# 6406 VISUALIZATION

# PROJECT REPORT

# VISUALIZING AIRPORT & RAILWAY STATION

**YAMUNA JAYABALAN – B00741912**

# CONTENTS

## 1. INTRODUCTION

This project focuses on visualizing the Airport and Railway station dataset using a novel and interactive design. Dataset is collected from http://openflights.org/data.html and it consists of more than 10,000 records which has entries for airports and train stations operating over the globe. This dataset is used in the International Journal *"Analysis of Airport Data using Hadoop-Hive: A Case Study"* [1] (reference section). The journal mainly states the uses of Big data and how Airport data is processed to provide various information using Big data analytics. But the concentration of this project is only on efficiently visualizing the dataset. The data has many fields which are described later in this report. Various stages of this project are pre-processing of the dataset, analysing them, selecting appropriate visualization, visualizing dataset, optimising the visualization. Two different visualizations are created, one using google maps API and another using bubble chart for airport and railway dataset. The visualization using google map provides the user with exact location of the airport or railway station. Bubble chart visualization helps the user to view the distribution of the airport and railway station based on the altitude and time-zone.

## 2. MOTIVATION

There are number of airports and railways throughout the globe, and there is a need to visualize the data using efficient and interactive visualization system. There are existing visualization systems with static maps, but there is no existing visualization system with the additional functionalities of google maps provided by google API. There is also a requirement to know the way in which airports and railways are distributed across the time-zones.

## 3. ANALYSING DATASET

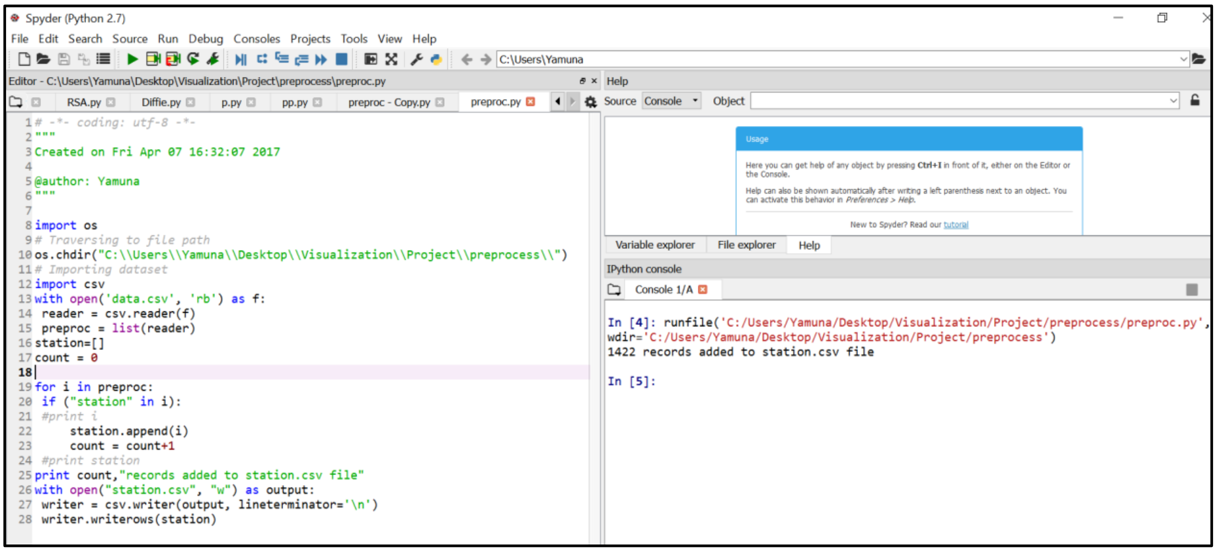
Dataset collected is updated as of January, 2017. Various fields of the dataset are described below:

| **Column** | **Description** |
|---|---|
| **Airport ID** | This is a unique identifier of OpenFlights, for a particular airport. |
| **Name** | It is the name of the airport. |
| **City** | It is the main city name where the airport is located |
| **Country** | Country where airport is located |
| **IATA** | IATA code which is in 3 letter format, can be Null when unknown |
| **ICAO** | ICAO code which is 4 letters, can be Null when unknown |
| **Latitude** | Latitude in degrees, which is usually up to 6 decimal digits. Negative values are for South, positive are North. |
| **Longitude** | Longitude in degrees, which is usually up to 6 decimal digits. Negative values are for West, positive are East. |
| **Altitude** | Height in feet at which the flight travels. |
| **Time zone** | Time zone in Hours. |
| **DST** | Daylight saving time. Single alphabet assigned for each country |
| | |
| **Tz dataset time zone** | Time zone in "tz" format. |
| **Type** | Type of entry. Eg. "airport" for air terminals and "station" for train stations. But for airports dataset all values will be "airport" by default. |
| **Source** | Source of the data entry and for airports dataset its "OurAirports" by default. |

The dataset has latitude and longitude values, the data can be efficiently visualized using a map based visualization system.

## 4. PRE-PROCESSING DATASET

Two datasets are used in this visualization. "airport.csv" is created by manually copying the data from "airports.dat","https://raw.githubusercontent.com/jpatokal/openflights/master/data/airports.dat". "station.csv" is created by using the python code which extracts the railway station records from "airport-extended.dat" file which has airports, railway stations and ferry terminals combined. The field names are manually added by referring the corresponding website. Pre-processing functionalities were discussed in detail before. The output of pre-processing code is shown below in Figure1.



Usage

Here you can get help of any object by pressing **Ctrl+I** in front of it, either on the Editor or the Console.

Help can also be shown automatically after writing a left parenthesis next to an object. You can activate this behavior in *Preferences > Help*.

New to Spyder? Read our tutorial

```
# -*- coding: utf-8 -*-
"""
Created on Fri Apr 07 16:32:07 2017

@author: Yamuna
"""

import os
# Traversing to file path
os.chdir("C:\\Users\\Yamuna\\Desktop\\Visualization\\Project\\preprocess\\")
# Importing dataset
import csv
with open('data.csv', 'rb') as f:
 reader = csv.reader(f)
 preproc = list(reader)
station=[]
count = 0

for i in preproc:
 if ("station" in i):
 #print i
     station.append(i)
     count = count+1
 #print station
print count,"records added to station.csv file"
with open("station.csv", "w") as output:
 writer = csv.writer(output, lineterminator='\n')
 writer.writerows(station)
```

```
In [4]: runfile('C:/Users/Yamuna/Desktop/Visualization/Project/preprocess/preproc.py',
wdir='C:/Users/Yamuna/Desktop/Visualization/Project/preprocess')
1422 records added to station.csv file

In [5]:
```

Figure 1 – Pre-processing dataset output screenshot

## 5. VISUALIZATION USING GOOGLE MAPS

Airport and Railway station data is visualized using google maps API. This part of visualization mainly uses the latitude and longitude parameters of the datasets. For airports, IATA code is displayed in the map near marker of each location corresponding to each record in the data. For railways, name is displayed in the map near the marker of each location.

The function used to mark the latitude and longitude on google maps is "google.maps.OverlayView()" function. Overlays are SVG objects that are red circles in this case placed on the map using SVG that are tied to latitude and longitude from the airport/station dataset. These overlays move along with the map when user drags or perform zoom on the map. Google Maps JavaScript API provides the OverlayView class for creating custom overlays. This OverlayView is a base class that provides methods that helps to implement custom overlays. The class also provides some methods that can be used to translate between screen coordinates and locations on the map.

### Using Custom overlay for Visualization

- Creating a new instance of custom overlay object google.maps.OverlayView() This will create a subclass of the original overlay class.
- Creating a constructor for the custom overlay.
- Implementing an onAdd() method within the prototype, and attaching the custom overlay to the map. OverlayView.onAdd() is called when the map is ready for the overlay to be attached.
- Implementing draw() method within the prototype, and creating SVG to place the circle on the map. OverlayView.draw() will be called when the circle is displayed.

*IEEE paper "Munin: A Peer-to-Peer Middleware for Ubiquitous Analytics and Visualization Spaces"* [2] (reference section) section 7 "MUNIN: VISUALIZATION LAYER" discuss about the Google maps API used and is explained. This paper provides a ubiquitous analytical environment that has multiple input and output systems. In the image below the system runs a PolyZoom on a 3 * 2 LCD display wall. Other android devices are used to interact with the map. The screen shows all the participants working in symphony. The google maps fetches these participants on a special node using the latitude and longitudinal data.

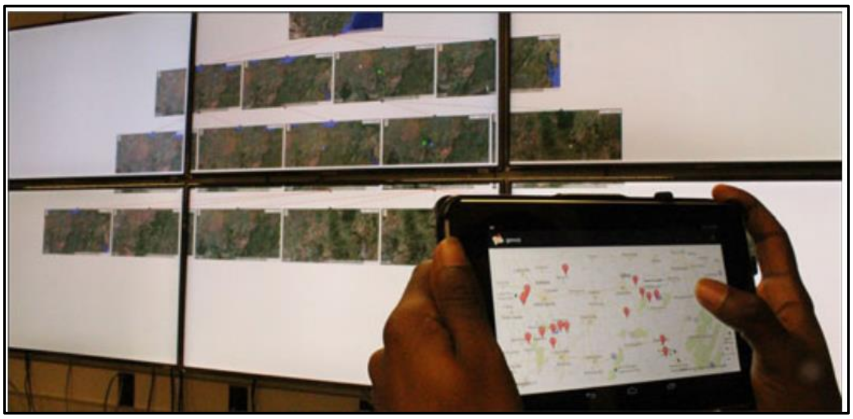


Figure 2

*"Mapmap.js: A Data-Driven Web Mapping API For Thematic Cartography"* [3] (reference section) This journal provides an idea to implement a java script that can be re-used by the users to draw maps. The user need not write code to draw a map from the scratch. The journal also discusses about google API which is another re-usable code that are used by experts to create maps easily. The paper also discusses about the requirement of better understanding of maps and visualizations using them.

## 6. VISUALIZATION USING BUBBLE CHART

The second visualization system that is used to visualize the dataset is bubble chart in d3. The bubbles are created for each record in the dataset. The radius of the bubble corresponds to the altitude at which the flight flies from the dataset. Name of Airport/station is shown upon mouse over activity. Force function is used to converge the bubbles. The bubbles are also separated based on the time-zone classification. The option to split or combine a bubble chart is provided in the visualization system.

Force layout in d3 is used in bubble chart visualization system. Force works based on the concept of tick of the clock for each iteration. For each tick, the object push/pull themselves until it finds its position on the layout. "Velocitydecay" attribute is used works like friction, adjusting the velocity of the nodes by multiplying by 1 - velocityDecay each tick.

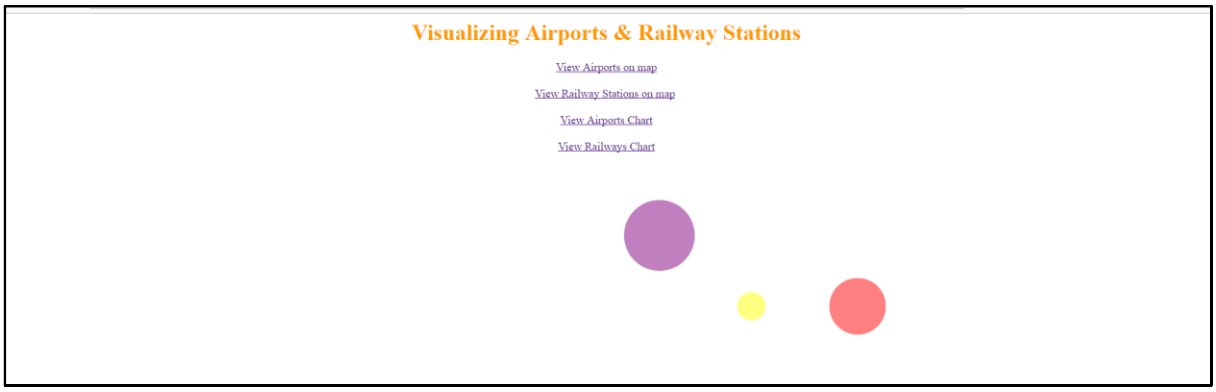
*IEEE paper "VisDock: A Toolkit for Cross-Cutting Interactions in Visualization"* [4], This paper presents a visualization toolkit called VisDock, which has visualization with interaction elements like selection, filtering, layer management and so on. It can be either used as a software by an end user to visualize data or it can be used by developers to develop web based visualizations. This implementation also uses bubble chart for visualization.

*IEEE paper "A methodology for airport arrival flow analysis using Track data – a case study for MDW arrivals"* [5] This paper discusses about the methodology for the flights arriving or departing at the airport. It describes a methodology for airport flow analysis to help use the airport space in efficient way. The analysis is done using National Offload Program (NOP) track data.

*IEEE paper "An exploratory data analysis of airport wait times using big data visualisation techniques"* [6] This paper talks about an interesting factor on airport wait times using big data and visualization techniques. Airport wait time is caused by multiple factors based on customer immigration, custom, dining, and it is also based on flight arrival time. Some of the busiest airports like Los Angeles and Chicago are considered in this paper for predicting wait time. For future implementation, all the airports can be take into consideration.

## 7. RESULTS & DISCUSSION

The first screen provides link to the four visualization pages



Visualizing Airports & Railway Stations

View Airports on map

View Railway Stations on map

View Airports Chart

View Railways Chart

*Figure 3 – Index page*

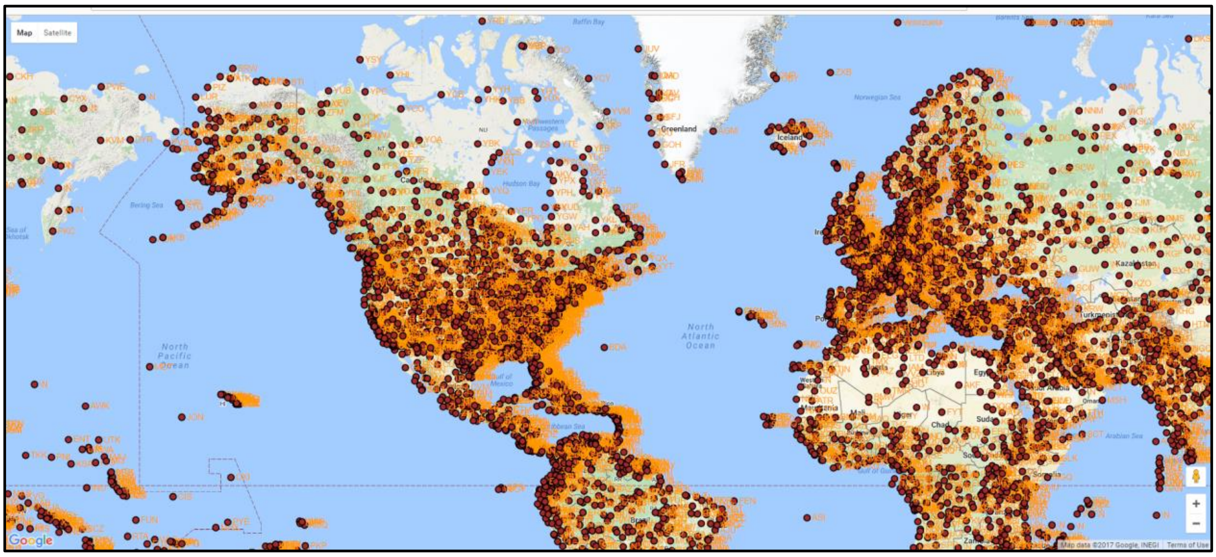


*Figure 4 – Airports Visualization using maps*

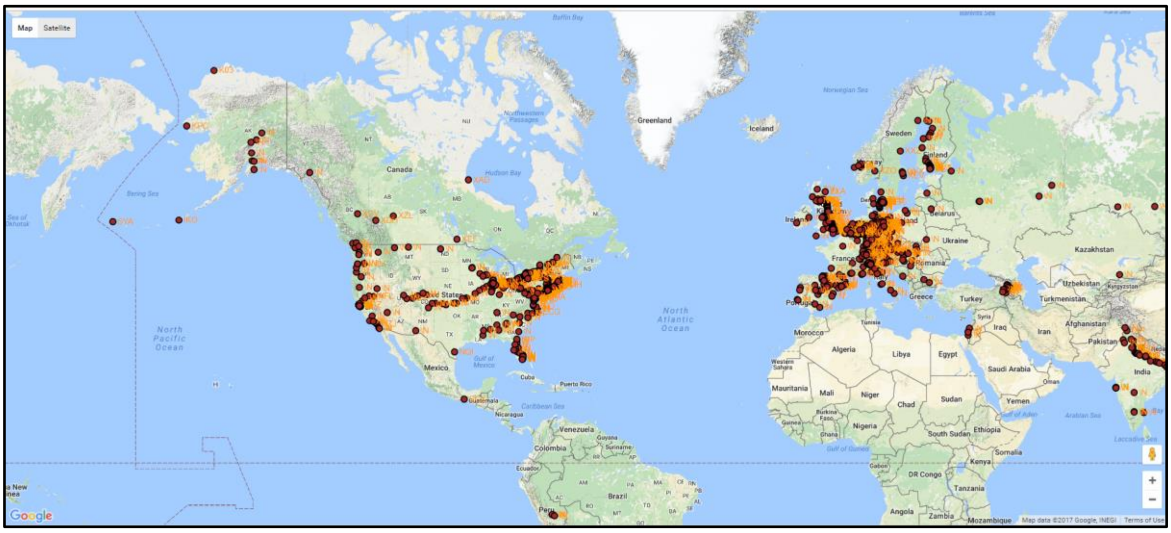
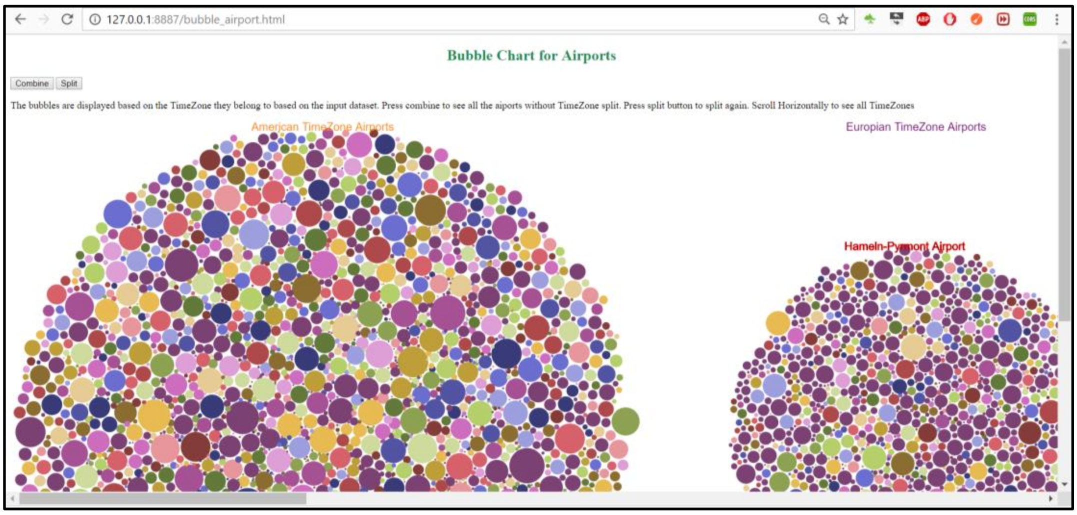


*Figure 5 – Railway Station Visualization using maps*

Bubble chart creates a bubble for each entry in the dataset. The radius of the bubble corresponds to the altitude. On mouseover activity, the airport name is displayed as shown in the figure below. Scroll horizontally to see the bubble chart for various timezones.



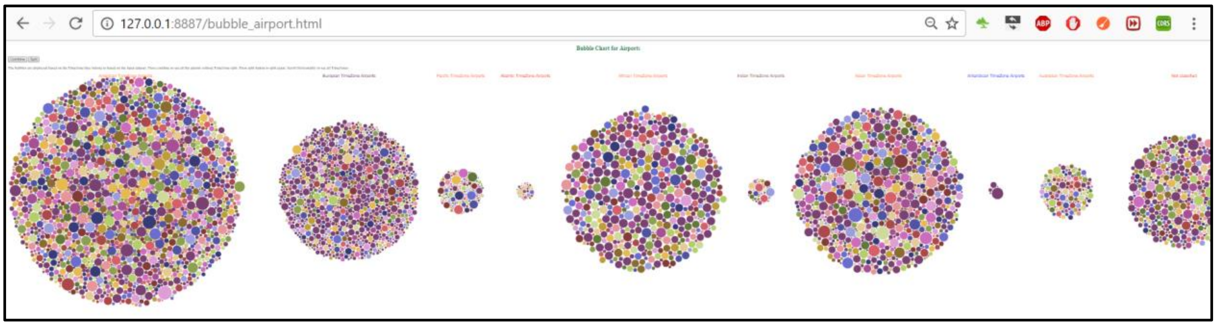
127.0.0.1:8887/bubble_airport.html

Bubble Chart for Airports

Combine Split

The bubbles are displayed based on the TimeZone they belong to based on the input dataset. Press combine to see all the aiports without TimeZone split. Press split button to split again. Scroll Horizontally to see all TimeZones

*Figure 6 – Airports Visualization using Bubble chart – TimeZone split*

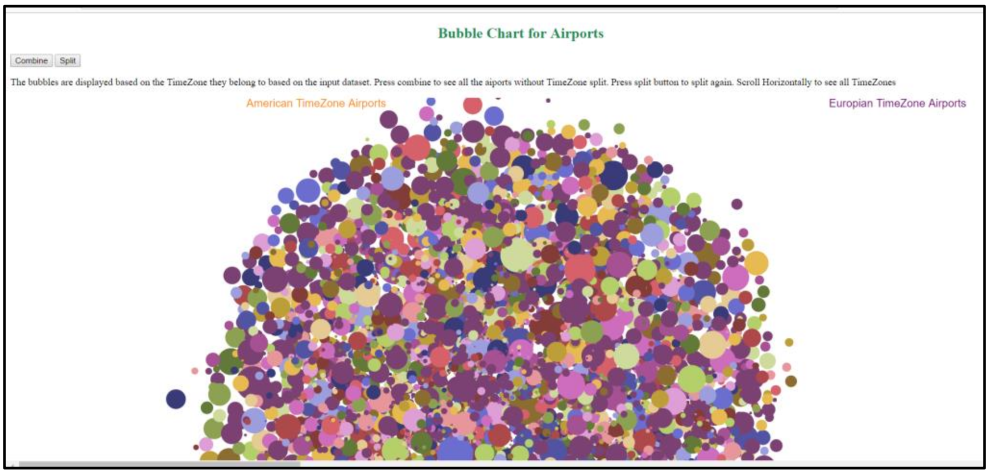


127.0.0.1:8887/bubble_airport.html

Bubble Chart for Airports

*Figure 7 – All TimeZones*

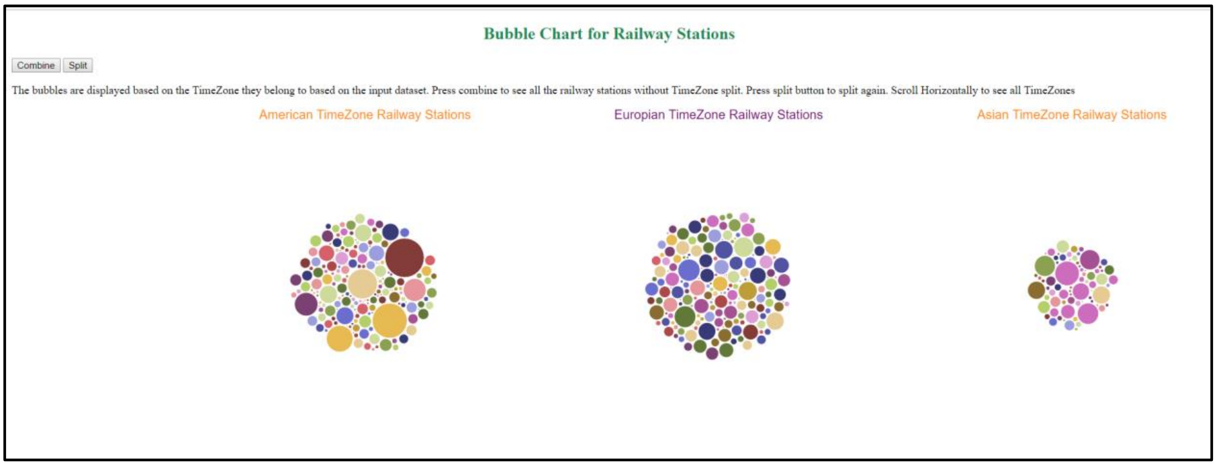
Click combined button to see the visualization shown below in Figure 8 and Figure 10



Bubble Chart for Airports

The bubbles are displayed based on the TimeZone they belong to based on the input dataset. Press combine to see all the aiports without TimeZone split. Press split button to split again. Scroll Horizontally to see all TimeZones

*Figure 8 - Airports Visualization using Bubble chart – Combined*

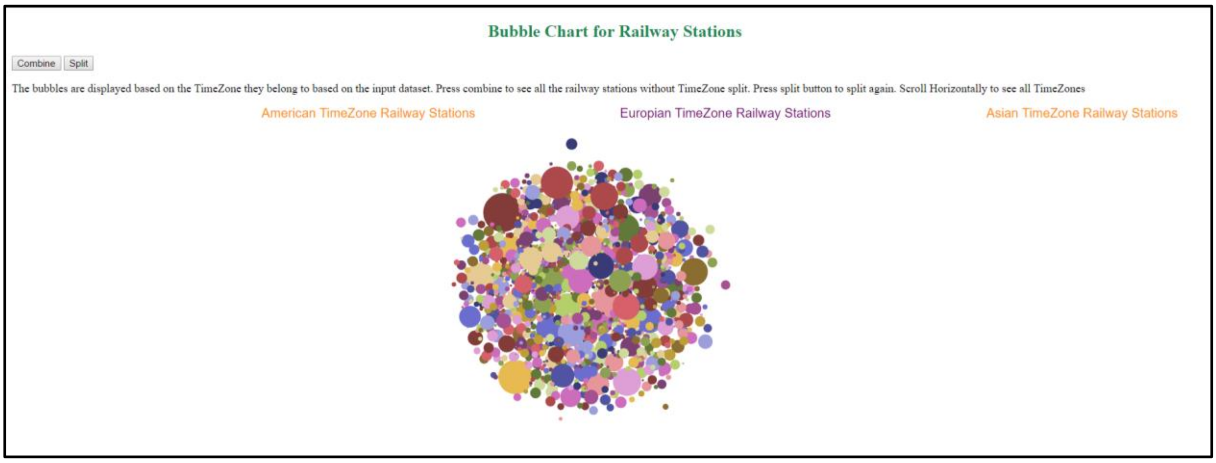


Bubble Chart for Railway Stations

The bubbles are displayed based on the TimeZone they belong to based on the input dataset. Press combine to see all the railway stations without TimeZone split. Press split button to split again. Scroll Horizontally to see all TimeZones

American TimeZone Railway Stations
Europian TimeZone Railway Stations
Asian TimeZone Railway Stations

*Figure 9 – Railway stations Visualization using Bubble chart – TimeZone split*



Bubble Chart for Railway Stations

The bubbles are displayed based on the TimeZone they belong to based on the input dataset. Press combine to see all the railway stations without TimeZone split. Press split button to split again. Scroll Horizontally to see all TimeZones

American TimeZone Railway Stations
Europian TimeZone Railway Stations
Asian TimeZone Railway Stations

*Figure 10 – Railway stations Visualization using Bubble chart – Combined*

## 8. CONCLUSION

The visualization system implemented is very novel, interactive and useful from the user's perspective. It has used the data in most effective way to provide the effective visualization of it. Many future analysis and implementations are possible with this system. It can be further improved to connect flight routes. The bubbles can be linked to the map visualization, so when a bubble is clicked, the corresponding marker on the map visualization is displayed.

## 9. REFERENCE

[1] International Journal of Computer Applications, "*Analysis of Airport Data using HadoopHive: A Case Study* ", 2016, http://research.ijcaonline.org/ncrtit2016/number2/ncrtit201634.pdf

[2] *"Munin: A Peer-to-Peer Middleware for Ubiquitous Analytics and Visualization Spaces"* http://ieeexplore.ieee.org/document/6851203/?reload=true

[3] "Mapmap.js: A Data-Driven Web Mapping API For Thematic Cartography", http://www.rbc.lsie.unb.br/index.php/rbc/article/viewFile/1322/851

[4] "VisDock: A Toolkit for Cross-Cutting Interactions in Visualization", IEEE, http://ieeexplore.ieee.org.ezproxy.library.dal.ca/stamp/stamp.jsp?arnumber=7063249

[5] "A methodology for airport arrival flow analysis using Track data – a case study for MDW arrivals", IEEE, http://ieeexplore.ieee.org.ezproxy.library.dal.ca/stamp/stamp.jsp?arnumber=6548547

[6] "An exploratory data analysis of airport wait times using big data visualisation techniques", IEEE, http://ieeexplore.ieee.org.ezproxy.library.dal.ca/document/7779379/

[7] Creating Animated Bubble Charts in D3,Jim Vallandingham http://vallandingham.me/bubble_charts_in_d3.html

[8] Contentshare – "Airport, airline and route data", 2017, http://openflights.org/data.html
[9] Google, "Google Maps APIs", https://developers.google.com/maps/documentation/javascript/tutorial

[10] Github, https://raw.githubusercontent.com/jpatokal/openflights/master/data/airports.dat