[[1]](#footnote-0)

Shodan’s SCADA Data Analysis(December 2016)

Authors : Rumela Saraswati, Chitra S. Kumar, Keerthi Bala Sundram, Madhurya Alur, Dhaval Adhav

*Abstract*— **Supervisory control and data acquisition (SCADA) is a**[**control system**](https://en.wikipedia.org/wiki/Control_system)**architecture that uses computers, networked data communications and**[**graphical user interfaces**](https://en.wikipedia.org/wiki/Graphical_user_interfaces)**for high-level process supervisory management. The SCADA concept was developed as a universal means of remote access to a variety of local control modules, which could be from different manufacturers allowing access through standard automation**[**protocols**](https://en.wikipedia.org/wiki/List_of_automation_protocols)**. Given the large number and complexity of production activities that influence yield in these industries, manufacturers need a more granular approach to diagnose and optimize production. Advanced analytics provides just such an approach.**

**In this project, we have used the concept of SCADA and applied advanced analytics to perform clustering based on the geo-coordinates of a location. By comparing the outputs of various clustering algorithms namely KMeans, Agglomerative Clustering and KNN we were able to achieve accurate results to identify relationships among data and then optimize the factors that prove to have the greatest effect on the live streamed dataset.**

***Index Terms*—Shodan, SCADA Data, Clustering, KMeans Algorithm, Agglomerative Clustering, KNN Algorithm**

# **INTRODUCTION**

**T**he term “SCADA” was coined in the early 1970s, and the rise of microprocessors and programmable logic controllers (PLCs) during that decade gave enterprises a greater ability to monitor and control automated processes than ever before. Modern SCADA systems allow real-time data from the plant floor to be accessed from anywhere in the world. This access to real-time information allows manufacturing units to make data-driven decisions about how to improve their processes. Without SCADA software, it would be extremely difficult to gather sufficient data for consistently well-informed decisions.

We have used Shodan in our project to gathering live streamed data about Siemens p7 device. Shodan is a search engine for Internet-connected devices. Unlike traditional websites, Shodan allows users to perform various activities like finding computer that run a specific software, know which version of a particular is most popular, check for security of the connected device.

The basic unit of data that Shodan gathers is the banner. The banner is textual information that describes a service on a device. We have used the banner CVE-2015-0204 though the REST and streaming API of Shodan to access the header of the data also known as the metadata. This was because the actual data was encrypted. The banner has several properties like hostnames, domains, location.longitude, location.latitude, ip1. We have performed clustering based on location.longitude, location.latitude properties. The clustering provided visual effects on which further decisions could be made.

The project was divided into 2 parts. The first part focused on live streaming of the Siemens data using Shodan. The second part focused on performing analysis on the live streamed data.

We have implemented the KMeans, KNN and Agglomerative clustering and achieved different accuracies for each approach. KNN gave an accuracy of 40%, Agglomerative clustering gave an accuracy of 80% while KMeans performed best and provided an accuracy of 95%. This was calculated based on the number of records that were plotted vs the total number of records that were streamed.

# **MOTIVATION**

The SCADA system is ensconced at a crucial layer of the enterprise hierarchy. Intelligent innovation in this space of industrial automation can provide remarkable benefits to the end-user community in terms of increased efficiency and improved profitability.

If we make can perform proper analyses of the gathered data in real-time, we can take more informed decisions given the fact that companies operate globally from several locations around the world round the clock.

This project was implemented in two parts:

Part 1) Live streaming of Siemens data using Shodan’s APIs and extracting useful properties namely location.longitude, location.latitude.

Part 2) Performing analyses on the data by means of several clustering algorithms like KMeans, KNN, Agglomerative clustering in order to achieve the best possible cluster.

Clustering was a way to group related data based on certain properties and take operational decisions. Performing accurate clustering on a live-streamed data for the Siemens SCADA system is a challenge from the large volume of encrypted data which was overcome by the extraction of available data and the implementation of the algorithms.

# **PROJECT PROBLEM**

The topic of the study was challenging because of the lack of number of resources and the difficulty of access to the data of large manufacturing units. Since the task was to perform analyses on live streamed data, proper APIs had to be used in order to gather the data. The available resources like Kafka did not have access to the Siemens dataset. We had to use Shodan APIs in order to access the dataset.

The problem can be formulated as below:

### Which data can we use for Siemens and how can we access the data? Can we use static data or data has to be live-streamed. We have chosen live-streaming in order to get the real-time data.

### Is the data gathered encrypted? If data is encrypted then what methods should we deploy to extract useful and sufficient information to perform clustering? The data we had streamed was encrypted and we did not have access rights to the full data. In that case we have extracted header or metadata to extract the properties.

### Is the data that we have extracted structured or semi-structured or unstructured? The data we have gathered was JSON which was semi-structured and how to extract properties from the data?

### Once the properties were extracted what kind of analysis we would do on the data? We have done clustering by implementing several algorithms like KMeans, KNN, and agglomerative clustering.

### What can we infer from the clusters formed? We have achieved upto 95% accuracy by implementing the KMeans algorithm.

# **IMPLEMENTATION DETAILS**

The project was largely divided into 4 stages:

**Stage 1) Live-streaming of Siemens data using Shodan’s APIs.**

1. Due to unavailability of free data and the lack of support of free existing technologies, we used Shodan’s APIs to get the streamed data.
2. The data gathered could not be accessed entirely due to access rights so we accessed the metadata.

**Stage2) Extracting properties from the metadata**

1. The metadata was in the format of JSON file which was semi-structured. In order to get the properties we have written Python script.

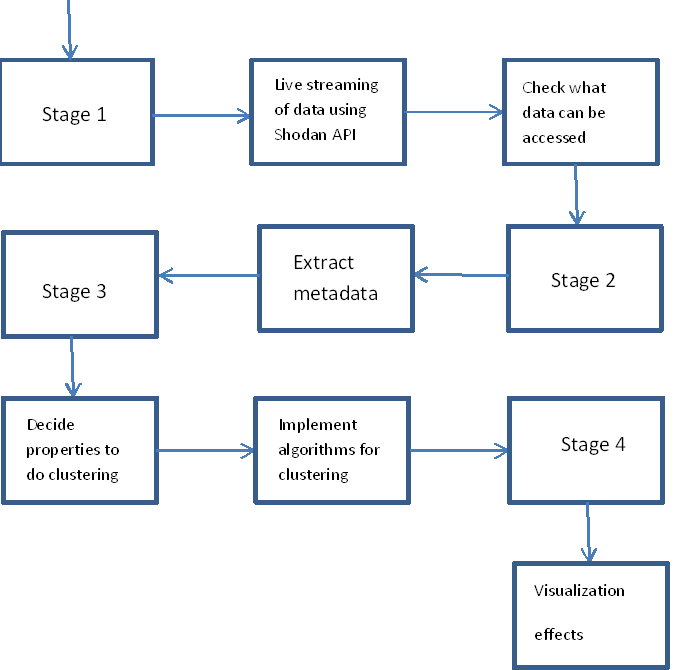
**Stage 3) Using the properties to form the clustering**

1. Once the properties were extracted we had to decide on which property we would do the clustering.
2. On inspection we found that location.latitude and location.longitude were 2 important parameters to check the connected devices across the globe, we decided to proceed with clustering based on these two parameters.
3. We implemented KMeans, KNN and Agglomerative clustering to compare the accuracies of the clusters formed and which algorithm would be best given our data to form the clusters.

**Stage 4) Creating visualizations for the clusters formed.**

1. Once the clusters were formed we have written Python scripts to perform the visualization across the globe for the various algorithms.
2. The Pyspark and mathplotlib were experimented to provide visualizations on the data clustered in the previous stage.
3. “ggplot” style was used to scatter the plot. From the json file, the location data was obtained in the form of list, which in turn had the geo-information stored in dict format.
4. Once the dict format was parsed, it was rightly converted to match the type of the parameter to be passed to the KMeans Clustering algorithm from the sklearn. cluster package. Similarly it was used for the Agglomerative and KNN clustering as well.
5. The no of clusters parameters for each algorithm was decided based on brute force technique. After deciding the number of cluster parameter for each algorithm, different coloring schemes were applied to the clusters to separate each clusters and provide them an identity.
6. The visualization obtained was able provided a clear, and an intuitive depiction of clusters.

# **Architectural Diagram**



# **Algorithms Implemented**

### **KMeans:**

Let  X = {x1,x2,x3,……..,xn} be the set of data points and V = {v1,v2,…….,vc} be the set of centers.

1) Randomly select ‘c’ cluster centers.

2) Calculate the distance between each data point and cluster centers.

3) Assign the data point to the cluster center whose distance from the cluster center is minimum of all the cluster centers..

4) Recalculate the new cluster center using:

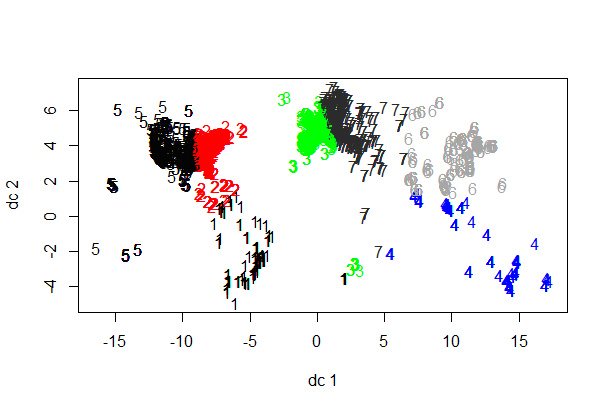
### https://sites.google.com/site/dataclusteringalgorithms/_/rsrc/1273048565389/k-means-clustering-algorithm/kmeans1.bmp

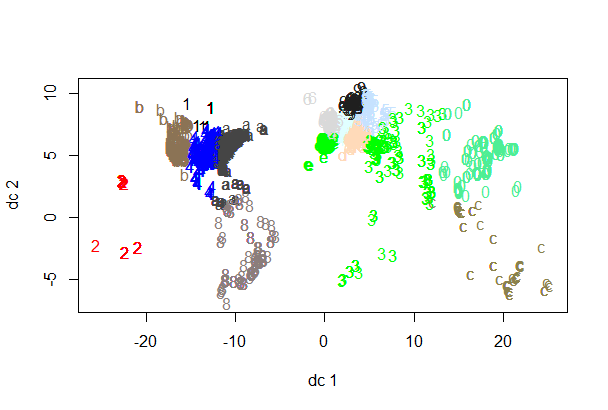
where, ‘ci’ represents the number of data points in ith cluster.

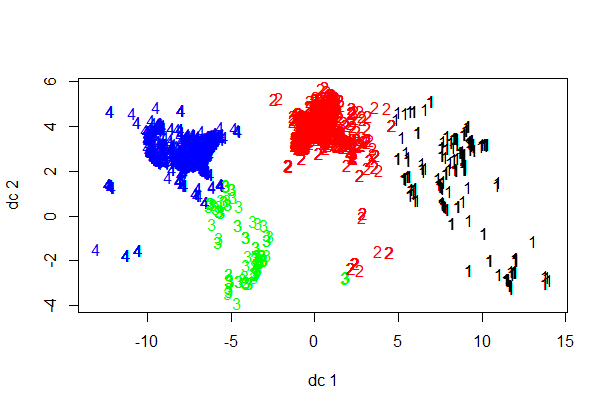
5) Recalculate the distance between each data point and new obtained cluster centers.

6) If no data point was reassigned then stop, otherwise repeat from step 3).

By using different means and iterations we got the below clusters:







### **2)** **KNN Algorithm:**

In [pattern recognition](https://en.wikipedia.org/wiki/Pattern_recognition), the ***k*-Nearest Neighbors algorithm** (or ***k*-NN** for short) is a [non-parametric](https://en.wikipedia.org/wiki/Non-parametric_statistics) method used for [classification](https://en.wikipedia.org/wiki/Statistical_classification) and [regression](https://en.wikipedia.org/wiki/Regression_analysis).[[1]](https://en.wikipedia.org/wiki/K-nearest_neighbors_algorithm#cite_note-1) In both cases, the input consists of the *k* closest training examples in the [feature space](https://en.wikipedia.org/wiki/Feature_space). The output depends on whether *k*-NN is used for classification or regression:

* In *k-NN classification*, the output is a class membership. An object is classified by a majority vote of its neighbors, with the object being assigned to the class most common among its *k* nearest neighbors (*k* is a positive [integer](https://en.wikipedia.org/wiki/Integer), typically small). If *k* = 1, then the object is simply assigned to the class of that single nearest neighbor.
* In *k-NN regression*, the output is the property value for the object. This value is the average of the values of its *k* nearest neighbors.

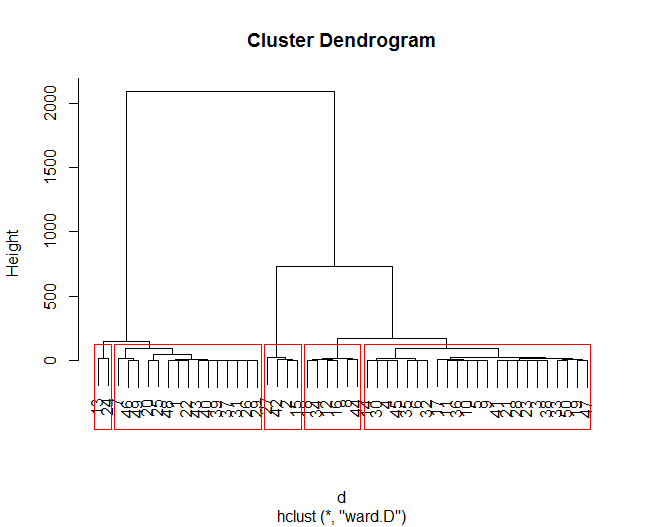
### **Agglomerative Clustering:**

This is a "bottom up" approach: each observation starts in its own cluster, and pairs of clusters are merged as one moves up the hierarchy.

Metric used to measure similarity is the Euclidean distance.



We have also used the Ward’s method to reduce the variance between the clusters.



**Conclusion:**

We can see that KMeans give the best visuals of the clusters based on the data.

# **CHALLENGES**

We faced several challenges during the project due to lack of accessibility to authentic data source. Below are the challenges we faced:

### Finding data source for live-streaming was a challenge due to unavailability of real data.

### Determining a technology to access the data. We used Shodan APIs since Kafka did not support accessing the data.

### Extracting the properties to be used from the live streamed data. This was a challenged since the live-streamed data came in the form of JSON files.

### Filtering and cleaning the data was difficult. We had to write scripts to clean and filter the data.

### Visualization was a challenge due to live streaming. Appropriate tools were not available to do the visualization so we used Python scripts to do the visualization.

# **CONCLUSION**

## Based on the analysis performed on the live-streamed data of Siemens we can see how KMeans outperformed other algorithms and how the changes are reflected for different means and iterations.

References

[1] <https://developer.shodan.io/api/banner-specification>

[2] <https://inductiveautomation.com/what-is-scada>

[3] <https://en.wikipedia.org/wiki/SCADA>

[4] <http://www.mckinsey.com/business-functions/operations/our-insights/how-big-data-can-improve-manufacturing>

[5] <http://www.remotemagazine.com/main/markets/scada/re-inventing-scada-for-the-next-generation-enterprise/>

1. December-12-2016. The University of Texas at Dallas, TX, USA

   Rumela Saraswati, The University of Texas at Dallas, TX, USA (e-mail: rxs145730@utdallas.edu).

   Chitra S. Kumar, The University of Texas at Dallas, TX, USA(e-mail: cxs154130@utdallas.edu).

   Keerthi Bala Sundram, The University of Texas at Dallas, TX, USA(e-mail:  [kxb152730@utdallas.edu](mailto:rxs145730@utdallas.edu)).

   Dhaval Adhav, The University of Texas at Dallas, TX, USA(e-mail:  [dsa150330@utdallas.edu](mailto:rxs145730@utdallas.edu)).

   Madhurya Alur, The University of Texas at Dallas, TX, USA(e-mail: mxv150730@utdallas.edu). [↑](#footnote-ref-0)