****

**Outline for Blog Post**

**Isolation Forests for Anomaly Detection**

or

**Anomaly detection using Isolation Forest**

1. Introduction **- Shreya**

Ever wondered if we can detect a fraud transaction, malfunction of equipment, reduction or increase in demand of products and detect ecosystem disturbances using the data? Yes, you got it right, we can do all these by using Data Science. The most useful techniques in Data Science for detecting the unusual behavior in the data is called Anomaly detection. In today’s world, Anomaly Detection is being used in every industry.

The method of using Isolation Forests for anomaly detection in online fraud prevention field is still relatively new. It’s no secret that detecting fraud, phishing and malware has become more challenging as cybercriminals become more sophisticated. We should be using the most advanced tools and methods to prevent current and future fraud.

Advanced outlier detection methods such as Isolation Forests are imperative for companies looking to reduce fraud because this method detects anomalies purely based on the concept of isolation without employing any distance or density measure —fundamentally different from all existing methods.

What is data?

Before anything else, we need to know what data in Data science is. Data is nothing but information of any kind. It can be a text, audio, video, picture, transaction logs, server logs etc. Data Science as the name implies requires data as a basic requirement. They reveal the user/system/market behaviour which can be analyzed to predict the future outcomes by building data and machine learning pipelines. But do all the data in a dataset reveal the right information?

What is the difference between an anomaly and a nominal point?

**What is Anomaly and Anomaly Detection?**

**Example:** Let’s take a simple day to day scenario of a coffee shop to understand this better. Data from the security camera inside the cafeteria is analyzed all the time. Now from the metadata collected from the camera, we can capture the number of people inside the shop at a time. Suddenly the number of customers inside the shop during evening 4 to 5 pm increased from usual and was found to be growing. Even though this is a positive sign for the cafeteria, it is still an anomaly as this trend is different when compared to the rest of the day. This inference can be used to understand the change in trend and the coffee shop owner can leverage this inference to increase his supply during these hours to meet the demands. In the above scenario, the coffee shop customer count between 4 to 5 pm is an anomaly.

The word anomaly means “unusual”. An unexpected change within the data patterns, or an event that does not conform to the expected data pattern, is considered an anomaly. **Anomalies** are also referred to as outliers, novelties, noise, deviations and exceptions.

**Anomalies as noise:** Anomalies sometimes act as noise in the dataset. Noise is a random error (or a modification of original values) that is not interesting or desirable. In such cases, we need to get rid of them before proceeding with the analysis.

**BEWARE: Wrong Information/Data in terms of noise might lead to wrong prediction of results.**

**Anomalies as outliers:** These are the once which are significantly different from the remaining dataset. Detecting unusual data patterns not only helps in preventing hazardous situations but also track the change in the trend of whatever is analyzed. An example of an outlier could be the unusual identifiable patterns of data seen in MRI scans that help detect the symptoms of disease, fraud detection etc.

1. Why is anomaly detection important? **- Archana**

As mentioned, an anomaly can be defined as a deviation from what is normal/expected. A data that has abnormality and is not regular to its similar data points can be considered as an anomalous data. When collecting data, we may encounter a number of anomalies which may sometimes be essential to comprehend any unusual trends or fraudulent behaviour, depending upon the case we perceive. In the above coffee shop scenario, we are able to identify the unusual number of customers during the time window of 4 to 5 pm. Though this data is essential for the coffee shop to make future decisions, there are many other scenarios where detecting an anomaly can help identify theft or fraudulent activity.

Why anomaly detection and what are its benefits?

* With the advancements in machine learning and data science domains, there are a variety of novel ways to detect and remove anomalies from data that are **effortless and cheap**.
* Anomaly detection on data before performing any complex tasks, reduces the possibility of errors, thus significantly **increasing the quality of the results** obtained. Compromising anomaly detection may lead to complex issues which may consume a large amount of time and money.
* It has been observed that when anomaly detection is performed in real-time, i.e as data is being read, the data was processed much **faster** as compared to a dedicated anomaly detection phase.
* Anomaly detection facilitates **easy identification of issues** - thus increasing the up-time and considerably reducing the down-time of the system.

1. Use cases for anomaly detection - cleaning a dataset before feeding it into model, cybersecurity - fraud detection, spam filters, medical (tumors as anomalies) **- Archana**

There are some areas where detecting anomalies can help identify a swindle activity:

* In the **Banking Secto**r - anomalies in card/online transactions can help detect theft of card/credentials.
* In **Science** - the identification of an anomalous point in space may lead to novel discoveries and theories.
* In **Computer Science** - to detect abnormality in the network traffic patterns which helps in identifying unauthorized access in that network. It is also used to detect and classify Spam messages in emails.
* In **Medical Sector** - anomaly detection is very critical in the areas of radiation oncology where the chances of errors are very rare, but when occurs can be fatal. It is used to identify tumors and trace logs of patients over a period of time.

**Intro to Isolation Forest:**

There are many ways in which we can determine if the given data point is an anomaly or not. Though most anomalies can be detected by **cluster analysis** and detecting the micro clusters formed by them, the most broadly used anomaly detection techniques are: Supervised, Semi-supervised and Unsupervised anomaly detection. Here in this blog, we concentrate on **Isolation Forest** - An Unsupervised anomaly detection algorithm, that makes use of unlabeled test dataset with the assumption that most of the instances in the dataset are normal.

Why isolation forest?

1. Isolation forest helps identifying outliers in a multidimensional space.
2. Explicitly identifies and categorizes data as outliers instead of profiling that data.
3. Makes use of Random partitioning which that generates a path from the root in turn easier to discover anomalies.
4. Low linear time complexity and a small memory requirement - eliminates major computational cost of distance calculation in all distance-based methods and density-based methods.
5. Capacity to scale up to handle extremely large data size.

**Applications of Isolation forest:**

* Used in Automation detection that identify abusive contents as Spam or Fake Engagement.
* Identify and rank the severity of fake accounts used for reviews.
* Detect network traffic and identifying intrusion in the system of an organisation.
* Detect automatic change in the values of a specific aspect/ condition over a period of time.
* Anomaly detection in time-series data.
* Credit card scam detection.
* Detect anomalies in data centre structure/schema.

1. Why are the isolation forest and EIF method beneficial in determining anomalies? **- Shreya**

Advantages of Isolation Forests:

* There exist supervised techniques such as SVM, Decision trees, KNN, logistic regression and others which can detect anomalies but all these methods require labelled dataset to form a classifier but finding labelled data for all cases is not easy. By using Isolation forests as an unsupervised learning algorithm we can do the anomaly detection on an unlabelled dataset.
* With this method, isolating anomaly observations is easier as it requires only a few conditions to separate such cases from the normal observations when compared to other methods which uses basic distance and density measures.
* The algorithm used in this method has a low linear time complexity and small memory requirement.
* Works well on multi-dimensional dataset

Why Extended Isolation Forest?

* Decision boundaries of the Isolation Forest are either vertical or horizontal due to which branches tend to cluster where the majority of the points are located. But as the lines can only be parallel to the axes, there are regions that contain many branch cuts and only a few or single observations, which results in improper anomaly scores for some of the observations.
* Resolves issues with assignment of anomaly score to given data points by identifying a random slope for hyperplanes which is used for splitting the data.

**Good for high dimension data**

**Quick**

**Built off of decision trees - easy to understand**

[**https://cs.nju.edu.cn/zhouzh/zhouzh.files/publication/icdm08b.pdf**](https://cs.nju.edu.cn/zhouzh/zhouzh.files/publication/icdm08b.pdf) **- This is the original 2008 paper for the algorithm**

[**https://arxiv.org/pdf/1811.02141.pdf**](https://arxiv.org/pdf/1811.02141.pdf) **- This is the introduction of the extended method**

1. Go over the actual algorithm **- Asha**

Isolation Forest builds an ensemble of “Isolation Trees” (iTrees) for the dataset, and isolates each point in the given dataset, then anomalies are those instances which have short average path length on the iTree.

Procedure:

1: Sampling for a training model.

2: Choose randomly two attributes.

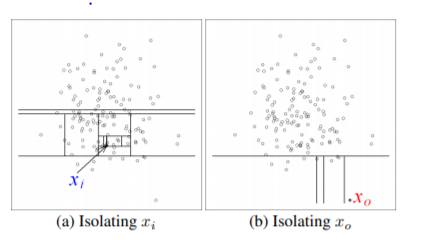
3: A split value between the maximum and minimum values of the selected attribute.

4: Partitioning of instances are repeated recursively until instances are isolated.

In the image below the author chooses two points.

Xi – Normal point and requires more partitions to be isolated.

X0 – Anomaly point and it requires less partitions to be isolated.



Here, the path length of xi is greater than the path length of x0.

The number of partitions required to isolate a point is equivalent to the path length from the root node to a terminating node.

In a data-induced random tree, partitioning of instances are repeated recursively until all instances are isolated. This random partitioning or isolation numbers produces shorter paths for anomalies.

How to calculate path length and anomaly scores:

Importance of anomaly scores :

The one way to detect anomalies is to sort data points according to their path lengths or anomaly scores so an anomaly score is required for any anomaly detection method and decision making.

Path Length h(x) of a point x is measured by the number of edges x traverses an iTree from the root node until the traversal is terminated at an external node. The difficulty in deriving such a score from h(x) is that while the maximum possible height of iTree grows in the order of n, the average height grows in the order of log n [7]. Normalization of h(x) by any of the above terms is either not bounded or cannot be directly compared. Since iTree have an equivalent structure to Binary Search Tree, the estimation of average h(x) for external node terminations is the same as the unsuccessful search in BST.

Provided a dataset of n instances we can calculate the average path length of unsuccessful search in BST as:

c(n) = 2H(n − 1) − (2(n − 1)/n) (1)

where H(i) is the harmonic number and it can be estimated by ln(i) + 0.5772156649 (Euler’s constant). As c(n) is the average of h(x) given n, we use it to normalise h(x). The anomaly score s of an instance x is calculated as:

s(x,n) = 2−E(h(x))c(n)  (2)

where E(h(x)) is the average of h(x) from a collection of isolation trees.

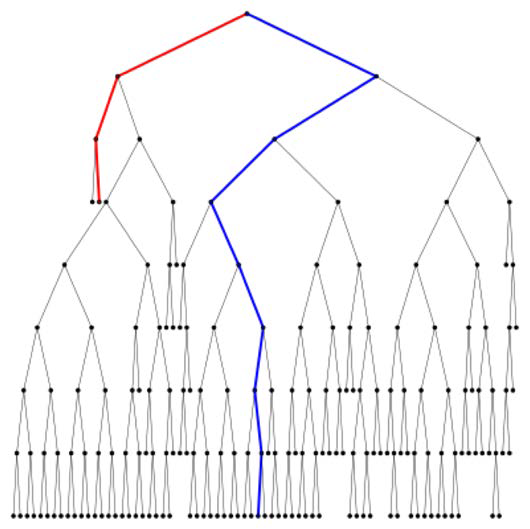
In Equation (2):

when E(h(x)) → c(n), s → 0.5; •

when E(h(x)) → 0, s → 1;

when E(h(x)) → n − 1, s → 0.

If s is very close to 1 indicates anomalies. Score is much smaller than 0.5 indicate they are normal observations. if all the instances return s = 0.5, then the entire sample does not have



In the above figure, we observe the trajectory of the data points after training. Since the height of the point along the red line is 3, which is the minimum, it is considered as an anomalous point. In contrast, the blue path shows the path of a normal point.

Isolation Forest: Among the different anomaly detection algorithms, Isolation Forest is one with unique capabilities.

Model free,Computationally efficient,Readily applicable to parallelization,Readily applicable to high dimensional data,Inconsistent scoring observed in score maps.

**Why was the extended isolation forest method introduced?**

As shown in the figures from Hariri et al., a situation arises with the traditional IF method where horizontal and vertical bands span from clusters of data points. That is, anomalies which occur within these bands, but are otherwise far away from nominal points, may be incorrectly classified as nominal. These bands are due to multiple straight lines drawn to separate the points which are parallel to the axes.

\*\*Show Figures\*\*

Such arises the necessity of the extended isolation forest (EIF) method in which hyperplanes (think separators between data points, but with dimensions up to one less than that of a dataset) with random slopes are used to slice the data, thus mitigating the issue of single dimension bands of nominality. A 2D hyperplane separating 3D data is shown in the plot below. It is a little more difficult to imagine or visualize hyperplanes of higher dimensions.

\*\*Example of 2D hyperplane\*\*

**Example**

In 2016, a traditional IF function was introduced in the scikit-learn package (link). EIF functions were developed shortly thereafter, with the ability to change the hyperplane dimension in hopes of eradicating the straight line issue.

Below is an implementation of the EIF algorithm using the package from here (<https://github.com/sahandha/eif>). The original implementation of EIF was generally reported as quite slow; however, the newest version released in November 2019 is built with C++ using Cython and is now a lot faster. The speed is even reported to be on par with scikit-learn’s original IF function when used on certain datasets.

We tested out EIF on a dataset of breast cancer masses with labeled anomalies (benign/non-cancerous mass = nominal, malignant/cancerous = anomaly) and 9 features including mass radius, smoothness and concavity (dataset reference). If you are looking for more datasets to practice anomaly detection on, check out this website (ODDS reference)!

The EIF package is very easy to use! All that you need to use it are Numpy and Cython.

First, we will create a dataframe from the csv file. The EIF function requires a numpy array, but the dataframe format will make plotting easier later.

\*\*\*Code\*\*\*

We now implement the EIF class. The ‘ExtensionLevel’ parameter indicates the dimension of the hyperplane to use, as discussed above. We can use anything from 0 to 8 (number of dataset features minus 1). A level of 0 would be just like using the traditional IF function. We will use the maximum value of 8 for this example.

\*\*\*Code\*\*\*

The plot below shows the anomaly score on the x-axis and the real classification from the dataset on the y-axis (0 for benign/nominal, 1 for malignant/anomaly). Again, a higher score indicates an increased likelihood that the point is an anomaly. As we can see, the EIF algorithm generally assigned higher scores to the data points which were labeled as anomalies, which is what we were hoping for! The results are by no means perfectly concrete; there is overlap between the scores of the nominal points and anomalies, and the scores of the anomaly points are not all that high, considering that the range is from 0 to 1.

\*\*\*Plot\*\*\*

1. Issues with EIF - bad luck on larger datasets?
2. Other methods for anomaly detection?? - If we have room :)

Isolation forests are one way to determine anomalies. There are several other methods including...

References:

Archana’s reference:

<https://blogs.oracle.com/datascience/introduction-to-anomaly-detection>

<https://en.wikipedia.org/wiki/Anomaly_detection>

<https://engineering.linkedin.com/blog/2019/isolation-forest>

<https://en.wikipedia.org/wiki/Isolation_forest>

<https://towardsdatascience.com/anomaly-detection-with-isolation-forest-visualization-23cd75c281e2>

Shreya’s reference:

<https://pdfs.semanticscholar.org/4013/bb59a49ceb4679cdf407e0b53c274026868e.pdf>

<https://quantdare.com/isolation-forest-algorithm/>

<https://blog.easysol.net/using-isolation-forests-anamoly-detection/>

<https://arxiv.org/pdf/1811.02141.pdf>

<https://cs.nju.edu.cn/zhouzh/zhouzh.files/publication/icdm08b.pdf>

<https://arxiv.org/pdf/1811.02141.pdf>

<https://github.com/sahandha/eif>

SKLearn

<https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+%28Original%29>

<https://github.com/criddel/projects/blob/master/bcisolationforest.ipynb>

Carmen’s code for last part ^^

<https://cs.nju.edu.cn/zhouzh/zhouzh.files/publication/icdm08b.pdf> - calculation of score

Image for tree:

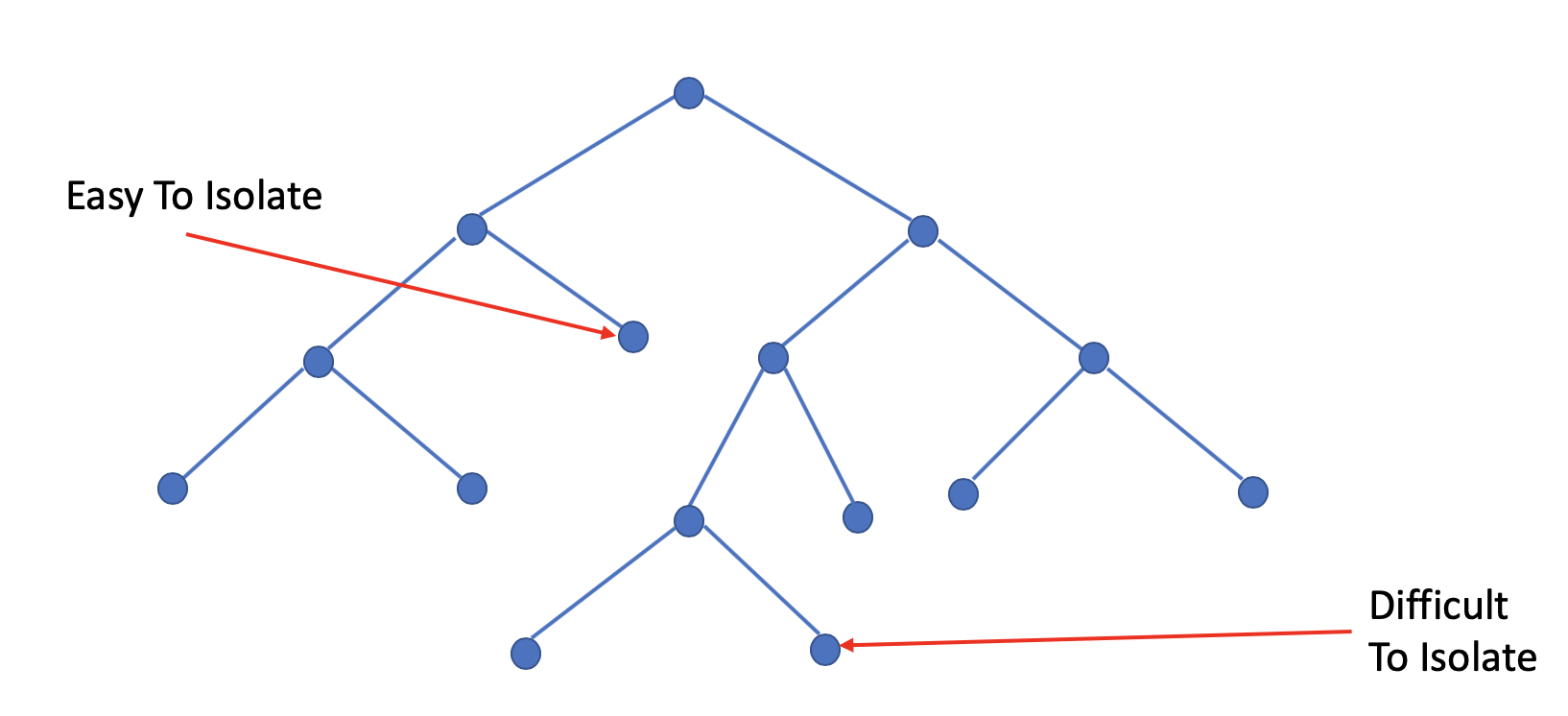


Image depicting an outlier:



