There are [3 main approaches](https://towardsdatascience.com/outlier-detection-theory-visualizations-and-code-a4fd39de540c) to detect anomalies.

1. **unsupervised clustering** : Determination of outliers without previous data information (anomalies).
2. Isolation Forest
3. Local Outlier Factor
4. Robust Covariance
5. One-Class SVM
6. One Class SVM(SGD)
7. **Supervised machine learning** issue with usable labels for both regular and anomalous classes. As anomalies are expected to be very few, this could be considered as binary classification with imbalanced data.
8. Learning only the labels of the normal class makes it a **semi-supervised** approach

**Supervised vs. unsupervised anomaly detection**

Most teams have sample sets they use to train the machine learning algorithm to detect anomalous data. Whether or not the data in these sample sets is labeled determines which of the two main anomaly detection types a system is—supervised or unsupervised.

Supervised anomaly detection involves training a model with pre-labeled data. These datasets contain predefined normal data and clearly labeled examples of anomalies. While this may make an anomaly detection platform better at identifying expected abnormalities in data, it won’t account for abnormalities security teams don’t anticipate or haven’t seen before. Plus, many labeled datasets don’t contain enough outlier data to effectively train the algorithm.

Most organizations don’t have pre-labeled data, so they do unsupervised anomaly detection to define system baselines. Teams may provide the algorithm with unlabeled data sets and allow the system to determine what data qualifies as outliers, or they may allow the algorithm to form organically by observing a system at work. With each alert, these teams will teach the system what data points are normal and abnormal, which can be time and resource intensive.

# **Supervised vs Unsupervised Anomaly Detection**

The most common version of anomaly detection is using the unsupervised approach. In there, we train a machine-learning model to fit to the normal behavior using an unlabeled dataset. In that process, we make an **important assumption that the majority of the data in the training set are normal examples**. However, there can be some anomalous data points among them (a small proportion). Then any data point which differs significantly from the normal behavior will be flagged as an anomaly.

In supervised anomaly detection, a classifier will be trained using a dataset that has been labeled as ‘normal’ and ‘abnormal’. When a new data point comes, it will be a typical classification application.

There are pros and cons in both of these methods.

The supervised anomaly detection process requires a large number of positive and negative examples. Obtaining such a dataset will be very difficult since anomalous examples are rare. Even though you obtain such a dataset, you would only be able to model the abnormal patterns in the gathered dataset. However, there are many different types of anomalies in any domain and also future anomalies may look nothing like the examples seen so far. It will be very hard for any algorithm to learn from anomalous examples; what the anomalies look like. That is why the unsupervised approach is popular. **Capturing the normal behaviour is much easier than capturing the many different types of abnormalities**.

## **Anomaly Detection Methods**

Some of the most common anomaly detection techniques are:

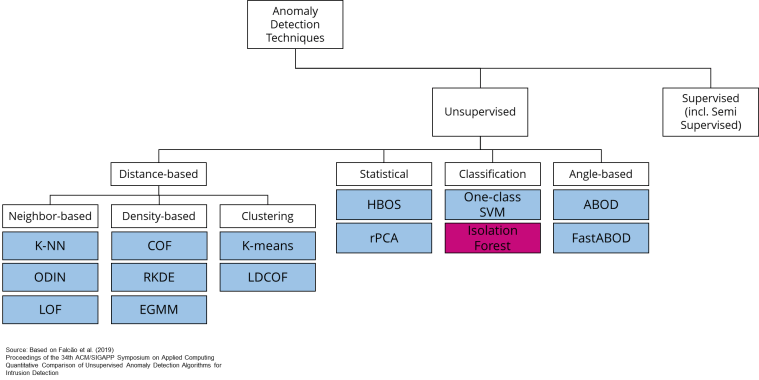
* **Density-based algorithms**: these anomaly detection approaches determine outliers based on whether a data point deviates beyond the normal—and subsequently denser—data population. [Isolation Forest](https://www.researchgate.net/publication/239761771_Isolation-Based_Anomaly_Detection) is a popular example that creates decision trees from a dataset by randomly selecting characteristics to detect similarities and isolate outliers.
* **Cluster-based algorithms**: these methods assign data points to clusters based on detected similarities. K-means is a popular example, where outliers are determined by how far they extend from a cluster group.
* **Bayesian-network algorithms**: these methods work by defining the probability that an event will occur based on the presence of contributing factors and detecting relationships with the same root cause.
* **Neural network algorithms: these methods use time-stamped data to forecast data patterns and identify outliers that don’t align with the historical data.**
* [**Long Short-Term Memory (LSTM)**](https://www.researchgate.net/publication/304782562_Long_Short_Term_Memory_Networks_for_Anomaly_Detection_in_Time_Series)**is a popular example that defines a sequence of events and detects outliers that do not follow the sequence.**

# **Popular Techniques**

* **Density-based techniques .**
  + **KNN,**
  + **Local Outlier Factor,**
  + **Isolation Forest, etc)**
* Support vector machines
  + **One Class SVM**
* **Angle Based:**
  + **ABOD**
* Statistical Based.
  + **HBOS**
* Cluster analysis based techniques
  + KMeans,
  + DBSCAN, etc)
* Bayesian Networks
* Neural networks, autoencoders, LSTM networks
* Hidden Markov models
* Fuzzy logic based outlier detection

There are several unsupervised anomaly detection techniques that can be implemented in Python. One way is to use the scikit-learn library which provides a set of machine learning tools for both novelty and outlier detection. These tools learn from the data in an unsupervised way using the fit method and new observations can then be sorted as inliers or outliers with a predict method.

There are also many other kinds of unsupervised methods for detecting anomalies like **Kernel Density Estimation, one-class Support Vector Machines, Isolation Forests, Self Organising Maps, C Means (Fuzzy C Means), Local Outlier Factor, K Means UNC, and Unsupervised Niche Clustering (UNC).**



## Advantages of Isolation Forest

Isolation Forest has a number of advantages compared to traditional distance and density-based models:

* Reduced computational times as anomalies are identified early and quick
* Easily scalable to high dimensional and large datasets
* Sub-samples the data to a degree which is not possible with other methods
* Works when irrelevant features are included

**How Does It Work:**

Univariate Anamoly Detection:

Call function get\_anamoly(data, column , evaluation\_model)

Return type is data frame with an additional columns containing anamoly label.

Input:

1. Data frame which contains the data to detect the anomaly
2. Columns name : for which yu need to find anamoly value
3. Technique Name: Predefine keys to be passed to evaluate anomaly detection. Pick any one from below table

Supported Techniques so far as below:

|  |  |
| --- | --- |
| Key | Technique Name |
| ABOD | Angle-based Outlier Detector (ABOD) |
| IF | Isolation Forest |
| CBLOF | Cluster-based Local Outlier Factor (CBLOF) |
| HBOS | Histogram-base Outlier Detection (HBOS) |
| KNN | K Nearest Neighbours (KNN) |
| OCSVM | One Class Support Vector Machine |

Output:

Dataframe. For a given technique dataframe will contain a columns for the given column name. column will have 2 values i.e.0,1.

Addition Column Name frmat : column + "\_" + model\_name + "\_Predictions"

0: Non Anamoly

1: Anamoly data.

get\_anamoly(df\_student\_anamoly,column , "IF")