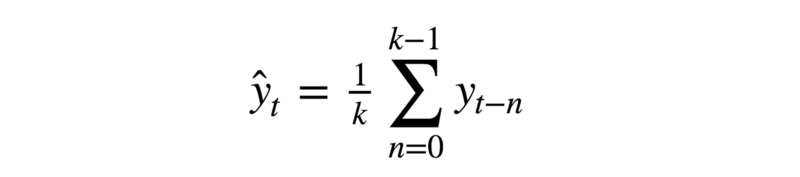
**Scikit-learn: Overview of outlier detection methods**



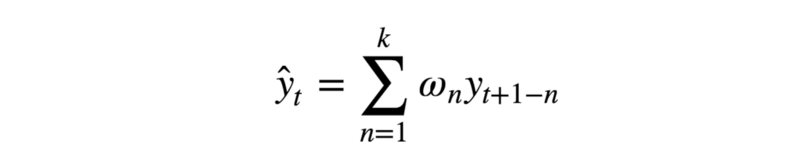
* ensemble.IsolationForest
* neighbors.LocalOutlierFactor
* covariance.EllipticEnvelope

**Formula**

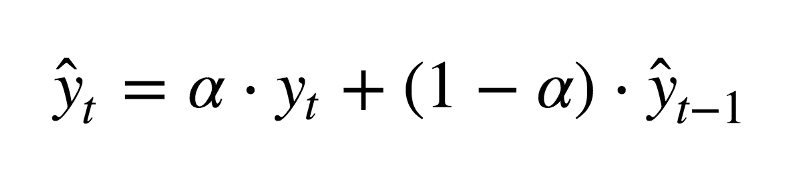
* Moving average



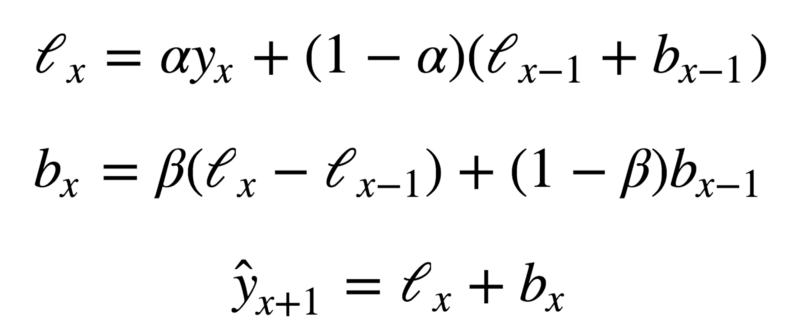
* Weighted average



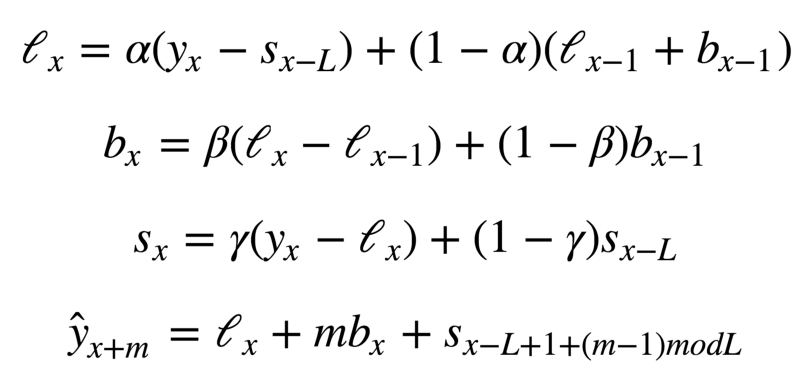
* Exponential smoothing, i.e. ARIMA(0,1,1) without constant



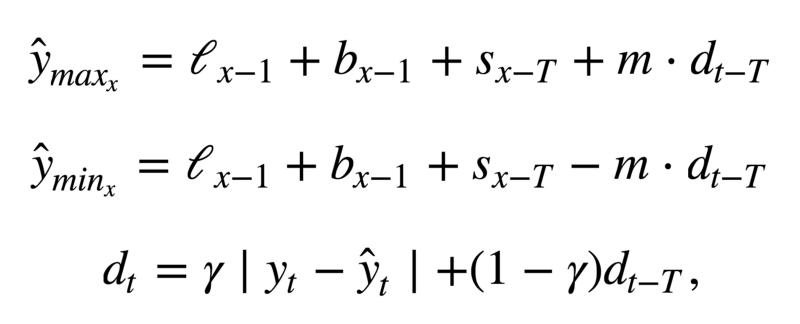
* Double exponential smoothing (Intercept/level, trend/slope), i.e. ARIMA(0,1,1) with constant



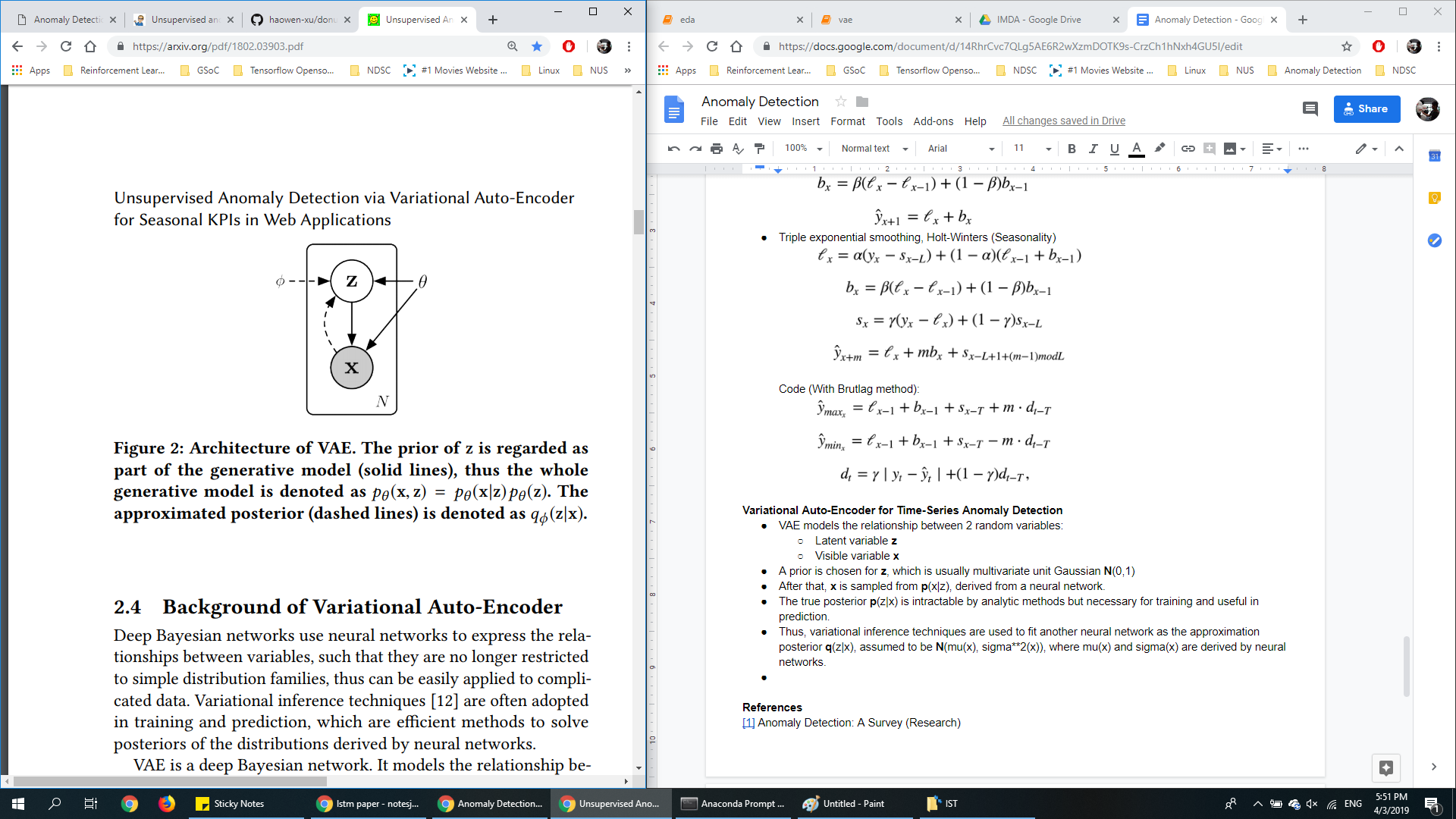
* Triple exponential smoothing, Holt-Winters (Seasonality)



Code (With Brutlag method):

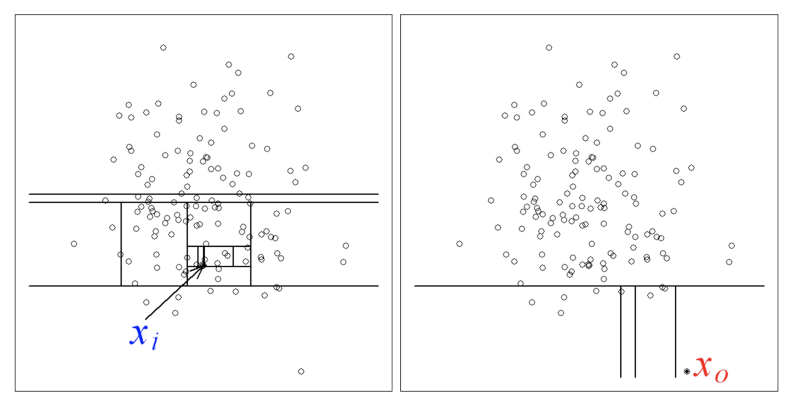


**Variational Auto-Encoder for Time-Series Anomaly Detection**



* VAE models the relationship between 2 random variables:
  + Latent variable **z**
  + Visible variable **x**
* A prior is chosen for **z**, which is usually multivariate unit Gaussian **N**(0,1)
* After that, **x** is sampled from **p**(x|z), derived from a neural network.
* The true posterior **p**(z|x) is intractable by analytic methods but necessary for training and useful in prediction.
* Thus, variational inference techniques are used to fit another neural network as the approximation posterior **q**(z|x), assumed to be **N**(mu(x), sigma\*\*2(x)), where mu(x) and sigma(x) are derived by neural networks.

**Isolation Forest**

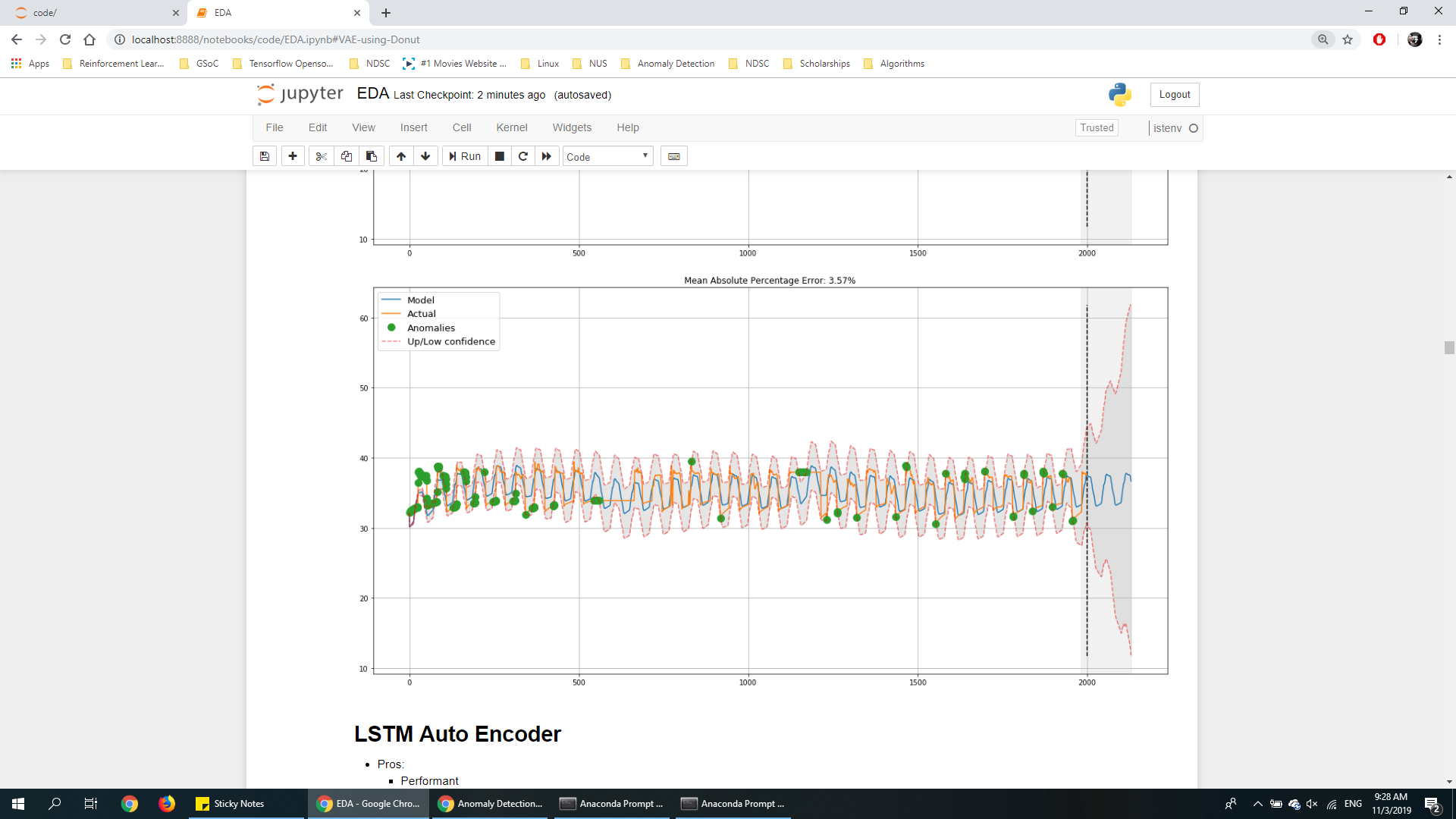
* Identifies anomalies instead of profiling normal data points
* 
* By random partitioning to identify points, anomalies (right) require less partitions to be identified than a normal point (left).
* Anomaly score: s(x, n) = 2 \*\* [-E(h(x)) / c(n)]
  + h(x) - path length of observation x,
  + c(n) - average path length of unsuccessful search in a binary search tree
  + n - no. of external nodes

**Decision Trees**

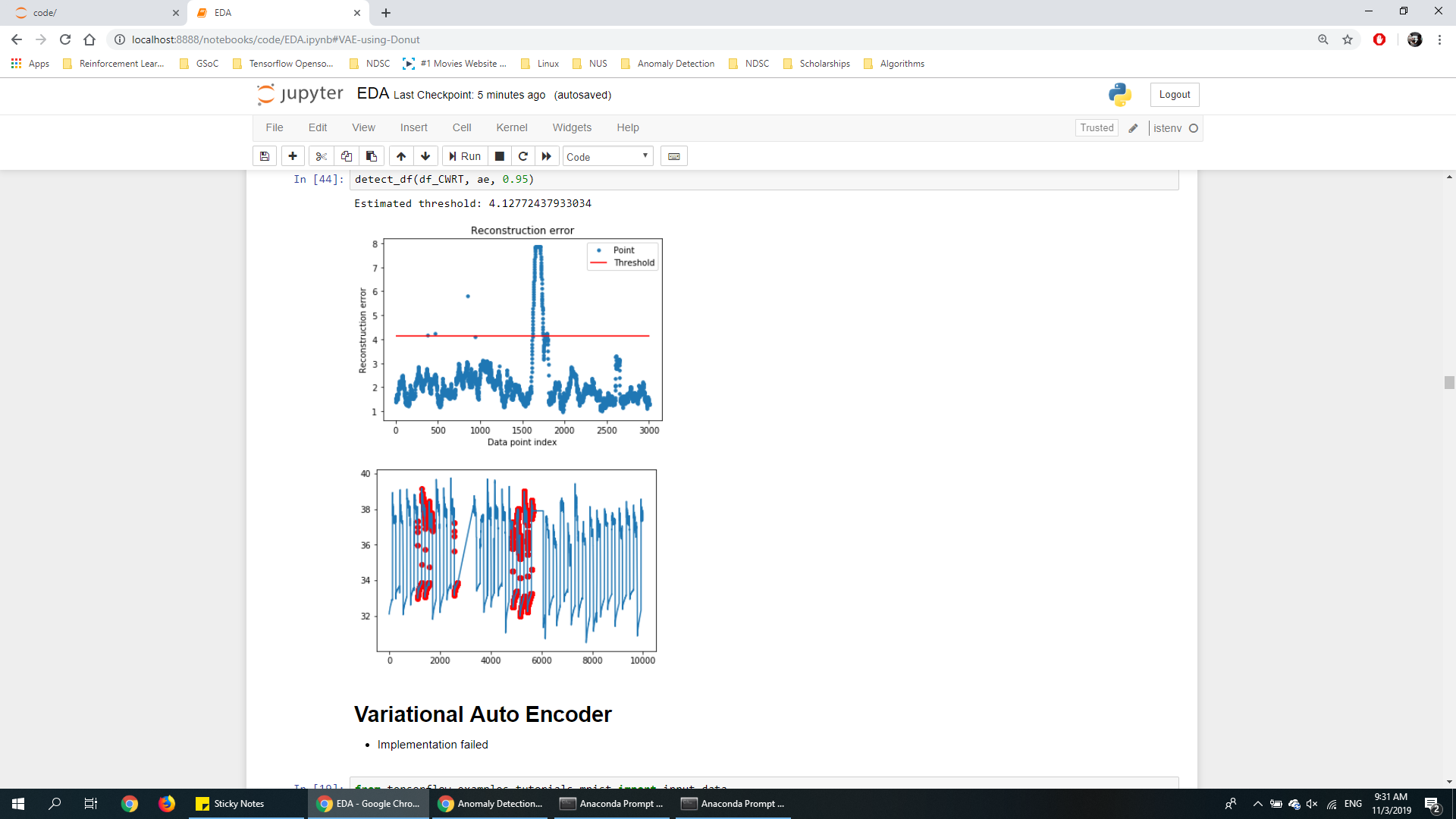
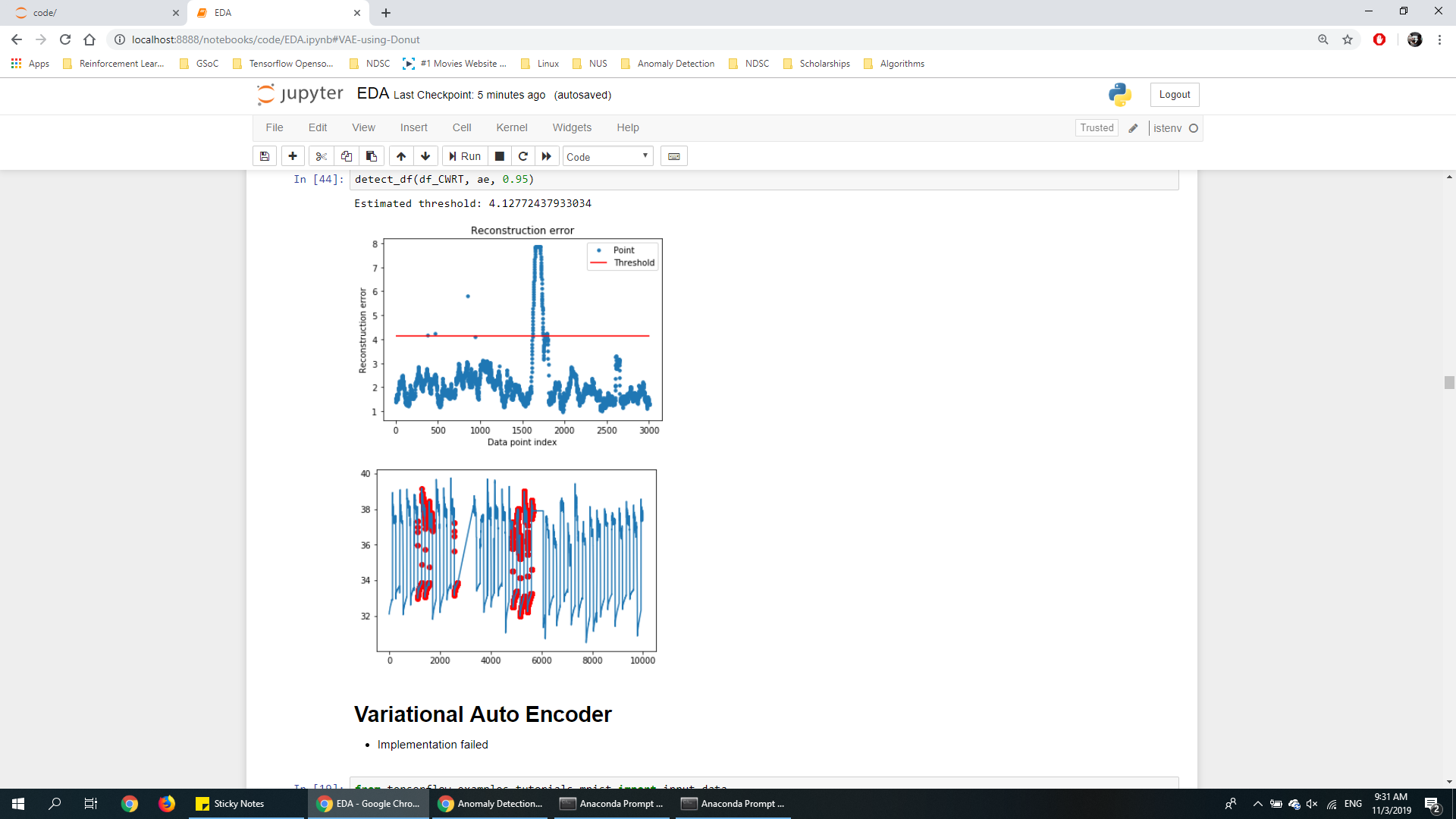
* Gini Importance / Mean Decrease in Impurity (MDI)
  + Feature importance = Sum over no. of splits (across all trees) that include the feature

**Implementations**

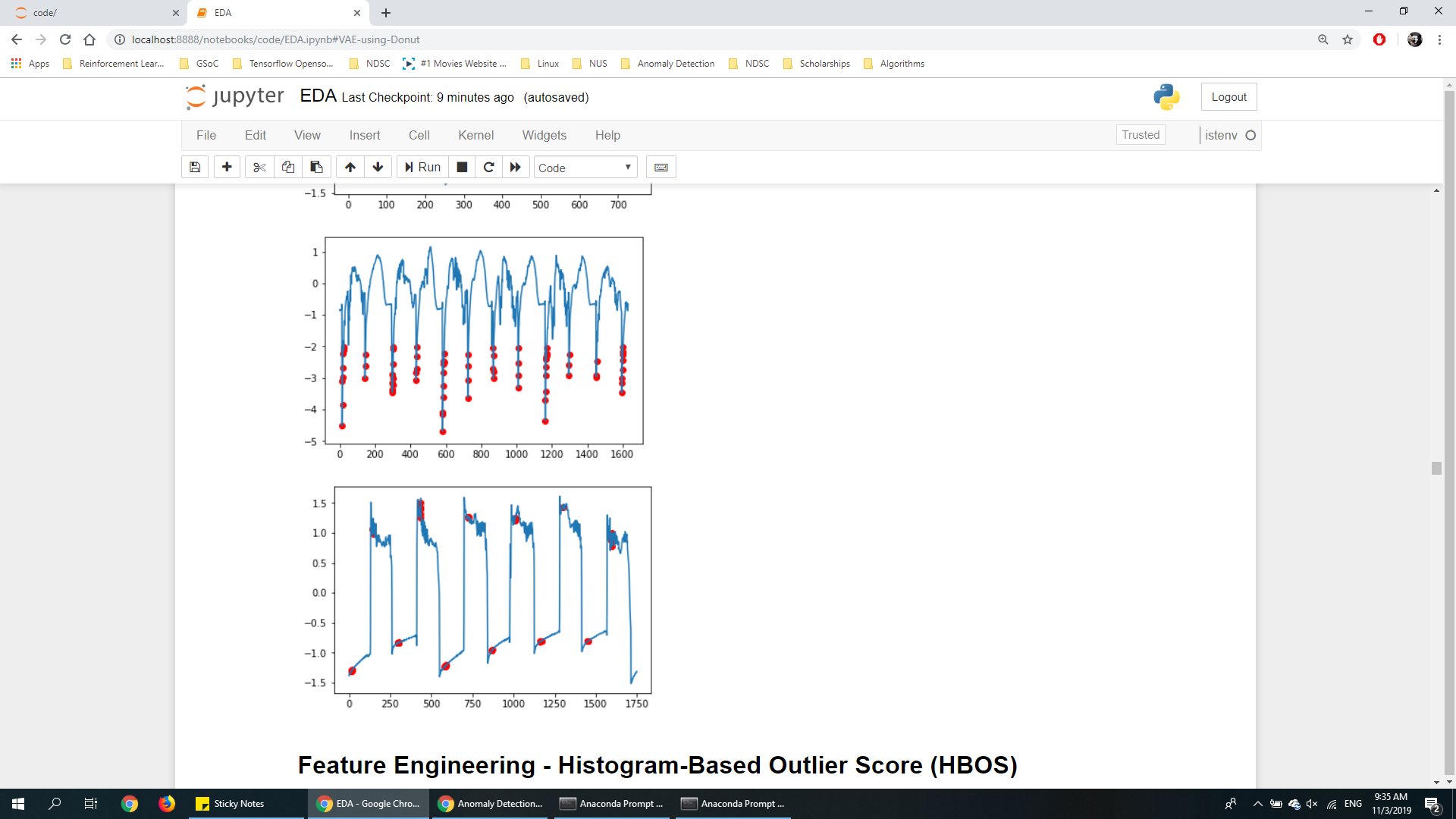
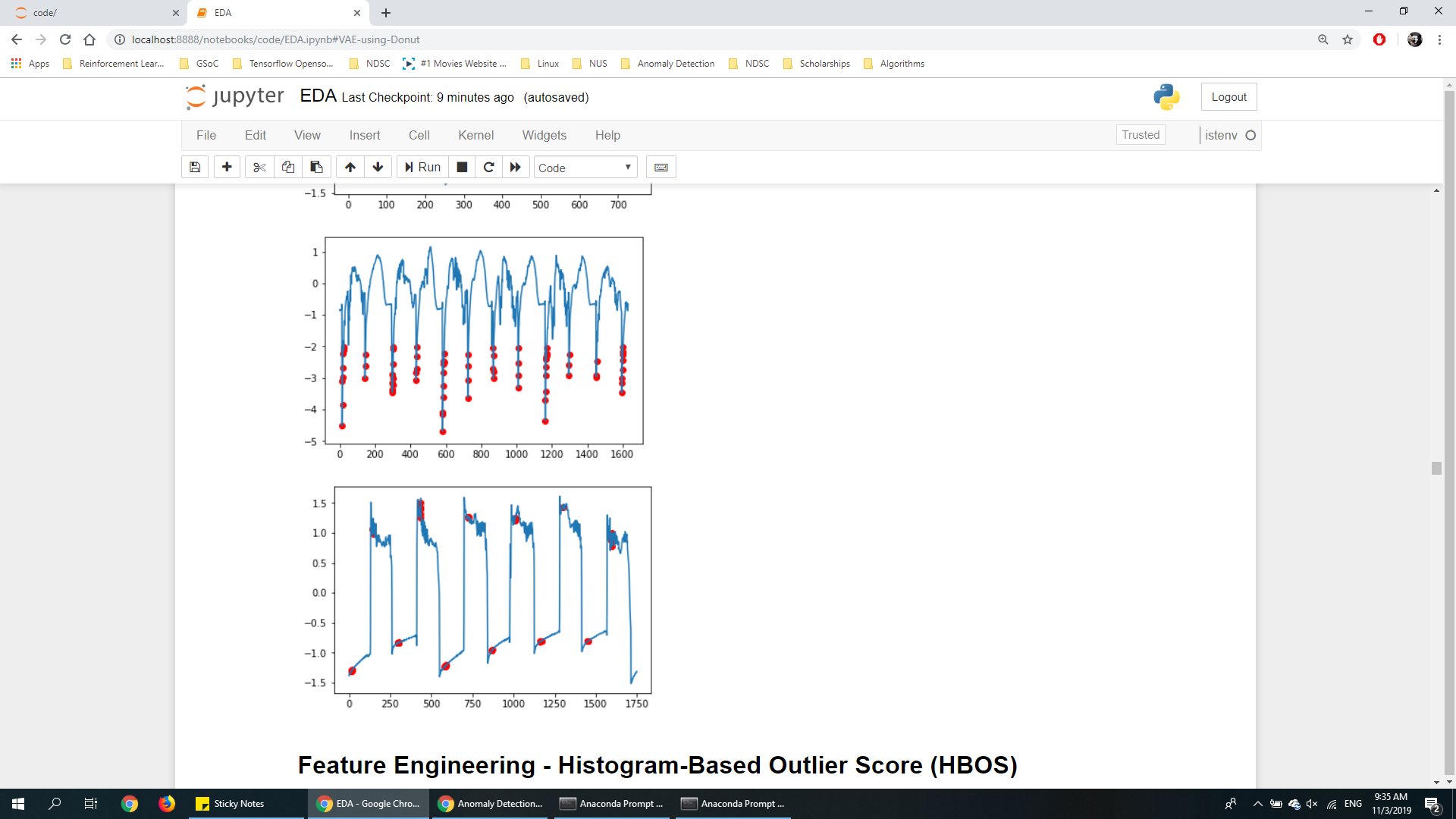
* **Holt-Winters**
  + Pros:
    - Simple statistical method
  + Cons:
    - Not robust to spike changes due to moving average mechanism
    - Cannot predict future anomalies



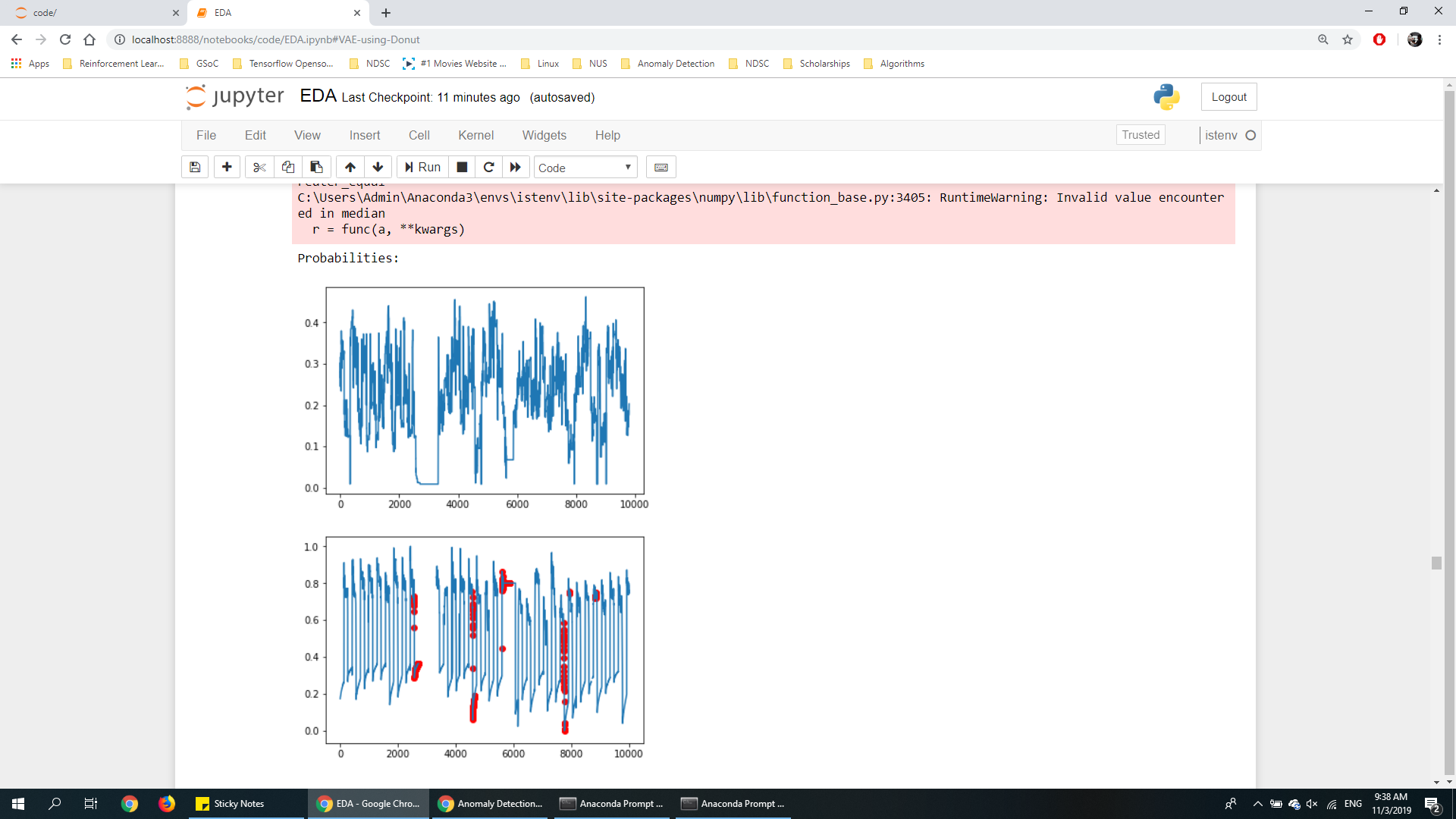
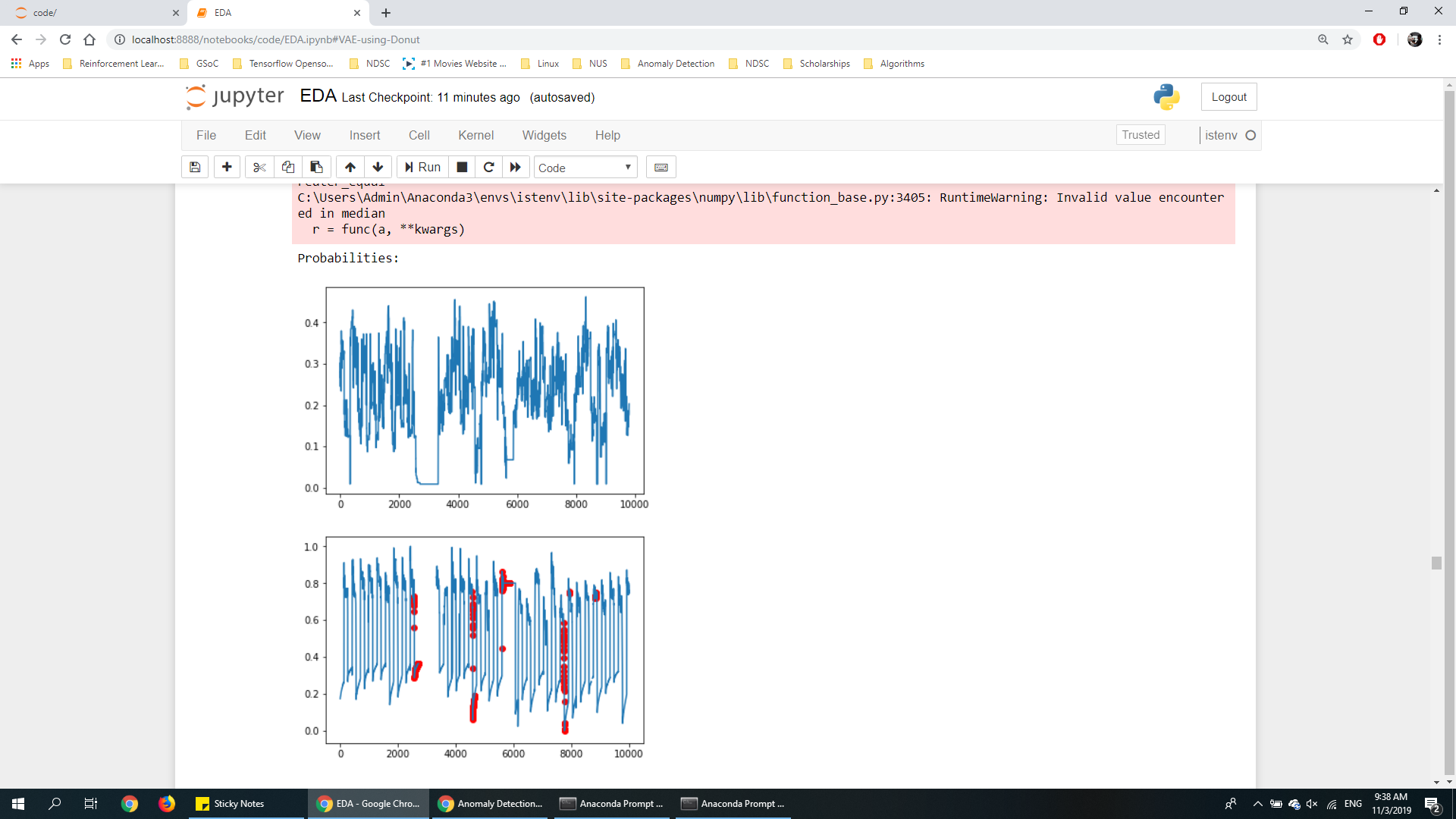
* **LSTM Autoencoder**
  + Pros:
    - Performant
  + Cons:
    - Unexplainable -- Attention?
    - Cannot predict future anomalies



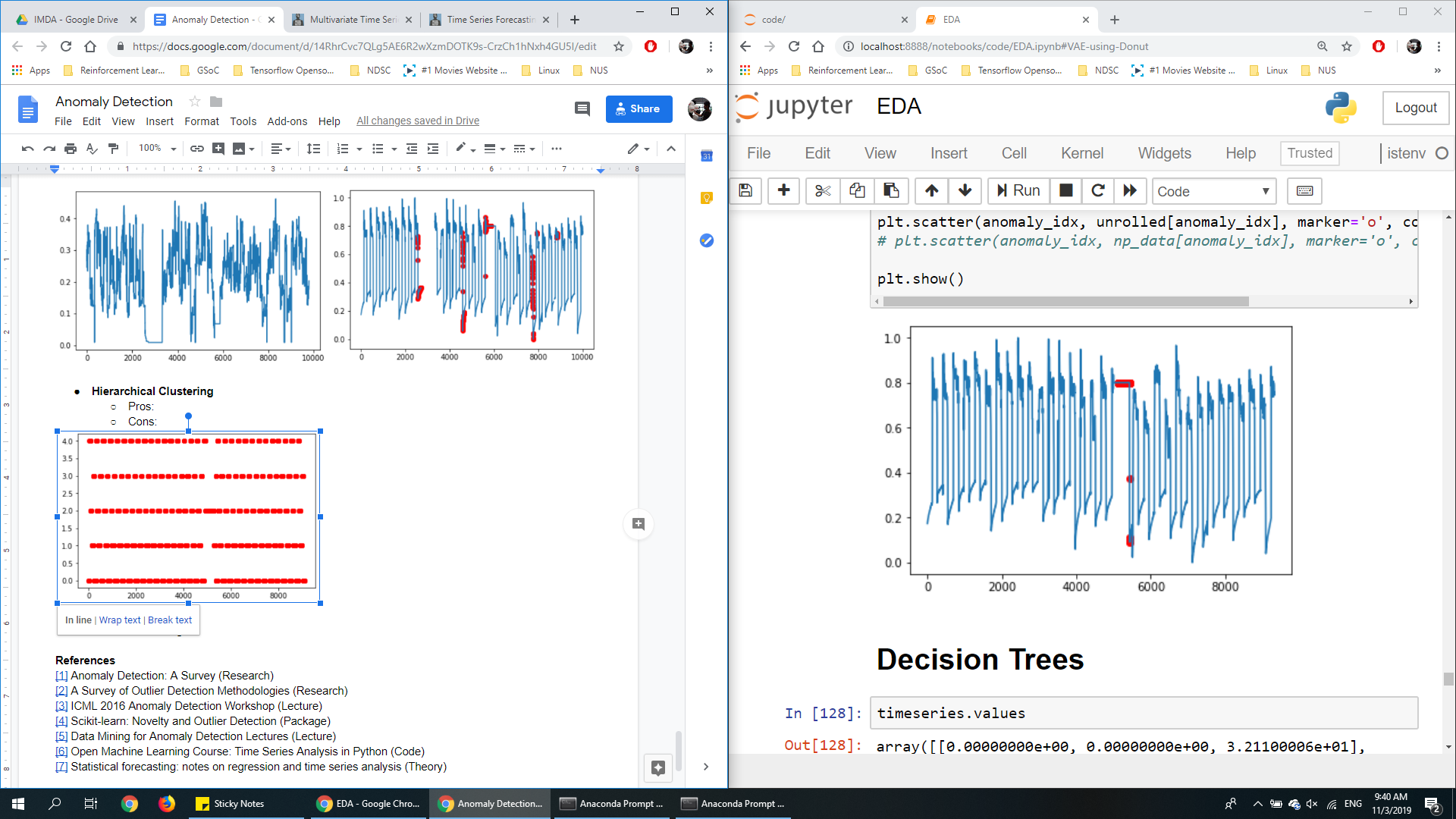
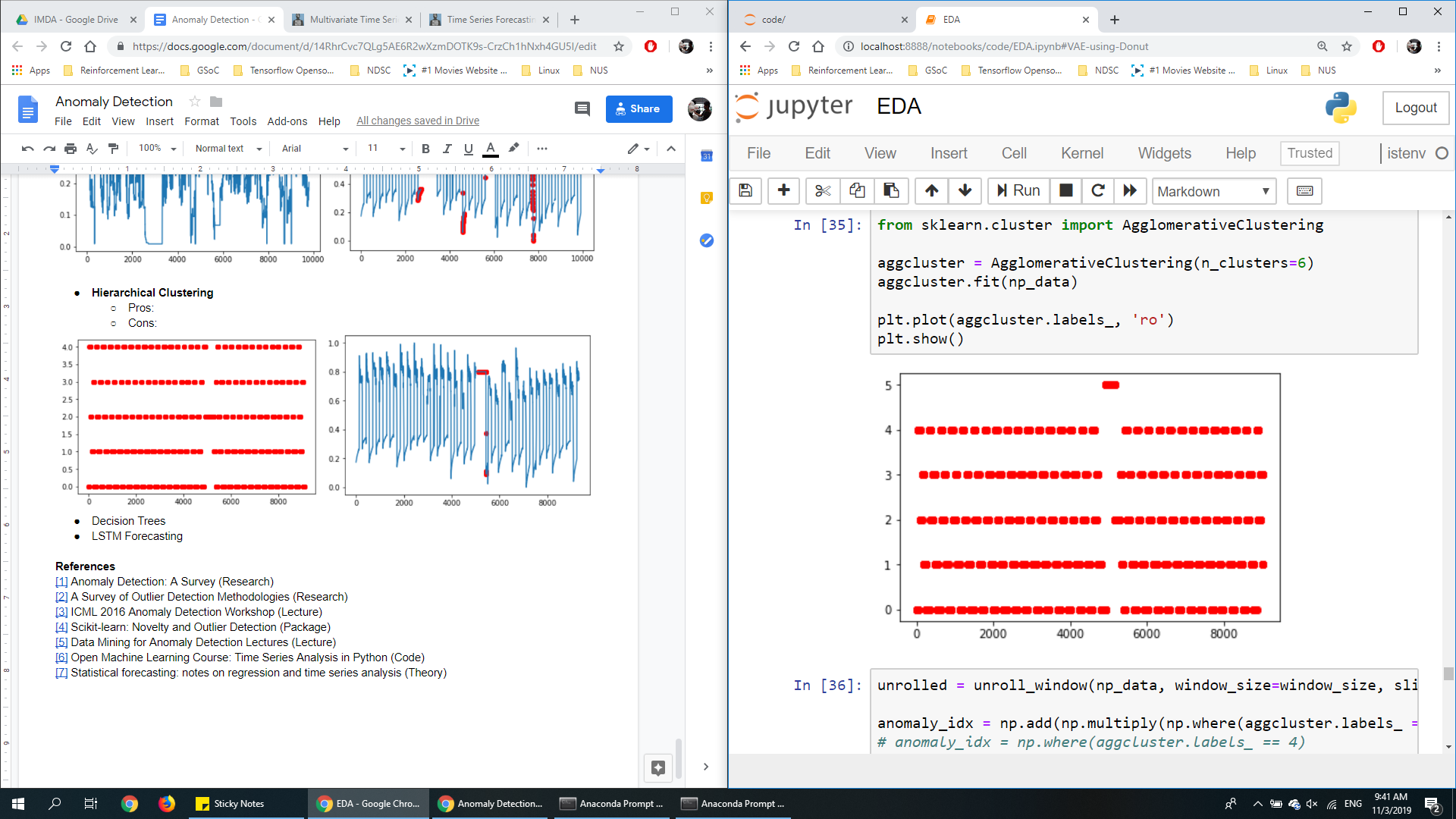
* **Variational Autoencoder (VAE)**
  + Cons:
    - Cannot predict future anomalies
    - VAE leads to local [PCA](https://arxiv.org/abs/1812.06775)



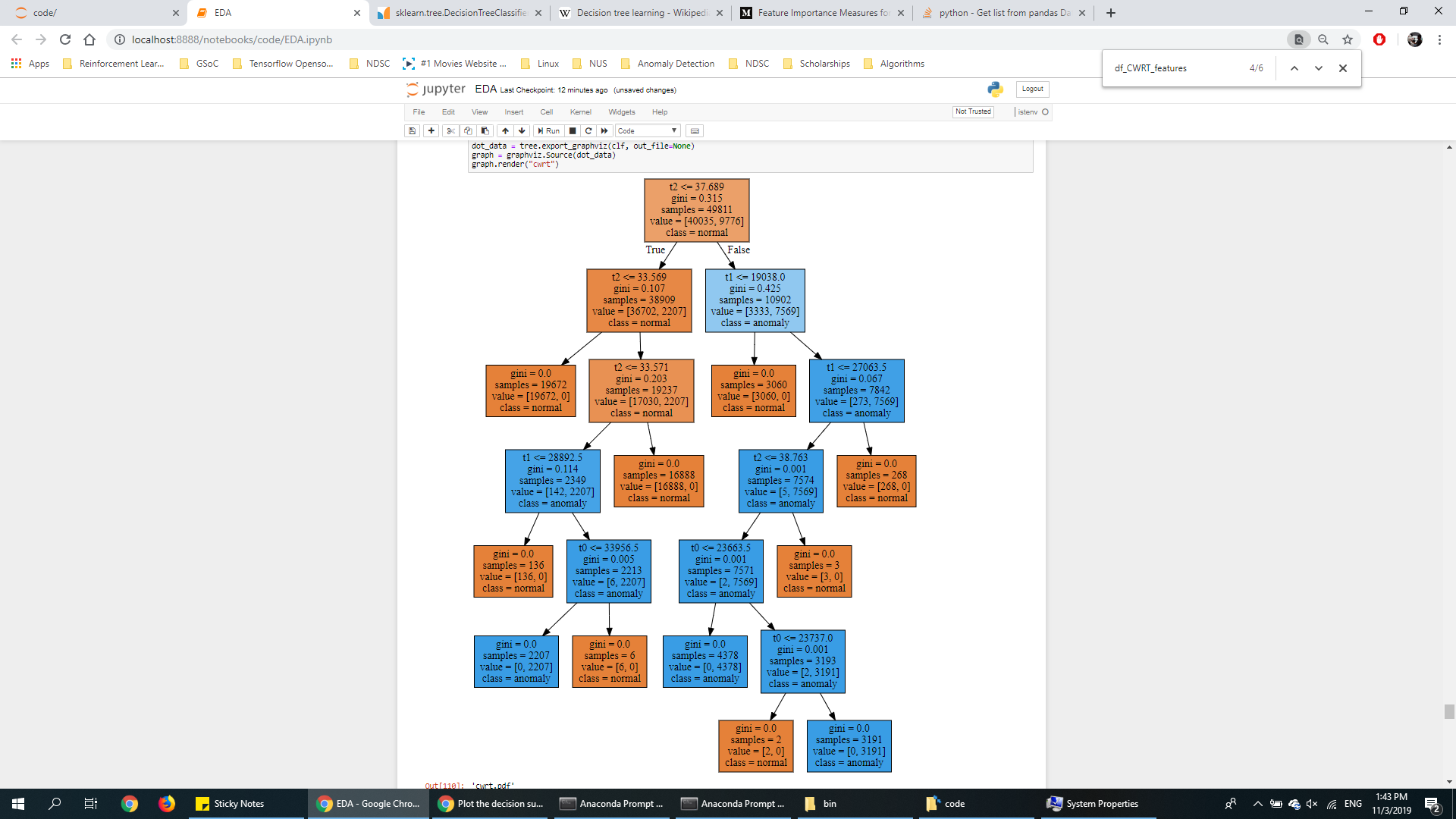
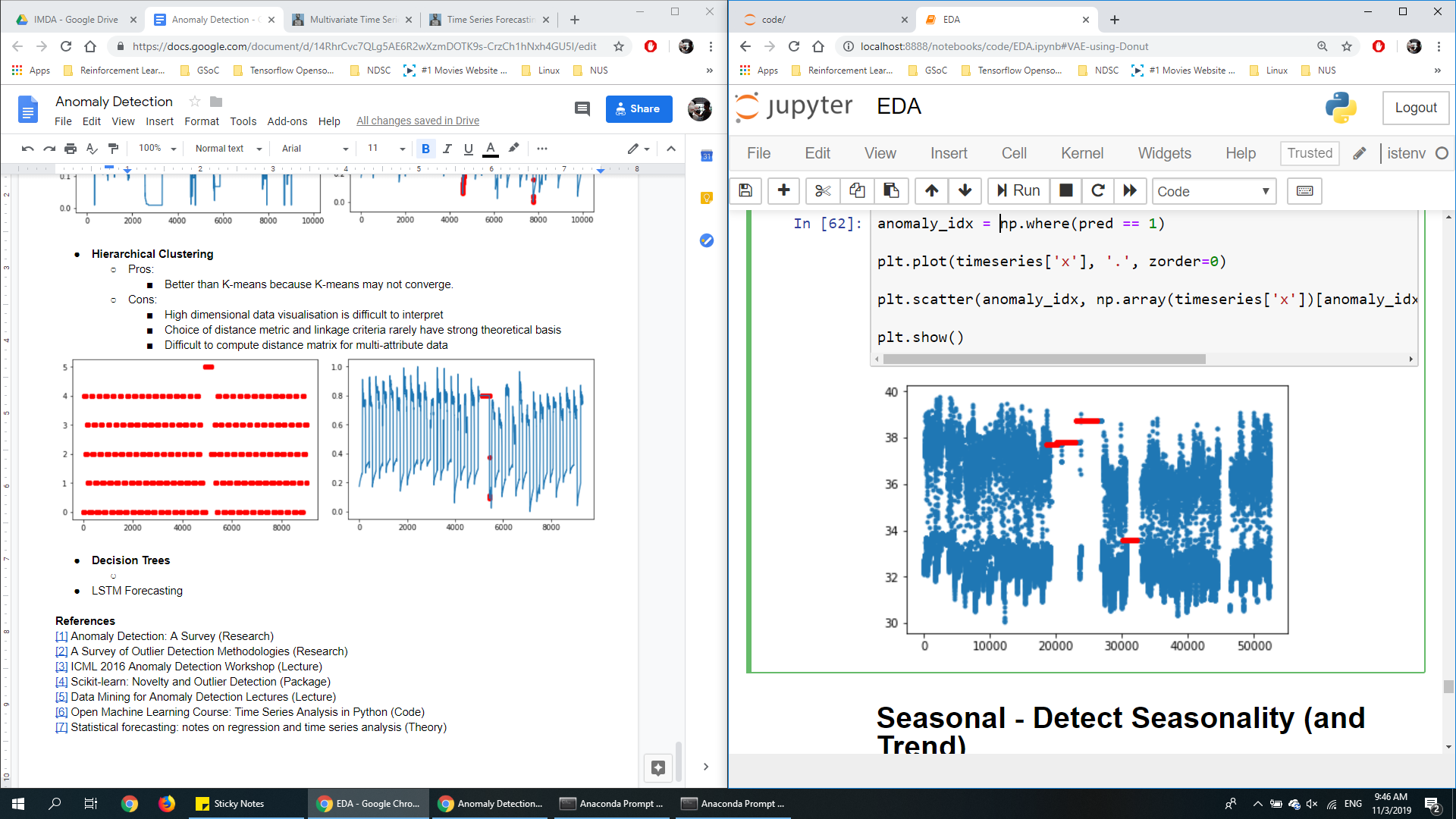
* **Histogram-Based Outlier Score (HBOS)**
  + Pros:
    - Explainable
    - Computationally cheap / fast
  + Cons:
    - Less performant
    - Cannot predict future anomalies
    - Fundamentally similar to LSTM and VAEs: Map an underlying distribution of time-series data.



* **Hierarchical Clustering**
  + Pros:
    - Better than K-means because K-means may not converge.
  + Cons:
    - High dimensional data visualisation is difficult to interpret
    - Choice of distance metric and linkage criteria rarely have strong theoretical basis
    - Difficult to compute distance matrix for multi-attribute data



* **Decision Trees**
  + Pros:
    - Explainable
    - Performant
  + Cons:
    - Requires truth labels

****

* **LSTM Forecasting**

**References**

[[1]](https://www-users.cs.umn.edu/~baner029/papers/09/anomaly.pdf) Anomaly Detection: A Survey (Research)

[[2]](http://eprints.whiterose.ac.uk/767/1/hodgevj4.pdf) A Survey of Outlier Detection Methodologies (Research)

[[3]](https://sites.google.com/site/icmlworkshoponanomalydetection/accepted-papers) ICML 2016 Anomaly Detection Workshop (Lecture)

[[4]](https://scikit-learn.org/stable/modules/outlier_detection.html) Scikit-learn: Novelty and Outlier Detection (Package)

[[5]](http://videolectures.net/ecmlpkdd08_lazarevic_dmfa/) Data Mining for Anomaly Detection Lectures (Lecture)

[[6]](https://medium.com/open-machine-learning-course/open-machine-learning-course-topic-9-time-series-analysis-in-python-a270cb05e0b3) Open Machine Learning Course: Time Series Analysis in Python (Code)

[[7]](https://people.duke.edu/~rnau/411home.htm) Statistical forecasting: notes on regression and time series analysis (Theory)