Exploratory Data Analysis

Preprocessing

The data obtained from the Kaggle dataset for New York City is preprocessed into the format required for performing exploratory data analysis. A preprocessing program was specifically written to convert to the format required. This format was specifically chosen to easily understand the data and for applying different processing methods for data analysis to get the best insights out of it.
The format is as shown for certain rows as below :

	datetime	Humidity	Pressure	Temperature	Wind_Direction	Wind_Speed
0	01-10-2012 13:00	58	1012	288.220000	260	7
1	01-10-2012 14:00	57	1012	288.247676	260	7
2	01-10-2012 15:00	57	1012	288.326940	260	7
3	01-10-2012 16:00	57	1012	288.406203	260	7
4	01-10-2012 17:00	57	1012	288.485467	261	6

Figure. Required Format for Data analysis

Cleaning data

The first step after the above step is to clean the data and deal with the missing data (Nan values). Missing data can reduce the statistical power of a study and can produce biased estimates, leading to invalid conclusions. To deal with this problem, two methods were used:

- Removing the rows with missing data
- Usage of Imputer function

The first method is effective if the data is very large. But the second method was implemented by imputing (replacing) the missing values with the mean of all the values since it removes the possibility of losing important data.

Time Series Data

From above it can be observed that for the data analysis, the given dataset is a Time Series Data as the data is a series of data points indexed in time order. A time series is a sequence taken at successive equally spaced points in time. Thus it is a sequence of discrete-time data.

The plot of all features as a time series is given below. The X-axis is usually time spread over between two values and Y-axis is the feature we are plotting. Refer to the image below.

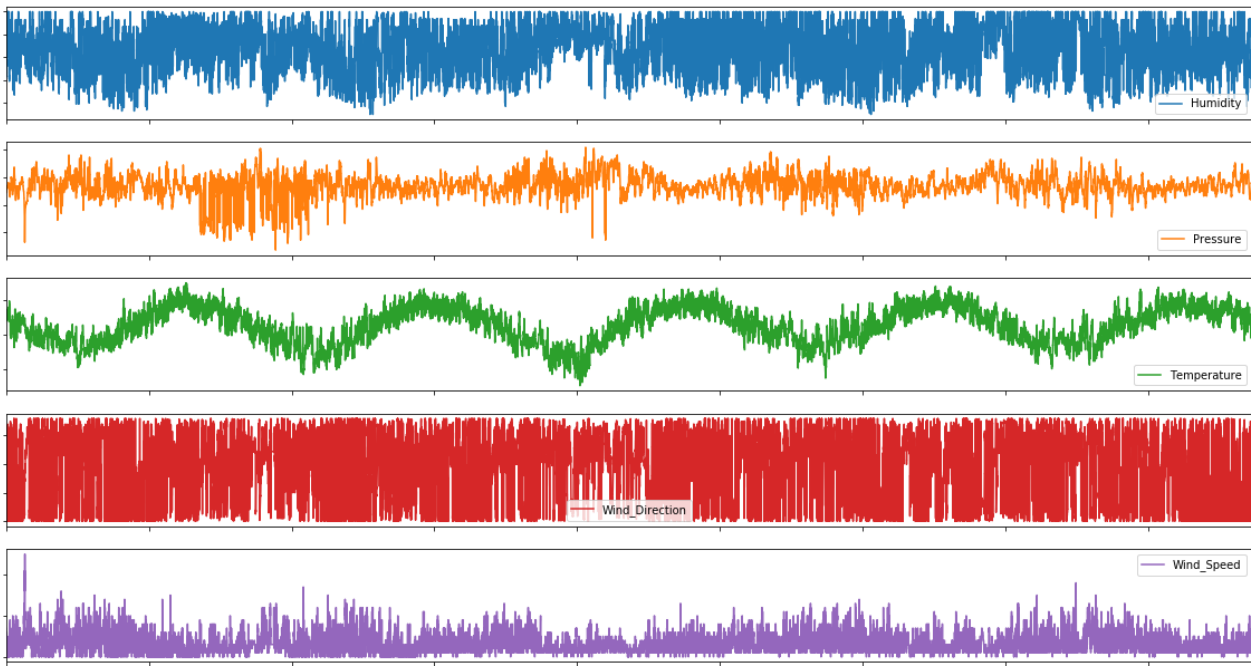


Figure. Time Series plots of all features

Correlation matrix

A correlation matrix is a table showing correlation coefficients between variables. Each cell in the table shows the correlation between the two variables. A correlation matrix is used to summarize data, as an input into a more advanced analysis, and as a diagnostic for advanced analyses. This is a very important preprocessing step. The figure below shows the correlation map for the dataset. As observed the data does not require Dimensionality reduction.

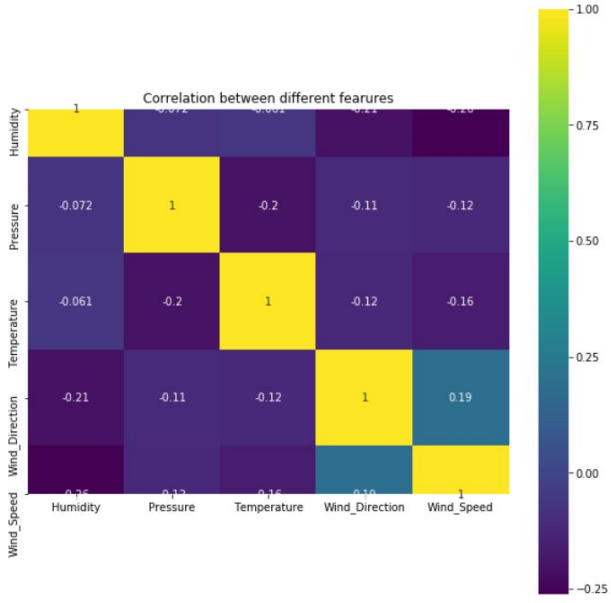


Figure. Correlation matrix of features

Plotting

Various plots were plotted using the data for understanding how the data was spread out.

1.Histogram Plots

Histograms are a type of bar plot for numeric data that group the data into bins.It is a type of bar plot where X-axis represents the bin ranges while Y-axis gives information about frequency.

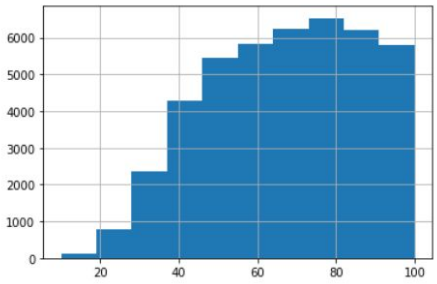


Figure. Humidity Histogram

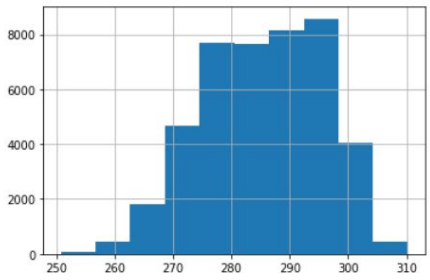
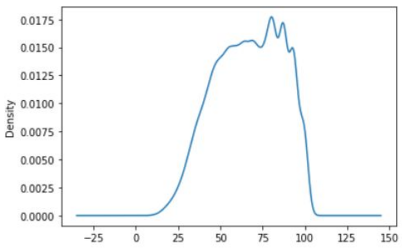


Figure. Temperature Histogram

2.Density Plots

A density plot is a representation of the distribution of a numeric variable. It uses a kernel density estimate to show the probability density function of the variable.It is a smoothed version of the histogram and is used in the same concept. Density plots are used to study the distribution of one or a few variables.



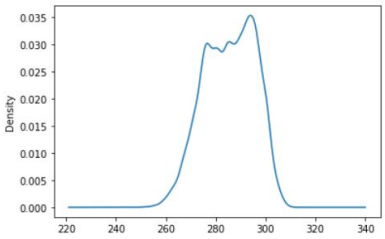


Figure. Density plots of Humidity and Temperature

3.Lag Plots

This type of plot checks whether a data set or time series is random or not. This plot is very useful in identifying the outliers. The points that lie distinguishably far from the population of data (i.e , where the data density is large on the plot) , are the outliers in the data.

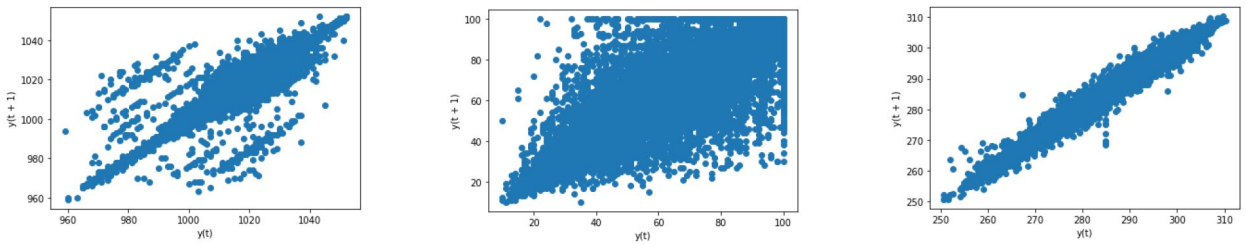


Figure . Lag Plots of Pressure , Humidity and Temperature in order

Anomaly Detection : What is it ?

Anomaly detection is a technique used to identify unusual patterns that do not conform to expected behavior, called outliers. It is used to find data points that significantly differ from a majority of data . Large, real-world datasets may have very complicated patterns that are difficult to detect by just looking at the data. That's why the study of anomaly detection is an extremely important application of Machine Learning.

As labeling the data or having just clean data is often hard and time consuming, at times for practitioners, the unsupervised approach is rather practical. This is the main reason why most anomaly detection algorithms are unsupervised in nature.

Since we are dealing with Time series data , the figure given below shows that a sudden hike is an anomaly. This is the best example of an anomaly.

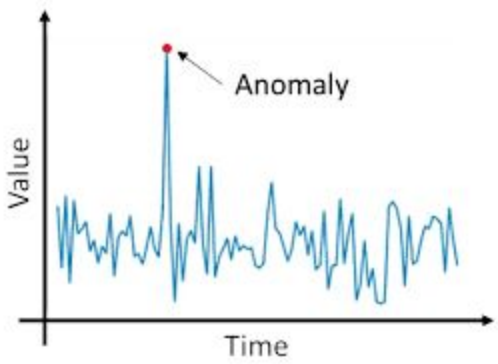


Figure . Anomaly detection in Time series

In time series data , the anomalies fall under three main categories :

1. Global Outliers : Easily differentiable outliers that exist far outside the entirety of a data set.
2. Contextual Outliers : Also called conditional outliers, these anomalies have values that significantly deviate from the other data points that exist in the same context. An anomaly in the context of one dataset may not be an anomaly in another. These outliers are common in time series data because those datasets are records of specific quantities in a given period.
3. Collective Outliers : When a subset of data points within a set is anomalous to the entire dataset, those values are called collective outliers.

To detect the outliers there are many algorithms. These algorithms fall into three categories based on the approach they follow.

1. Density Based
2. Distance Based
3. Isolation Based

Based on the way the math at core of the each algorithm is formulated , it can be classified into the above three types.

There are many use cases of isolation forest in fields of Healthcare, Business and Intrusion Detection.

But there are quite a few challenges in the field of anomaly detection. It is a tedious job to perform.It also requires domain knowledge and—even more difficult to access—foresight.

In this project we have used four anomaly detection methods:

1. Isolation Forest
2. DBSCAN
3. One Class SVM
4. Local Outlier Factor

Isolation Forest

Isolation forest is a machine learning algorithm for anomaly detection. It is an unsupervised learning algorithm that identifies anomaly by isolating outliers in the data. This algorithm is unique in a sense that it rather explicitly isolates anomalies instead of profiling normal instances.

The term isolation means ‘separating an instance from the rest of the instances’. Since anomalies are ‘few and different’ and therefore they are more susceptible to isolation. Thus the core of the algorithm is to “isolate” anomalies by creating decision trees over random attributes.

Issues with Usual Approaches

There are a few drawbacks with the traditional and usual approaches (density and distance) .

1. Too many false alarms: The usual methods are not optimized to detect anomalies, instead, they are optimized to find normal instances, because of which the result of anomaly detection either contains too many false positives or might detect too few anomalies.These are called swamping and masking effects.
2. Many existing methods are constrained to low dimensional data and small data size because of the legacy of their original algorithms.
3. There are violations in the assumption for density and distance measure anomaly detection, e.g., high density and the short distance do not always imply normal instances; likewise, low density and long distance do not always imply anomalies.
4. Most methods are slow and inefficient.

Isolation forests solve the above problems.

Understanding Decision Trees : Core Principle of Isolation Forests

Isolation Forest uses an ensemble of Isolation Trees for the given data points to isolate anomalies. The core data structure used is a decision tree. The decision tree is constructed by splitting the sub-sample points/instances over a split value of a randomly selected attribute such that the instances whose corresponding attribute value is smaller than the split value goes left and the others go right, and the process is continued recursively until the tree is fully constructed. The split value is selected at random between the minimum and maximum values of the selected attribute.

There are two types of nodes in a decision tree : Internal and External node. Internal nodes are parents that have two child sub trees. External nodes are those that cannot be split anymore.

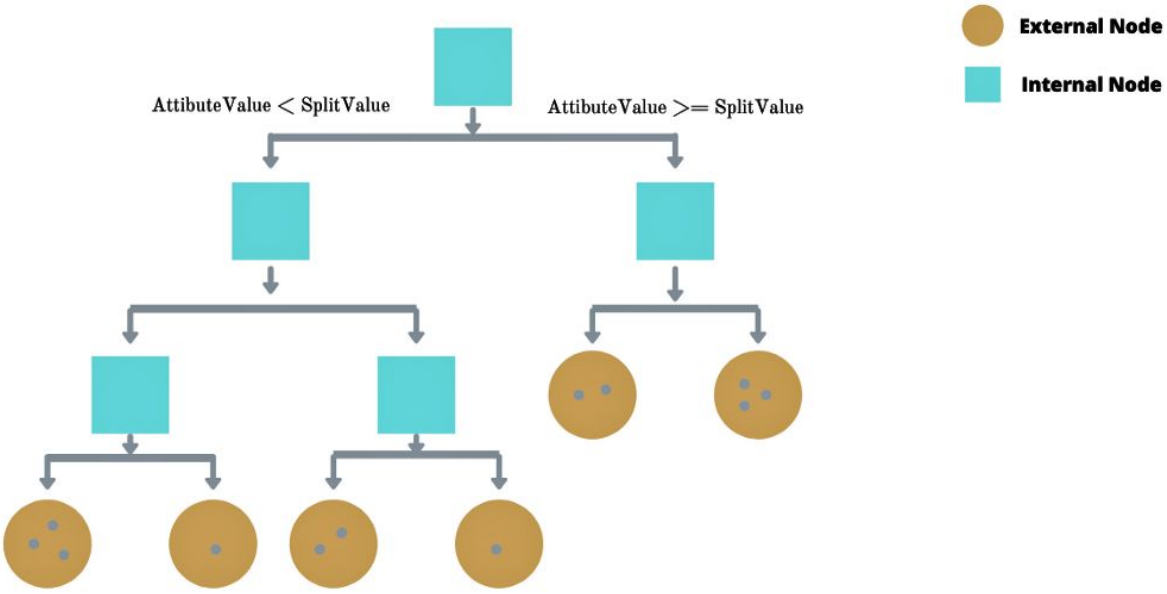


Figure . Decision Tree

The above figure is a diagramatic representation of the Decision Tree. The figure below is the mathematical algorithm behind it.

Inputs: X - input data, e - current tree height, l - height limit
Output: an iTree

if $e \geq l$ or $|X| \leq 1$ **then**
  return $exNode\{Size \leftarrow |X|\}$
else
  let Q be a list of attributes in X
  randomly select an attribute $q \in Q$
  randomly select a split point p from max and min values of attribute q in X
  $X_l \leftarrow filter(X, q < p)$
  $X_r \leftarrow filter(X, q \geq p)$
  return $inNode\{Left \leftarrow iTree(X_l, e+1, l),$
    $Right \leftarrow iTree(X_r, e+1, l),$
    $SplitAtt \leftarrow q,$
    $SplitValue \leftarrow p\}$
end if

Figure. Math behind Decision Tree

Since anomalies are susceptible to isolation and have a tendency to reside closer to the root of the decision tree, the decision tree is constructed till it reaches a certain height and not

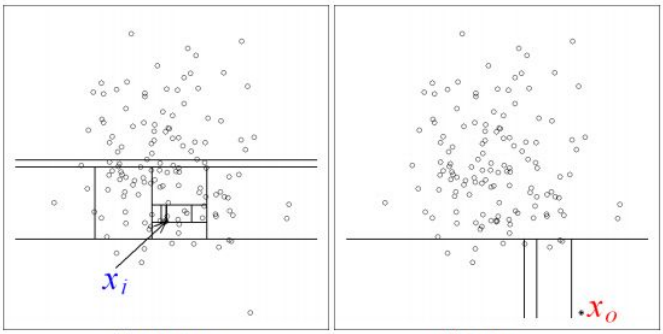
split points further. This is due to the fact that the distance to reach an anomaly is much less compared to reach a normal data point.

The process of tree construction is repeated multiple times and each time a random sub-sample is taken to construct the tree. There are no strict rules to determine the number of iterations, but in general, the more the better.

Steps in working of Isolation Forest

The following steps show the working of Isolation Forest:

1. Random and recursive partition of data is carried out, which is represented as a tree (random forest). This is the training stage where the user defines the parameters of the subsample and the number of trees.
2. The end of the tree is reached once the recursive partition of data is finished. It is expected that the distance taken to reach the outlier is far less than that for the normal data (see the below figure).



(a) Isolating x_i (b) Isolating x_o

3. The distance of the path is averaged and normalized to calculate the anomaly score. The path length is calculated by the formula given below.

Inputs : x - an instance, T - an iTree, e - current path length; to be initialized to zero when first called
Output: path length of x

1: **if** T is an external node **then**
2: return $e + c(T.size)$ {$c(.)$ is defined in Equation 1}
3: **end if**
4: $a \leftarrow T.splitAtt$
5: **if** $x_a < T.splitValue$ **then**
6: return $PathLength(x, T.left, e + 1)$
7: **else** $\{x_a \geq T.splitValue\}$
8: return $PathLength(x, T.right, e + 1)$
9: **end if**

Figure.Formula for path length

The anomaly score is mathematically calculated as below.

$$Anomaly\ Score\ (S) = 2^{\frac{-E(h(k,m,N)}{c(n)}}$$

$$,where\ c(n) = 2(\ln(n-1) + 0.5772156649) - 2\left(\frac{n-1}{n}\right)$$

$$,where\ n\ is\ a\ number\ of\ data\ points\ in\ a\ chosen\ sample$$

$$,where\ E(h(k,m,N)) = \frac{\sum_{i=1}^{N} \begin{cases} if\ k == 1, \sum_{j=1}^{M} 1 \\ else,\ \sum_{j=1}^{M} 1 + c(k) \end{cases}}{N}$$

$$,where\ N\ is\ a\ total\ number\ of\ trees$$

$$,where\ M\ is\ a\ total\ number\ of\ binary\ splits$$

$$,where\ k\ is\ a\ total\ number\ of\ data\ points\ in\ the\ final\ node\ (exit\ node)$$

Figure. Formula for anomaly score

4. The judgment (final decision) of the outlier is carried out on the basis of the score. Mathematically the value received for an outlier when the anomaly score is calculated shoulde be close to 1.

Exploitation of Sub Sampling

The Isolation Forest algorithm works well when the trees are created, not from the entire dataset, but from a sub-sampled data set. This is very different from almost all other techniques where they thrive on data and demands more of it for greater accuracy.

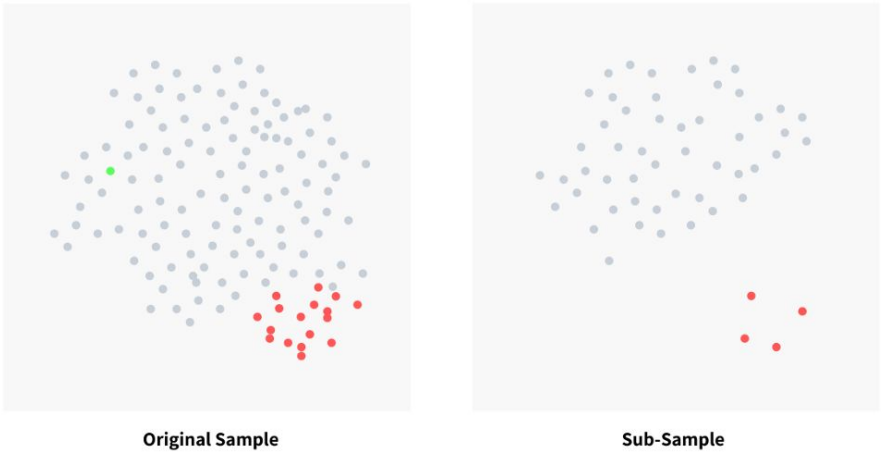


Figure. Sub Sampling

The image above shows how sub-sampling actually makes a clear separation between normal points and anomalies. In the original dataset, the normal points and very close to anomalies making detection tougher and inaccurate (with a lot of false negatives). Because of sub-sampling,a clear separation of anomalies and normal instances is seen. This makes the entire process of anomaly detection efficient and accurate. This way Isolation forest exploits the

sub sampling to achieve low linear time complexity and a small memory requirement and to deal with the effect of swamping and masking effects.

Implementation using Scikit Learn

Using the scikit learn python library , from the ensemble sub library , Isolation Forest is imported. The function is defined with the default values of 100 estimators and 256 samples. These values are taken as they are considered to give the best results. Given below is the implementation of the main function.

```
model=IsolationForest(n_estimators=100, max_samples='auto',
contamination=float(0.1),max_features=1.0)
```

The features were taken all together and separately one at a time. Given below are the visualizations of the results of the program in detecting anomalies in Temperature , Pressure and Humidity over a short period of time. The Blue line represents how the data is spread over the time period and the red dots represent the points of anomaly.

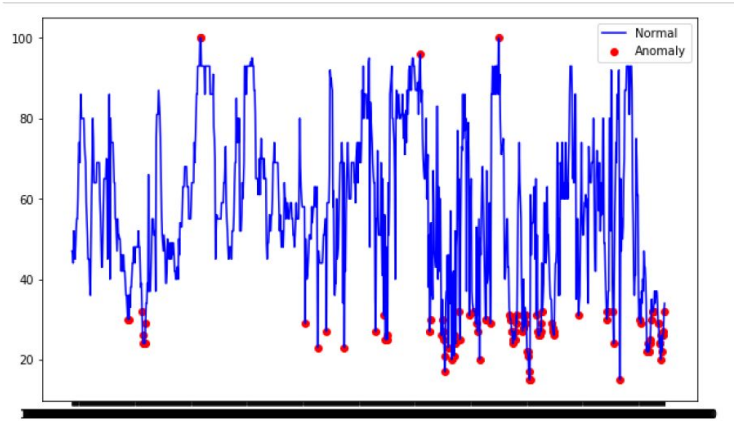


Figure. Anomalies detected in Humidity

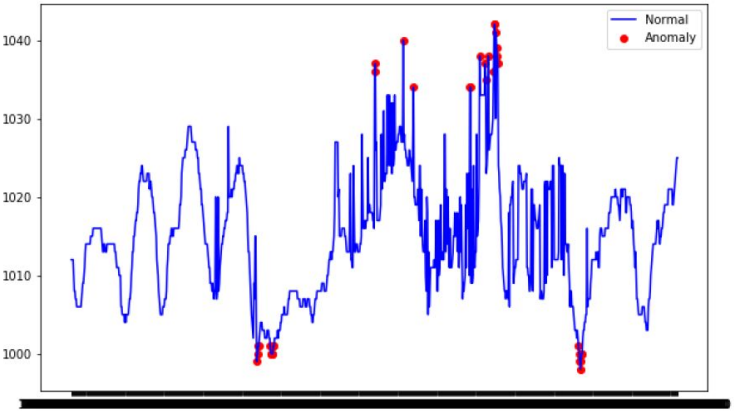


Figure . Anomalies detected in Pressure

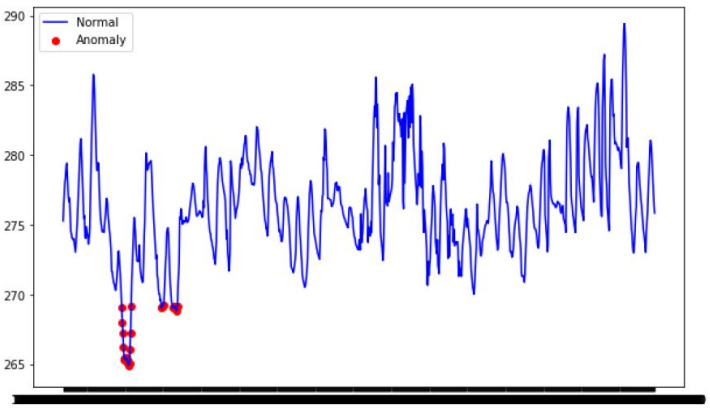


Figure. Anomalies detected in Temperature

Results

The Isolation forest found the following number of anomalies as shown below.

Data	Number of Outliers
All Data together	4361
Temperature	4353
Humidity	4334
Pressure	4084
Wind Speed	3621

Figure. Results of Isolation Forest

Conclusions on Isolation Forest

Some of the pros of this method are:

1. no need of scaling the values in the feature space.
2. Effective method when value distributions can not be assumed
3. There are few parameters, this makes this method fairly robust and easy to optimize

Some of the disadvantages of this method:

1. Visualization is complex
2. If not correctly optimized , it will become computationally expensive.

For further information about Isolation forest , it is recommended to refer the paper :
Liu, Fei Tony, Kai Ming Ting, and Zhi-Hua Zhou. "Isolation forest." 2008 Eighth IEEE International Conference on Data Mining. IEEE, 2008.