

Visualization Research Center (VISUS)

University of Stuttgart
Allmandring 19
70569 Stuttgart
Germany

Master Thesis Nr. 0 1414 0001

Online Visualization of German Power Plants and Their Production

Kazi Riaz Ullah

Course of Study: INFOTECH

Examiner: Prof. Dr. Daniel Weiskopf

Supervisor: Dipl. -Inf. Rudolf Netzel,
Dipl. -Inf. Nils Rodrigues,
Prof. Dr. Bruno Burger,
Dipl. -Inf. Alexander Schultz

Commenced: September 01, 2016

Completed: March 03, 2017

CR-Classification: H.5.2

Acknowledgments

Acknowledgement goes here.....

Abstract

Summary of the thesis goes here

Contents

1	Introduction	1
2	Related Work and Foundation	3
2.1	Interactive Maps	3
2.1.1	Visualization Techniques	6
2.2	Time series visualization	8
3	Background of This Project	11
3.1	Concept	11
3.2	Tools	12
3.3	Leaflet	12
3.4	Other Libraries and Plugins	13
3.5	Working Environments	16
3.6	File Format	17
3.6.1	Advantage	19
3.6.2	Disadvantage	19
3.7	Power line(GeoJSON) File Source	19
4	Software System	23
4.1	Requirement Analysis	23
4.2	Architecture	25
4.3	User Interface	25
4.3.1	Control layer	26
4.3.2	Map API	29
4.3.2.1	Functionality	31
4.3.2.2	Connectivity between map and energy-charts	32
4.3.3	Comparison List	34
4.4	Data query process on energy charts	35
4.5	Optimization of power line visualization	41
4.5.1	Map Bounds	42
4.5.2	The Length of Power line	43
4.6	Summary and Discussion	45

5	Evaluation	47
5.1	Online survey setup	47
5.1.1	Question groups	48
5.2	Survey Results and Discussion	49
5.2.1	Summary of participants and their backgrounds	49
5.2.2	Working with the Interactive Map	50
5.3	User Feedback	55
5.4	Summary	55
6	Discussion	57
6.1	Advantage of the Tool	57
6.2	Weakness	57
7	Conclusion and Future Work	59
7.1	Summary	59
7.2	Future Work	59
	Appendix	61
	Bibliography	63

List of Figures

2.1	Interactive map showing the electricity production of Germany	4
2.2	Interactive map showing how the United States generates its electricity	5
2.3	Interactive map : Menu selection mechanism	7
2.4	Interactive map with Selectable layers and Legend.	8
2.5	UK electricity generation by fuel type. Source:[SE15]	9
2.6	U.S Dry natural gas production by type. Source:[Eia17]	10
3.1	Impact of using marker cluster plugin	14
3.2	Impact of using marker cluster and Leaflet label plugin	15
3.3	Overpass turbo query wizard for generating query	20
3.4	Overpass turbo API query (left) with rendered vector layer of 220kV power line (right) on the Openstreetmap	20
4.1	The architecture of the visualization tool	26
4.2	The main structure of the visualization page	27
4.3	Navigation menu	27
4.4	Tool tip for Check-boxes	28
4.5	Navigation menu with all power plant selected.	28
4.6	Navigation menu where one energy-source is deselected.	28
4.7	A German state (Baden-Württemberg) is highlighted	29
4.8	Interactive map area	30
4.9	Showing the geographical location of all power plants with markers.	31
4.10	Showing all nuclear power plants.	32
4.11	Marker Pop-up box	33
4.12	Buttons inside pop-up box	34
4.13	Wind-offshore(3) power plants are listed on the comparison table	35
4.14	Energy Chart showing the hourly production data	36
4.15	Hyperlink of "Go to Energy Charts"	37
4.16	Steps of adding power plants to the comparison list	38
4.17	Selected power plant data from the comparison table of Figure 4.16	40
4.18	All possible data format that be extracted from the Overpass-turbo API	42
4.19	Filtering power lines depending on the map bounds	43
4.20	Filtering power lines by coordinate length	44

5.1	Age ration and professional background of survey participants	49
5.2	Familiarity with German power plants and their production	50
5.3	The result of online survey - participants frequency of using interactive map per week	51
5.4	The result of online survey - map usability	51
5.5	The evaluation result of online survey	52
5.6	The evaluation result of complete visualization tool	54
5.7	The result of online survey - complete map usability and usefulness	54

List of Listings

3.1	An example of JSON-object for Brown Coal(lignite) power plant	17
3.2	An example GeoJSON-object for 110kV power line inside Germany	18
4.1	URL variables are handled using JavaScript	39
4.2	Example of hourly production JSON data object	40
4.3	JavaScript pseudo code for checking unit ids and enabling their visibility on the charts	41
4.4	Power line filtering algorithm depending on the map bound	44

1 Introduction

Data visualization allow us to take valuable information from itself instead of being bored from reading rows upon rows of spreadsheet data. It also allows extracting complex data into graphic visual representations and to communicate information clearly and efficiently via interactive graphical presentations. This visual communication process becomes more efficient when interactivity is added to it. Since July 2014, the Fraunhofer Institute of Solar Energy Systems (ISE) has been providing interactive charts on electricity production and other related information regarding electricity and power generation in Germany. These charts became very popular and widely used by people from different professions namely scientists, politicians, journalists as well as online/printed media. Due to the high popularity of these interactive energy charts, an additional interactive map is developed to make the whole framework more informative and interesting for the users. Thereby the aim of this visualization tool was to develop an interactive map to illustrate the geographical location of power plants in Germany on the OpenStreetMap. On top of that, this data layer is providing necessary information about power plants, source, capacity, operator, start date and other additional information that user might be interested in. Furthermore, this new visualization framework is interconnected with the existing energy charts which make the framework stronger in principle.

However, the data set used for this visualization does not include all renewable and non-renewable power plants. Basically, the data is periodically collected from different neutral sources which include power plants that have an output of more than 100MW. Their reporting is obligatory to our source. In addition, the data set also contains information on voluntary reporting companies which are generating a relatively low power output. Apart from this, this interactive map also illustrates the power lines of Germany, specifically the 110KV, 220KV and 380KV lines. The data set for these power lines is extracted using Overpass turbo API.

Due to the lack of frameworks that supply a sufficient visualization for this kind of scenario and data, it seemed to be essential to develop some additional interacting factors that can be added to present the data in a way which tells the story of the data set more clearly. On the other hand, there is a scope of giving the user more control to make the interactive chart more useful. The main focus of this project was to visualize the data overlaid on a geographical projection using open source data mapping tool and deploy this framework to the web environment. In addition, a fully interactive data visualization is provided by adding custom API which allows technical and non-technical experts easy to get to grips with and custom made glyph icons are used to grasp the context.

Outline

This structure of the thesis as follows:

Chapter 2 – Related Work and Foundation: In this chapter, .

Chapter 3 – Background of This Project: In this chapter, .

Chapter 4 – Software System In this chapter, .

Chapter 5 – Evaluation In this chapter, .

Chapter 6 – Discussion In this chapter, .

Chapter 7 – Conclusion and Future Work In this chapter, .

Acronym

In this paper several abbreviations are used which may not be commonly known to everyone. Therefore, the following list is added to clarify the used terms and to avoid confusions with similar abbreviations.

Acronym	Definition
OSM	Open Street Map
UI	User Interface
JSON	JavaScript Object Notation
API	Application Programming Interface

2 Related Work and Foundation

The research on data visualization was settled in the context of Power Plants and their production mapping based on the geographical information system. This chapter discusses related work about interactive maps and time series visualization.

2.1 Interactive Maps

Maps are one of the more popular ways to visualizing geolocated data. There exist many web based dynamic interactive maps to visualize large, complex and compressed data set with the powerful interface. These are used to present economical, cultural, or scientific information about different geographical objects or locations such as cities, districts, or countries. It requires less search time to find objects and location of interest than a non-interactive map. These interactive maps are equipped with a variety of features and glyphs to simplify visual comparison and to speed up the search among displayed entities. Nowadays maps are light weight and portable. Unlike old techniques, new generation maps provide users an environment where the user can manipulate the data by clicking, scrolling, zooming, and get the details on demand. Color mapping, transparency, and glyph allow guiding users in an understandable way to grasp information from the whole context of visualization of a large data set, which is presented on the display. Modern interactive maps also allow filtering and retrieving information in an efficient manner. This flexibility of modern facility is attained by providing a clear and concise UI.

In September 2016, CarbonBrief [SE16] has published an interactive map on their official website with an article named "How Germany generates its electricity". This is built using Leaflet and mapbox two open source libraries which illustrates all renewable and non-renewable power plants inside Germany. This map is based on data published by the German Federal Network Agency. A circle area marker is used to locate the power plants and its area represents the capacity, as shows in the Figure 2.1. Unfortunately, most of the power plant locations are based on postcodes which are converted to degrees latitude and longitude. Therefore, their location is not accurate. Within this map, a drop-down selector is used for interaction. Users can filter data on the map by selecting different power sources. An unique color is used for each source. But its significance is not explained in the online article.

2 Related Work and Foundation

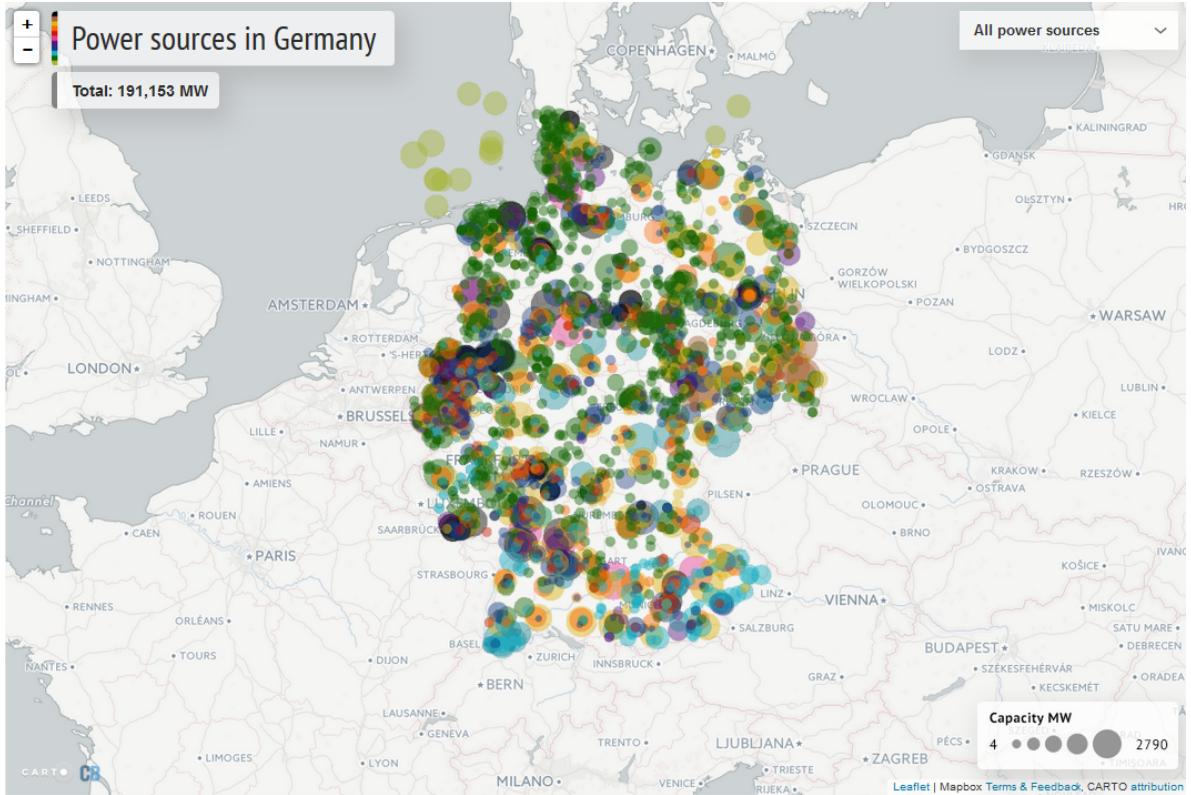


Figure 2.1: Interactive map showing the electricity production of Germany

Another interesting approach to the interactive map was published on July 2015 in the Washington Post [JMG15] under the article of "how United states generate its electricity". This map provides information available power sources and its capacity in MW per US state. Like the previous example, circular area marker is also used here to locate power plants. Unfortunately, it's not possible to know the capacity of each source individually. A Choropleth map is used to render different states inside the USA. The user can get information about the total power generated in a state by hovering over any state inside the map. The details of Choropleth map is discussed later in this section. A label is bound to each state which shows the total power generated by each source category and the area of the state is filled with darker color according to the principle of choropleth map. Unlike the previous example, no additional control layers are provided along with the map which makes this less interactive. However, meaningful and self explanatory glyph are used to define each power plants. The color of the logo of a power plant and color of the circular marker of the same power plant is the same. This scheme made easier to focus on each power plant category from a large dense area of the map..

Choropleth map, Propotional Symbol map, Pinpoint maps, Connection maps, Isopleth maps are common types. In our case, particularly Pinpoint map and Choropleth are focused for

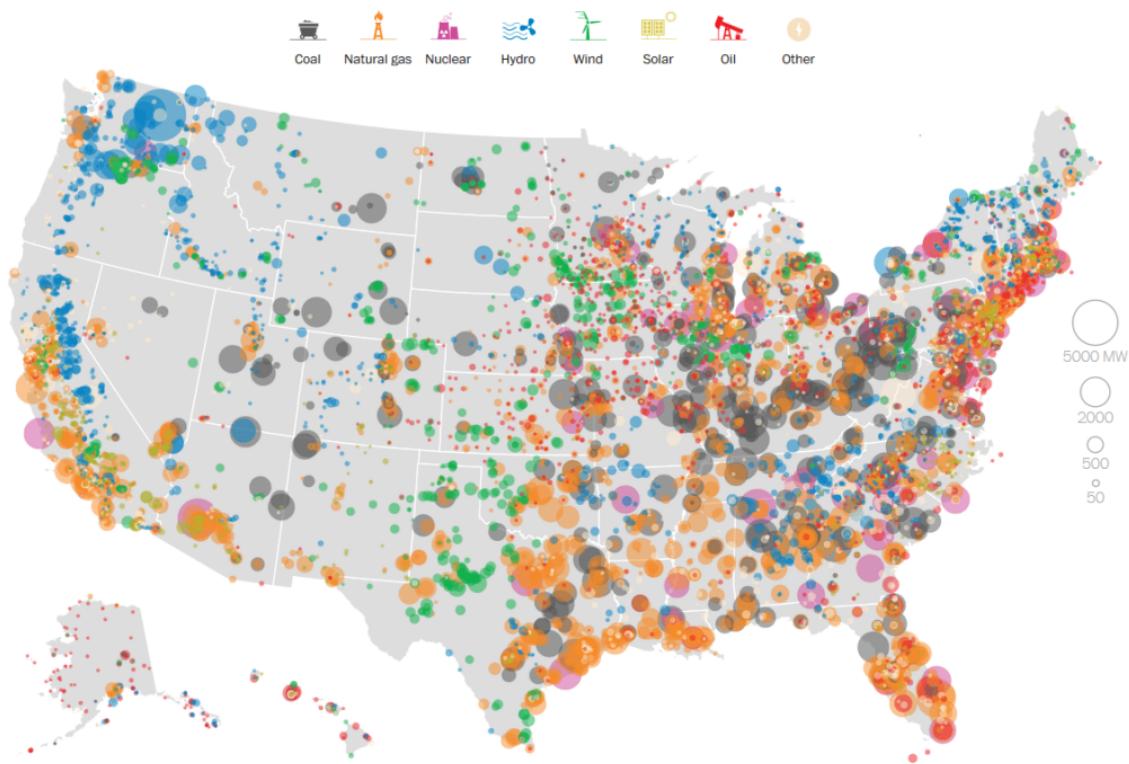


Figure 2.2: Interactive map showing how the United States generates its electricity

locating the power plants and visualize their density inside Germany. In general, Choropleth Map displays geographical area surrounded by the border or area is colored or shaded in relation to a data variable. This mapping techniques provide a way to visualize density/value over an area. A color progression scale is used to represent the value in each area. On the other hand, pinpoint mapping is used to visualize the exact location of things. Currently, this mapping technique has become more popular. For example, Google Maps show the exact locations of points of interest. Companies from different sectors and service providers are creating their own applications using Google Maps or OSM to provide and share locations of different places such as banks, ATM machines, restaurants, hospitals, shopping malls, roads, rail ways, and shortest paths. We have used this mapping technique to visualize the exact geographical location of the power plants in Germany.

2.1.1 Visualization Techniques

Glyph-Based Visualization

In different approach for locating renewable and non-renewable power plants and their production different countries have adapted different approaches. A common technique that has been observed from research, such as [SE16] and [JMG15], is a circular area on the map. Circular area is the most used marker on the map. Basically, the center of the circle is locating the position and its area is describing the electricity production capacity of that corresponding power plant. These circular areas are filled with colors to categorize the power plants. Circles filled with different colors are easy to identify without deep investigation. To visualize the capacity along with the location of the power plant an ordinal data type is used to compare the size of a circle which is relative to a certain amount of Gigawatts or Megawatts. Mostly a legend is added to the map which provides useful meaning about the circles and helps the user to perceive useful information from the map. On the other hand symbols or glyph are used to represent the object which is the most fundamental way to show the position of power plants on the map. Markers with different colors are also used for easier navigation. Essentially markers can be used to represent anything that has a global position in latitude and longitude coordinates. Every interactive map generation tool provides default markers, which are familiar to most users. However, developers are able to change the markers or add own meaningful marker images. Along with the marker additional label or snippet added to provide more information or description about the marker. This title or snippet is displayed in an info window, a bubble appears over the marker when the user clicks or hovers over the marker.

CarbonBrief [SE16] and the Washington Post [DE16] have introduced two interactive maps. They provide information on how Germany generates its electricity. Both maps have used a colorful circular marker to locate the position of the plant inside Germany as well as the area of the circle representing the capacity. An internal drop-down navigation menu is added to the map of [SE16] where the user can filter the plants by selecting different sources from the menu. On the other hand, [DE16] is showing 380kV and 220kV power lines inside Germany along with the circular marker locating the plants and their generation. But there are no menu selection mechanism or filters for showing different power plants on the map. The available interaction techniques might be difficult for the users who are not expert enough using this type of interactive visualization tools.

However, those approaches are missing the opportunity to engage user into the map on initial view. Circular markers with different colors might confuse the users to interpret the specific type of power plants. In particular, there is more than one unit of power plants are located in the same area. A plant with smaller generation capacity can be overlapped by a plant with larger capacity because of its larger circular area. Therefore, pin point marker is used in our visualization framework to locate the power plants on the map. Logo inside the marker is self-explanatory and color of the markers is provided by the data set.

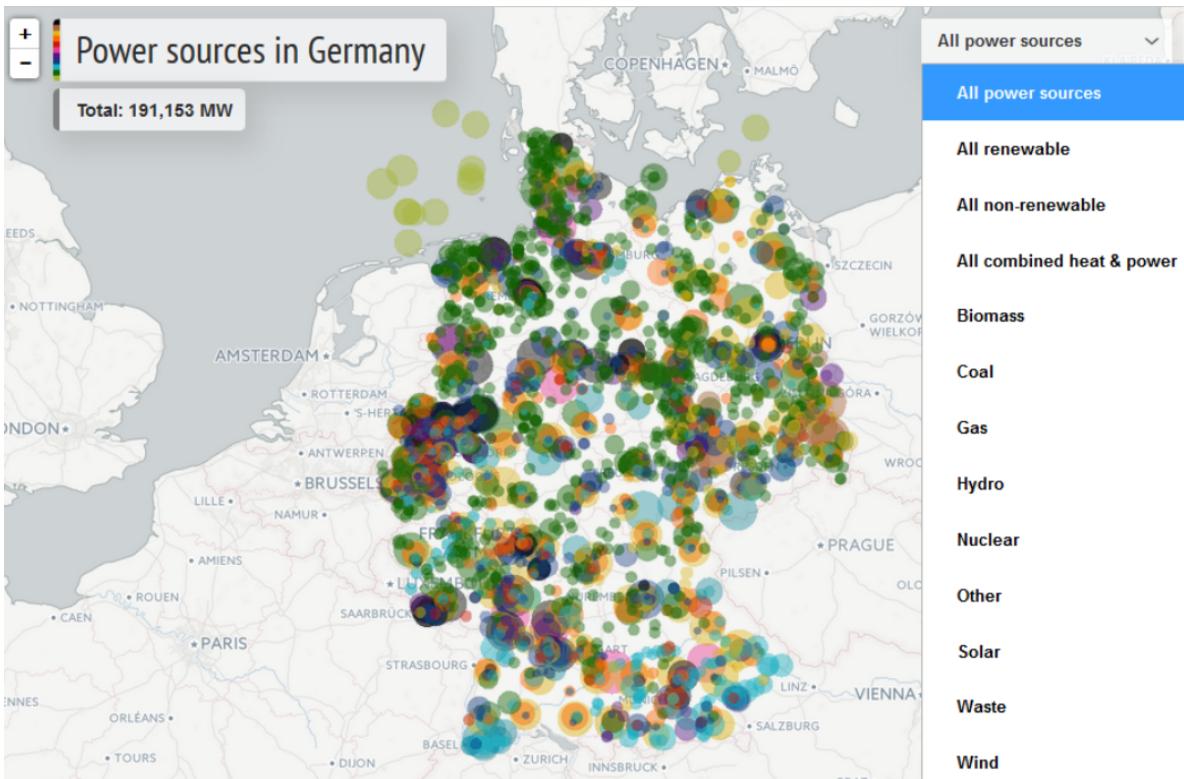


Figure 2.3: Interactive map : Menu selection mechanism

Interface and Interaction

In CarbonBrief [SE16] a full width map is used with menu selection mechanism (Figure 2.3). With this menu selection, the user can select different categories of power sources. All this menu are hidden in the drop down section. Users are able to select only one category at a time from the menu. There is no possibility to select multiple categories of sources and render them into the map. This restricts users to view their desired power plants on the map. On the other hand, the Washington Post in [JMG15] has published the map with not much functionality. No additional navigation menu is provided along with the map. To get details, user needs to hover over the map.

In Geoportal [DE16] also published a web based interactive map and providing information with a large and complex interface. Their interactive map has multiple functionality but still, lacks some important features. For example, they are visualizing the power plants, which has the capacity above 100MW or equal as we are doing with our visualization framework. But there is no scope of power plant selection mechanism (Figure 2.4). A descriptive but rather complex legend is added on the map. The user might find it difficult and it would take more time to explore the desired output out of this information visualization technique.

2 Related Work and Foundation

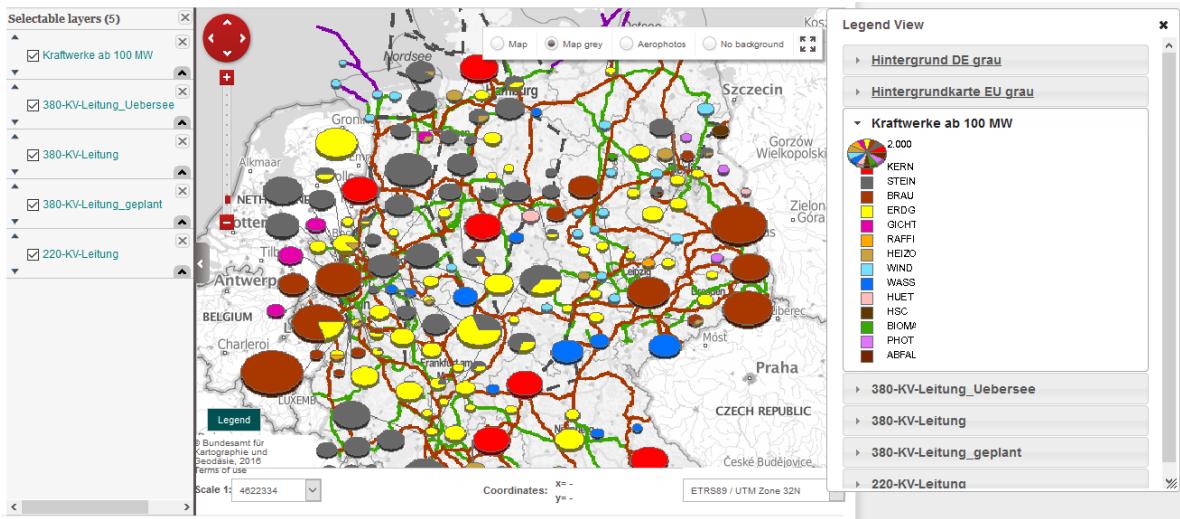


Figure 2.4: Interactive map with Selectable layers and Legend.

All these approaches and visualization techniques indeed enable users to extract the desired information from the map. However, in our visualization tool, we tried to make a clear, concise and efficient user interface. It might be easier for the user to figure out how our application works. We also tried to use familiar and consistent approaches to make the interface intuitive and given some more functionality to the users which are somehow missing in [SE16], [JMG15] and [DE16].

2.2 Time series visualization

There are countless visualization techniques that show time series data in the charts. One of the unique time series visualization is ThemeRiver [HHWN02]. This visualization depicts the thematic variations and changes over time within a large collection of data. The changes are shown in the context of a time line. The main focus of this visualization is to allow users to discern patterns which help to understand trends of the data-set. The theme river flow is directed from left to right. The variation of width represents the variation in the degree of representation unit. The horizontal width between two points on the theme river represents time interval and vertical distance or height of the river at any point of time indicates the strength of that point of the corresponding data set. It shows several streams, i.e. variables which changes over time and lays on top of each other as a layer. In visualization methods for time dependent data, Mueller and Schauman in [MS03] have discussed different conventional approaches for visualizing time dependent data. ThemeRiver is one of the most well-known techniques to visualize multivariate data over Time. An intuitive interpretation of temporal

changes can be observed by using this technique. In February 2008 New York Times depicting the “Ebb and Flow of Movies: Box Office Receipts Over Past 20 Years” [MB08]. It was an interactive ThemeRiver visualization which illustrates the pattern of the amount of money films over a 21 years period make at the box office. This total figure is shown by varying heights and width that reach over time. A color scale is used to reveal the amount of revenue range. The online response to this visualization was drastic and controversial. Marco and Yifan also mention in [DBH16] about this New York Times(NYT) publication. Later Byron and Wattenberg [BW08] outlined three issues which affected the aesthetics of the graph, e.g, the ordering different layers, the shape of the lowest curve, which suppose to be the baseline and the labels of the layers. ThemeRiver fits in a general mathematical framework. A standard way to visualize time series data is to plot them on a Cartesian graph, which has time on the x-axis and the numeric values on the y-axis. A flat baseline allows the user to easily read the total diagram.

Later on, this flat baseline strategy has become very popular to visualize power production from renewable and non-renewable energy sources over time this visualization technology has been widely used. In another article of CarbonBrief [SE15], ThemeRiver visualization technique (see Figure 2.5) is used to illustrate how UK is generating electricity over the last 50 years. Around 30% of the UK electricity came from coal, which is clearly visible from the illustration. On the other hand, US energy information administration in their Annual Energy Outlook 2017 [Eia17] this flat base ThemeRiver is used to visualize the data for different aspects. Figure 2.6 illustrates an example of it.

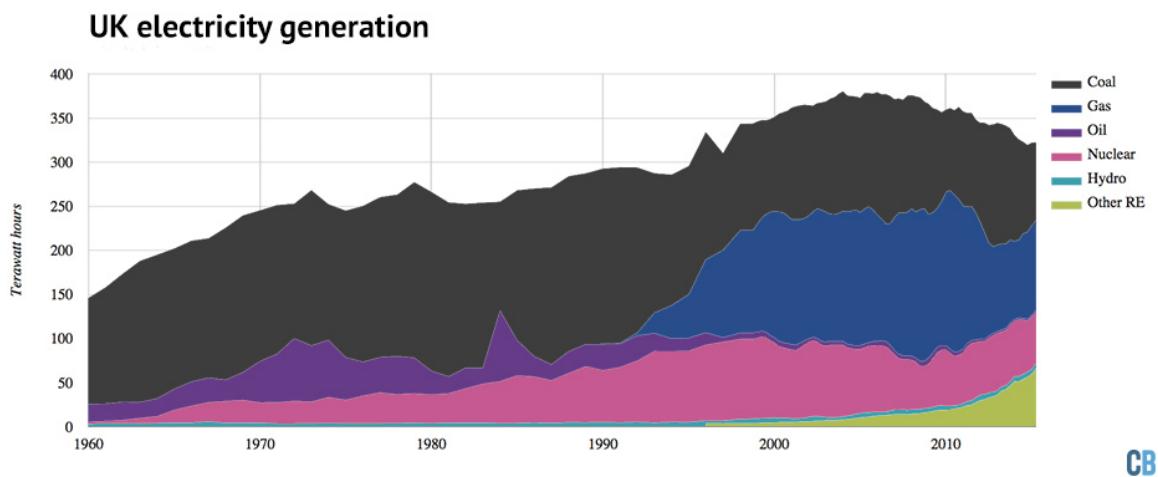


Figure 2.5: UK electricity generation by fuel type. Source:[SE15]

2 Related Work and Foundation

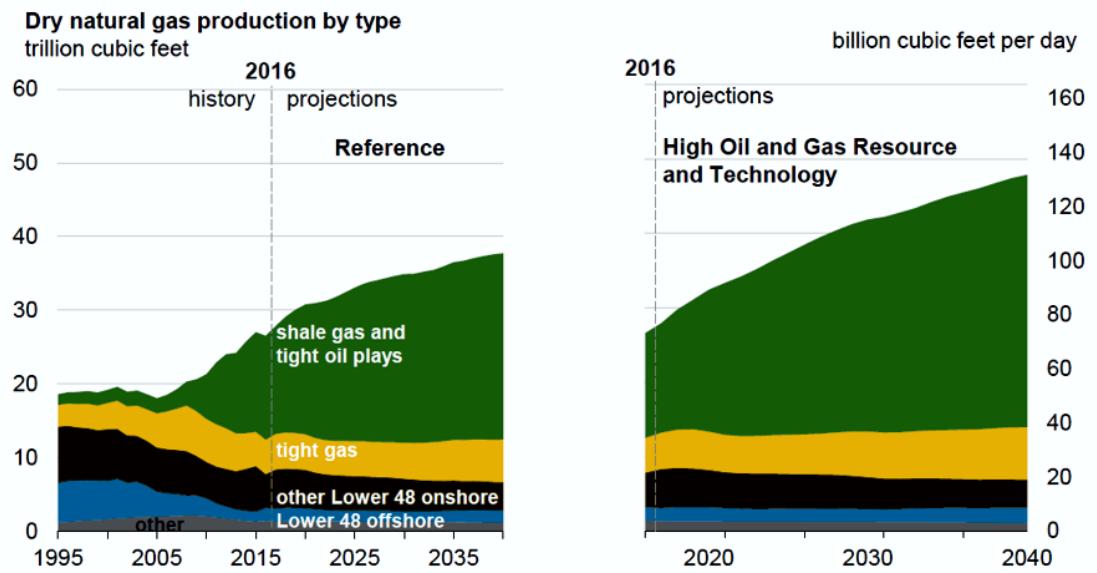


Figure 2.6: U.S Dry natural gas production by type. Source:[Eia17]

3 Background of This Project

3.1 Concept

The main objective of this project was to make an interactive map with OpenStreetMap to visualize the location of the conventional and renewable power plants. In addition providing the type, production, status and other additional information, which might be interesting for the users. A convenient navigation menu needs to be added to manipulate the data set. Markers dedicated for each power plant would be intuitive. Information for power plants such as the name of the plants, start of operation, fuel type, installed power, grid operator should be displayed in a window or tooltip. Additionally, the high voltage line should be displayed on the map. In our case, three voltages (110kV, 220kV, and 380kV) power lines are displayed in three different unique colors. All these items need to be included in the selection mechanism. The user can select or deselect multiple power plants as well as the power lines according to their interest. As mentioned, information can be extracted from the map with just a few clicks of the mouse. Finally, a tight linking between the 2D graphical map and the temporal plots on www.energy-charts.de must be established. This will help the user to see the hourly production for each power plant of different sources. At the end, this interactive map needs to be merged with the www.energy-charts.de.

Therefore, the creation of this web application was divided into some smaller parts. First, selection of geographical information visualization tool that supports the web environment. Formulation of data in a structured way that supports the visualization tool and web environment, in our case it is JSON. Second, filtering and categorizing the different power plants and power lines from the available data sources. Create a selection mechanism for each category. Third, to map each power plant and power line according to the category and select unique colors for them. Finally, render everything on the map.

3.2 Tools

Our online visualization tool is based on JavaScript. For visualizing geographical data Leaflet.js¹ version 0.7.7 framework is used. Some additional plugins are used to enhance the performance of the Leaflet map. They are available on the Leaflet website. All the plugins that have been used in the framework will be discussed in Section 3.4. JSON Data files of power plants are stored on the FTP server.

3.3 Leaflet

Leaflet is an open source JavaScript library for building interactive maps. It is relatively new as it was first released in 2011. It is popular for its flexibility and is used by renowned websites, like Pinterest, Flickr, The Washington Post and Foursquare. It has all features developers need to make an interactive map. It can handle various basic tasks mouse interaction and different map layers can be used. It is easy to extend with a variety of plugins which are available on its web page and plenty of opportunities to extend the basic functionality. It works efficiently across all the browsers as well as mobile platforms.

The map is rendered using tiled layers. Its basic display is implemented by one default basemap. Its build in capabilities enables creating thematic map layers for JSON and GeoJSON data. It also has the ability to draw points, circles, polylines, polygons, custom markers and styling these features dynamically. Among these features, custom markers are used for locating power plants and polylines are used for rendering the power plants on the map. Leaflet also supports Web Map Service(WMS), Vector layers and Tile layers. Power lines on the map are rendered as a vector layers in our framework. Additionally, different tile layers are available along with OpenStreetMap in our framework like streetview, grayscale and, outdoor. Leaflet has a controller that allows the users to see different tile layers as a base layer of the map. It also provides the functionality to add popups to the markers, overlay lines, and shapes. It is extremely light weight and there are no dependencies for this open source library.

The purpose of using this library is because it is widely used and open source. It supports desktop as well as modern mobile devices. As a whole it is smaller in size, and it takes advantage of the new features of JavaScript and HTML. It is very easy to use and its API is easy to understand for performing common mapping tasks like changing base map, zooming, panning. Leaflet also provides a nice documentation which makes a low barrier to develop new applications.

¹Official Leaflet.js Website, <http://leafletjs.com/> (last accessed on February 24, 2017)

3.4 Other Libraries and Plugins

The significant strength that made leaflet much more powerful is its ability to extend its classes and functionality with third party plugins. There is a large community behind leaflet. At the time of writing, there are more than 100 plugins available on the official website of Leaflet. A couple of plugins are also used in this tool which has a great impact on the performance. The plugins used for our visualization tool are mentioned below.

Plugins for Leaflet framework:

- **Leaflet Marker Cluster**
- **Leaflet Label**

External plugins:

- **jQuery.js**²
- **Bootstrap**³
- **Underscore.js**⁴

Leaflet Marker Cluster

A very useful plugin for Leaflet is Leaflet.MarkerCluster. It helps to cluster a large number of markers on a map by providing an animated marker. Its clustering functionality for leaflet has a great impact on the performance of interactive map. It takes less processing time to load the data set and finally it helps to make the map look cleaner. The plugin is available for download from the **Leaflet.markercluster**⁵ github page. Its CDN link is also provided, which can be used right after the declaration of Leaflet files in the header of the HTML. By default the plugin is providing some nice functionality such as **showCoverageOnHover**: helps to show the boundary that is been covered by the marker, **zoomToBoundsOnClick**: when a cluster is clicked it zooms to its bounds.

²Official jQuery.js Website, <https://jquery.com/> (last accessed on February 24, 2017)

³Official Bootstrap.js Website, <http://getbootstrap.com/> (last accessed on February 24, 2017)

⁴Official underscore.js Website, <http://underscorejs.org/> (last accessed on February 24, 2017)

⁵Github Leaflet.markercluster Website, <https://github.com/Leaflet/Leaflet.markercluster> (last accessed on February 24, 2017)

3 Background of This Project

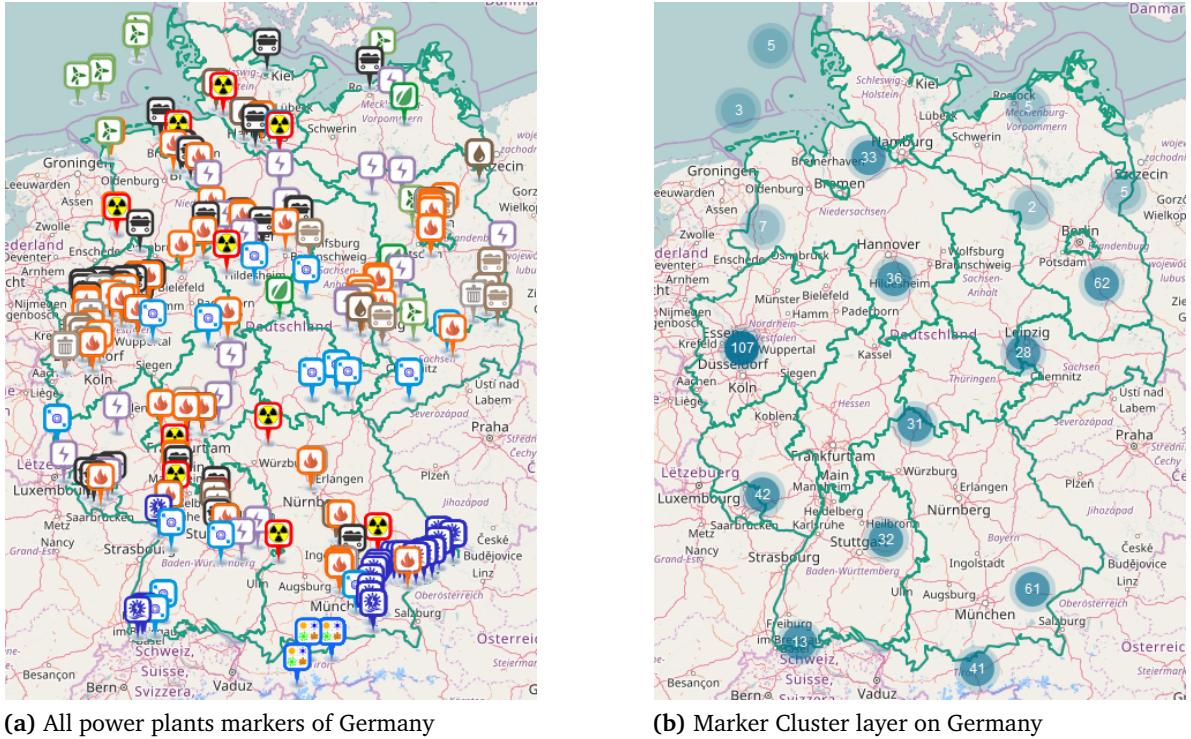


Figure 3.1: Impact of using marker cluster plugin

Figure 3.1a shows all the power plants with pinpoint marker inside Germany and Figure 3.1b shows the impact of using the marker cluster plugin. Clustered animated markers can be customized. Most importantly, it helps to visualize if there are more than one marker rendered in the same geographical location on the map. A cool built in feature spiderfies the child markers of that cluster. Figure 3.2a shows that there are in total 13 power plants located in the same area. Hence, to reduce complexity they are registered on a unique latitude and longitude.

Leaflet Label

Leaflet label is a plugin for adding labels to markers & different shapes such as paths, polygons, and circles on leaflet powered maps. **Leaflet.Label**⁶ development versions are available on GitHub . There is a default style for the label as you can see in the Figure 3.2a that power

⁶Github Leaflet.markercluster Website, <https://github.com/Leaflet/Leaflet.label/tree/0.2.1> (last accessed on February 24, 2017)

plants are located in Nordrhein-Westfalen. When the user hovers the cursor on the displayed objects of the map, this label appears with the data underlying for that particular object.

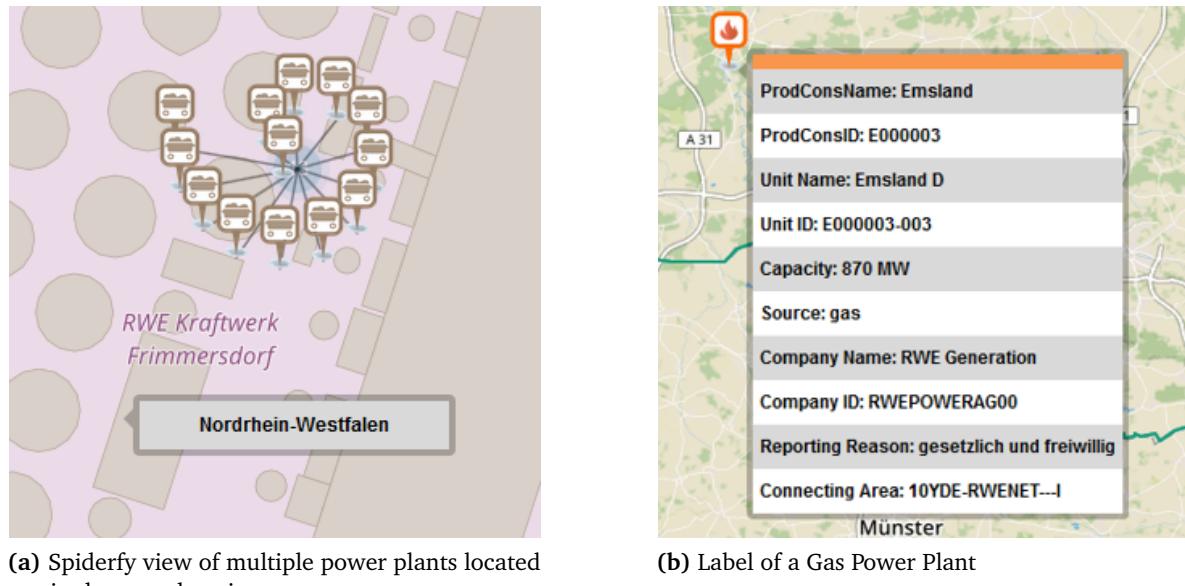


Figure 3.2: Impact of using marker cluster and Leaflet label plugin

jQuery

jQuery is the most popular JavaScript library. Its features are rich. Its syntax is designed to make it easy for HTML document manipulation, handling events, animations, and developing Ajax applications. This free and open-source software enables developers for creating plugins on top of any JavaScript library. In our framework, jQuery is used for DOM element selection and manipulation and for handling events. For example, jQuery is used for the navigation menu in our framework. Our navigation menu is equipped with check-boxes and a drop-down menu. Effect of toggling check-boxes, effect of selecting drop-down menu options, and handling these events on different zoom level are all handled by jQuery. Other applications of jQuery will be discussed in details in Chapter 4

Bootstrap

Bootstrap is well known and free open-source framework. Its popular as front-end framework for designing websites. It is based on HTML & CSS design with JavaScript extensions. It is

dependent of jQuery It provides a responsive structure to the website and compatible with almost all the latest versions of browsers. In our framework, we have used bootstrap to make our website responsive. In particular, it is incorporated for designing the navigation menu of our map. Bootstrap buttons, check boxes and drop-down navigation menus styling provides a great performance on the websites. These components are interactive and made with JavaScript plugins which are bundled in the Bootstrap package. Furthermore, one of the main reasons to use this plugin is, it offers a good documentation for almost every element what a typical web application requires. So its easy to customize and understandable by any web developer. For creating a nice user friendly interface for web applications with less effort, Bootstrap is a great solution.

3.5 Working Environments

Before developing the online visualization tool of German power plants Fraunhofer Institute for Solar Energy System (ISE) was providing interactive charts on electricity production, electricity stock market prices and import/export of electricity at the website Energy-Charts⁷. This whole framework is developed based on JavaScript libraries. For visualizing data, bar chart, ThemeRiver, stacked Multi-Bar chart, and Pie chart are used. To make the chart interactive D3.js⁸ and NVD3.js⁹ JavaScript libraries are taken into account. Electricity production of power plants, stock market prices, and import/export data are stored in the web server. There is no database for storing all these data. Therefore, the data used for our visualization tool is also stored in the web server. Other required JavaScript libraries, plugins, dependencies are stored in the webserver of Energy-Charts.

⁷Energy-Chart official webpage, <http://www.energy-charts.de/> (last accessed on February 24, 2017)

⁸d3.js official website, <https://d3js.org/> (last accessed on February 24, 2017)

⁹NVD3.js official website, <http://nvd3.org/> (last accessed on February 24, 2017)

Listing 3.1 An example of JSON-object for Brown Coal(lignite) power plant

```
1 var Power_Plants = [
2   {
3     "ProdConsName" : "Frimmersdorf",
4     "ProdConsID" : "E000005",
5     "UnitName" : "Frimmersdorf P",
6     "UnitID" : "E000005-012",
7     "Source" : "lignite",
8     "color" : "rgb(150,125,100)" ,
9     "Capacity" : 289,
10    "CompanyName" : "RWE Generation",
11    "CompanyID" : "RWEPOWERAG00",
12    "ReportingAvailableCapacity" : "True",
13    "WGS84Latitude" : 51.05629,
14    "WGS84Longitude" : 6.57703,
15    "Country" : "DE",
16    "ReportingReason" : "gesetzlich und freiwillig",
17    "ConnectingArea" : "10YDE-RWENET---I",
18    "TSO" : "Amprion GmbH",
19    "Commercialisation" : 1,
20    "StartDate" : "2009-10-26T00:00:00+01:00",
21    "EndDate" : "2017-09-30T00:00:00+02:00"
22  },
23  {....},
24  {....}
25 ]
```

3.6 File Format

All the data sources that have been used for power plant visualization and their production in the enrgy charts are in the JSON¹⁰(JavaScript Object Notation) format. This is data format is language-independent and use the extension .json. The MasterData power file is received in CSV (Comma-seperated values) format which is downloaded from the EEX¹¹ server. Fraunhofer ISE is authorized to download the files from the FTP server in csv format and later it is converted into JSON data. List 3.1 shows an example how power plant files are structured in JSON. As we are using JavaScript libraries like Leaflet and NVD3, both support JSON data as input. It is easier to extract information from a large file if data and features are organized in such structured way.

¹⁰JSON official website, <https://json.org/> (last accessed on February 24, 2017)

¹¹European Energy Exchange, <https://www.eex.com> (last accessed on February 24, 2017)

3 Background of This Project

Listing 3.2 An example GeoJSON-object for 110kV power line inside Germany

```
1 var _110KV_layer = [
2   {
3     "type": "Feature",
4     "id": "way/18957128",
5     "properties": {
6       "@id": "way/18957128",
7       "cables": "3",
8       "frequency": "50",
9       "power": "line",
10      "voltage": "110000"
11    },
12    "geometry": {
13      "type": "LineString",
14      "coordinates": [
15        [
16          [
17            [
18              [
19                [
20                  [
21                    [
22                      [
23                        [
24                          [
25                            [
26                              [
27                                [
28                                ]
29                              ]
30                            ]
31                          ]
32                        ]
33                      ]
34                    ]
35                  ]
36                ]
37              ]
38            ]
39          ]
40        ]
41      ]
42    }
43  ]
44]
```

Another format that has been used for rendering power lines of Germany on the map is GeoJSON¹². It is a popular data format among GIS (Geographic Information System) technology and services. Leaflet is very good at handling this GeoJSON format. Power lines inside Germany are created from GeoJSON objects. According to [HB08], GeoJSON object is a format for encoding a variety of geographic data structures in the form of geometry and collection of features. GeoJSON supports points, LineString, Polygon, Multipoint, MultiLineString, Multipolygon. Features in GeoJSON contain a geometry object and additional properties and a feature collection represents a list of features. In the list 3.2, an example of GeoJSON feature is shown for 110kV power line. These GeoJSON objects are added to the map through creating a GeoJSON layer. The line feature showed in the list 3.2 belongs to the 110kV layer. Its geometry type is LineString and coordinates are given under the type of geometry. Leaflet creates a GeoJSON layer and adds it to the map.

¹²GeoJSON official website, <https://geojson.org/> (last accessed on February 24, 2017)

3.6.1 Advantage

JSON is language independent and programming language model data interchange format [Cro06]. This is one of the main reasons for using JSON. It provides a compatibility library for JavaScript, which really supports for reading JSON files using JSON.parse. It is also compatible with browsers and web applications. JSON has a natural way of evolving the data and information. It is very simpler to work with in conjunction with the browser scripting. JSON can be imported into any website, smaller and can represent all Unicode characters. It's also trivial to use it in AJAX applications.

On the other hand, GeoJSON is also a data interchange format. It has a standard way of passing information. Leaflet can adopt it well and its performance is very good compare to another available format. It doesn't matter from where it is generated. As long as it is GeoJSON, users can render power lines on the map. It is also easily explorable because it's a regular JavaScript object. This makes the handling easy of large files. Filtering, Styling and then rendering is easy on Leaflet using this file type.

3.6.2 Disadvantage

JSON is hard to read by a human and it is confusing. Its enormous number of braces, brackets, and commas made it almost unreadable. Therefore, sometimes it is difficult to generate the proper command for parsing the target object from a complex JSON file. It is also not suitable for Big data. An article [Zei12] has described how big is too big for JSON. Test results show that JSON file larger than 15MB is barely usable in modern browsers. Apparently, in our case, there was no data base available at the time of working. Hence, data are stored on the web server in the format of JSON or GeoJSON. Particularly, GeoJSON data of power lines are larger in size. Therefore, this large data file causes longer execution time for rendering. As a result, the browser crashes sometimes and performance of the map is affected after rendering a lot of polylines on the map. On the other hand, user would never be interested in downloading a large file which consumes so much data, specially on the mobile devices with such limited bandwidth.

3.7 Power line(GeoJSON) File Source

In this section, the source of GeoJSON data for power lines and how it is generated using Overpass turbo are discussed. Data for power lines are extracted from the overpass-turbo¹³ API. Overpass API acts like a database of OSM. It also offers search possibilities of power lines.

¹³Overpass turbo API, <https://overpass-turbo.eu/> (last accessed on February 24, 2017)

3 Background of This Project

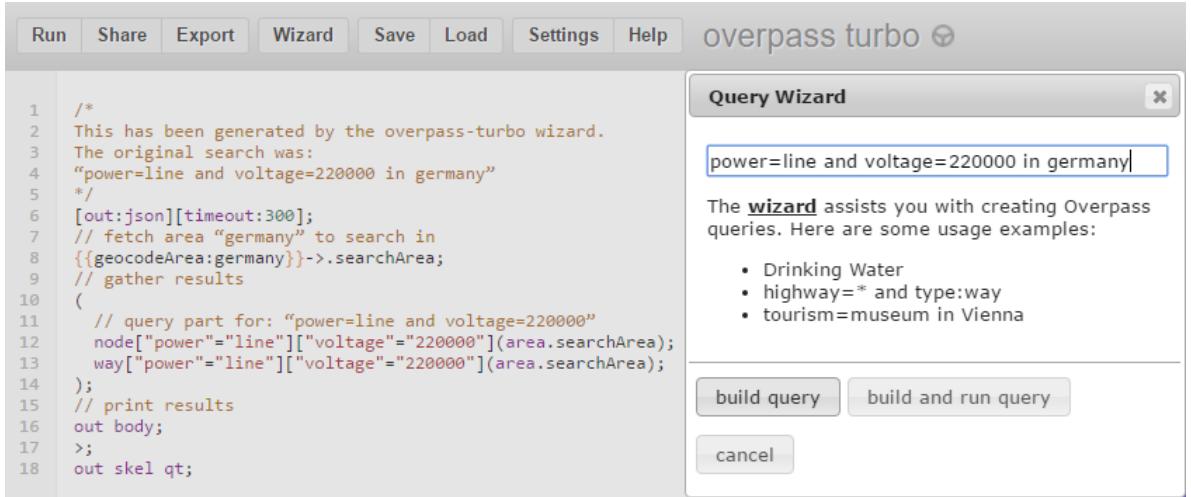


Figure 3.3: Overpass turbo query wizard for generating query

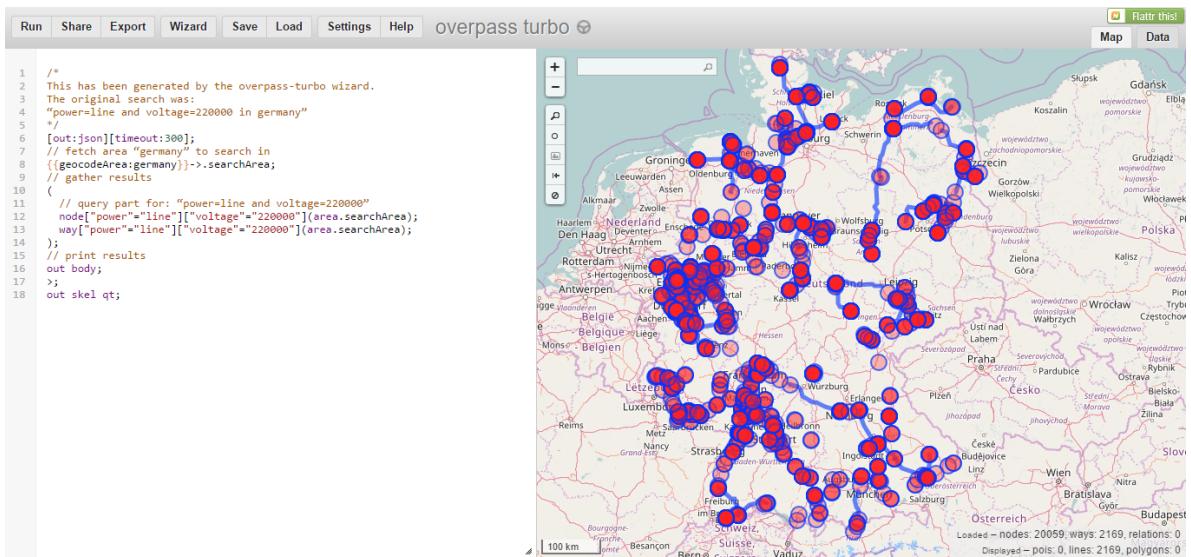


Figure 3.4: Overpass turbo API query (left) with rendered vector layer of 220kV power line (right) on the Openstreetmap

For querying power lines a unique tag value must be used. A tag consists of two parts, one is a key and another is value. They are used together separated by an equal. Overpass API uses the tag and serves the part of the OSM map and render it on the map. Therefore, Overpass turbo is used to run the overpass API query. There is a query wizard which assists one generating proper overpass queries. An example is shown in the Figure 3.3. To get power line for 220kV,

one have to fire up the wizard and enter `power=line` and `voltage=220000`. Here "power" is a key and "line" is a value of the tag. In addition, fetch area can be selected by simply typing the name of the country inside two curly brackets.

After the execution of the query, filtered results are rendered on the OSM. There is an option for exporting rendered data. We exported the data as GeoJSON and used it in our framework. This operation is done twice for exporting the GeoJSON of other two (110kV and 380kV) available power lines. The structure of this GeoJSON is already discussed on the list 3.2.

4 Software System

The online visualization toolkit is implemented as an additional tool for accessing plant data on Energy-Charts. Energy-Charts is build using JavaScript libraries. Therefore, same, or corresponding technologies were used for our interactive visualization map.

The vision of this project is to provide an interactive map for users who are interested in Germany's electricity production. This visualization tool provides multiple possibilities to explore the Germany's electricity generation and its distribution. First, users can get an overview of power plants. It shows the location, operating status, and generation capacity of all renewable and conventional power plants in Germany. Second, this interactive map is associated with ThemeRiver or stacked area charts where users can see an hourly production of electricity for each power plant. Finally, the tool also visualizes the power transmission lines (110kV, 220kV, and 380kV) of Germany. Furthermore, users can select multiple power plants and list them in a table chart for comparison. It is also possible to compare those selected units and based on their hourly production with Energy-Charts.

4.1 Requirement Analysis

As mentioned before, to implement Energy Chart JavaScript library is used. Plant data are stored as JSON files on the server. Hence, for creating an interactive map as an extension to the Energy Chart and to match the functionality with the existing charts, JavaScript mapping library is also used for this project. Our goal was to create an interactive map which is dedicated to the power plants of Germany. Before implementing the tool, the basic must-have components for the map must be specified. Therefore, following things were considered as a list of requirements for the map to enable a quick exchange of information between users and the map.

System Requirements

- Need to develop an interactive map with OpenStreetMap open source JavaScript libraries.
- For having no data base, our tool must be able to access JSON files from the server using AJAX query and deliver it to the map.

- The interactive map should run on web platform, support web technologies and usable on modern smart devices.

Interface Requirements

- To visualize the location of the plants, glyph based visualization technique can be used. A meaningful map marker must be used for each fuel source category. A fixed color code is assigned to each source category.
- In the case of mouseover or clicking event a marker, window, or tool-tip must appear displaying the basic information of the power plants. Displayed information needs to be extracted from given JSON files.
- In addition, the high voltage power lines should be displayed on the map for three voltages 110kV, 220kV, and 380kV.
- To complete the interface, a simple, clean, and effective navigation menu must be designed and should be placed above the map to display or hide different types of power plants and high voltage power lines.

Functional Requirements

- Internal communication between the interactive map and Energy-Charts must be established. Thus, the users can query plant data, see their hourly production, and can make a comparison.
- The map and energy charts have to be linked. Therefore every selection is reflected in both and that people should be able to chose individual power plants.
- Navigation menu and its utility should offer adequate possibility to allow user for controlling the view on the map API

Therefore, to achieve and implement the above-mentioned requirements we decided to use a simple lightweight JavaScript mapping library – Leaflet. Finally, we end up with a fully-clickable interactive map as a composite image consisting of a map, location markers and poly lines.

In the following, we will give an overview of the architecture and explain the graphical user interface of the visualization tool.

4.2 Architecture

The diagram in Figure 4.1 shows the architecture of the complete interactive map and its connection between server-side and client-side.

Back-end is developed based on JavaScript along with the Leaflet. Additionally, jQuery and underscorejs are used to eliminate the necessity to reprogram the basic operation and functionality. Multimedia contents and JSON files are loaded on the FTP server.

Initially, leaflet and additional JavaScript code for creating the map API are executed on the page load. By default, JavaScript code makes request to the file server over HTTP and get JSON files as a response. Location of all power plants are preserved in that JSON file. Leaflet parses the JSON file and render the power plants on the map. During this operation program also set the marker properties. Each source category has a unique marker image which are also stored in the server. At the end users get an interactive interface and can see the map on the screen with some colorful markers of power plants locating its location on the map.

There is an individual script that executes after page load for establishing the connection between front-end and back-end. All the JavaScript events for navigation menu, buttons and map zoom-in/out events are registered there. The large GeoJSON files for high voltage power lines are not initially loaded. User needs to use the navigation menu for sending a new HTTP request to see them on the map. Server identifies the path of the requested file through scripts and returns the GeoJSON file to the client side. Finally leaflet parses the GeoJSON data and render it to the map.

4.3 User Interface

Based on the back-end the front-end is entirely developed in HTML, CSS, and JavaScript using the existing template for Energy Charts. Bootstrap design system was chosen for interface(e.g navigation menu, buttons and drop down) design and making the web page responsive. Furthermore, the user interface of interactive map is designed for desktop computers as well as smart mobile devices. In the following, the main structure of the graphical user interface containing the navigation menu, map API as well as the comparison chart will be presented. The main structure is shown in the Figure 4.2. As this interactive map is integrated into the Energy-Charts page, therefore the same template is used for the header section. Our graphical UI starts right after the navigation menu of the Energy-Charts page. Our interactive visualization tool starts with a navigation menu which can also be called as a control layer for the map. Check-boxes are used for the navigation menu area to externally control the map. They are decorated with unique colors. These colors are matched with the colors that have been assigned to each source category in the JSON file. The map is located under the control layer where markers, power lines, and their basic information are displayed. An area

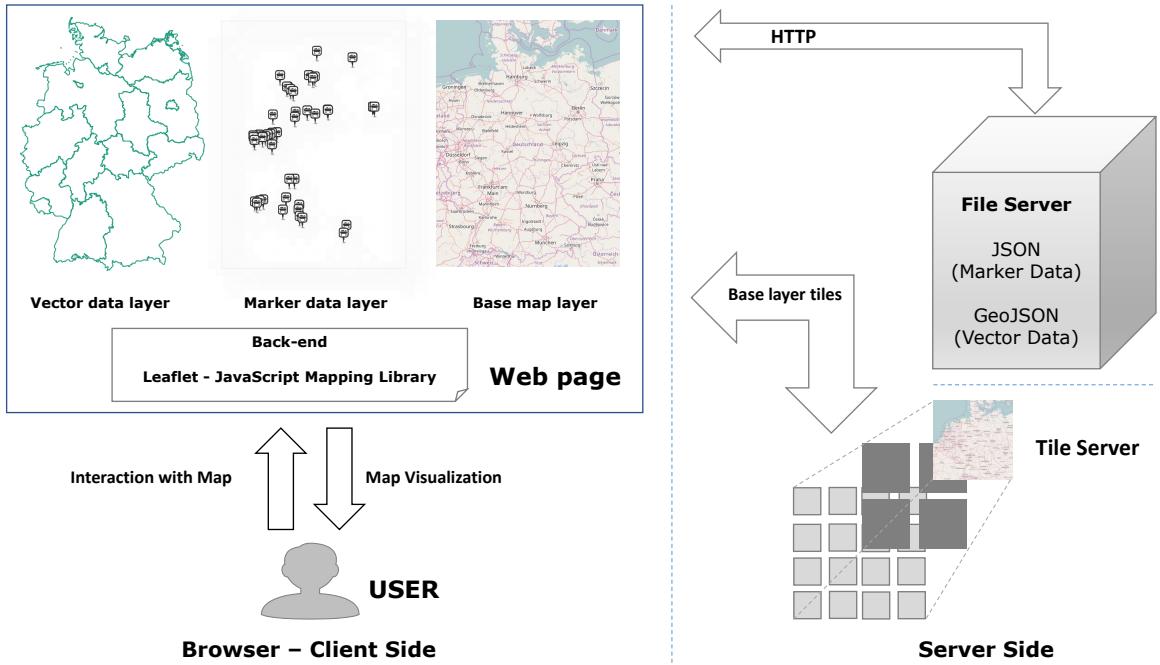


Figure 4.1: The architecture of the visualization tool - connection between back end and front end

for comparison list is allocated on the web page. Comparison table chart only appears when a user selects a power plant for comparison. Depending on the device, the appearance of the comparison list may change. Hence, our online interactive data mapping tool consists of three main parts:

- Control Layer: Navigation menu with two subsections
- Map API: Cluster layer, markers, high voltage power lines, legends and basic information in the pop-up box
- Comparison List: Comparison table chart for comparing power generation of the power plants that fall under same source category.

In the following these parts are particularized in details.

4.3.1 Control layer

Control layer provides the functionality to manage the displayed content on the map. Checkboxes are used to control the map using external JavaScript events. It is possible to design

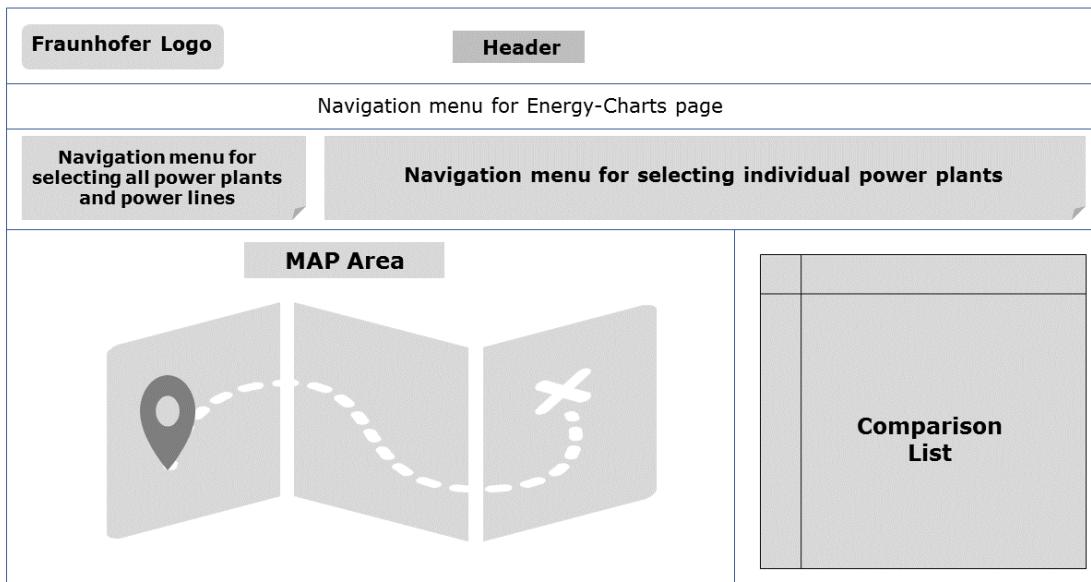


Figure 4.2: The main structure of the visualization page

these boxes in different ways. This circular design is inspired by the legends of energy power chart. For this reason, we designed this external navigation menu for controlling the map. Control layer is divided into two sections (See Figure 4.3).

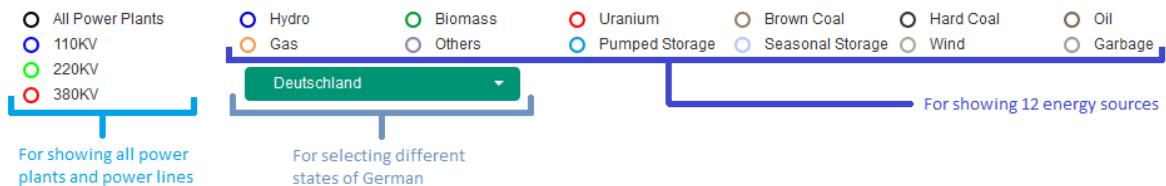


Figure 4.3: Navigation menu

Check-boxes for selecting all power plants at once, and for high voltage power lines are located on the left side of the navigation menu. In the remaining space, check-boxes for selecting individual energy sources category are placed on the right side of the navigation menu. As provided in the JSON data, there are 12 energy source categories are provided in the navigation menu. Check-boxes are decorated with a circle with a border and a label next to the circle. A tool tip appears under it if the user hovers the mouse over the check-box (see Figure 4.4). Check-box area includes the circular check-box and its label.

4 Software System



Figure 4.4: Tool tip for Check-boxes

A drop-down selection menu is designed to locate and select 16 different states of Germany on the map.

Control Layer Properties

Control layer is an essential part of the interactive map. To perform an action, the user needs to select check-boxes. When the status of the check-box is not selected or unchecked, its appears as an empty white circle with a colored border. On selection, it is filled with its previously assigned border color (see Figure 4.5) and reveals the power plant on the map. For example, to show all power plant on the map the user must click on the check-box labeled with “All Power Plants”. This action triggers all 12 check-boxes of power plants and fills the circles with colors (see Figure 4.6). While continuing this operation if any one of the power plants is unchecked or deselected, then “All Power Plants” is automatically deselected. On the other hand, if all 12 check-boxes are selected manually, this action also selects the check-box of “All Power Plants”. Therefore, based on selection the content of the map will be updated. Same properties and behaviors are also applicable for individual high voltage power lines. To see the power lines on the map, the user must select the check-boxes assigned for each power line.



Figure 4.5: Navigation menu with all power plant selected.



Figure 4.6: Navigation menu where one energy-source is deselected.

The drop-down selection mechanism for German states is also a part of the control layer. It was added to give users an opportunity to control the map and find different states inside

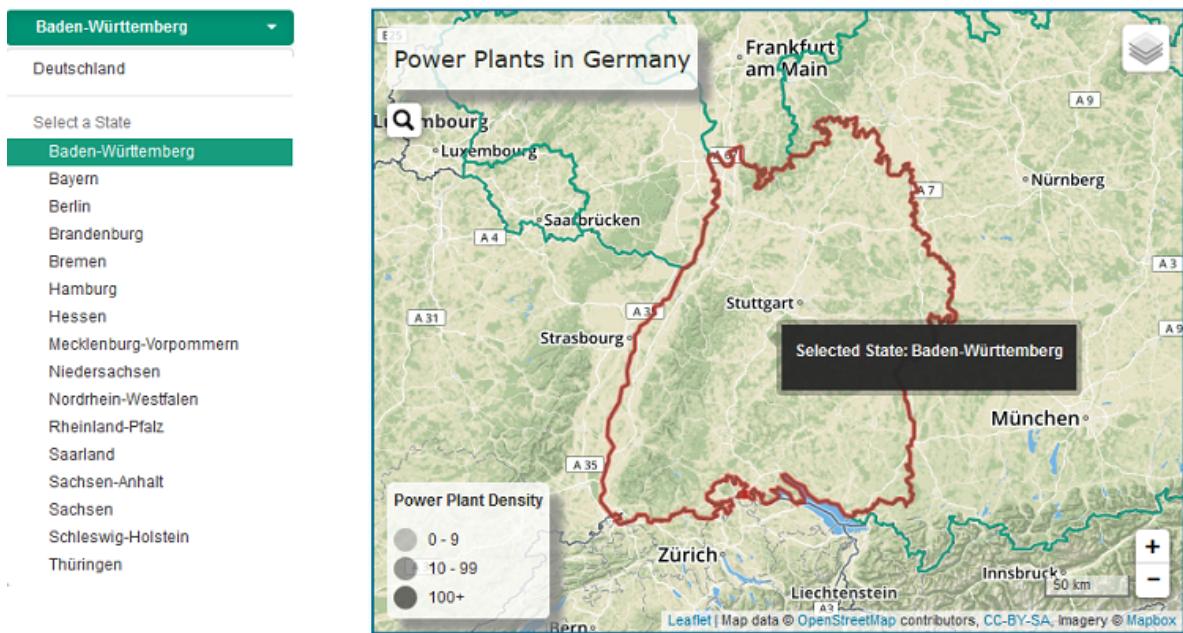


Figure 4.7: A German state (Baden-Württemberg) is highlighted, when it is selected from the drop-down navigation menu

Germany. Users can choose between 16 states from the menu for zooming in and to have the local view of that particular state. This animated zoom-in option uses a feature of Leaflet to get the bounds of the selected state data layer and fits on the map with the maximum possible zoom level. This will help the user to get an idea about the power plant density inside the selected state of Germany. The user needs to select “Deutschland” to restore the global view of the German map. Figure 4.7 shows an example of state selection mechanism. On the selection of Baden-Württemberg from the drop-down list, it is highlighted, and map zoomed in to fit it on the map screen.

4.3.2 Map API

The interactive map appears under the navigation menu on the web page. Our initial view of the map, when the page opens, is illustrated in Figure 4.8. After initializing the map, the default view of the map is set to Germany with a zoom level 6. Therefore, the entire map of Germany is visible in the map area. The area is equipped with legends, zoom controller, and map scale controller. The custom control layer is collapsed and placed on the top-right corner of the map. Beside these, another information layer is added by default to highlight the border of Germany and its states. This vector data layer is a collection of polygons and

4 Software System

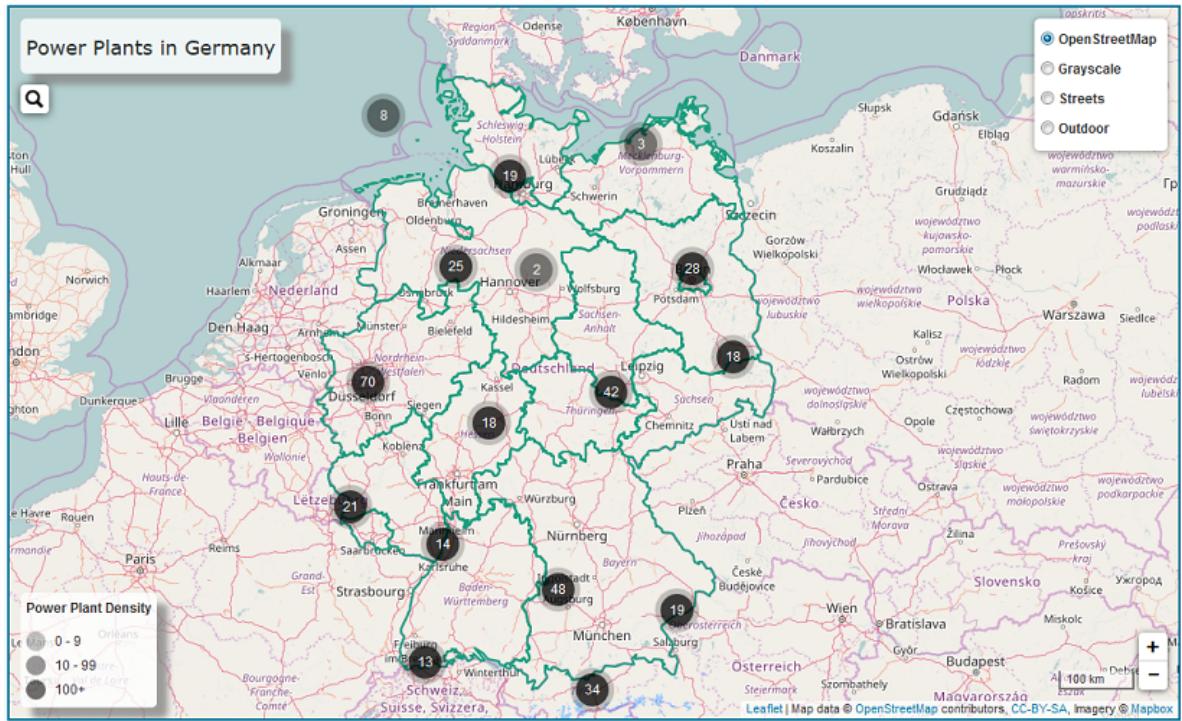


Figure 4.8: Interactive map area equipped with legends on (top-left and bottom-left), initial cluster view, zooming and map scale control (bottom-right), different base map layer controller (top-right)

stored as GeoJSON on the server. This GeoJSON was extracted from GitHub¹. In addition, a cluster view of all markers was created using Leaflet.MarkerCluster plugin and was added to the map. This cluster view could help the user to understand the density of power plants of a particular area. The number written inside of a cluster represents the total number of markers contained under this. Cluster layer is filled with black color and its opacity depends on the density of the markers. A custom control legend is added to the bottom-left corner of the map area describing the color scale. When user mouse over a cluster it shows the bound of its markers and if the user clicks on a cluster it zooms to its bounds. As discussed in Chapter 3 Section 3.4, this plugin is also helpful to find if there are more than one plant located at the same position on the map.

¹Databundeslander, <https://gist.github.com/oscar6echo/4423770> (last accessed on February 24, 2017)

4.3.2.1 Functionality

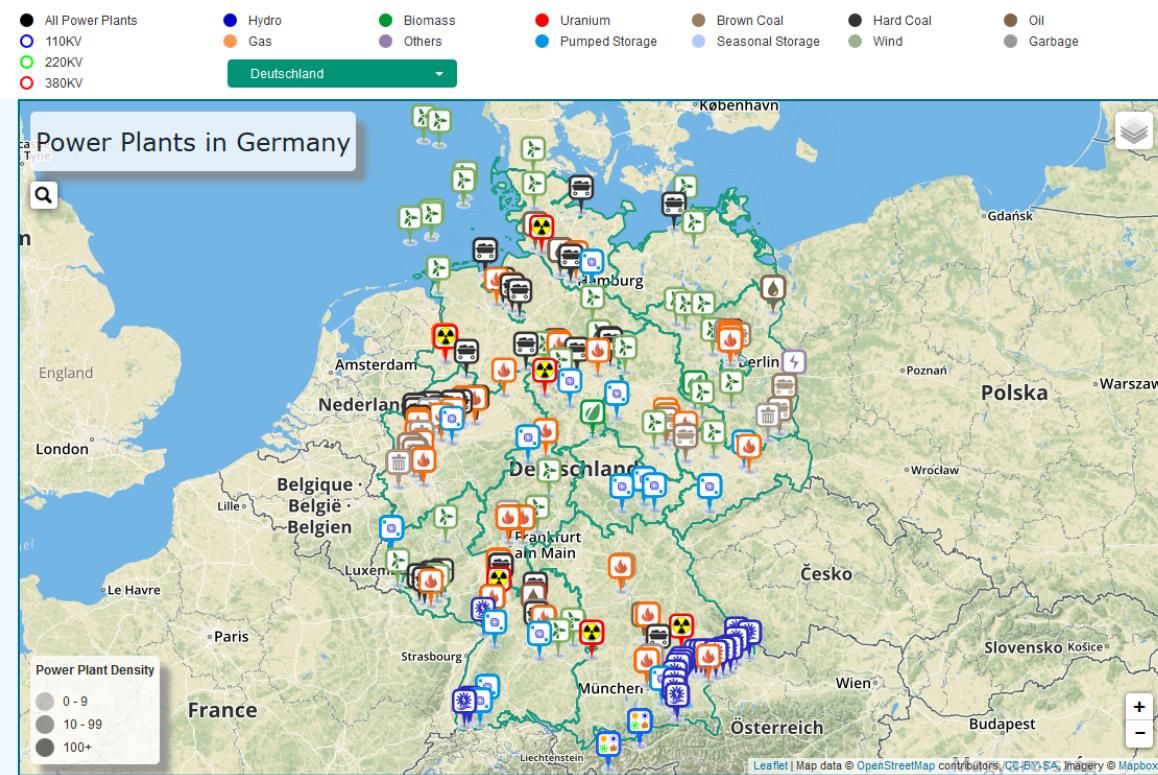


Figure 4.9: Showing the geographical location of all power plants with markers.

In this section, the functional aspects of our visualization tool in the combination with our control layer are discussed.

As discussed in Section 4.3.2, by default a cluster view of all power plants are rendered on the map. This is added as a layer on the map. Cluster view interprets that there are no check-box selected or no action was performed externally on the map. From this position, users can zoom in by clicking the zoom-in button or even by scrolling the mouse. Markers smoothly split from the cluster on zoom-in and merge to its cluster on zoom-out. If any check-box is selected, this action will remove the cluster layer from the map. In the case of viewing all power plants on the map, the user needs to select “All Power Plant” and as a result, all the markers are visible on the map (see Figure 4.9). For Each energy source category, a unique color and marker logo are assigned. At the time of reading the JSON data, markers are categorized and stored in groups. Each object of layer group is rendered on the map as a check-box input. For example, to see all nuclear power plants, the user needs to select the check-box labeled “Uranium”. In that case, only the nuclear marker layer object is added to the map (See Figure 4.10. Here,

4 Software System

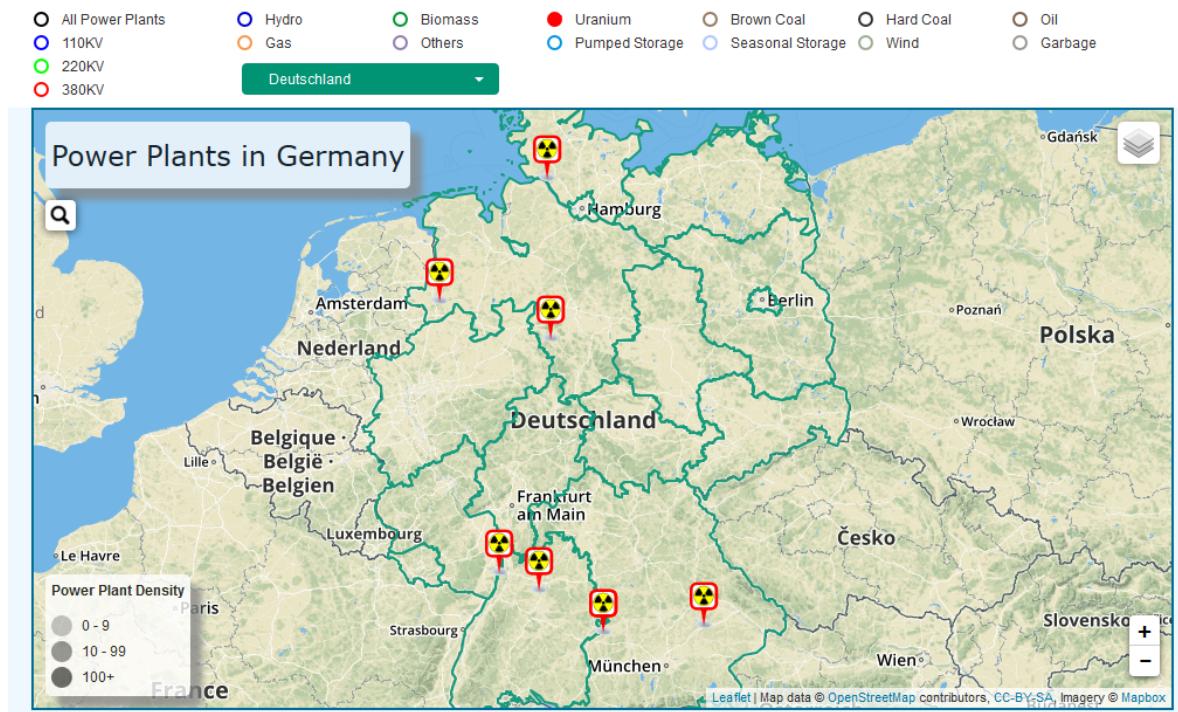


Figure 4.10: Showing all nuclear power plants.

jQuery is used for targeting the html DOM (Data Object Model) input check-box object and leaflet functionality is used to add it to the map.

Along with the geographical location representation, information about each power plants have also been attached to its location marker. Every time a user clicks on a marker a pop-up is attached to it with a specified HTML content and finally, the pop-up appears on the center of the map. We have used a custom design for the pop-up. It is a table with details on the power plant, two buttons, and colored border. Figure 4.11 shows two different pop-up boxes for two different power plant categories. It is clearly visible that they are similar in structure but not in color. The color of the border and the button varies for different categories. This pop-up box is used for displaying the basic information of power plants from the JSON. Basic information includes unit name, unit id, source, capacity, company name, start date and some other additional information.

4.3.2.2 Connectivity between map and energy-charts

In this section, the connectivity between interactive map and energy chart is discussed. This was one of the main functional requirements for our interactive map application.

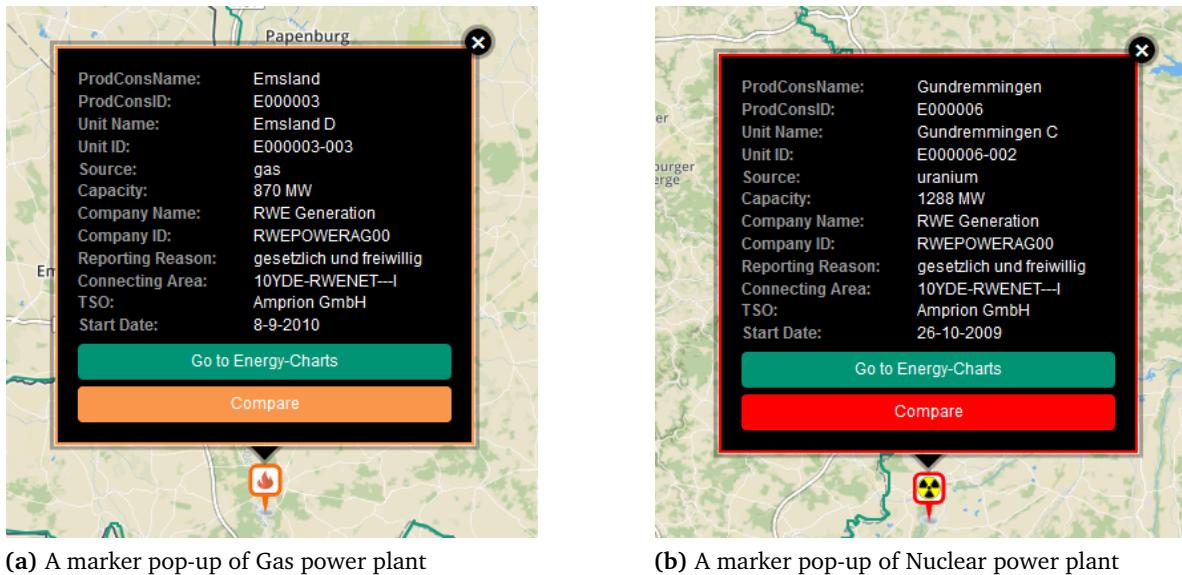


Figure 4.11: Marker Pop-up box

Our goal was to access and view the hourly electricity production data of each power plant from the interactive map. Therefore, an internal connection is established. Users can select power plants and send a query to the energy charts to see their production. If the data of that specific power plant unit is available on the server, then hourly production of that power plant will be illustrated on energy charts. Users are informed over a browser pop-up window on the energy chart page if there is not data available.

The interactive map provides two possibilities to access power plant data on the energy charts. First approach, using the “Go to Energy-Charts” button, which acts as a link (see Figure 4.12). Users can send a query by clicking on the “Go to Energy-Charts” button to directly view the time line visualization of electricity production. Each of this button has a unique link for sending queries. These links are dynamically generated while inserting the basic information of power plants in the pop-up box. This link delivers information to the chart. These information are *source* and *ID*. Each power plant has a unique unit id and source category, which are provided inside the power plant JSON data as an object. The hourly production JSON data also have the same object for unit id but its source category is specified inside the file name. Therefore, the value of *source* variable is required to query the desired file and the value of *ID* is required to match the desired power plant data. Hence, only one power plant production data is visible on the hourly production chart and others visibility are hidden. Second approach is using the “Compare” button, which leads to the Comparison List. Comparison list and its application for accessing electricity production data is discussed in the Section 4.3.3.

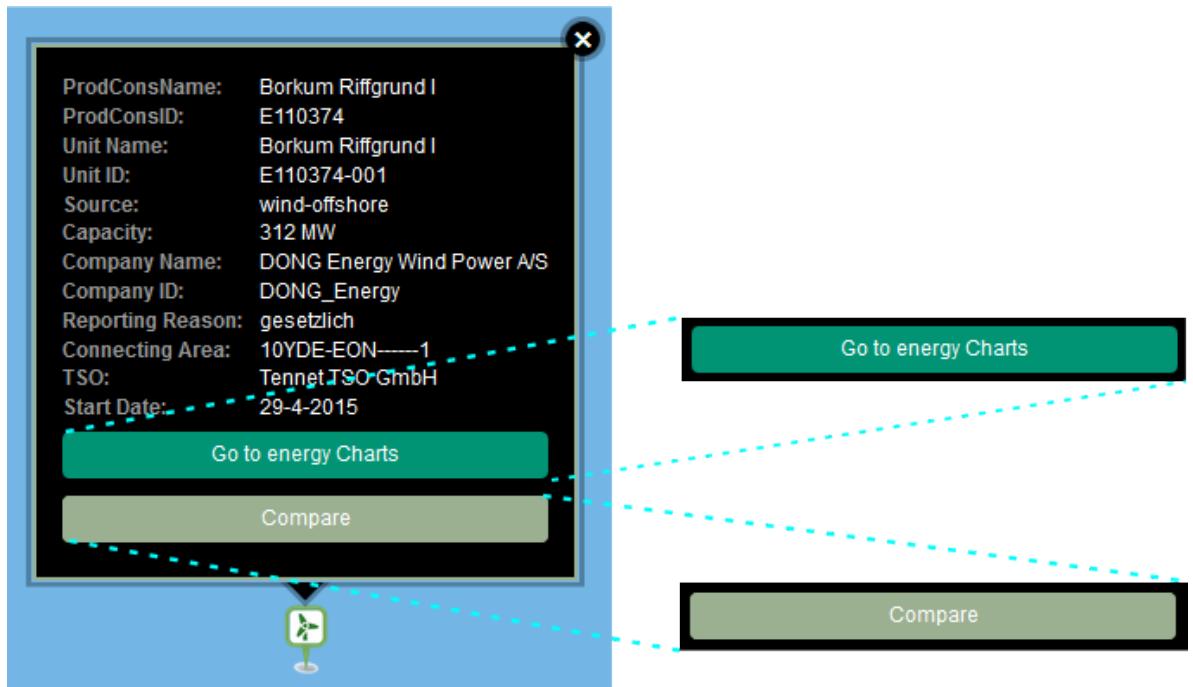


Figure 4.12: *Go to energy charts* button - to see its electricity production on energy chart and *Compare* button - to add this plant to the comparison list

4.3.3 Comparison List

Our interactive map provides an additional feature, where users can compare multiple power plant, and in the end, users can view their hourly electricity generation on the chart. A click on the “Compare” button (see Figure 4.12) generates a comparison list (see Figure 4.13) on the interactive map page. The comparison list will appear next to the map (or below, for mobile devices). Users can dynamically add more power plants to the list. Newly added plants are listed as a new entry on the table. If a plant is added to the table then its “Compare” button is disabled and marked as “Selected”. This function prohibits adding the same element several times to the list. The hourly production data of the power plants in the comparison list can also be shown in the Energy Charts by clicking the “Compare on Energy Charts”. This describes the second of approach for accessing hourly electricity generation data. Like “Go to Energy-Charts” button, “Compare on Energy-Charts” button also acts as a link. This new button also passes information upon clicking it. Unlike the first approach, this link passes the information from the comparison table. Therefore, more than one power plant data is visible on the charts. Figure 4.14 shows the hourly production data for the wind-offshore power plants because only those three power plants are listed in the comparison table. This function runs when users click on the "Compare on Energy Charts" button.

The screenshot shows a user interface titled "Comparison List". Below it is a table with five columns: NAME, UNIT NAME, SOURCE, CAPACITY (MW), and REMOVE. The table contains three rows, each representing a wind-offshore power plant from DONG Energy Wind Power A/S. The first two rows have "Borkum Riffgrund I" and "Gode Wind II" as unit names, while the third has "Gode Wind I". All three units are categorized as "wind-offshore". Their capacities are 312 MW, 252 MW, and 330 MW respectively. Each row features a red "X" button in the "REMOVE" column. Below the table is a green button labeled "Compare on Energy Charts".

NAME	UNIT NAME	SOURCE	CAPACITY (MW)	REMOVE
DONG Energy Wind Power A/S	Borkum Riffgrund I	wind-offshore	312 MW	X
DONG Energy Wind Power A/S	Gode Wind II	wind-offshore	252 MW	X
DONG Energy Wind Power A/S	Gode Wind I	wind-offshore	330 MW	X

Compare on Energy Charts

Figure 4.13: Wind-offshore(3) power plants are listed on the comparison table. Table also includes a button (*Compare on Energy Charts*) to see their hourly production on energy chart.

Only power plants from the same source category can be compared on the comparison list. Users can add as many power plants as needed in the comparison list from different source category but this approach disables the functionality to compare them on energy charts page. For example, if users are interested to view the electricity production of Nuclear(Uranium) power plants then users should include only Nuclear power plants to the comparison list and use the "Compare on Energy Charts" to see their production. Currently, the energy charts can only illustrate the hourly data for a single source category. There is a form, inside the gray area with some radio buttons, for controlling the chart and to see the electricity production on the chart for different sources (see Figure 4.14). Energy charts provides the unit wise electricity production data for Hydro, Nuclear, Brown Coal (lignite), Hard Coal, Gas, Pumped storage, Oil, Wind-offshore, and Wind-onshore power sources. At present there is no data available for Biomass, Seasonal storage, Garbage, and Others source category. Therefore, it is not possible to see their production and compare them on energy charts.

4.4 Data query process on energy charts

In the Section 4.3.2.2, how the interactive map is connected to the energy charts is already discussed. The algorithm used for solving this functional requirement is separated from that discussion and described here in details.

Electricity production from wind offshore in Germany in week 6 2017

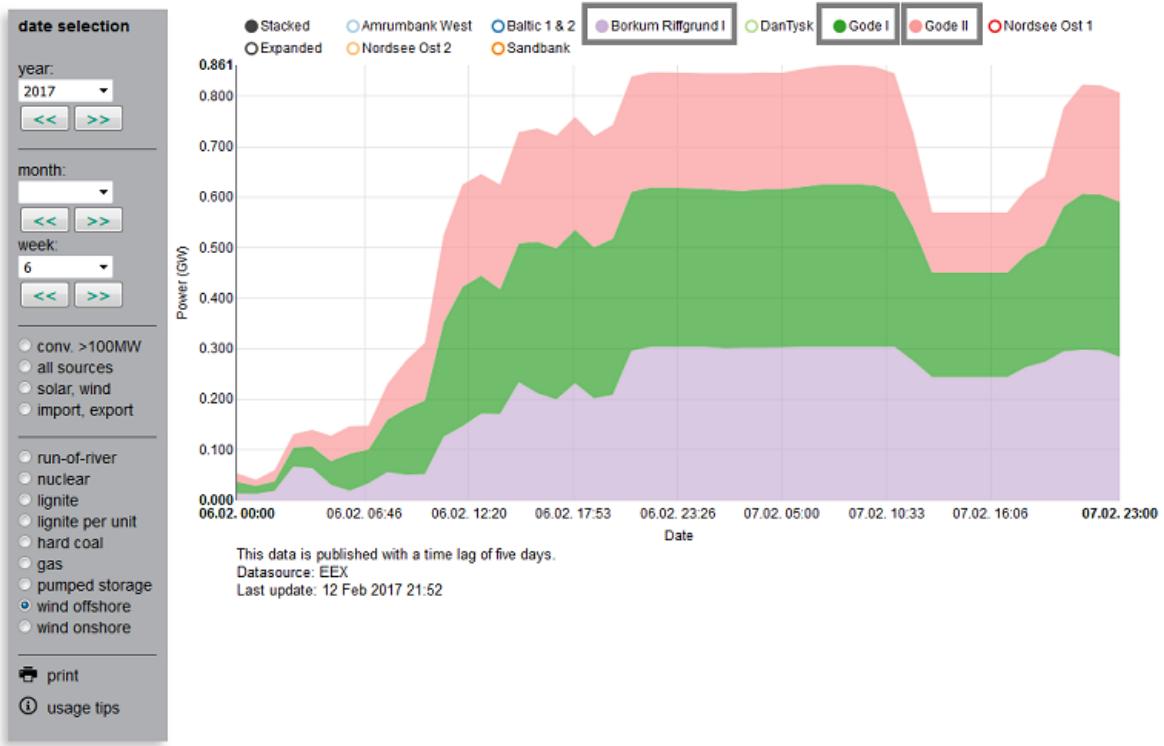


Figure 4.14: The chart is created after clicking the "Compare on Energy charts" button in Figure 4.13. Wind-offshore is selected in the left navigation area (source) and only listed power plants are selected and the rest are deselected.

As mentioned before there are two different ways to access and view the hourly electricity production data. The basic difference between these two approaches is not much. “Go to Energy Charts” button passes only two variables (*source* and *ID*) over URL and direct to energy chart page. Figure 4.15 shows how the underlying link of the button is updated dynamically.

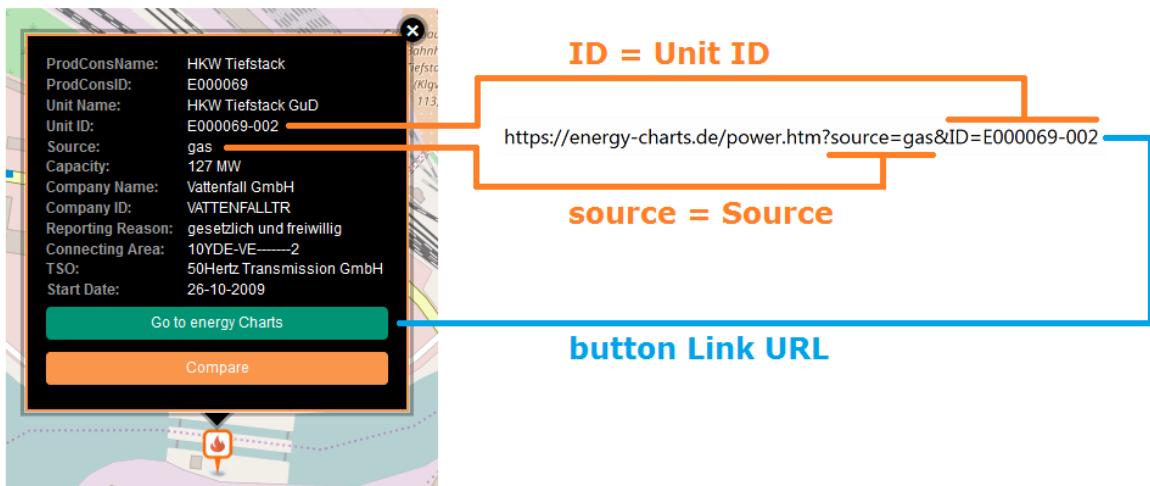


Figure 4.15: Hyperlink of "Go to Energy Charts"

The functionality of "Compare on Energy Charts" button is also discussed before in Section 4.3.2.2. The functionality of this button is to pass one or more unit ids over the URL. When a user adds a new power plant to the comparison list, the unit id of that power plant is added inside the *ID* variable. Consequently, the link of the "Compare on Energy Charts" button is updated in the background. Figure 4.16 shows different steps of adding more than one power plant in the comparison list and how the link is updated in every step. This approach starts when users want to compare one or more than one power plants. For example, a user wants to compare the capacity and hourly production of three Nuclear power plants. In Step 1, user opened a pop-up window of a Nuclear plant and added this to the comparison list by clicking on the "Compare" button. Step 2 and Step 3 can be done in the similar way. Therefore, two more power plants are added to the comparison table. Thus, in total three unit ids are added to the *ID* and the redirect link for "Compare on Energy Charts" is updated as showed in the Figure 4.16.

4 Software System

Step 1: one power plant is selected

The screenshot shows a map interface where a nuclear power plant has been selected. A detailed information box is displayed, listing the following details:

ProdConsName:	Isar 2
ProdConsID:	E000130
Unit Name:	Isar 2
Unit ID:	E000130-001
Source:	uranium
Capacity:	1410 MW
Company Name:	PreussenElektra GmbH
Company ID:	PreussenElektra
Reporting Reason:	gesetzlich
Connecting Area:	10YDE-EON—1
TSO:	Tennet TSO GmbH
Start Date:	10-8-2016

Buttons at the bottom of the box include "Go to energy Charts" and "Selected".

To the right, a "Comparison List" window is shown with the following table:

NAME	UNIT NAME	SOURCE	CAPACITY (MW)	REMOVE
PreussenElektra GmbH	Isar 2	uranium	1410 MW	X

A blue bracket highlights the URL in the browser: <https://energy-charts.de/power.htm?source=uranium&ID=E000130-001>. A callout text "Unit ID copied and inserted" points to the URL.

Step 2: one more is selected to compare

The screenshot shows a map interface where a second nuclear power plant has been selected. A detailed information box is displayed, listing the following details:

ProdConsName:	Gundremmingen
ProdConsID:	E000006
Unit Name:	Gundremmingen C
Unit ID:	E000006-002
Source:	uranium
Capacity:	1288 MW
Company Name:	RWE Generation
Company ID:	RWEPOWERAG00
Reporting Reason:	gesetzlich und freiwillig
Connecting Area:	10YDE-RWENET—1
TSO:	Amprion GmbH
Start Date:	26-10-2009

Buttons at the bottom of the box include "Go to energy Charts" and "Selected".

To the right, a "Comparison List" window is shown with the following table:

NAME	UNIT NAME	SOURCE	CAPACITY (MW)	REMOVE
PreussenElektra GmbH	Isar 2	uranium	1410 MW	X
RWE Generation	Gundremmingen C	uranium	1288 MW	X

A blue bracket highlights the URL in the browser: <https://energy-charts.de/power.htm?source=uranium&ID=E000130-001,E000006-002>. An orange bracket highlights the URL in the browser. A callout text "Unit ID copied and inserted" points to the URL.

Step 3: Two more is selected to compare

The screenshot shows a map interface where a third nuclear power plant has been selected. A detailed information box is displayed, listing the following details:

ProdConsName:	Grohnde
ProdConsID:	E000129
Unit Name:	Grohnde
Unit ID:	E000129-001
Source:	uranium
Capacity:	1360 MW
Company Name:	PreussenElektra GmbH
Company ID:	PreussenElektra
Reporting Reason:	gesetzlich
Connecting Area:	10YDE-EON—1
TSO:	Tennet TSO GmbH
Start Date:	10-8-2016

Buttons at the bottom of the box include "Go to energy Charts" and "Selected".

To the right, a "Comparison List" window is shown with the following table:

NAME	UNIT NAME	SOURCE	CAPACITY (MW)	REMOVE
PreussenElektra GmbH	Isar 2	uranium	1410 MW	X
RWE Generation	Gundremmingen C	uranium	1288 MW	X
PreussenElektra GmbH	Grohnde	uranium	1360 MW	X

A blue bracket highlights the URL in the browser: <https://energy-charts.de/power.htm?source=uranium&ID=E000130-001,E000006-002,E000129-001>. An orange bracket highlights the URL in the browser. A purple bracket highlights the URL in the browser. A callout text "Unit ID copied and inserted" points to the URL.

Figure 4.16: Step 1: a nuclear plant is added into the comparison list - unit id added to the link; Step 2: another nuclear plant is added - new unit id added; Step 3: The last one is added into the list - unit id added and link is updated

Listing 4.1 URL variables are handled using JavaScript

```

1 var _unitID = []; // Empty array for storing ID from the link
2 var _sourceName = getUrlVariables()["source"]; // Variable for storing the source name from
      the URL variables
3
4 _unitID = getUrlVariables()["ID"]; // get the ID/IDs from the link and stores inside this
      array
5
6 if (_sourceName != undefined && _unitID != undefined) { // checks if source and id are
      available inside the url
7
8 var defaultradiob = "disp_"+_new_sourceName; // Selects the radio button for selecting
      source
9
10 var filepath = "./power_unit/week_"+_path_source_name+"_"+defaultyear+"_"+_0"+ defaultweek
      +" .json"; //file on first-load using link variables
11 }

```

Electricity production data are all stored in JSON files. Charts always load the data dynamically and plot these files according to user selection. Every JSON file name is written in several parts. It has a unique source name, week and year. The week and year is manually set when the data for that particular week is available on the server. On having URL variables, energy chart uses these and creates the chart accordingly. URL variables are handled by JavaScript and stored it locally for further operation. “source” variable is used to select the radio button of that particular source category on the navigation menu and also to complete the filepath (see List 4.1). Radio button initializes the filepath using the source name. List 4.1 shows an instance of the first step of handling URL variables. This operation is similar for both approaches.

In the next step, algorithm handles unit ids from the URL. "Go to Energy Charts" link passes only a single unit id and "Compare on Energy Charts" link passes an array of unit ids. Figure 4.16 shows the impact of URL parameters to the electricity production chart. Source name from URL activates the radio button from the chart control area and unit ids are used to enable the visibility of those power plants on the chart. Therefore, circular legends are filled with colors. Other available power plant data is hidden from the chart thus circular legends are transparent. If there is no data available for selected units in JSON, then chart will load its default view instead. Default view of this chart enables the visibility of all power plants on the chart.

NVD3² JavaScript library has been used for creating chart on Energy-Charts page. It is a widely used charting library which is written in d3.js³. An instance of hourly production data file is shown in the List 4.2. This input JSON array has four attributes but only *eexUnitID* is required

²Official NVD3 Website, <https://nvd3.org/> (last accessed on February 24, 2017)

³Official D3 Website, <https://d3js.org/> (last accessed on February 24, 2017)

4 Software System



Figure 4.17: Selected power plant data from the comparison table of Figure 4.16

to fulfill our requirements. The *eexUnitID* of the hourly electricity production data and the unit id of the power plant location data (see Figure 4.16 are both the same ids. Our algorithm checks whether this eexUnitID exists inside the array of *query list* or not. If exists then it is rendered on the chart.

Listing 4.2 Example of hourly production JSON data object

```

1 // example of
2 var inputData = [
3 { "key": [{ "en":"Name_en" , "de":"Name_de" , "fr":"Name_fr" , "it":"Name_it" }] ,
4 "color" : "rgb(number,number,number)" ,
5 "eexUnitID" : "E000###-00#",
6 "KwNrBNetzA" : "BNA####" ,
7 "values" : [ [ 1485903600000 , 0 ] , [ 1485907200000 , 0 ] , [.....]]}
8 ]

```

[language=JSON,firstnumber=1]

Listing 4.3 JavaScript pseudo code for checking unit ids and enabling their visibility on the charts

```
1 IF link variables exists THEN
2   IF Only one eexUnitID exists THEN
3     loop1: REPEAT UNTIL < query list length
4     loop2: REPEAT UNTIL < eexUnitID length
5     IF eexUnitID exist in query list
6       enable series visibility
7       BREAK loop1
8     end IF
9   ELSE Disable series
10  END loop2
11  END loop1
12 end IF
13 ELSE //for more than one eexUnitID
14   REPEAT REPEAT UNTIL < query list length
15   IF eexUnitID exist in query list
16     Enable Series
17     BREAK loop2
18   end IF
19   ELSE Disable series
20 END for
21 end IF
22
23 ELSE Enable Series
```

This part of the code is the new contribution to the existing energy chart. Line 1 of List 4.3 checks the existence of parameter on URL. This function runs if URL parameters are available. Line 2 checks the number of the URL parameter (less or greater than 1). If the type of *ID* is object then starts executing from line 2. Conversely, the function starts executing from line 13 if the number of *ID* elements are more than one. Subsequently, function repeatedly checks the availability of unit ids from URL inside the data series. For this purpose, Line 6 enables the visibility of that data series on the chart if its unit id matches whereas unmatched data series are disabled on line 9. After executing this code users get the actual desired data illustrated on the chart which were queried over the link.

4.5 Optimization of power line visualization

In addition to the visualization of power plant location, showing the power transmission line is also a part of functional requirements. Therefore, our application provides an option for visualizing power transmission lines for 110kV, 220kV, and 380kV. Information about the data source is already discussed in Chapter 3 Section 3.7. Data was extracted in the format of geoJSON from the openstreetmap database. Figure 4.18 showing the available data formats that can be extracted using the overpass-turbo API. Leaflet supports GeoJSON and provides

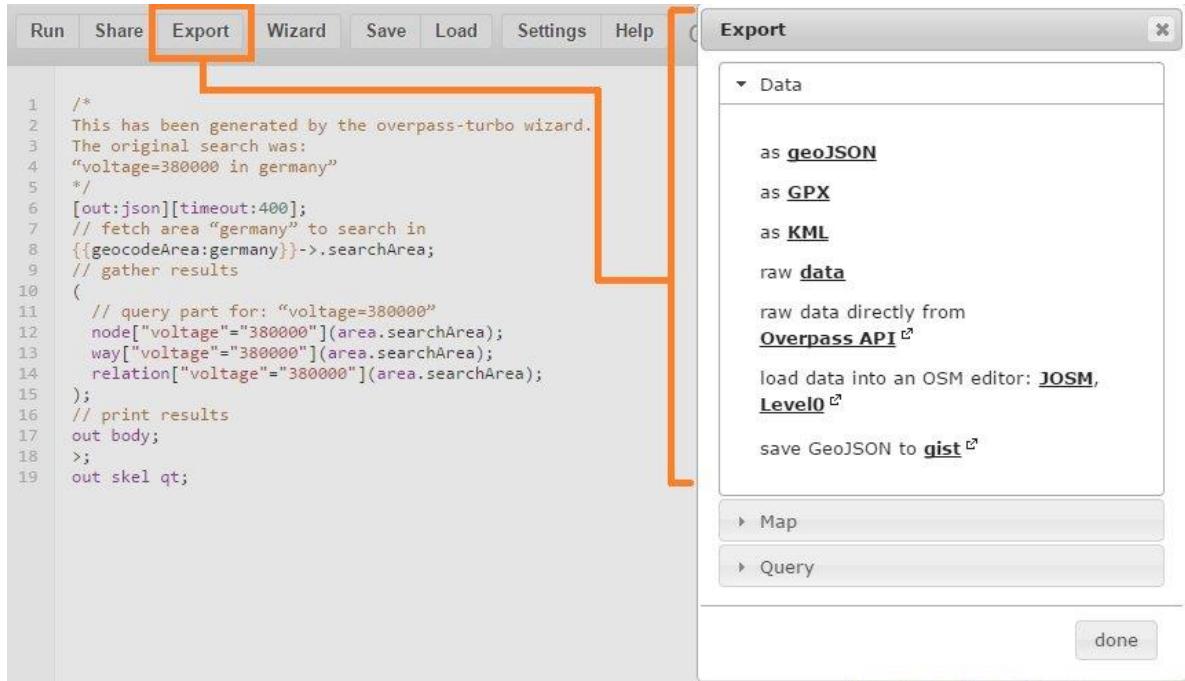


Figure 4.18: All possible data format that be extracted from the Overpass-turbo API

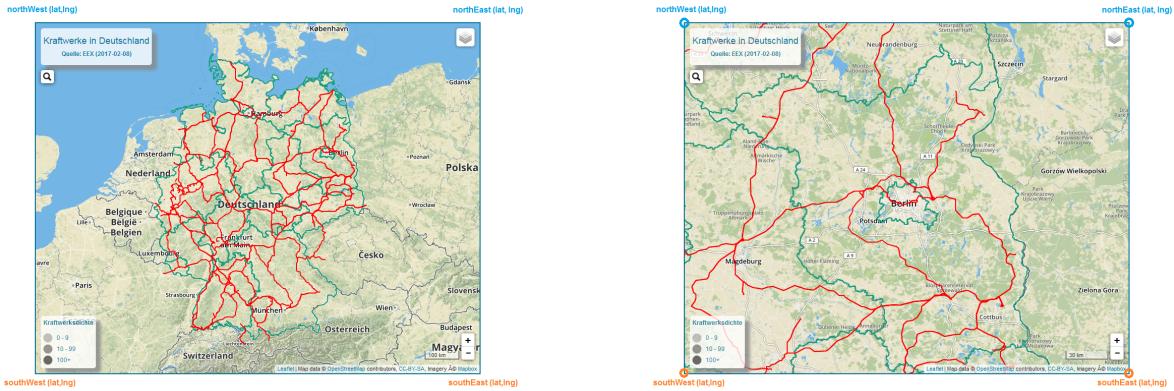
flexibility to map GeoJSON file there we have decided to use this format. Leaflet parses GeoJSON data and creates feature layers. Finally, the function displays the power transmission line to the map.

The extracted GeoJSON data are too big in size. For the desktop computers it takes less than 2 seconds to load but Smart-phones and tablets take around 2 to 5 seconds for loading. Thus, it affects the performance of the map while zooming-in/out. Hence, it was required to optimize the process of mapping power lines. The following steps were taken to optimize the visualization of power lines.

4.5.1 Map Bounds

Leaflet provides extensive opportunities to control the map and provides current scale information about the map view. For our operation map bounds were considered for displaying the power transmission line on the map. It provides the lowest and highest value of latitude and longitude of the map view. Map bounds are changed every time the map is moved (either by zoom or pan). Therefore, we used a function to reconsider the map bounds in every zoom level and display the GeoJSON data accordingly. Each time the map is moved, the map bounds is stored. If the power line data contains the map bound then the line is displayed otherwise it is skipped from mapping. Figure 4.19 showing the power lines (380 kV) on 2 different zoom

4.5 Optimization of power line visualization



(a) Complete 380kV power line in the Germany (zoom level 6, no filtering)

(b) 380kv power line through Berlin (zoom level 8, filtered by map bounds)

Figure 4.19: Filtering power lines depending on the map bounds

levels. In figure 4.19a, map zoom level is 6 and whole Germany is visible inside the map bounds. Therefore, all the power lines data contains inside the current map bounds. Figure 4.19b represents the map on zoom level 8 and showing the power lines that contains inside the current map bounds. Every time the map is moved, power lines are update according to the map bound. The Power lines which are outside the map bounds are skipped from operation. As a result, performance of the map was improved and loading time of the power lines were faster than before. List 4.4 showing the pseudo code for filtering the power line by utilizing the map bounds. Line 1 checks if the map has moved or zoomed-in/out and line 2 checks whether any power line is selected by users. A function for updating the power lines is called in line 4. Map update function is represented between line 8 to line 14 which is doing the filtering operation and return the layer to the map.

4.5.2 The Length of Power line

The length of power transmission line is also considered for optimization. A long power transmission line is a collection of small sub layers of line string. These smaller line strings are affecting the performance of the map while loading. For this purpose, smaller power lines are hidden on smaller zoom levels (when zoom level ≤ 8). Small power lines contain a length between 1m to 500m. These smaller power lines are visible on higher zoom level (zoom level > 8). This optimization is applied only for the 110kV power line. 110kV geoJSON data is relatively bigger in size and has more details than 220kV and 380kV transmission lines.

This file has massive amount of entries for power lines and power stations. Polygons are describing mostly the power stations or substations inside the GeoJSON data and line-strings

4 Software System

Listing 4.4 Power line filtering algorithm depending on the map bound

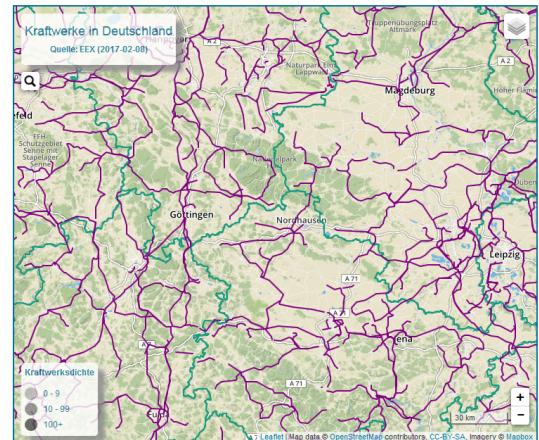
```

1 ON map move ends or map zooming ends
2 IF power line input selected
3 DO remove previous layer
4 CALL updateMap function
5 DO add new power line layer
6 end IF
7 // function for updating the map
8 FUNCTION updateMap
9   GET map bounds
10  CREATE power line layer
11  IF layer contains map bounds
12    ADD to layer
13    RETURN layer
14 end IF

```



(a) 100kV power line (zoom level < 6, coordinate length > 45)



(b) 100kV power line (zoom level 8, coordinate length > 20)

Figure 4.20: Filtering power lines by coordinate length

are describing the power transmission line. To visualize the proper routing and angle of power lines on the map, they are divided into many smaller parts inside the GeoJSON file. Our algorithm is hiding the power stations or substation, which are polygons, in smaller zoom level. Because, this smaller entries doesn't provide any useful feedback to the user. These smaller lines are available and added to the map on higher zoom level.

4.6 Summary and Discussion

In this chapter, the implementation of the back-end and the front-end of the system was explained. Furthermore, an overview of the graphical user interface as well as a detailed impression of the architecture , interactivity scripting , interconnection process between map and chart was given. The visualization application solves the interface and functional requirements described in the Section 4.1. User are able to locate power plants and their hourly production data on energy chart. Furthermore, we assume that our visualization tool will help user to access the information of power plants as well as their hourly electricity production data.

5 Evaluation

The online interactive map was evaluated in two different studies. Beforehand, it was published online and merged to the existing energy chart website to make it available for the large audience. In addition, an online survey was prepared to evaluate the map and deployed online to assess the impact, usefulness, and usability of the visualization tool. This online study evaluates the map on two different characteristics. The first study was focusing on the representational aspects of the map and the second was aiming at the usability of the map and its interactive functionality. The goal of this survey was to accumulate qualitative feedback on the map interface, visualization techniques, usability, and scopes of accessing data. A wide range of users assessed the survey. Some of them were from the field of renewable energy study and research, university professors, engineers, journalist, public service holders, businessman, and students. In this chapter, the survey studies and their results are analyzed, explained, and discussed in detail.

5.1 Online survey setup

The online survey evaluation process was divided into four different question groups with a brief introduction about the purpose of this evaluation and a little usage guide of the visualization tool. Therefore, users can explore the map and discover the functionality before participating in the survey. The participants were invited over the email and social media. They did not get any reward for participating in the online survey. None of the survey questions were mandatory, therefore participants were not required to answer every question.

In the short introduction, all the participants were informed about the purpose of making this visualization tool followed by a short description about the usability, and some tasks they need to perform for using the map and accessing the data out of it. The tasks were divided into the following steps:

Exploring power plant information:

In the description, participants were asked to click on the map marker to view the information of each power plants from the pop-up box.

Use navigation menu for interaction:

In the description, participants were told to use the navigation menu for selecting each source category and for rendering the power lines on the map.

View hourly production data:

In the description, participants were told to use the “Go to Energy Charts” link to view the hourly production data on the energy charts.

Compare power plants production:

In the description, participants were told to compare between power plants in a comparison table by using the “Compare” button from the pop-up box.

View hourly production data of multiple power plants:

In the description, participants were told to use “Compare on Energy Charts” link to see the production data of the power plants in the comparison list.

After the short introduction, some questions were asked to the participant to empirically evaluate the interface and usability of the map. In total 20 questions were asked to the survey participants and participation time was estimated about 5 minutes.

5.1.1 Question groups

Introduction

The first question group was about the self-introduction. Participants were asked to mention their gender, age, and professional area of working, besides that their frequency of using interactive maps and their acquaintance with the German power plants and its electricity production were asked.

User Interface evaluation

In this group, participants were asked to evaluate the user interface and its components especially about the cluster view, map markers, power lines, and comparison table. A collection of the demographic question, rating scale, and multiple choice question were asked for this evaluation.

Map Usability test

In this group, participants were asked to evaluate the usability and usefulness of this visualization tool. Participants were also requested to leave some comments if they found the tool difficult to use.

Comments or Suggestion

Participants were asked to comment on this complete work package and its features as well as requested to provide suggestions on adding new features or ideas.

5.2 Survey Results and Discussion

In this section, the result of survey questionnaire of each question group and the qualitative comments and consequences derived from these visions are discussed.

5.2.1 Summary of participants and their backgrounds

The goal of this question group was to get an idea about the participants, their professional background, age ratio, and their familiarity with German power plants. In addition, how frequently they use interactive maps in their daily life. For our online research study, 23 participants (19 males, 2 females, 2 did not answer) from different level expertise participated in the survey. Participants belong to different age groups where 50% with an average age of 27 years, 35% are with an average of 53 years and 15% are with an average age of 36 years (see Figure 5.1). In total, 19 full and 4 partial survey reports were submitted.

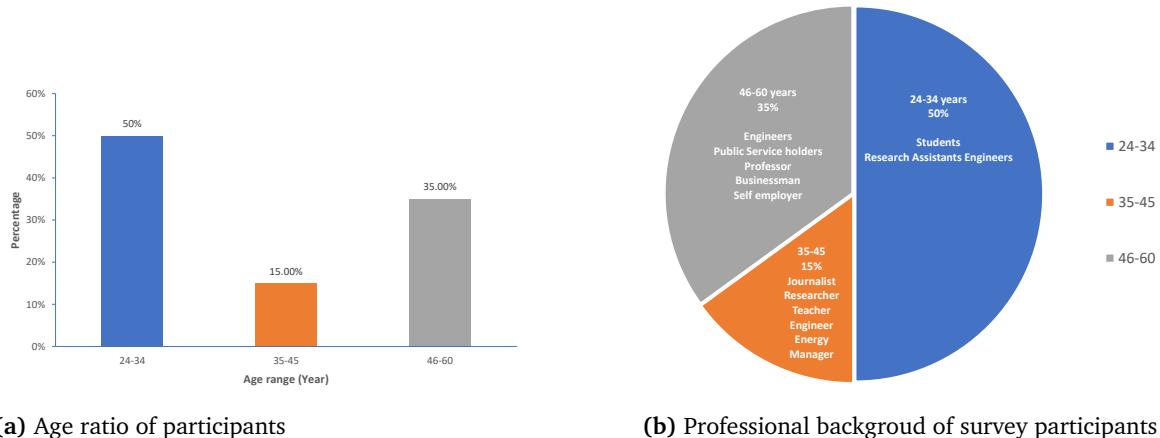


Figure 5.1: Age ration and professional background of survey participants

Summary of this question groups tells us that almost 50% of the participants who are comparatively younger than other participants are familiar with German power plants and their production to some extent whereas other participants, who belong to older age groups, are

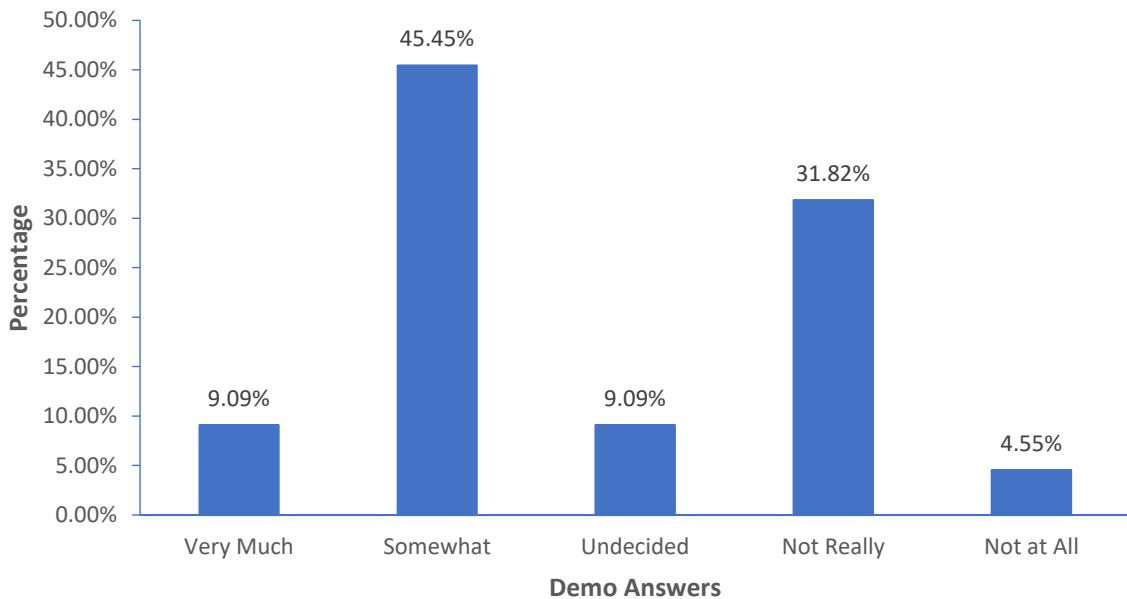


Figure 5.2: Familiarity with German power plants and their production

somewhat or very much familiar with it (see Figure 5.2). After digging up the data more deeply it has been noticed that younger participants are mostly students, research assistants, and engineers however participants older than other group are professors, businessman, and working in the energy industry. Nevertheless, the participants having no idea about German power plants also exists in all age groups. On the other hands, concerning the frequency of using interactive map per week, a maximum number of participants (around 70%, see Figure 5.3) use interactive map at least once or twice in a week. Therefore, we assumed that user would find the tool and its functionality easy to use.

5.2.2 Working with the Interactive Map

Evaluation of User Interface(UI)

We assumed that before evaluating the UI, user followed the steps described in the short introduction and gone through the interface accordingly. The purpose of this questionnaire was to evaluate the map UI and its complexity, comprehensibility, construction as well as visual aesthetics. The participants were very positive after using the map and its interface. They found it intuitive and easy to use which is confirmed by the around 91% participants. Where 77.27% participants were agreed and around 13.64% were strongly agreed on the question “*I found the map easy to use*” of the survey (see Figure 5.5a). Similar results are also noticed where

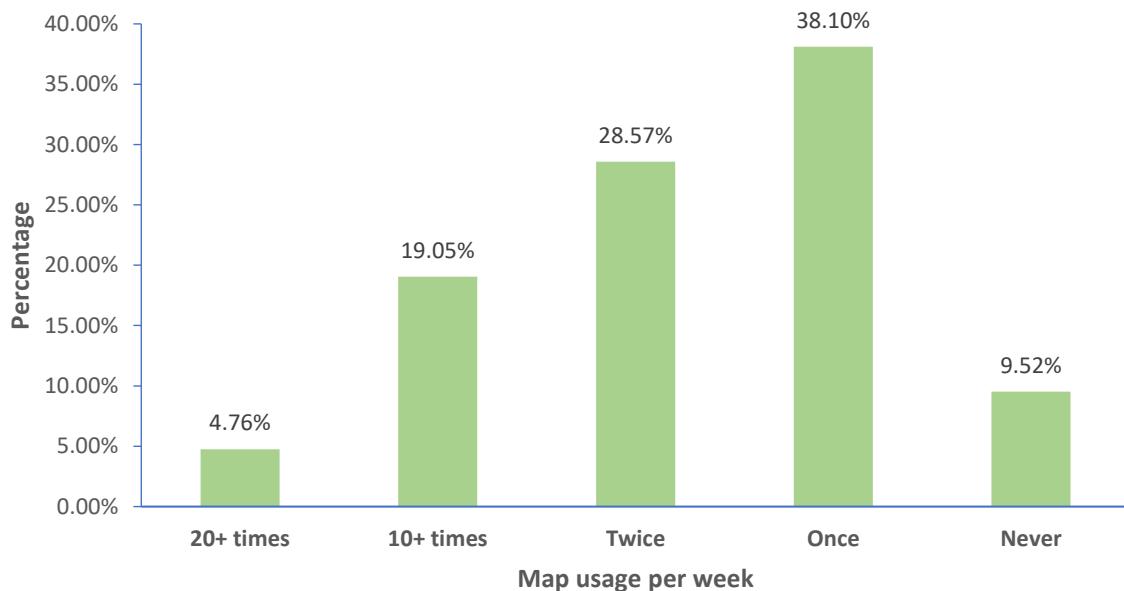


Figure 5.3: The result of online survey - participants frequency of using interactive map per week

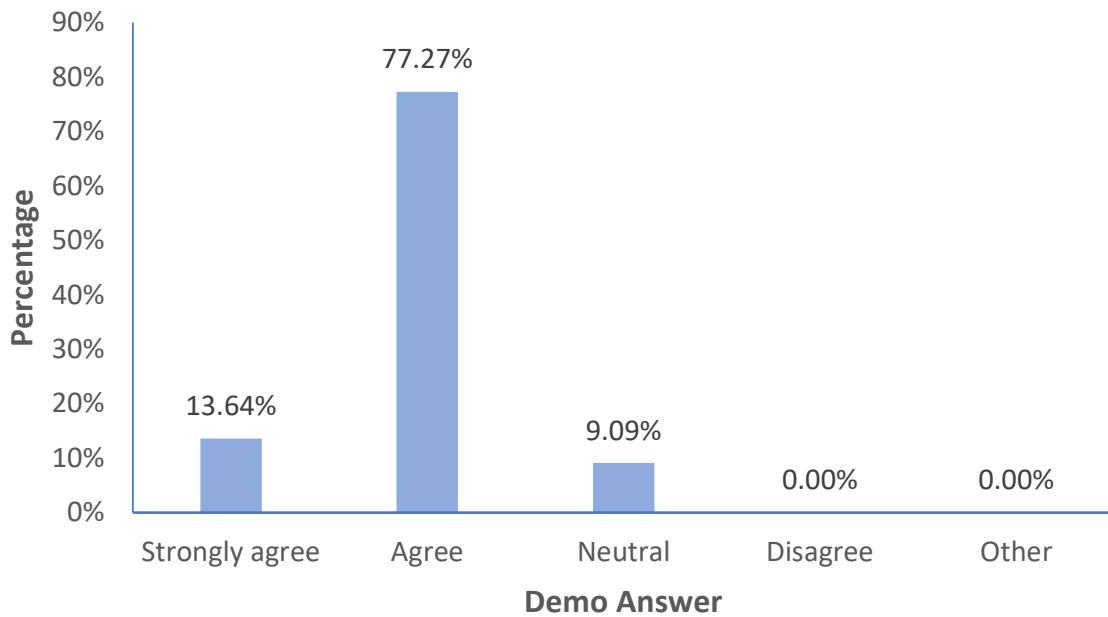
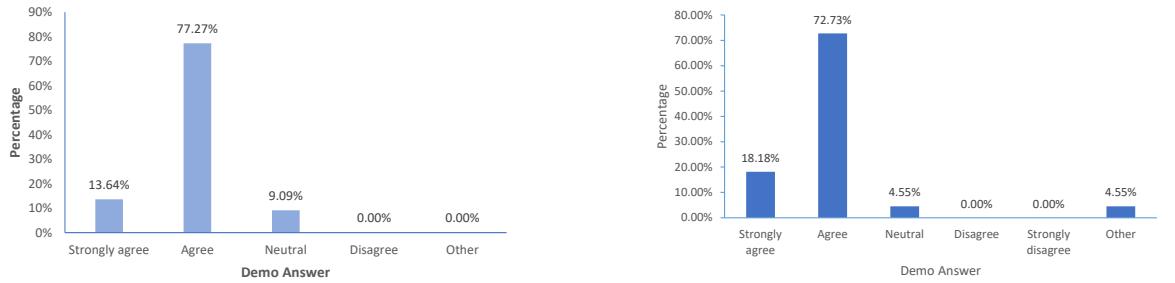


Figure 5.4: The result of online survey - map usability

5 Evaluation



(a) The evaluation result of map usability by participants

(b) The evaluation result of interface and functionality by participants

Figure 5.5: The evaluation result of online survey

participants were asked about the structure of map especially the navigation menus, markers, marker pop-ups, buttons, and comparison table. This is shown by the 91% (about 72.73% agreed and 18.18% strongly agreed) participants on the question "*Interface and functions of this map are self-explanatory*" of the survey (see Figure 5.6). There was also a comment section where participants can write about their confusions instead of answering the question. P12 (a participant) took this opportunity and wrote, "*Definitely they are self-explanatory.*". P2 said, "*navigation menu could have a logo next to each label*".

Evaluation of Map API

The participants rated the initial cluster by the average score of 3.81 (Standard deviation = 1.40, lowest(strongly disagree) – 1.0, highest – 5.0(strongly agree)). This score falls between neutral and agree. A mixed response from the participants on the cluster view is noticed from the high standard deviation. Some of them found it interesting but at the same time, some found it confusing and unnecessary. P12 again commented, "*The power plant density function should be removed, it is only confusing*". P13 noted the cluster view appears on higher zoom level and said, "*The power plant density representation is not always helping, its appearance can be resolved at higher zoom level*". The Participants found the size of the marker '*Just right*' although some participants found it little bigger. They were satisfied with the icons used inside the marker. The question "*Icons used for power plants are self-explanatory*" got an average score of 4.13 (SD = 0.71). Participants also rated the readability of the marker pop-up by the average value of 3.9 (Standard deviation = 0.9) and clarity of the navigation menu by the average value of 4.3 (Standard deviation = 0.67). Participants found the power line visualization interesting and rated by the average value of 4.5 (Standard deviation = 0.60) but not highly satisfied with the loading time in the mobile devices. P2 tested the tool on the mobile device and commented, "*The application crashes very often on tablets while loading power lines*". This result was expected as we did not use any database for storing large GeoJSON files of power lines. Therefore, It

takes between 3 - 5 seconds to load and. All in all, the participants found the user interface and other elements of the interface very user friendly and easy to understand.

Evaluation of Comparison table

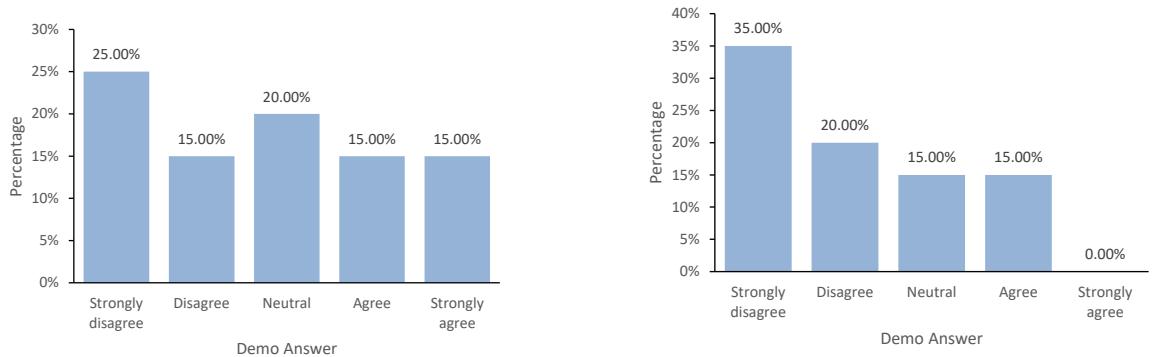
The participants also rated the comparison table on the question "*I found the comparison table useful*". Not everyone participated in this question. Some participants did not find it useful and it was difficult for find the comparison list. P2 said, "*where is the comparison table?*". A reason for this could be short usage time and overlooking the short introduction at the beginning. On the other hand, P19 noticed the feature and said, "*Pity that you can compare only the same energy sources*". All together participants rated this feature by the average score of 3.77 (Standard deviation = 1).

Usability Evaluation

The purpose of this question group was to evaluate the interactive functions, its usability, and usefulness. This study would tell whether the interactive techniques and tool supported well for exploring information. From this individual test records, we observed that 75% of the participants found this map very informative and agreed on the question "*I found this map very informative*". Others selected neutral as an answer to this question. P9, P11, P2 and P22 mentioned about the information on solar power plants which is missing in our visualization tool. P2 was mostly interested in the high voltage transmission line visualization and requested to have more information regarding the transmission lines. P9 said, "*It would be more informative marker pop-up shows the name of the location*". Around 40% Participants disagreed when they answered the question "*I found the interactivity very complex*" (see Figure 5.6a). 30% participants found it difficult to use because of having less experience in using interactive maps and others decision were neutral. Participants were also asked about the necessity of having a short tutorial beforehand or example of how to use the map. In this case, 55% participants disagreed just like before and around 15% were neutral or couldn't decide. Around 15% participant agree on the question of "*I think that I would need a tutorial before using this visualization tool*" (see Figure 5.6b).

In figure 5.7, the evaluation results are shown ordered by participants based on the map usability and its usefulness. 22 participants completed this question group. Total usability test got a score of 73 where 60% of the participants rated above than average score and they think the map is very informative and its interactivity is not really complex. 40% of the participants found it difficult to use and rated less than the average score.

5 Evaluation



(a) The evaluation result of - "I found the interactivity very complex"

(b) The evaluation result of - "I think that I would need a tutorial before using this visualization tool"

Figure 5.6: The evaluation result of complete visualization tool

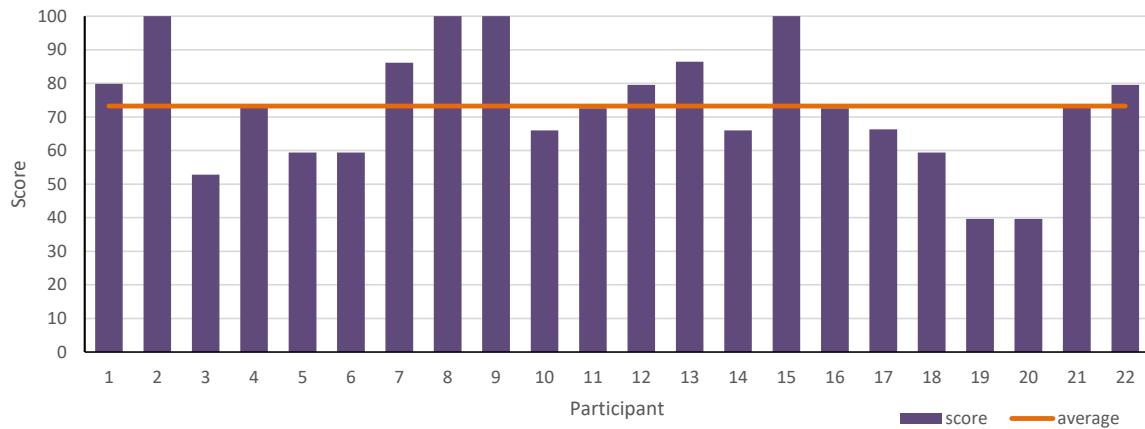


Figure 5.7: The result of online survey - complete map usability and usefulness

5.3 User Feedback

Besides having this information about the power plants the participants in general desired more information from the system. Because of missing information on solar power plants, they proposed to include solar in the information list. Participants also requested to include the city where the plant is located in the marker pop-up. Participants got confused when they saw the marker cluster layer and proposed to omit this feature. The participants also mentioned about difficulties for finding some features, for example, comparison table. Participants also talked about the limitation of comparison table and energy charts. They are interested to compare power plant units from different energy sources on energy chats. They are also interested to see power plants production in bar charts. Participants also requested to have this map in German language. People from social media also shared their thoughts and ideas who didn't participate in the survey. Their thoughts were quite similar with the survey participants. A user said, "*No information on solar!*". Another social media use provided some important information regarding the correctness of geographical location of power plants. Another user commented on the 220kV and 380kV power lines and their information. Information provided for the power lines in the map are outdated. However, users from social media noticed the work, got fascinated by the tool and appreciated saying, "*Great Stuff*", "*The Map is very informative*".

5.4 Summary

The online interactive map was evaluated using an online survey for one week. In this very short time we received 23 survey reports . The study was focused on the qualitative feedback of the map user who are renewable energy professionals, professors, students, engineering working in the energy industry, journalist, and teacher. We were able to reach them through this survey and obtained various feedback, thoughts, ideas, suggestion, usability evaluation result as well as appreciations. The qualitative feedback however uncovered some gaps and limitations within the tool which can be included in the list of future work.

6 Discussion

6.1 Advantage of the Tool

6.2 Weakness

7 Conclusion and Future Work

7.1 Summary

7.2 Future Work

Appendix

Bibliography

- [BW08] L. Byron, M. Wattenberg. Stacked Graphs – Geometry & Aesthetics. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1245–1252, 2008. doi:10.1109/TVCG.2008.166. URL <http://dx.doi.org/10.1109/TVCG.2008.166>. (Cited on page 9)
- [Cro06] D. Crockford. JSON: The Fat-Free Alternative to XML, 2006. URL <https://www.json.org/fatfree.html>. (Cited on page 19)
- [DBH16] M. Di Bartolomeo, Y. Hu. There is More to Streamgraphs than Movies: Better Aesthetics via Ordering and Lassoing. *Computer Graphics Forum*, 35(3):341–350, 2016. doi:10.1111/cgf.12910. URL <http://dx.doi.org/10.1111/cgf.12910>. (Cited on page 9)
- [DE16] G. DE. Kraftwerke ab 100 MW, 2016. URL <http://www.geoportal.de/DE/Geoportal/Karten/karten.html>. (Cited on pages 6, 7 and 8)
- [Eia17] Eia. Annual Energy Outlook 20017, 2017. URL [http://www.eia.gov/outlooks/aoe/pdf/0383\(2017\).pdf](http://www.eia.gov/outlooks/aoe/pdf/0383(2017).pdf). (Cited on pages IX, 9 and 10)
- [HB08] M. D. H. Butler. The GeoJSON Format Specification, 2008. URL <http://www.geojson.org/geojson-spec.html>. (Cited on page 18)
- [HHWN02] S. Havre, E. Hetzler, P. Whitney, L. Nowell. ThemeRiver: visualizing thematic changes in large document collections. *IEEE Transactions on Visualization and Computer Graphics*, 8(1):9–20, 2002. doi:10.1109/2945.981848. (Cited on page 8)
- [JMG15] D. K. John Muyskens, S. Granados. Mapping how the United States generates its electricity, 2015. URL <https://www.washingtonpost.com/graphics/national/power-plants>. (Cited on pages 4, 6, 7 and 8)
- [MB08] L. B. Mathew Block. The Ebb and Flow of Movies: Box Office Receipts 1986–2008, 2008. URL http://www.nytimes.com/interactive/2008/02/23/movies/20080223_REVENUE_GRAPHIC.html. (Cited on page 9)

Bibliography

- [MS03] W. Muller, H. Schumann. Visualization methods for time-dependent data - an overview. In *Proceedings of the 2003 Winter Simulation Conference, 2003.*, volume 1, pp. 737–745 Vol.1. 2003. doi:10.1109/WSC.2003.1261490. (Cited on page 8)
- [SE15] R. P. S. E. Mapped: How the UK generates its electricity, 2015. URL <https://www.carbonbrief.org/mapped-how-the-uk-generates-its-electricity>. (Cited on pages IX and 9)
- [SE16] R. P. S. Evans. Mapped: How Germany generates its electricity, 2016. URL <https://www.carbonbrief.org/how-germany-generates-its-electricity>. (Cited on pages 3, 6, 7 and 8)
- [Zei12] J. Zeigler. How Big is Too Big For JSON?, 2012. URL <http://josh.zeigler.us/technology/web-development/how-big-is-too-big-for-json/>. (Cited on page 19)

All links were last followed on February 24, 2017.

Declaration

I hereby declare that the work presented in this thesis is entirely my own and that I did not use any other sources and references than the listed ones. I have marked all direct or indirect statements from other sources contained therein as quotations. Neither this work nor significant parts of it were part of another examination procedure. I have not published this work in whole or in part before. The electronic copy is consistent with all submitted copies.

place, date, signature