# Liam Reid Final Year Project Research

## SOM Algorithm

Self-Organizing Map

### SOM Description

SOM is an unsupervised learning machine. It is a type of artificial neural network (ANN). It is a 2 dimensional representation of a higher dimensional dataset. It can make high dimensional data easier to visualize and analyze.

It operates in 2 modes:

* Training
  + Input dataset to generate lower dimensional representation of input data (map space).
* Mapping
  + Classifies additional input data using a generated map.

(Ralhan, 2018)

### Learning in SOM

Goal:

* Cause different parts of the network to respond similarly to certain input patterns.

Neurons:

* (Weights) Usually a random small value.
  + Slow
* Sampled evenly from subspace spanned by 2 largest principle component eigenvectors.
  + Principle component: Direction of a line that best fits.
  + Eigenvectors: Nonzero vector changes by scalar factor when linear transformation applied.
  + Faster

Network must be fed a large amount of vectors (kinds of vectors expected during mapping). Examples are administered many times as iterations.

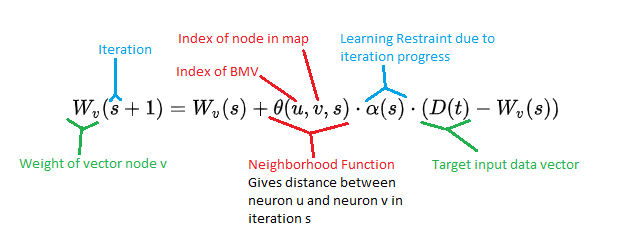
Learning utilizes competitive learning (form of unsupervised learning where nodes compete for the right to respond to a subset of the input data).

### SOM Algorithm

1. Each of the nodes weights are initialized.
2. An input vector is randomly picked from training data set.
3. Each node is then examined.
   1. Euclidian distance is sued to find similarity between input vector and map’s node’s weight vector.
   2. The node that produces the smallest distance is the winning node and known as the Best Matching Unit (BMU).
4. Neighborhood of BMU is calculated. The amount of neighbors decreases over time.
5. Winning weight is rewarded by becoming more like the sample vector.
   1. Neighbors also become more like sample vectors.
   2. The closer it is to BMU, the more weights get altered.
   3. The further away a neighbor is from the BMU means it learns less.
6. Repeat step 2 for N iterations.

(Ralhan, 2018)

### Mathematical Algorithm for SOM



Wikipedia. (n.d.) Self-organizing map. (Online) Available at: <https://en.wikipedia.org/wiki/Self-organizing_map> (Accessed: 20th October 2021)

Ralhan, A. (2018) Self Organizing Maps. (Online) Available at: <https://medium.com/@abhinavr8/self-organizing-maps-ff5853a118d4> (Accessed: 20th October 2021)

## K-Means Clustering

“A cluster refers to a collection of data points aggregated together because of certain similarities.”

### K-Means Clustering Description

Unsupervised learning algorithm used in solving clustering problems.

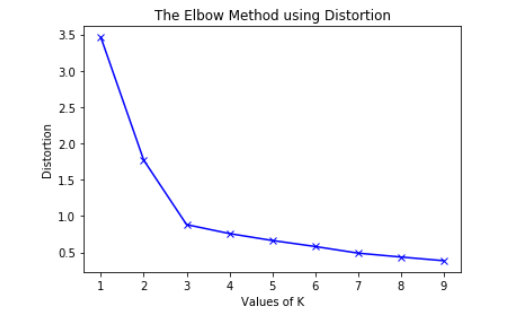
### K-Means Clustering Algorithm

1. Select number K (number of clusters).
2. Select random K points (centroids).
3. Assign each data point to their closest centroid which will form the pre-defined k clusters.
4. Calculate variance and place a new centroid for each cluster.
5. Repeat step 3 – Reassign each data point to the new closest centroid of each cluster.
6. If any reassignment occurs, got to step 4 else the model is ready.

(Garbade, 2018)

### How to Choose ‘K’ Number of Clusters

Elbow method

* Uses WCSS – Within cluster sum of squares
* Formula example for 3 clusters
  + Where
* For k values 1-10, WCSS calculated and plotted on a graph against k.
  + Figure 1: Graph showing WCSS values plotted against values of k. (GeeksForGeeks, 2021)

Garbade, M. (2018) Understanding K-means Clustering in Machine Learning. (Online) Available at: <https://towardsdatascience.com/understanding-k-means-clustering-in-machine-learning-6a6e67336aa1> (Accessed: 20th October 2021)

GeeksForGeeks Website. (2021) Elbow Method for optimal value of k in K-means. (Online) Available at: <https://www.geeksforgeeks.org/elbow-method-for-optimal-value-of-k-in-kmeans/> (Accessed: 20th October 2021)

## Local Outlier Factor

(LOF)

### K Distance and K Neighbors

K value determines how many neighbors we are looking at, K distances are calculated according to this value. If k=5, the distance to the nearest fifth neighbor is looked at. If k is small, it may become sensitive to noise. On the other hand, if it is large, it may not detect local anomalies.

### Reachability Distance

This is the maximum distance of 2 points and the k-distance of the second point. Distance can be Euclidean.

### Local Reachability Distance

How far we need to go from the point we are at to reach the next point or set of points. Reachability distances are used to calculate the local accessibility density. We calculate the density by taking the inverse of the summing and dividing the reachability distances of a points neighbors by the value of k.

### Local Outlier Factor

Local reachability density compared to local reachability of k’s nearest neighbors. Density of each neighbor is summed and divided by number of neighbors.

### Local Outlier Factor Algorithm

1. Local density determined by estimating distances between data points that are neighbors.
2. Local density calculated for each point.
3. Check which data points have similar densities and which have less densities than its neighbors.
4. Ones with lesser densities are considered outliers.

### Suitability

* Advantages
  + LOF can determine local outliers, can be very accurate compared to other algorithms
* Disadvantages
  + Detection accuracy affected in higher dimensions.
  + Not suitable for large datasets, high time complexity.

(Ferhatmetin, 2012)

Ferhatmetin. (2012) LOCAL OUTLIER FACTOR. (Online) Available at: <https://medium.com/datasciencearth/local-outlier-factor-7821b5651bc5> (Accessed 20th October 2020)

## K-Nearest Neighbor

KNN

### Description

KNN is a supervised learning algorithm. KNN works by assuming similar things exist in close proximity to each other.

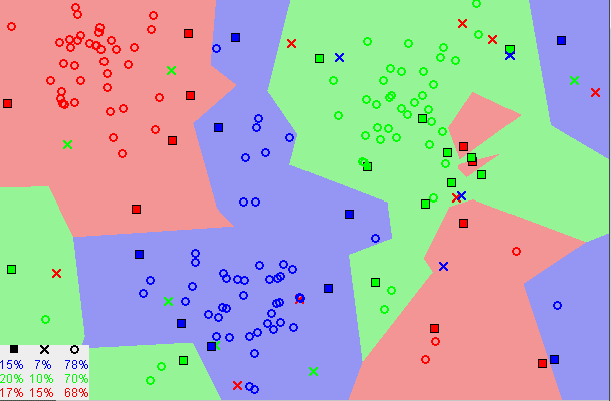


Figure 2: Image showing how similar data points typically exist close to each other. (Harrison, 2018)

### Algorithm

1. Select number K of neighbors.
2. Calculate Euclidean distance of k number of neighbors.
3. Take k nearest neighbors as per calculate Euclidean distance.
4. Among these k neighbors, count number of data points in each category.
5. Assign new data points to the category for which number of neighbor is max.

### Suitability

* Advantages
  + Simple/easy to implement
  + Can be used for classification and regression
* Disadvantages
  + Algorithm gets significantly slower as the number of variables increase

Harrison, O. (2018) Machine Learning Basics with the K-Nearest Neighbors Algorithm. (Online) Available at: <https://towardsdatascience.com/machine-learning-basics-with-the-k-nearest-neighbors-algorithm-6a6e71d01761> (Accessed 20th October 2020)

## Incremental Anomaly Detection Model for Virtual Machines

Notes on the research article.

### IISOM

* Traditional SOM algorithm
  + Takes a long time to train a detection model.
  + Low accuracy
  + Low scalability
* SOMSA
  + SA = Simulated annealing
  + Optimization algorithm
* IISOM
  + Improved incremental SOM.
  + Weight Euclidean distance (WED).
  + WED speeds up training process.
  + Heuristic based initialization algorithm.
  + More suitable for anomaly detection.
  + Weights are estimated instead of random, quicker to train.
* Neighborhood-Based Training Domain Searching Algorithm
  + Reduces search space and lessens search time.
  + Decreases computation complexity
  + Shortens training time.

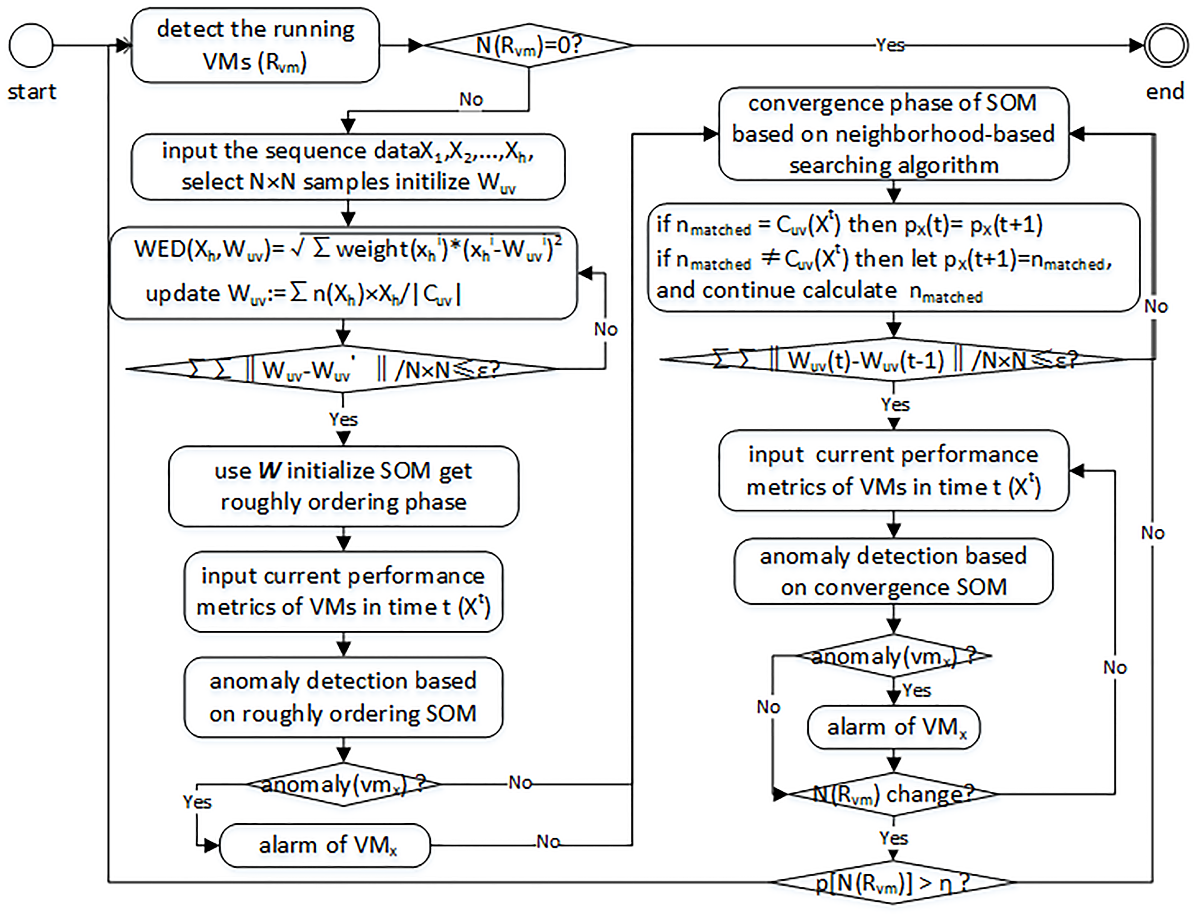


Figure 4: Flowchart of anomaly detection for virtual machine based on IISOM. (Zhang, 2017)

### Experiments

IISOM is fastest in experiments. Neighborhood based domain searching controls searching domain into small range for every input sample, detection time is faster.

IISOM is most accurate. It has the highest true positive rate and lowest false positive rate. Also highest true negative rate.

### Performance

IISOM algorithm performance good whether intrusion is large or not. Initial learning radius most efficient at 1.

Zhang, H., Chen, S., Liu, J., Zhou, Z., Wu, T. (2017) An incremental anomaly detection model for virtual machines. (Online) Available at: [https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0187488](https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0187488%20) (Accessed: 25 October 2021).

## 

## Time Series Anomaly Detection

### What is a Time Series Anomaly

An anomaly in time series data is a data point that does not follow the common pattern of the entire dataset. It is ‘significantly’ distinct from the rest of the data. Look below at figure 4 and see the anomalies marked by the red circles.

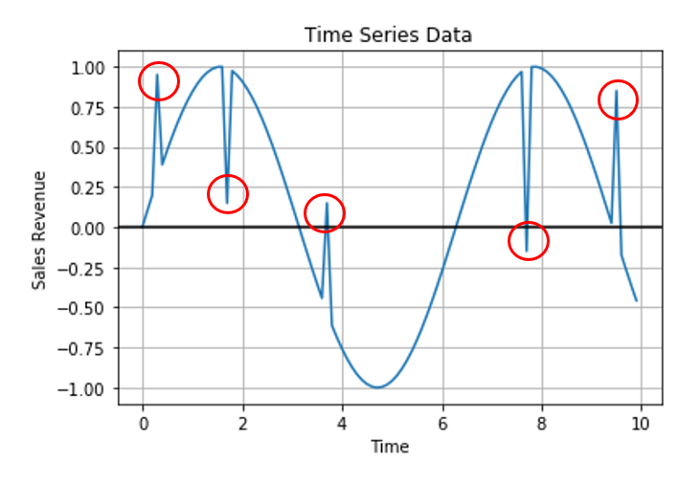


Figure 4: Time series plot highlighting anomaly data points. (Bhattacharya, 2020)

### Time Series Anomaly Detection

Three main ways:

* Predictive confidence
  + Predictive model gets an estimate of an overall common trend.
  + Future behaviour is forecast based on error rates.
  + We come up with a ‘confidence band’ and any data points that fall beyond it are seen as anomalies.
  + Loopholes in predictive models can give false positives and negatives.
* Statistical profiling
  + Statistical values like mean or median moving average of historical data to come up with a band of statistical values that make up bounds .
  + Anything falling beyond these bounds are seen as anomalies.
  + Can not detect local outliers.
* Clustering based
  + Unsupervised so very handy for time series data.
  + Hard to determine the number of clusters with time series data.
  + Density Based Spatial Clustering of Applications with Noise (DBSCAN) becomes the choice for determining clusters.
    - Easy to tune and fast in performance.
    - Anomaly points marked many times may be seen as the ‘new normal’ and not detected as an anomaly.

Bhattacharya, . (2020) Effective Approaches for Time Series Anomaly Detection. (Online) Available at: [Effective Approaches for Time Series Anomaly Detection | by Aditya Bhattacharya | Towards Data Science](https://towardsdatascience.com/effective-approaches-for-time-series-anomaly-detection-9485b40077f1)(Accessed: October 27th 2021)

## Numenta Anomaly Benchmark

NAB - Benchmark for evaluating anomaly detection in streaming data.

This link will allow me to enter my algorithm [NAB Entry Points · numenta/NAB Wiki · GitHub](https://github.com/numenta/NAB/wiki/NAB-Entry-Points)

### Features

* Scoring mechanism
  + Scoring mechanism to reward early detection, penalise late or false results, gives credit for online learning.
* Open source
* Contains real world data

### Cloning and using the NAB software

Running NAB using the provided data sets with the provided algorithm.

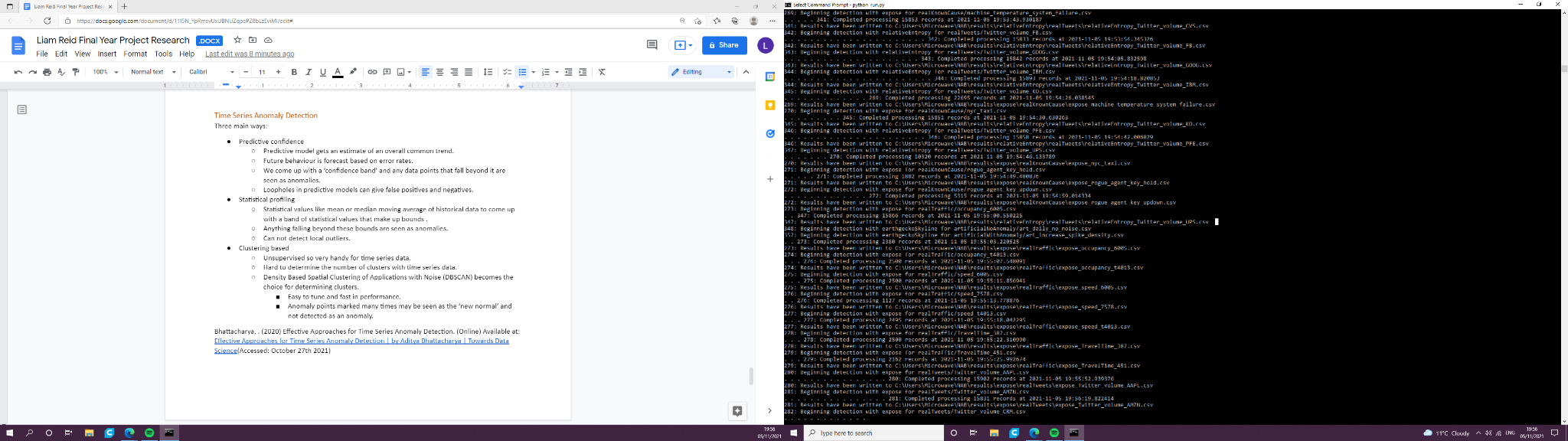


Figure 5. Screenshot of NAB running.

## 

## Ensemble Outlier Detection

Outlier Detection in Sensor Data using Ensemble Learning.

Ensemble learning obtains better prediction accuracy using a range of learning algorithms.

### Proposed Ensemble Method

The proposed “ensemble method based on unsupervised machine learning algorithms for predicting outlier(s) in a time series. The ensemble method detects the outliers by applying clustering prior to classification.”

Simplified in two steps:

* Step 1 (Batch Processing)
  + Create structures over the entire dataset.
  + Classical clustering applied to identify clusters.
  + Merging/Relocating of clusters.
  + Training a one class classifier over common data structures/clusters.
* Step 2 (Real-time processing)
  + Create structures over recent incoming data.
  + Outliers are detected by applying the classification model.

### Structure

Sequences of data are grouped. A structure pattern is produced for each group. The data retains its separate identity. Sliding windows are used so that data is not looked at as a sequence of arrays of data. A sliding window will have the size of ‘n’ observations. Each window can be considered a snapshot in time.

### Clustering

Classical clustering can identify patterns by applying clustering of rows and columns of a data matrix. Biclustering can identify local patterns by allowing simultaneous clustering of the objects and attributes. It can detect abnormalities in the earliest stages of machine fault. Can reduce dimensionality by picking only small sets of objects and attributes.

### Computational Steps for Proposed Algorithm

* Algorithm 1
  + Slide across readings with a sliding interval of one second. Construct data structures using a time-based window of size n.
    - Use classical clustering to assign each structure to a cluster.
    - Merge clusters of similar structures using cluster analysis or a voting technique for relocating clusters.
* Algorithm 2
  + Use normal structures to train a one class classifier.
  + If a newly arrived pattern does not belong to one of the normal structures it is classified as an outlier.

Iftikhar, N., Baatrup-Andersen, T., Nordbjerg, F., Jeppesen, K. (2020) Outlier Detection in Sensor Data using Ensemble Learning. (Online) Available at: <https://www.sciencedirect.com/science/article/pii/S1877050920320123> (Accessed: 6th November 2021)