## **Requirement**

## Analyse/observe the behavior of the managed devices/elements in the network (managed by netalytics) over time and detect (?) & predict anomalies, future behavior and raise alerts in time to allow the network operator to a take timely action.

## Analyze the SNMP messages/notifications that devices raise & predict/discover/derive future useful insights/behaviors using Machine Learning algorithms/techniques.

## **Netalytics Use cases**:

* A certain system (laptop, desktop or server) typically shows a certain kind of usage (CPU in the range of 75 to 100 on an average from 9 am to 9 pm) and much below that range during non-working hours (9 pm to 9 am). It would be deemed as an anomaly if during this non-working hours the system shows a CPU in the range of 75 to 100.
* A certain router has a packet drop rate (per hour() in a certain range on an average during busy traffic hours. It would be deemed as an anomaly if, during this busy traffic hours, packet drop rate (per hour) were to exceed the average.
* In a load-balancer setup, typically requests from clients were being distributed almost evenly across the servers in the load-balancer setup. It would be deemed as an anomaly if the requests are now being routed to a single server instead of the even distribution.
* Typically cameras for surveillance are supposed to be kept on, throughout the day or day and night, as per setting. An anomaly would be if the cameras were to be in an off state during the time-frame when they were supposed to be on (we can flag the anomaly if they were off for say more than 5 minutes continuously).

## **Apply ML to Netalytics use cases**

Collect Performance parameters of a system (CPU, Memory, Disk, Network) over 1 day, 10 days, 20 days, 30 days. Build a model to predict if a system is **up & running** or **not.** This is a classification problem. Logistics regression with regularization can be applied here.

| **Use case** | **Training examples** | **Output/ target** | **Prediction** |
| --- | --- | --- | --- |
| System - CPU: CPU pattern over a typical day - identify the working and non-working hours is   * defined * To be discovered   If the CPU pattern doesn’t fit this model, then flag an alert! | CPU during non peak hours | Anomaly? | If the CPU pattern is not following the desired pattern for a long duration, predict/recommend a change in load pattern |
| Router-Packet drop rate: Router packet drop rate  is   * defined * To be discovered | Packet drop rate | Anomaly? |  |
| Load Balancer - distribution:  LB distribution pattern is   * defined * To be discovered | LB distribution pattern | Anomaly? |  |
| Camera - downtime: Camera - downtime is defined | Camera - downtime | Anomaly? |  |

## **Anomaly detection**

Anomaly detection detects data points in data that does not fit well with the rest of the data.

the precision ( given model predicted an anomaly, how likely it is to be true) and recall (how much anomalies the model will catch

### Popular machine learning-based techniques

* Density-Based Anomaly Detection: Normal data points occur around a dense neighborhood and abnormalities are far away. k-NN (k Nearest Neighbour), local outlier factor (LOF)
* Clustering-Based Anomaly Detection: k-Means
* SVM, One-Class SVM

**Anomalies or outliers come in three types**

* **Point Anomalies.** If an individual data instance can be considered as anomalous with respect to the rest of the data (e.g. purchase with large transaction value)
* **Contextual Anomalies**, If a data instance is anomalous in a specific context, but not otherwise ( anomaly if occur at a certain time or a certain region. e.g. large spike at the middle of the night)
* **Collective Anomalies**. If a collection of related data instances is anomalous with respect to the entire dataset, but not individual values. They have two variations.
  + Events in unexpected order ( ordered. e.g. breaking rhythm in ECG)
  + Unexpected value combinations ( unordered. e.g. buying a large number of expensive items)

## **Monitoring computers in a data center**

Apply Anomaly detection algorithm

**Preventative maintenance system** We can also observe behaviors of other things such as machines. If you run a large machine in a factory and you want to predict if a machine is about to fail or have a fault, then just by observing the behavior of a machine, you can then record a dataset like this. There's a machine ID, there's a temperature of the machine, there's a pressure within the machine, and then did the machine fail or not. If your application is prevent the maintenance, say you want to figure out if a machine is about to fail, then you could for example, choose this as the input A and choose that as the output B to try to figure out if a machine is about to fail in which case you might do preventative maintenance on the machine.

### Identify features

X i = Features of machine i

X1 = memory use

X2 = number of disk accesses/sec

X3 = CPU load

X4 = network traffic

X5 = CPU load/network traffic

X5 = (CPU load)^2/network traffic

Choose features that are indicative of anomalous examples i.e. they fit a bell-curve. Choose features that might take on unusually large or small values in the event of an anomaly

### Data Collection and processing

Collect the above identified features.

Divide into training, CV and test sets

### Fit parameters on training set

Mu and sigma^2

### Use CV set to evaluate the algorithm

* Predict for CV set examples
* Evaluate algorithm using metrics
  + TP,FP, FN, TN
  + Precision/Recall
  + F1-score
* Choose a value for epsilon

**Testing**

## **Machine Learning**

**Machine Learning** is "the field of study that gives computers the ability to learn without being explicitly programmed”

**Supervised Learning**

we are given a data set and already know what our correct output should look like, having the idea that there is a relationship between the input and the output.

* **Regression**: predict results within a continuous output, meaning that we are trying to map input variables to some continuous function
* **Classification**: predict results in a discrete output. In other words, we are trying to map input variables into discrete categories.

**Unsupervised Learning:** approach problems with little or no idea what our results should look like. We can derive structure from data where we don't necessarily know the effect of the variables.

* Clustering:the data based on relationships among the variables in the data
* Non-clustering:

**Linear Regression Algorithm**

* **Gradient Descent**
* **Normal Equation**

**Logistic Regression (for Classification)**

* **Gradient Descent**

**Optimized Algorithms**

* **Conjugate gradient**
* **BFGS, L-BFGS**

**Problem of overfitting** occurs when the model fits the training set correctly but fails to predict correctly for new data. Solutions are to reduce the feature set or use **regularization**.

**Neural Networks**

**Prediction Errors**

**Bias**

**Variance**

**Analytics** is the use of data, [machine learning](https://www.educba.com/big-data-vs-machine-learning/), statistical analysis and mathematical or computer-based models to get improved insight and make better decisions.

**Data analytics** is to summarize or turn data into relevant information. Raw data is churned into clean data. Collect → Inspect → Clean → Transform → Reach conclusions. It can be used to find hidden patterns, unidentified correlations, customer preferences, market trends and other useful information that can help to make more informed decisions for businesses.

**Predictive analytics** uses historical data to predict future events. Clean data is used for predictive analytics.Inspect → Model the data → Train the model → Predict → Forecast

Artificial intelligence techniques are also used. It helps to answer questions such as “what will happen if demand goes down by 10% or if supplier prices go up by 5%?” “What do we presume to pay for fuel for next few months?” What will be the risk of losing money in a new business enterprise?”