

# A Tool for Data Visualization in Smart Grids

By

Alejandro A. Sánchez Fernández

A project submitted in partial fulfillment of the requirements for the degree  
of

MASTER OF ENGINEERING

in

COMPUTER ENGINEERING

UNIVERSITY OF PUERTO RICO

MAYAGÜEZ CAMPUS

2017

Approved by:

---

Wilson Rivera Gallego, Ph.D.  
President, Graduate Committee

---

Date

---

Manuel Rodríguez Martínez, Ph.D.  
Member, Graduate Committee

---

Date

---

Marla Pérez Lugo, Ph.D.  
Member, Graduate Committee

---

Date

---

Efraín O'Neill Carrillo, Ph.D.  
Member, Graduate Committee

---

Date

---

José Cruz Cruz, Ph.D.  
Graduate Studies Representative

---

Date

---

José G. Colom Ustariz, Ph.D.  
Department Chairperson

---

Date

Abstract of Project Presented to the Graduate School  
of the University of Puerto Rico in Partial Fulfillment of the  
Requirements for the Degree of Master of Engineering

## **A Tool for Data Visualization in Smart Grids**

by

*Alejandro A. Sánchez Fernández*

Chair: Dr. Wilson Rivera Gallego

Department: Electrical and Computer Engineering Department

Data in the smart grid is generated from various sources including for example, energy consumption data measured by smart meters, operation data from devices in the power infrastructure, data from sensors distributed over transmission and distribution networks, and auxiliary data such as weather, geographic information system, social media, and energy market data. With so much data being generated related to smart grids, users may not be able to make informed decisions related to energy consumption and/or production. In this project, we tackle this problem by creating an easy to use dashboard that presents smart grid data. The two main objectives of this project are to: (1) study and review the best practices in the collection, analysis and visualization of data from smart grids; and (2) implement a smart grid data workflow that includes data collection from diverse sources and interactive visualization. We have implemented a dashboard called the OASIS dashboard using various tools and techniques for data analysis and visualization. Using this dashboard users may reach a better understanding of data related to smart grids and potentially make conscious decisions regarding to energy consumption and/or production.

Resumen de proyecto presentado a la Escuela Graduada  
de la Universidad de Puerto Rico como requisito parcial de los  
requerimientos para el grado de Maestría en Ingeniería

## **Una Herramienta para la Visualización de datos en la red inteligente**

by

*Alejandro A. Sánchez Fernández*

Consejero: Dr. Wilson Rivera Gallego

Departamento: Ingeniería Eléctrica y Computadoras

Los datos en la red inteligente son generados de diversas fuentes. Entre estas se encuentra la data de consumo obtenida por los metros inteligentes, data operacional, data de sensores distribuidos a través de redes de transmisión y distribución, data auxiliar como el clima, data geográfica, data de redes sociales y data del mercado de energía. Con la gran cantidad de datos siendo generados relacionados a las redes inteligentes, los usuarios quizás no puedan tomar decisiones informadas relacionadas a su consumo o producción de energía. En este proyecto nos enfrentamos a este problema creando un dashboard que es fácil de utilizar y entender, en donde se presenta la data relacionada a las redes inteligentes analizada y de forma visual. Los dos propósitos de esta investigación son: (1) Investigar y determinar las mejores prácticas en la colección, análisis y visualización de data relacionada a las redes inteligentes; e (2) Implementar un flujo de trabajo relacionado a las redes inteligentes que incluye la colección de data de diversas fuentes y la visualización interactiva de esta data. Implementamos el OASIS dashboard utilizando varias técnicas de data análisis y visualización. Utilizando este dashboard los usuarios podrán obtener un mejor conocimiento sobre la data relacionada a las redes inteligentes y potencialmente podrán tomar decisiones conscientes con relación a su consumo y/o producción de energía.

Copyright ©2017

*by*

*Alejandro A. Sánchez Fernández*

*I would like to dedicate this project to my friends and family who helped me through this process.*

## Acknowledgments

I would like to give thanks to my advisor Wilson Rivera for guiding me through the process of researching tools and techniques to properly finish this project. Without his help and belief in me this project would have not been possible.

Also, I would like to thank the members of my committee for giving me ideas for helping me see the social and electrical aspects of the project.

Many thanks to the ADM lab team members, Jessica Cotrina, Howard Martinez, Cristian Garzón and Dan Rosa, for always supporting me through the project. Finally, I would like to acknowledge the National Science Foundation for funding this project, under the grant #ACI-1541106. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the NSF.

# Table of Contents

Abstract English	ii
Abstract Spanish	iii
Acknowledgments	vi
List of Figures	x
List of Abbreviations	xii
<b>1 INTRODUCTION</b>	<b>1</b>
1.1 Contribution of the project . . . . .	2
1.2 Document Organization . . . . .	3
<b>2 CONCEPTS AND TOOLS</b>	<b>4</b>
2.1 Smart Grids . . . . .	4
2.2 Big Data . . . . .	5
2.2.1 Definition . . . . .	5
2.2.2 Big Data and Smart Grids . . . . .	5
2.2.3 Big Data Analysis . . . . .	6
2.2.4 Sentiment Analysis using Twitter Data . . . . .	7
2.3 Data Visualization . . . . .	8

2.3.1	Definition . . . . .	8
2.3.2	Techniques . . . . .	8
2.3.3	Visualization and Smart Grids . . . . .	9
2.4	Web Dashboard Development Tools . . . . .	9
2.4.1	Python . . . . .	10
2.4.2	JavaScript . . . . .	13
2.4.3	HTML . . . . .	14
2.4.4	CSS . . . . .	14
2.5	Related Work . . . . .	14
2.5.1	Big Data Analysis in Smart Grids . . . . .	14
2.5.2	Big Data Visualization in Smart Grids . . . . .	15
<b>3</b>	<b>DASHBOARD FOR VISUALIZING SMART GRID DATA</b>	<b>17</b>
3.1	Energy Data Section . . . . .	17
3.2	Social Data Section . . . . .	18
3.3	Weather Data Section . . . . .	20
3.4	Market Data Section . . . . .	23
3.5	Data Integration Section . . . . .	25
3.6	How to access the web dashboard . . . . .	27
<b>4</b>	<b>WEB DASHBOARD DEVELOPMENT PROCESS</b>	<b>30</b>
4.1	Energy Data . . . . .	30
4.1.1	Data Gathering . . . . .	30
4.1.2	Data Modification . . . . .	31
4.1.3	Data Visualization . . . . .	32
4.2	Social Data . . . . .	33
4.2.1	Data Gathering . . . . .	33
4.2.2	Data Cleaning . . . . .	34



4.2.3	Data Analysis . . . . .	35
4.2.4	Data Visualization . . . . .	36
4.3	Weather Data . . . . .	37
4.3.1	Data Gathering, cleaning and analysis . . . . .	37
4.3.2	Data Visualization . . . . .	38
4.4	Market Data . . . . .	38
4.4.1	Data Gathering . . . . .	39
4.4.2	Data Modification . . . . .	39
4.4.3	Data Visualization . . . . .	40
4.5	Comparison between datasets . . . . .	40
4.5.1	Data Integration . . . . .	41
4.5.2	Data Visualization . . . . .	41
<b>5</b>	<b>CONCLUSION AND FUTURE WORK</b>	<b>42</b>
5.1	Conclusion . . . . .	42
5.2	Future Work . . . . .	43
	<b>Bibliography</b>	<b>45</b>

# List of Figures

2.1	Tools for developing the OASIS dashboard . . . . .	10
2.2	Basic Flask Application . . . . .	11
3.1	Energy data section from the OASIS Dashboard . . . . .	18
3.2	Pie chart from social data section of the OASIS Dashboard . . . . .	19
3.3	Donut charts from social data section of the OASIS dashboard . . . . .	20
3.4	Twitter data in csv format . . . . .	20
3.5	Current weather, todays forecast and 12-hour forecast from weather data section of the OASIS Dashboard . . . . .	21
3.6	4-day forecast and Puerto Rico map showing temperature data from weather data section of the OASIS Dashboard . . . . .	22
3.7	Weather data in csv format . . . . .	22
3.8	Load data from market data section of the OASIS Dashboard . . . . .	23
3.9	Zonal LBMP data from market data section of the OASIS Dashboard . . .	24
3.10	Load vs Zonal LBMP data from market data section of the OASIS Dashboard	24
3.11	Process of creating graph from data integration section of the dashboard .	26
3.12	Process of creating a graph in the data integration section of the OASIS Dashboard . . . . .	26
3.13	Data comparison between humidity in Adjuntas Puerto Rico, positive tweets and negative tweets in csv format . . . . .	27

4.1	AEE energy data in csv file . . . . .	31
4.2	Example AEE JSON data . . . . .	31
4.3	Example JSON that NVD3 is expecting . . . . .	32
4.4	Example of a tweet stored in tweets.txt . . . . .	34
4.5	Example cleaned tweet . . . . .	34
4.6	Example cleaned tweet . . . . .	36
4.7	JSON File of weather data . . . . .	38
4.8	Example JSON of how load data looks like . . . . .	39
4.9	Modified JSON of load data . . . . .	40

## LIST OF ABBREVIATIONS

NIST	National institute of standards and technology
OASIS	Open access smart grids
NLTK	Natural language toolkit
WSGI	Web server gateway interface
D3	Data driven documents
HTML	Hypertext markup language
CSS	Cascading style sheet
AEE	Autoridad de energia electrica (Puerto Rico Power Authority)
CSV	Comma separated value

# Chapter 1

## INTRODUCTION

Electric energy networks are the cornerstone of the civil infrastructure of our society.<sup>1</sup> These networks provide the energy essential for carrying out daily operations in education, health care, commerce, entertainment, defense, and government. However, electric energy markets, due to their vertical integration often exclude customers from the processes associated with energy production, pricing, transmission and distribution.

A new trend in energy infrastructure management has been proposed. This update to the energy grid is commonly known as the smart grid. In a smart grid, computers and communications networks are embedded in the power generation, transmission, distribution and user applications, establishing a mechanism to gather information, control generation, control demand, diagnose problems, bid for prices in energy markets, and forecast energy consumption. As the number of sensors and the ability to collect large amounts of data increases, analysis and visualization techniques are required to support decision making.

OASIS (Open Access Smart Grids) is a project from the University of Puerto Rico (UPRM) that aims to study smart grids and the interdependency between the energy grid and the IT infrastructure that is setup to manage it. This project champions a transfor-

---

<sup>1</sup> Interdependent Electric and Cloud Services for Sustainable, Reliable, and Open Smart Grids: [https://www.nsf.gov/awardsearch/showAward?AWD\\_ID=1541106](https://www.nsf.gov/awardsearch/showAward?AWD_ID=1541106)

mation of the electric grid, moving it away from being centered on centralized utilities that supply most, if not all, power services. Instead, the grid becomes a marketplace of third-party power-service suppliers, who compete to sell their electric services over the Internet. These electric services include generation, storage, billing, weather forecasting, and other ancillary services. As part of the activities in this project we are interested in the analysis and visualization of smart grid data. This report describes our work in this direction.

## **1.1 Contribution of the project**

The two main objectives of this project are to: (1) study and review the best practices in the collection, analysis and visualization of data from smart grids; and (2) implement a smart grid data workflow that includes data collection from diverse sources and interactive visualization.

A dashboard called the OASIS dashboard has been implemented using various tools and techniques for data analysis and visualization. The dashboard contains 5 areas of interest: (1) Energy Data, (2) Social Data, (3) Weather Data, (4) Market Data and (5) Data Integration. The purpose of this dashboard is to help users in the open energy marketplace, currently being generated by the OASIS team, to make more conscious decisions when purchasing and selling energy.

The dashboard generated on this project will be utilized as a proof of concept to demonstrate how and what data related to smart grids can be visualized interactively. This document will also describe why we chose the technologies for implementing the dashboard and lessons learned.

## **1.2 Document Organization**

This document is organized as follows: Chapter 2 discusses the concepts and tools utilized for implementing this project. Chapter 3 describes in detail the web dashboard created in this project and each of its sections. Chapter 4 explains in detail the development process for the dashboard. Finally, chapter 5 shows the conclusions drawn from the finished project and some potential future work. The references are shown at the end of this document.

# Chapter 2

## CONCEPTS AND TOOLS

Chapter 2 describes all the concepts and tools related to this project. We begin by describing smart grids in section 2.1, then we describe what is known as big data and how it is related to smart grids in section 2.2. Section 2.3 defines visualization and describes some visualization techniques. The tools for developing the dashboard are explained in section 2.4. Finally, section 2.5 shows some related work to our project.

### 2.1 Smart Grids

Smart grids have been proposed as a mechanism to modernize energy grids. In a smart grid, sensors, computers and communication networks are integrated into the power generation, transmission, distribution, and load elements. This enables devices to gather information, control generation and demand, diagnose problems, and forecast energy consumption.[\[1\]](#)

A smart grid enables bidirectional flows of energy and uses two-way communication and control capabilities. This is a departure from traditional power grids with one way communication and limited control. The NIST smart grid conceptual model [\[1\]](#), consists of seven domains all interconnected through a communication network. The domains are the following: (1) Customers, which are the users of electricity (2) Markets which



are the participants in electricity markets, (3) Service Provider which is the organization providing services to customers, (4) Operations that manage the movement of electricity, (5) Bulk Generation which are the generators of electricity, (6) Transmission which handles the transference of energy and (7) Distribution which distributes energy to customers. Each domain encompasses several challenges and opportunities. For example, distributed generation (DG) can be implemented taking advantage from distributed energy resources (DER) to improve power quality and use local resources, representing a challenge to traditional bulk generation participants.

## **2.2 Big Data**

### **2.2.1 Definition**

According to NIST [50], Big Data can be defined as the inability to manage data in traditional data architectures like databases. This is due to the following four factors: (1) Volume, (2) Variety, (3) Velocity and (4) Variability. Volume refers to the size of the dataset, Variety refers to the multiple sources from which data can be collected, Velocity refers to how frequently the data is being generated and Variability refers to how different the datasets may be. An example of big data is data generated in Twitter. The amount of twitter data generated every second is enormous, which may or may not contain attributes like retweets and favorites.

### **2.2.2 Big Data and Smart Grids**

Big data in the smart grid is generated from various sources such as energy consumption measured by smart meters, operation data from devices in the power infrastructure, data from sensors distributed over transmission and distribution networks, and auxiliary data such as weather, geographic information system, social media, and energy market data.

A smart grid network brings a whole set of new issues since the sheer volume of data

that needs to be collected from the power grid requires big data techniques to manage it. In addition, the large number of elements that must be tracked require parallel processing techniques to add redundancy and to speed up system analysis.

The distributed resources in a smart grid may include renewable sources of energy such as solar and wind energy. However, the effective utilization of intermittent renewable sources poses a challenge. Several solutions have been proposed to model short and long term renewable source patterns. Examples of such solutions are Gaussian distributions [2], finite state Markov models [3], and online learning techniques [4].

Another example, now in the service provider domain, is the use of data from smart meters [5], sensors [6] and phasor measurement units (PMU) [1] to estimate energy prices and consumer behavior [8, 9]. It enables providers to capture the interdependency between the energy demand and the prices [10] enhancing the decision-making process.

The previous examples illustrate the importance of big data analysis in smart grids to facilitate decision making. It is worth mentioning that estimates of the size of the global market related to smart grid data analytics will be around USD \$3.8 billion by 2020 [11].

### 2.2.3 Big Data Analysis

There are many tools utilized for big data analysis. Among them are data mining algorithms; these are applied across data sets to extract patterns. Data mining algorithms can be categorized in four different approaches: association rules, classification, clustering, and sequential patterns. Association rules [12, 13] are used to determine correlations between different attributes in a data set. Classification algorithms such as decision trees [14], Bayesian classification [15], and support vector machines [16] are designed to predict one or more discrete variables based on the other attributes in the dataset. Clustering [17, 18, 19] algorithms categorize subsets of data with similar properties. Finally, sequential patterns [20, 21] enable the analysis of time and spatial series of data. Data mining is

being applied to diverse application domains. For example, sentiment analysis [22] refers to the use of natural language processing and data mining to extract subjective information from documents and social media. Examples of how other researchers are applying these techniques are shown in section 2.5.1 Big Data Analysis in Smart Grids.

Despite major advances in data analysis for smart grids we can identify some challenges that remain: (1) In order to fully realize the smart grid vision, scalable and adaptive algorithms are required to deal with the dynamic and uncertainty nature of the smart grid. For example, the need for optimal coordination of intermittent distributed energy resources, while modeling dynamic stakeholders behavior suggests a hierarchical control architecture. (2) The wide spread use of sensors and smart meters would increase the need for privacy and security strategies. In turn, new data privacy mechanisms are required to deal with the very complex and dynamic smart grid infrastructure.

The OASIS project<sup>1</sup> aims at designing and developing the basic models and infrastructure prototypes that can help implement and manage an Open Access Smart Grid to create truly sustainable energy markets. The project focuses on the distribution level of the power grid since it provides the best opportunities for the integration of alternative power sources, customer participation, and energy storage. OASIS platform takes advantage of big data techniques to provide trending information, past performance data, and prediction on possible prices, energy availability, customer signup, and customer satisfaction ratings. It uses multivariate linear regression, neural network algorithms, and machine learning tools to enable consumers of resources and policy makers to have a more in-depth understanding of the services available in the marketplace.

#### **2.2.4 Sentiment Analysis using Twitter Data**

Sentiment Analysis can be defined as the process of reading and analyzing a sentence and classifying as negative or positive. This type of analysis is very popular and useful with

---

<sup>1</sup> OASIS Project website: <http://oasis.uprm.edu/>

twitter data. Real time twitter data is easily collected using the twitter streaming API. We can also apply filters to the types of tweets we want to gather. For example, in this project we are particularly interested in tweets related to energy data from Puerto Rico. The process of gathering these tweets is described in section 4.2.1.

Sentiment Analysis in Twitter is a completely different process from other investigations such as the ones in [51]. This is due to the many limitations of a tweet. One of these limitations is that a tweet may only have 140 characters, excluding Twitter username and links, therefore many users abbreviate words or use emoji to express a sentiment. Another factor that makes this process is the ambiguity of the text. For example, the text may contain sarcasm, making it difficult to determine its polarity. Twitter Sentiment analysis in this project is done in two phases: First, identifying sentiments and expressions and second determining the polarity of the express sentiment in the tweet. For this project, we are using sentiment analysis to find the opinion of users related to energy production and consumption, smart grids and other concepts related to energy.

## **2.3 Data Visualization**

### **2.3.1 Definition**

Data visualization can be defined as a graphical representation of data. These graphs are created with the purpose of helping people understand data more easily. With data visualization patterns or hidden data can be easily discovered, potentially helping users to gain more insight on the data they are analyzing. Examples of graphs are bar graphs, stacked charts, scatter charts, line charts, pie charts, etc.

### **2.3.2 Techniques**

There are many techniques used for visualizing data. Among them are: query-driven visualization for information exploration, event-driven visualization for situational awareness

and visual data mining for knowledge discovery. These are explained in more detail in section 2.5.2 Big Data Visualization in Smart Grids. The Oasis [44] project, will utilize as guidance the work discussed in this section to create a web dashboard that will be customizable and will change based on different user profiles. The goal is to have an easy to use and understand dashboard that is tailored to each users specific needs and problems.

### **2.3.3 Visualization and Smart Grids**

Currently, there is no consensus among the community on how to visualize data related to smart grids or what kind of data should be shown to users. However, we will discuss some of the solutions or methods that other studies have implemented and use them as a guide for the dashboard that will be implemented in the Oasis project.

The work in [48] shows some of the challenges of visualizing data with Big Data, specifically outlining how to handle each of the Vs in Big Data. Another problem is that data may be 1-dimensional or multi-dimensional and that there is real-time data and static data. Each of these datasets are managed in this project.

## **2.4 Web Dashboard Development Tools**

The Oasis dashboard is a web dashboard that was designed to be a companion to the electricity marketplace from the OASIS project. This project was developed using Python and JavaScript programming languages. For Python, we used the framework Flask, and the libraries NLTK, Tweepy and Forecastio-py. For JavaScript, we are utilizing the jQuery and D3. We are also utilizing the markup languages: HTML and CSS. Detailed descriptions of these tools are provided in the next subsections. In Figure 2.1 we summarize the development tools used in this project.

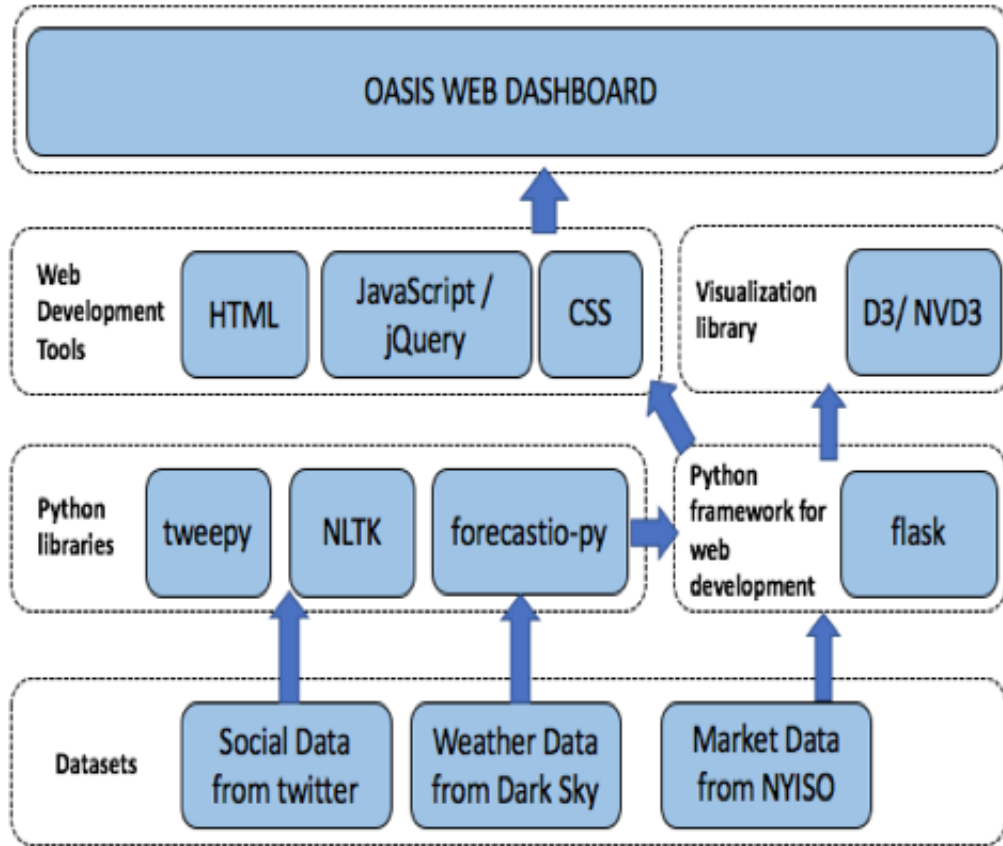


Fig 2.1: Tools for developing the OASIS dashboard

The dashboard created in this project could also be created with other programming tools such as Java, instead of Python.

#### 2.4.1 Python

For this project, we are utilizing Python 2.7.10. Python is a general-purpose language that can be utilized for web development. It is an interpreted language, this means that it does not have to be compiled into machine code. To run a Python program the user just types the command `python <filename>.py`. Python 2.7 was chosen mainly because it is very well-documented legacy language and because it has many libraries, like the ones described below, that facilitate web development, data gathering and data analysis.

## Flask

Flask is a web development framework built exclusively for Python. With Flask, developers can create blogs, websites, dashboards, etc. It provides many tools and libraries that facilitate development. For example, Flask provides a server to run the web application. It is considered a micro-framework, which is a framework that has little to none dependencies, because it only has two dependencies: Werkzeug, which is a WSGI utility library and Jinja2 which is a templating engine. Flask works utilizing routes to successfully render a template. A template can be thought of as a web page, for example, when the user clicks a button in the web page, this button may call a route inside flask that then executes some code and returns it to the webpage. To run flask the developer needs to import the flask library, then create the application, create the routes for this application and then run the application, all these steps are shown in Figure 2.2.

```
#!/usr/bin/env python

import flask

# Create the application.
APP = flask.Flask(__name__)

@APP.route('/')
def index():
    """ Displays the index page accessible at '/'
    """
    return flask.render_template('index.html')

if __name__ == '__main__':
    APP.debug=True
    APP.run()
```

Fig 2.2: Basic Flask Application

This micro-framework was selected as the basis for the OASIS dashboard because it is relatively easy to use and because it is well documented in the internet which can help facilitate development. Flask also provides many tools that made dashboard development easier. Among the tools that we utilized are the following: *jsonify*, which converts data into JSON, *request*, which gets data from templates that can be used for other templates

and *send\_from\_directory*, which is useful for downloading files from the dashboard. Other tools that can be used to implement the dashboard could be Java Play framework for the backend and AngularJs for the frontend.

### *NLTK*

NLTK stands for Natural Language Toolkit. This is a library for Python that is utilized for working with human natural language. For this project, we are using NLTK to apply sentiment analysis to the tweets being gathered. NLTK was chosen because it provides a naïve bayes classifier, that can be easily trained and utilized for classifying tweets as positive or negative. The process of how we applied sentiment analysis is described in section 4.2.

### *Tweepy*

Tweepy is a library that manages the Twitter API for Python. With this library, we can use the Twitter Streaming API in our project. The Twitter Streaming API permits developers to gather the real-time tweets that are being published to Twitter. In this project, we utilized tweepy by using the streaming API to gather data that is related smart grids using the list of energy keywords that are shown at the beginning of the document. This library was chosen among other Python Twitter clients, because it is the most well documented.

### *Forecastio-py*

Forecastio-py is a library for accessing weather data using the Dark Sky API. It provides the necessary tools for communicating Python with the API. With this library, developers can access current forecast, 48-hour forecast, 7 day forecast and historic weather data for various locations based on the latitude and longitude. We chose this library because it the most well document and it is easy to use with Python 2.7.



### 2.4.2 JavaScript

JavaScript is a programming language that is mainly utilized for web development. It is an interpreted language, just as python. JavaScript was chosen because it is, currently, the most popular language for web development. This means that this language is well documented and there are many examples in the internet that facilitate development. There are many libraries that improve JavaScripts core capabilities, among these are jQuery and D3, which are described below.

#### *jQuery*

jQuery is a JavaScript library that makes many of the core functionality of JavaScript more accessible. For example, event handling (what happens when a button or link is click in the webpage) and ajax requests (getting data from a server to the webpage) are very common in webpages and the code for implementing these features is greatly reduced with jQuery. In this project, we utilize jQuery mainly for manipulating some of the content displayed on the dashboard and to handle the ajax requests which are mainly GET requests to the server.

#### *D3*

D3 stands for data driven documents. This JavaScript library is utilized for visualizing data in an interactive manner. In recent years, it has become very popular because it provides a relatively easy way to visualize data interactively. With D3, a developer can gather data from various sources like JSON, csv or txt, among others file formats and then visualize that data in graphs. Some of the graphs that may be generated are: bar charts, line charts, stacked charts, pie charts and donut charts.

For this project, we are utilizing a library called NVD3. We chose this library instead of plain D3 because it simplifies the process of creating charts. With NVD3, the template

for the chart already comes preloaded, the developer simply needs to manipulate the data so that it can be visualized correctly.

### **2.4.3 HTML**

HTML is a markup language utilized for generating web pages. It stands for Hypertext Markup Language. This is one of the three most important languages utilized for the creation of web pages, the other two being JavaScript and CSS. In our project, we are creating template files using HTML which then are being rendered utilizing Flask.

### **2.4.4 CSS**

CSS is a stylesheet language that is used in web pages to define how the contents of a HTML file should look like. It stands for Cascading Style Sheets. CSS can be thought of as a companion for HTML; it is what makes webpages more accessible. In this project, we utilize a CSS library called Bootstrap, which, currently, is the most popular web developing framework used styling webpages. Bootstrap is also very popular because it works with different aspect ratios and it adapts to the screen that is accessing the webpage. We are also utilizing the normal CSS for styling the dashboard.

## **2.5 Related Work**

### **2.5.1 Big Data Analysis in Smart Grids**

There are several examples of the application of data mining analysis in the power analysis domain. Simmhan et. al. [23] developed a cloud-based software platform to address the dynamic demand response problem in which supply-demand mismatch must be detected and pre-emptively corrected by initiating demand-side management from consumers. At the core of the platform the authors use regression-tree machine-learning and auto-regressive integrated moving average (ARIMA) time-series to predict the energy con-

sumed at different spatial and temporal granularities for individual or aggregate buildings. Balac et. al [24] describes a research platform driven by the University of California, San Diego (UCSD) microgrid for developing large scale, predictive analytics for real-time energy management. Big data analysis has been applied in several scenarios including for example sustainable power grids [25], energy load forecasting [26-36], renewable energy resources [37, 38], and secure smart grids [39].

### 2.5.2 Big Data Visualization in Smart Grids

The research conducted by [46] utilizes power consumption data gathered from smart meters, which is stored in a database and then analyzed. This data is analyzed in three different times: daily, weekly and yearly. Using these three different analyzed data, the visualization will be divided into three different layers. Each layer will work with the respected analyzed data. Layer 1, will visualize the daily data, by showing users power consumption for each hour. Similarly, layer 2 will create a visualization of each day of a week, and layer 3 each month of the year.

The work by [47] proposes a visualization system for real-time monitoring of smart grids for control room operators of large amounts of distributed power generation. The proposed methodology for visualization utilizes an analysis pipeline of three stages: filtering, diagnosis and recommendation. The user can trace the different inputs for each of these stages as visualizations and see if something is wrong with the inputs. If something is wrong with the inputs, the user can change them and the visualization will change in real-time.

One proposed way of visualizing data is using a data-driven approach. This approach is explained with more detail in [48]. The goal of this approach is to help users find and visualize data that is deemed interesting specifically for them. There are three types of data-driven approaches that can be taken:

1. *Query-driven visualization for information exploration*: it focuses on providing patterns for defining, finding and visualizing interesting data for users. It is well suited for performing analysis and visualization on large and complex datasets.
2. *Event-driven visualization for situational awareness*: it is particularly useful for providing effective decision-support for performing time-critical tasks. For example, if its about to rain, the system could generate an alert to show the user that their solar panels will not be receiving sunlight.
3. *Visual data mining for knowledge discovery*: utilizing this method, the system can find interesting patterns that otherwise might be hidden and then visualize the data generated from these patterns.

Another important factor that needs to be visualized is how pricing for energy is changing. The work by [49] focuses precisely on this. They propose the gridSMART RTP system which gathers data from customers and then visualizes their energy consumption price. Other factors such as weather and social data need could affect how a user benefits from the smart grid, therefore this data must also be analyzed and shown to the user.

## Chapter 3

# DASHBOARD FOR VISUALIZING SMART GRID DATA

Chapter 3 focuses on stating what is the importance of this investigation and the composition of the web dashboard. Dashboard sections are explained sections 3.1 to 3.5 of this document. Finally, in 3.6 we describe how a user can access the dashboard from a web browser and how to install the dashboard in either Ubuntu or Mac OS.

### 3.1 Energy Data Section

The Energy Data area will show users AEE (Puerto Rico Electric Power Authority) historic data that was gathered from 2000 to 2015. This data is being shown in a stacked chart created in NVD3. The graphs being shown are the following:

- Active customers by service class
- Basic income of customers
- Consumption by service class
- Demand and generation

- Income derived from purchased energy
- Income from fuel adjustment

This data is also exportable as a csv file for users. The process of gathering, modifying and visualizing this data is described in section 4.1. Figure 3.1 shows how the Energy Data section looks like.

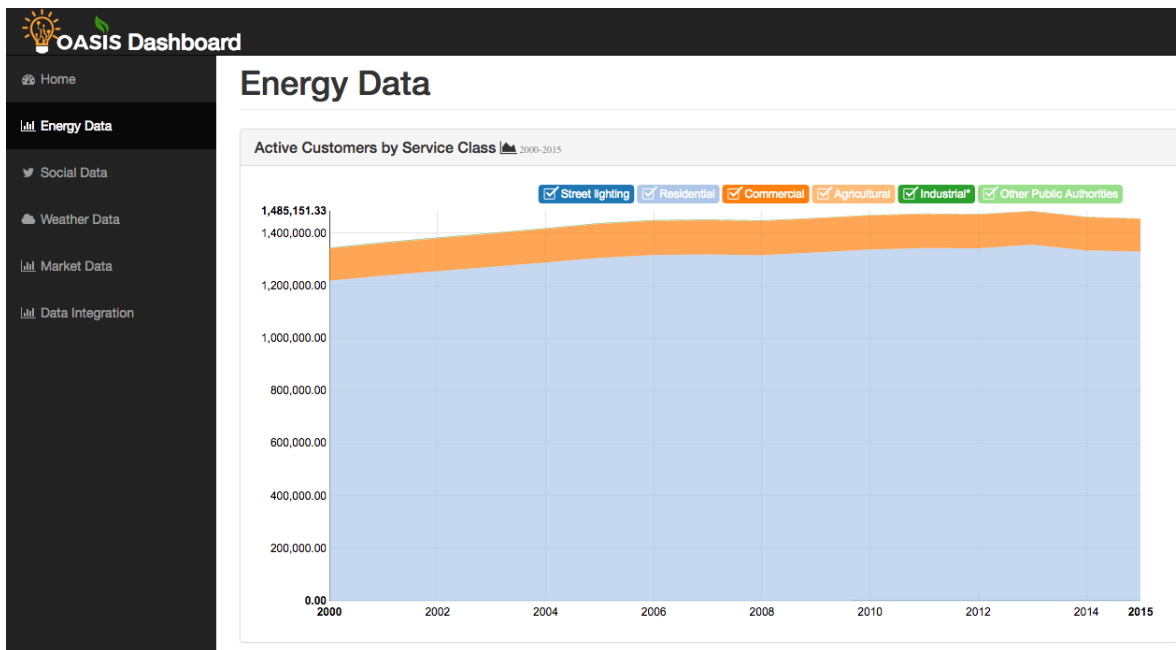


Fig 3.1: Energy data section from the OASIS Dashboard

### 3.2 Social Data Section

Social data is being gathered from Twitter. Tweets from Puerto Rico related to smart grids and energy are being streamed, stored and classified as positive or negative using a naïve bayes classifier. The process of classifying tweets as positive or negative is called sentiment analysis. Sentiment analysis is defined in section 2.2.4 and the description of how this technique is implemented and integrated is explained in section 4.3. A pie chart shows the amount of positive and negative tweets, two donut charts show the top 10 most positive and negative words and the tweets classified as positive or negative are

being shown in near real-time. Positive tweets are colored blue and negative tweets are colored red. The process of visualizing this data is shown in section 4.3. Figures 3.2 shows a button for downloading twitter data and a pie chart from the dashboard that shows the percentage of positive and negative tweets in Puerto Rico related to smart grids and figure 3.3, shows two donut charts. Figure 3.4 shows how the twitter data looks when downloaded. The csv data has the date when the tweet was created, the location where the tweet was created, the text of the tweet and its polarity.

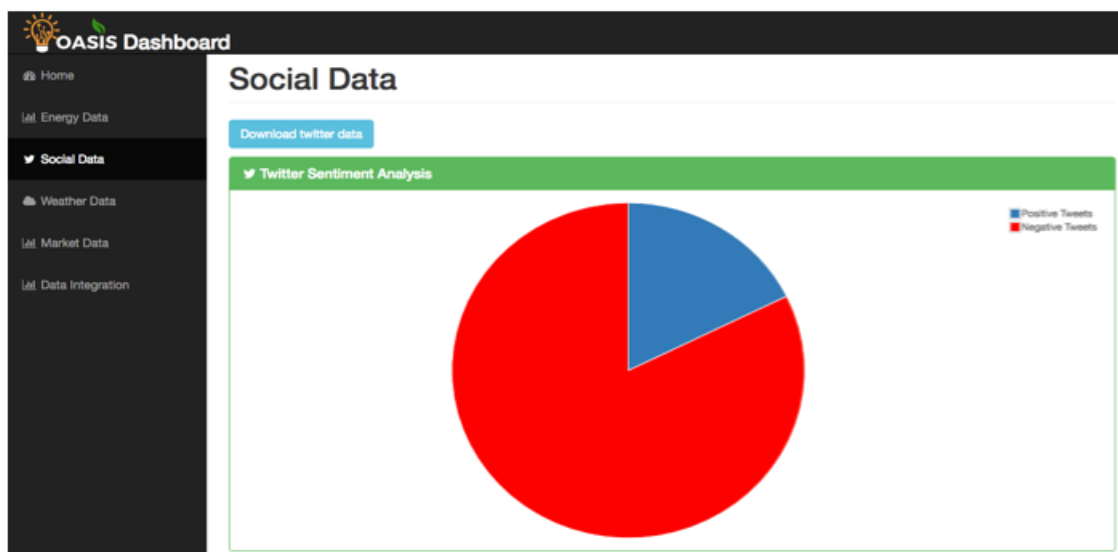


Fig 3.2: Pie chart from social data section of the OASIS Dashboard

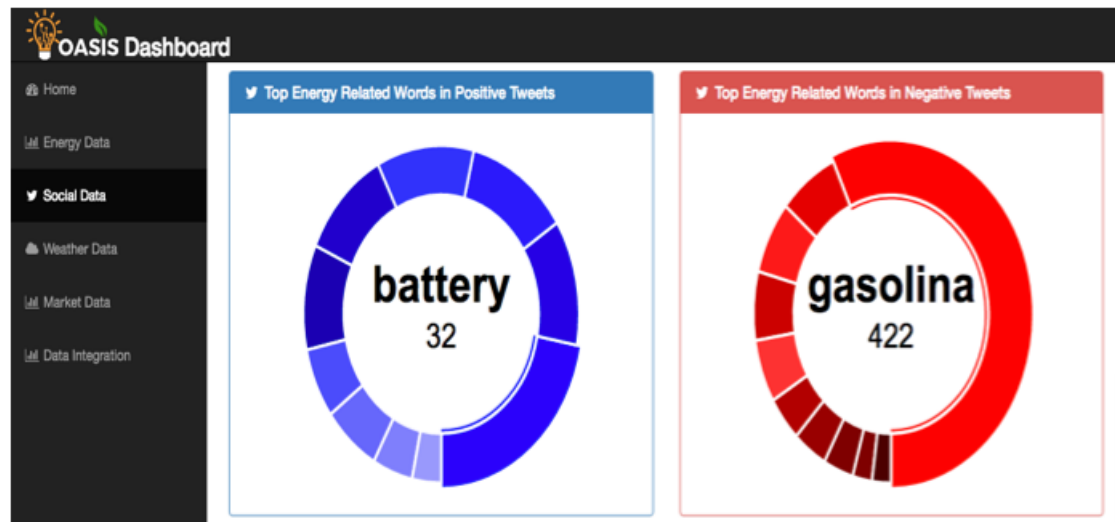


Fig 3.3: Donut charts from social data section of the OASIS dashboard

	A	B	C	D
1	created_at	location	text	polarity
2	Fri Mar 03 18:14:58 +0000 2017	Puerto Rico, USA	RT @pitbull:	positive
3	Fri Mar 03 18:16:44 +0000 2017	San Juan, Puerto Rico	RT @_misiass	positive
4	Fri Mar 03 18:54:36 +0000 2017	Puerto Rico, USA	RT @gamein	positive
5	Fri Mar 03 21:46:01 +0000 2017	Puerto Rico	RT BleacherF	positive
6	Fri Mar 03 21:48:07 +0000 2017	Puerto Rico	Shaq vows to	positive
7	Fri Mar 03 22:20:08 +0000 2017	Luquillo, Puerto Rico	RT @chrislha	positive
8	Sat Mar 04 00:35:13 +0000 2017	Puerto Rico, USA	RT @untouch	positive
9	Sat Mar 04 15:54:16 +0000 2017	Puerto Rico and Tampa	RT @chrislha	positive
10	Sat Mar 04 16:29:13 +0000 2017	Puerto Rico	Steelers, Tita	positive
11	Sat Mar 04 17:03:13 +0000 2017	Puerto Rico & USA	RT @TheDoc	positive
12	Sat Mar 04 17:34:35 +0000 2017	Puerto Rico	US shale pro	positive
13	Sat Mar 04 18:10:26 +0000 2017	Puerto Rico	RT @insidecl	positive
14	Sat Mar 04 19:19:14 +0000 2017	Puerto Rico	RT @greenp	positive
15	Sat Mar 04 22:01:42 +0000 2017	Puerto Rico	Anyone here	positive
16	Sun Mar 05 16:46:14 +0000 2017	Puerto Rico, USA	no mires que	positive
17	Sun Mar 05 17:07:30 +0000 2017	Puerto Rico	Check out	positive
18	Sun Mar 05 17:08:09 +0000 2017	Puerto Rico	Check out	positive
19	Sun Mar 05 17:17:54 +0000 2017	Fajardo, Puerto Rico	RT @OWante	positive
20	Sun Mar 05 21:15:10 +0000 2017	Puerto Rico	RT @Megalo	positive
21	Mon Mar 06 15:39:29 +0000 2017	Puerto Rico	WHO: Polluti	positive
22	Mon Mar 06 15:39:29 +0000 2017	Puerto Rico	WHO: Polluti	positive
23	Mon Mar 06 15:42:15 +0000 2017	Puerto Rico	RT @ruthart	positive
24	Tue Mar 07 17:43:11 +0000 2017	Puerto Rico	still no	positive
25	Tue Mar 07 17:45:35 +0000 2017	Puerto Rico, USA	@Watt pad	positive

Fig 3.4: Twitter data in csv format

### 3.3 Weather Data Section

Weather data is being gathered using the Dark Sky API. With this API we can present users with the current condition, today's forecast; today's 12-hour forecast with the date brief description of the forecast, precipitation percentage, temperature in Fahrenheit or



Celsius, humidity percentage and wind speed and direction; and 4-day forecast for a given town. Users may search for a town by name or zip code. This area also shows users radar images for Puerto Rico. The hourly forecast for the next 24 hours can be exported as a csv file. The process of gathering, analysis and visualization of this data is explained in section 4.2. Figure 3.5 shows the first portion of the weather data section, which includes the current weather for a town, today's forecast and a 12-hour forecast. Figure 3.6 shows the second portion of the weather data section, which shows the 4-day forecast and a map of Puerto Rico showing details like precipitation and temperature. Figure 3.7 shows the downloaded data in csv format.

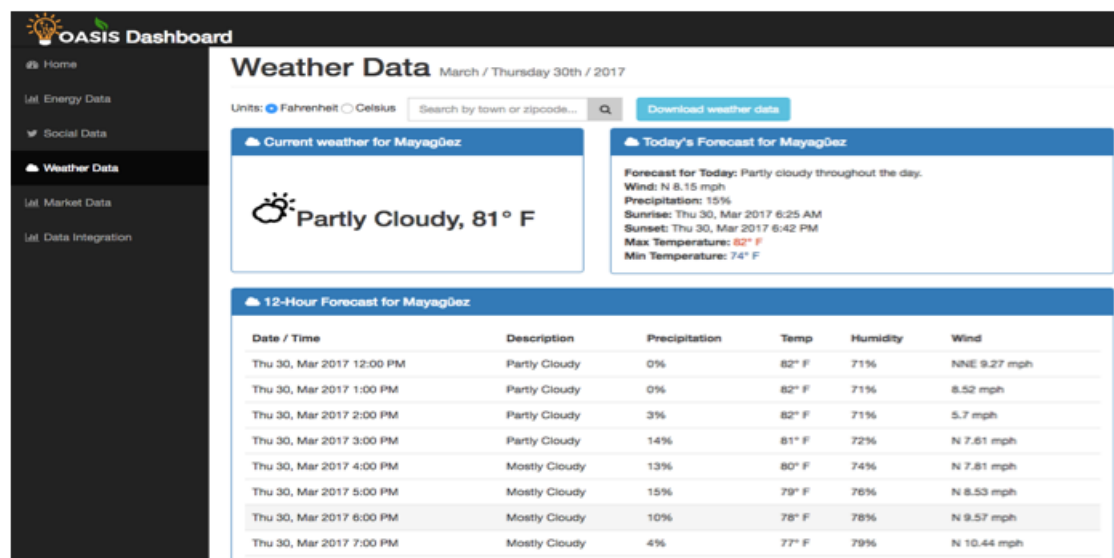


Fig 3.5: Current weather, today's forecast and 12-hour forecast from weather data section of the OASIS Dashboard

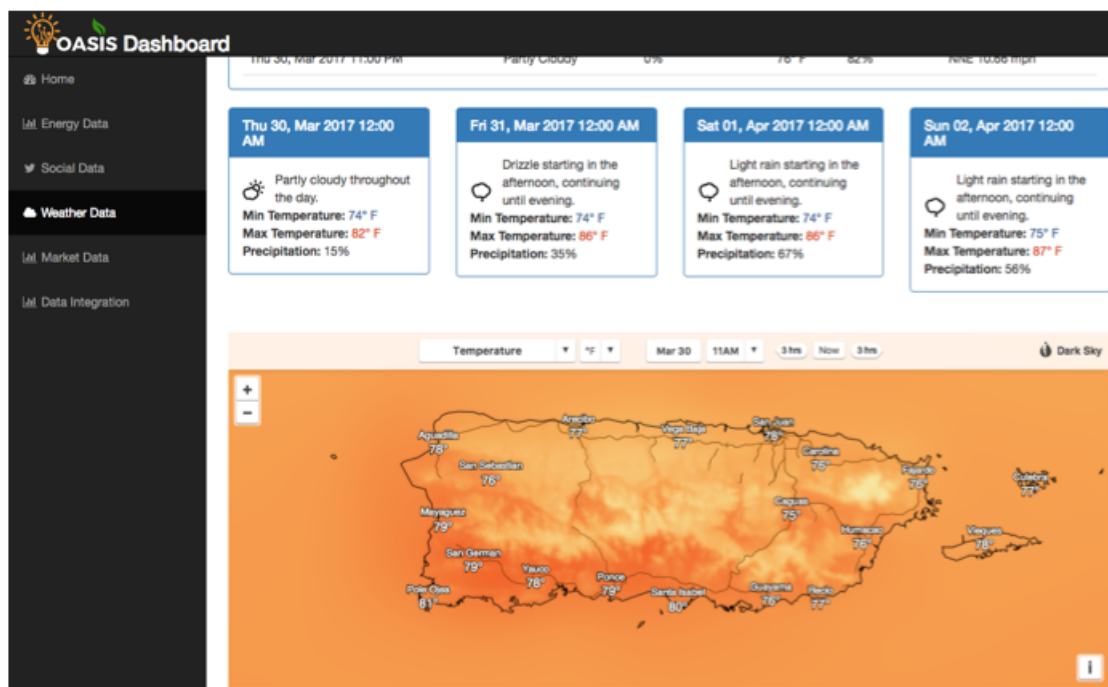


Fig 3.6: 4-day forecast and Puerto Rico map showing temperature data from weather data section of the OASIS Dashboard

	A	B	C	D	E	F
1	date	description	precipitation	temperature	humidity	wind
2	Thu 30, Mar 2017 12:00 AM	Partly Cloudy	0%	75 F	82%	6.56 mph
3	Thu 30, Mar 2017 1:00 AM	Partly Cloudy	0%	75 F	83%	6.73 mph
4	Thu 30, Mar 2017 2:00 AM	Partly Cloudy	0%	75 F	85%	N 6.98 mph
5	Thu 30, Mar 2017 3:00 AM	Mostly Cloudy	0%	75 F	85%	N 7.07 mph
6	Thu 30, Mar 2017 4:00 AM	Mostly Cloudy	5%	75 F	85%	N 7.06 mph
7	Thu 30, Mar 2017 5:00 AM	Mostly Cloudy	0%	75 F	86%	N 7.33 mph
8	Thu 30, Mar 2017 6:00 AM	Mostly Cloudy	0%	75 F	86%	N 7.32 mph
9	Thu 30, Mar 2017 7:00 AM	Mostly Cloudy	0%	76 F	85%	N 7.58 mph
10	Thu 30, Mar 2017 8:00 AM	Partly Cloudy	0%	79 F	81%	N 8.03 mph
11	Thu 30, Mar 2017 9:00 AM	Mostly Cloudy	0%	81 F	79%	NNE 8.69 mph
12	Thu 30, Mar 2017 10:00 AM	Mostly Cloudy	0%	83 F	76%	NNE 8.92 mph
13	Thu 30, Mar 2017 11:00 AM	Mostly Cloudy	0%	84 F	74%	N 9.55 mph
14	Thu 30, Mar 2017 12:00 PM	Mostly Cloudy	0%	85 F	73%	N 10.1 mph
15	Thu 30, Mar 2017 1:00 PM	Partly Cloudy	0%	86 F	72%	N 10.14 mph
16	Thu 30, Mar 2017 2:00 PM	Partly Cloudy	1%	85 F	73%	N 9.92 mph
17	Thu 30, Mar 2017 3:00 PM	Partly Cloudy	2%	82 F	74%	N 10.26 mph
18	Thu 30, Mar 2017 4:00 PM	Partly Cloudy	3%	80 F	75%	N 10.71 mph
19	Thu 30, Mar 2017 5:00 PM	Mostly Cloudy	3%	78 F	75%	NNE 11.17 mph
20	Thu 30, Mar 2017 6:00 PM	Partly Cloudy	2%	78 F	78%	NNE 11.37 mph
21	Thu 30, Mar 2017 7:00 PM	Partly Cloudy	1%	77 F	80%	NNE 11.28 mph
22	Thu 30, Mar 2017 8:00 PM	Partly Cloudy	0%	75 F	81%	NNE 11.27 mph
23	Thu 30, Mar 2017 9:00 PM	Partly Cloudy	0%	75 F	82%	NNE 11.24 mph
24	Thu 30, Mar 2017 10:00 PM	Partly Cloudy	0%	74 F	81%	NNE 11.21 mph
25	Thu 30, Mar 2017 11:00 PM	Partly Cloudy	0%	74 F	83%	NNE 11.08 mph

Fig 3.7: Weather data in csv format

### 3.4 Market Data Section

The Market data section will show users loads in MWhr and Day-ahead Locational Based Marginal Pricing gathered from the NYISO. Also, a graph comparing loads with Day-Ahead Locational Based Marginal Pricing (LBMP) is shown. This data will be shown in bar charts and line charts. Also, users will be able to compare load times vs price by viewing a bar-line chart that shows both the load and prices. This data will be exportable as a csv file. The process of gathering this data, cleaning, analysis, and visualization is explained in section 4.4. Figure 3.8 shows a bar graph comparing loads per hour in a day, while Figure 3.9 shows a line chart comparing the LBMP for different zones per hour in a day. In Figure 3.10 a bar-line chart compares the load data with the zonal data.

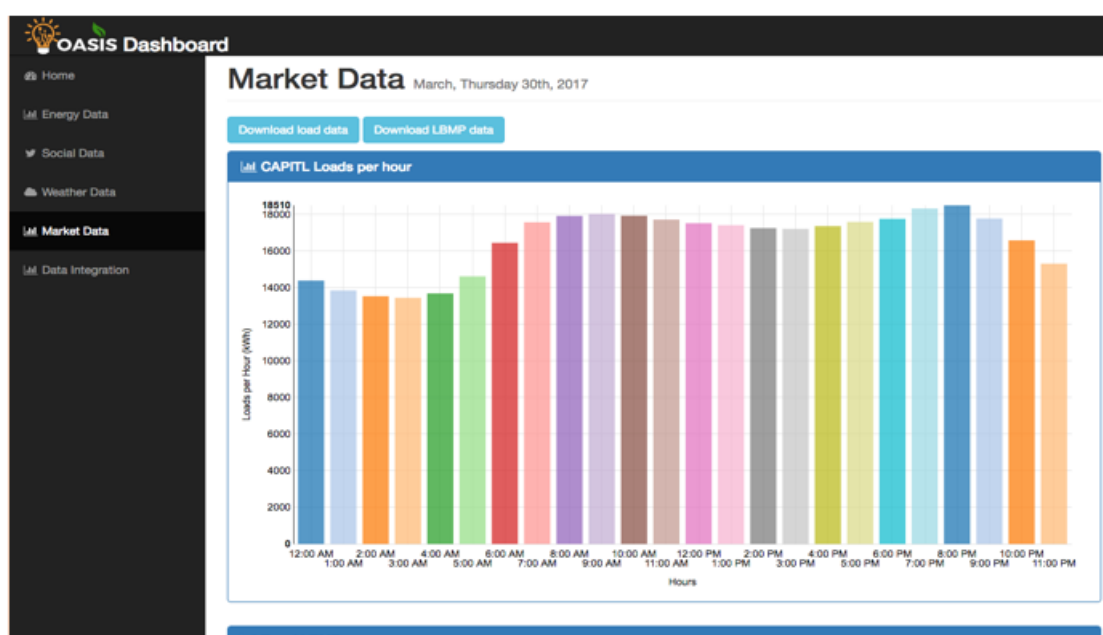


Fig 3.8: Load data from market data section of the OASIS Dashboard

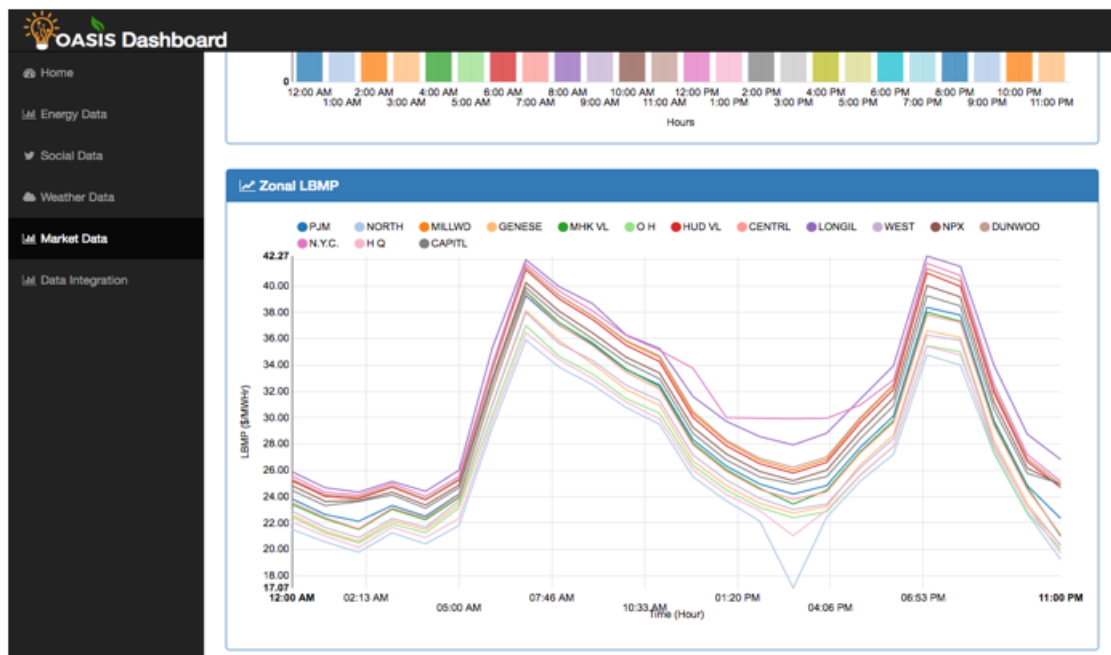


Fig 3.9: Zonal LBMP data from market data section of the OASIS Dashboard

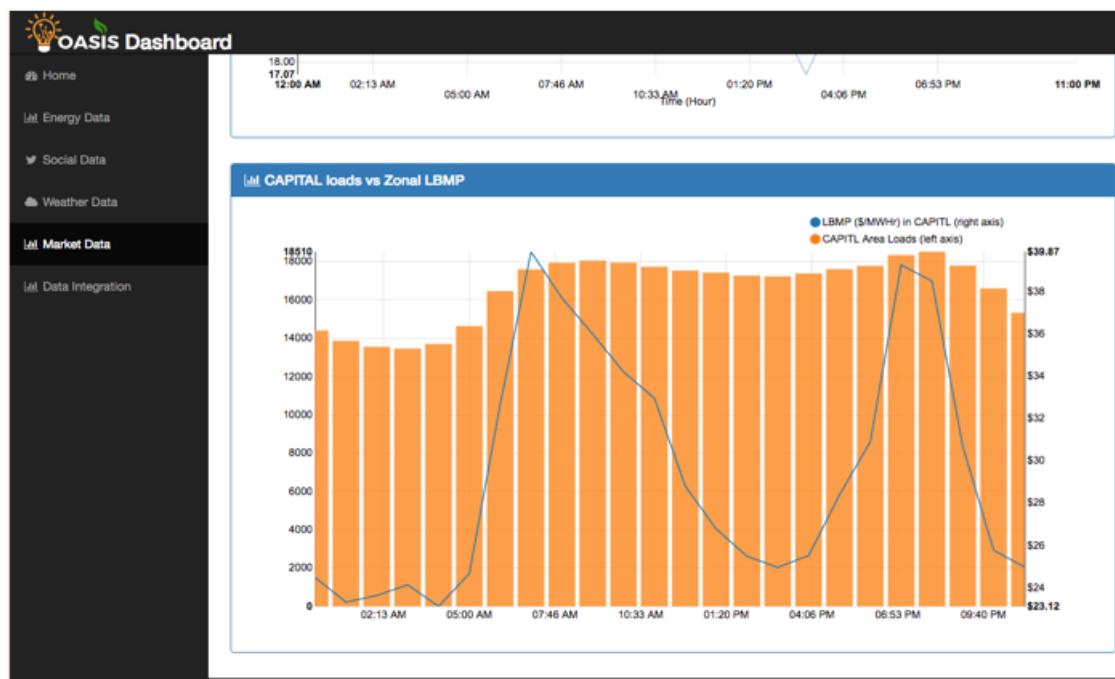


Fig 3.10: Load vs Zonal LBMP data from market data section of the OASIS Dashboard

For this project, we decided to use the NYISO market data because there is no market data from Puerto Rico. One of the problems in Puerto Rico regarding energy, is that the

market is essentially the AEE and the consumers. Another problem is that daily data from the AEE is nonexistent. This is one of the problems that the OASIS project is planning on resolving with an open market for electricity consumers and producers.

### 3.5 Data Integration Section

The Data Integration section is the area in the dashboard where users will be able to make comparisons between the different datasets. Users will be able to compare the following:

- Humidity data per hour for a given town
- Precipitation data per hour for a given town
- Wind data per hour for a given town
- Temperature data per hour for a given town in Fahrenheit or Celsius
- Load data
- Load based marginal pricing data per zone
- Positive Tweets
- Negative Tweets

Up to four of these datasets may be compared. These datasets may be imported or exported. The process of integrating, analyzing and visualizing this data is explained in section 4.5. Figure 3.11, shows an example comparison line chart between positive and negative tweets. Figure 3.12 shows an example of how create a graph based on the datasets provided by the dashboard. Figure 3.13 shows how the compared data looks in csv format when a user downloads it.

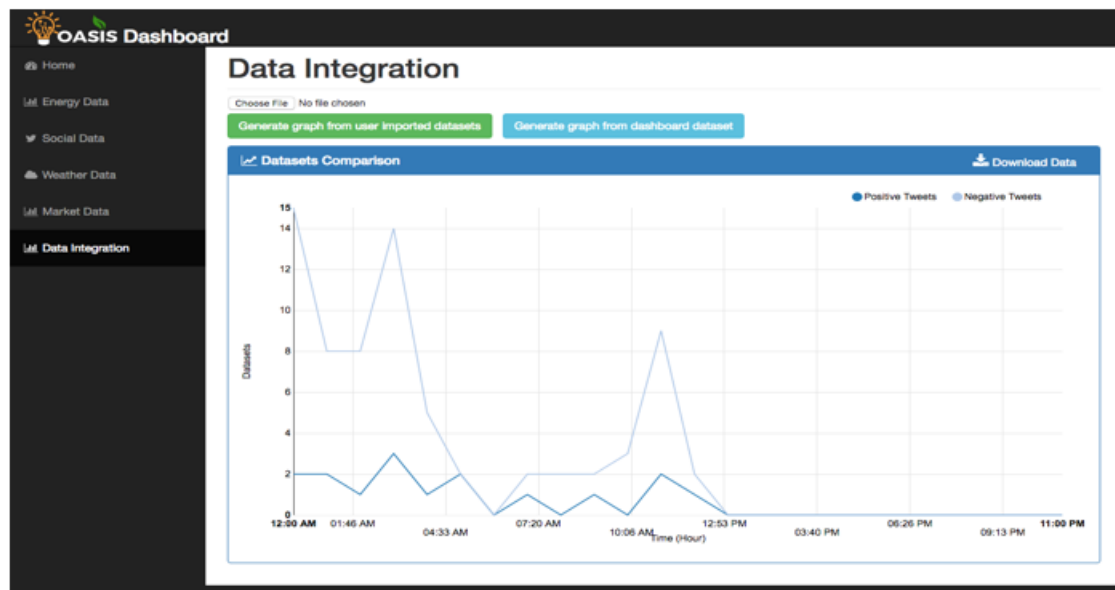


Fig 3.11: Process of creating graph from data integration section of the dashboard

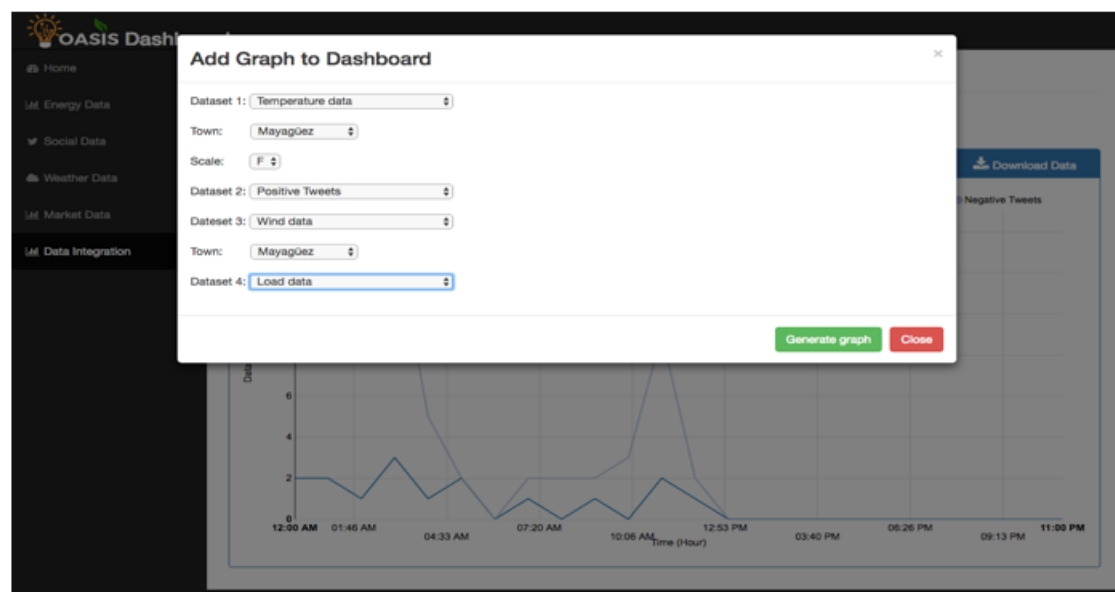


Fig 3.12: Process of creating a graph in the data integration section of the OASIS Dashboard

	A	B	C	D
1	date	humidity_Adjuntas	Positive Tweets	Negative Tweets
2	3/29/17 0:00	0.91	0	0
3	3/29/17 1:00	0.91	0	0
4	3/29/17 2:00	0.91	8	22
5	3/29/17 3:00	0.91	6	9
6	3/29/17 4:00	0.91	5	14
7	3/29/17 5:00	0.91	2	7
8	3/29/17 6:00	0.91	1	2
9	3/29/17 7:00	0.91	3	3
10	3/29/17 8:00	0.88	1	2
11	3/29/17 9:00	0.83	0	11
12	3/29/17 10:00	0.78	3	17
13	3/29/17 11:00	0.75	4	8
14	3/29/17 12:00	0.73	1	8
15	3/29/17 13:00	0.73	1	9
16	3/29/17 14:00	0.75	4	15
17	3/29/17 15:00	0.76	6	9
18	3/29/17 16:00	0.78	5	40
19	3/29/17 17:00	0.81	1	6
20	3/29/17 18:00	0.84	0	0
21	3/29/17 19:00	0.87	0	0
22	3/29/17 20:00	0.9	0	0
23	3/29/17 21:00	0.9	0	0
24	3/29/17 22:00	0.9	0	0
25	3/29/17 23:00	0.9	0	0

Fig 3.13: Data comparison between humidity in Adjuntas Puerto Rico, positive tweets and negative tweets in csv format

### 3.6 How to access the web dashboard

There are two ways in which users can install the web dashboard. Users can either enter the to the following url: <http://oasisdash.uprm.edu/> on a web browser or manually install the dashboard in their machine. The dashboard may be installed to run in MacOS or Ubuntu.

To install the dashboard in Mac or Ubuntu, first the user must have installed Python in their computer. By default, MacOS and Ubuntu come with Python preinstalled so this should not be a problem. After this, the user must make sure that the following Python libraries are installed: Schedule, Forecastiopy, NLTK, Tweepy, Unidecode and Flask. To install each of these libraries, the users must first install pip, which can be done in Ubuntu by opening and typing in the terminal:

```
sudo apt-get install python-setuptools python-dev build-essential  
sudo easy_install pip
```

For Mac users, open the terminal and type:

```
curl https://bootstrap.pypa.io/ez_setup.py -o | sudo python  
sudo easy_install pip
```

With pip is installed, the user can proceed to install the necessary libraries by typing the following commands in the terminal:

```
sudo pip install schedule  
sudo pip install python-forecastio  
sudo pip install nltk  
sudo pip install unicode  
sudo pip install tweepy  
sudo pip install flask
```

After the user has finished typing all those commands, the user may now download the dashboard by going to the desired folder in the terminal where the dashboard will be stored. For example, the dashboard may be stored in */User/me/Desktop*, where *me* is the name of the user of the computer. Assuming we want to store the dashboard in the Desktop in a folder called *oasis\_dashboard*, we type the following in the terminal:



```
cd  
  
cd Desktop/  
  
mkdir oasis_dashboard  
  
cd oasis_dashboard  
  
git init  
  
git clone https://github.com/alesanz237/oasis_dashboard.git  
  
python startDashboard.py
```

Once the user has finalized typing these commands in the terminal, the dashboard should be successfully installed. To verify that the dashboard is running, the terminal should show say that the dashboard is running on port 5000. If everything was installed successfully, the user may go the web browser and type in the url localhost:5000 to use the dashboard.

# Chapter 4

## WEB DASHBOARD

## DEVELOPMENT PROCESS

As stated previously, this dashboard visualizes energy data, weather data, social data, market data and the comparison between each dataset. Chapter 4 describes for each of these datasets how the data was gathered, cleaned, analyzed and visualized; this is described in sections 4.1 to 4.4. Finally, we describe how we integrate each of these datasets, clean them and visualize them in section 4.5.

### 4.1 Energy Data

In section 3.2, we described how the energy data looks like. In this section, we describe how we gather this data, explaining the process of how we convert csv data to JSON data and how we visualize that JSON data.

#### 4.1.1 Data Gathering

The data for the Energy Data section in the dashboard is gathered from a csv file provided by the AEE with historic data from Puerto Rico. To gather this data, we converted the

data from csv to JSON data using a class that we had implemented in Java. The data in a csv file looks as shown in Figure 4.1. Using the Java class, we successfully transformed the data into a JSON data file which looks as shown in Figure 4.2.

Generación Neta													Acumulado Año Fiscal a octubre	% cambio anual fiscal	Año Fiscal	% cambio fiscal	Año Natural	% cambio Natural
Año Fiscal	julio	agosto	septiembre	octubre	noviembre	diciembre	enero	febrero	marzo	abril	mayo	junio						
2000	1,765.3	1,835.6	1,787.9	1,780.8	1,653.0	1,651.2	1,551.7	1,526.2	1,632.5	1,644.4	1,792.0	1,774.4	7,169.6		20,397.0		20,728.9	
2001	1,631.6	1,907.4	1,795.3	1,834.9	1,696.2	1,736.3	1,665.3	1,501.9	1,727.1	1,673.9	1,841.4	1,793.3	7,369.2	2.78	21,008.8	3.00	21,203.2	2.29
2002	1,848.1	1,947.8	1,838.8	1,889.1	1,730.7	1,748.7	1,716.2	1,641.2	1,730.6	1,730.5	1,892.2	1,874.9	7,520.7	2.06	21,485.7	2.27	22,086.1	4.16
2003	1,945.1	2,010.0	1,930.5	1,981.5	1,869.2	1,864.2	1,794.6	1,659.1	1,885.8	1,816.8	1,966.7	1,917.9	7,867.1	4.81	22,841.4	5.38	22,884.4	3.61
2004	1,962.0	2,045.5	2,026.1	2,059.5	1,832.5	1,896.0	1,772.8	1,717.5	1,854.3	1,967.6	1,947.1	2,016.4	8,125.0	3.28	23,515.1	1.85	23,304.9	0.93
2005	2,025.2	2,135.3	1,834.8	2,048.4	1,847.6	1,942.0	1,780.9	1,623.5	1,985.4	1,993.3	2,076.7	2,077.6	8,043.6	-1.00	23,375.6	1.87	23,946.6	3.66
2006	2,050.3	2,166.7	2,116.7	2,046.5	1,955.7	1,925.3	1,821.8	1,758.1	1,923.7	1,876.3	2,072.1	2,045.8	8,420.2	4.72	23,793.9	1.62	23,839.8	-0.03
2007	2,050.3	2,166.7	2,085.6	2,129.7	1,999.8	1,969.4	1,868.0	1,759.0	1,961.8	1,903.5	2,120.0	2,060.0	8,420.9	-0.03	24,062.4	1.30	23,720.2	-0.90
2008	2,085.1	2,132.7	2,024.5	2,056.1	1,889.4	1,858.2	1,780.9	1,682.5	1,788.3	1,813.9	1,961.4	1,939.7	8,300.3	-1.43	23,014.7	-4.35	22,337.2	-6.83
2009	1,977.4	2,033.8	1,883.0	1,941.6	1,782.1	1,752.5	1,721.3	1,557.6	1,686.4	1,736.8	1,816.9	1,873.7	7,835.9	-6.60	21,763.2	-5.44	22,022.4	-1.41
2010	1,992.5	2,018.8	1,939.6	1,997.9	1,819.4	1,861.3	1,732.1	1,659.9	1,901.0	1,812.9	1,922.5	1,903.5	7,948.9	1.44	22,561.6	3.67	22,216.1	0.88
2011	1,942.4	2,014.4	1,924.0	1,944.8	1,750.2	1,758.2	1,688.1	1,554.8	1,732.8	1,711.1	1,809.4	1,896.1	7,825.7	-1.55	21,836.4	-4.10	21,283.6	-4.20
2012	1,848.5	1,870.7	1,863.5	1,892.5	1,748.3	1,757.8	1,636.1	1,563.8	1,676.7	1,689.1	1,819.5	1,897.1	7,475.3	-4.48	21,203.6	-2.00	21,296.3	0.06
2013	1,854.8	1,906.2	1,879.8	1,896.2	1,774.2	1,791.1	1,636.9	1,490.1	1,670.6	1,681.1	1,773.6	1,732.7	7,548.9	0.99	21,509.0	-0.92	20,620.9	-3.17
2014	1,790.1	1,866.7	1,816.1	1,844.4	1,676.0	1,629.8	1,609.7	1,467.2	1,674.9	1,649.3	1,715.3	1,755.8	7,330.3	-2.90	20,506.2	-2.36	20,272.2	-1.09
2015	1,803.2	1,807.4	1,732.3	1,823.5	1,623.7	1,609.9	1,568.2	1,496.0	1,591.2	1,621.0	1,736.2	1,734.6	7,166.4	-2.24	20,107.3	-4.29		
2016	1,790.3	1,796.7	1,749.5	1,792.1									7,116.7	-0.67				

Fig 4.1: AEE energy data in csv file

```
{
  "Demand and Generation(*millions of kilowatts per hour(mkWh))" : {
    "Gross Generation" : {
      "2000" : {
        "07" : 1859.43646,
        "08" : 1927.77842,
        "09" : 1878.75264,
        "10" : 1876.87378,
        "11" : 1746.47682,
        "12" : 1747.2306,
        "01" : 1637.01906,
        "02" : 1607.41398,
        "03" : 1716.46042,
        "04" : 1722.45842,
        "05" : 1875.81493,
        "06" : 1865.703801
      }
    }
  }
}
```

Fig 4.2: Example AEE JSON data

#### 4.1.2 Data Modification

After gathering the data, we proceed to modify the data so that it can be visualized. The NVD3 stacked chart requires the data to look as shown in Figure 4.3. The data in Figure 4.3 can be interpreted as an array containing multiple dictionaries, each dictionary has two key value pairs. The first pair, called key, is the label that is going to be displayed

for the graph. For example, Net Generation will be displayed for the values in the first dictionary, Gross Generation in the second and so forth. The second pair, called values, is an array of arrays which contains two values: a long number which is the date presentation in epoch time, and the value for that time. We used Python to convert the data into the format needed for the chart. This modified data is then made available so that it can be visualized as soon as the user enters the energy data section in the dashboard.

```
[
  {
    "key": "Net Generation",
    "values": [[ 1138683600000, 27.38],
               [ 1141102800000, 27.72],
               [ 1143781200000, 26.91]]
  },
  {
    "key": "Gross Generation",
    "values": [[ 1138683600000, 35.31],
               [ 1141102800000, 27.37],
               [ 1143781200000, 29.30]]
  },
  {
    "key": "Max Demand (MW)",
    "values": [[ 1138683600000, 27.23],
               [ 1141102800000, 25.73],
               [ 1143781200000, 27.10]]
  }
]
```

Fig 4.3: Example JSON that NVD3 is expecting

### 4.1.3 Data Visualization

After the data has been converted to JSON and modified, we can begin the process of visualizing the energy data. As described in section 3.1, we created 6 stacked charts showing the data from the AEE. To visualize this data, we used NVD3 and the data that was gathered and cleaned, as described in the two subsections above. These graphs are generated every time as soon as the user enters the energy data section.

These stacked charts are interactive, and users can scroll through the chart and a popup that shows relevant information about the dataset. Aside from that the user may select any of the buttons in the top right side of the graph, which represent attributes of

the data, and remove the datasets that are not important.

## 4.2 Social Data

### 4.2.1 Data Gathering

To gather twitter data, we executed the following steps:

1. Create a Twitter developer account
2. Acquire the following credentials for utilizing the Twitter API
  - (a) access\_token
  - (b) access\_token\_secret
  - (c) consumer\_key
  - (d) consumer\_secret
3. Generate a list of words in English and Spanish related to energy data and smart grids. This list contains 98 English words and 27 Spanish words for a total of 125 words.
4. Using the list generated in step 3 and the Twitter Streaming API we listen to the tweets in real time that are being published to twitter and contain any of the words in the energy keywords list. Those tweets are stored in a file called tweets.txt. Each tweet is a JSON object that looks like the one in figure 4.4.



### 4.2.3 Data Analysis

The twitter data is analyzed in two steps: (i) Training set creation and (ii) Tweet classification. To generate the training set we use the Sentiment Labelled Sentences dataset provided by University of California, Irvine. This dataset contains 3 files with 1000 sentences each. These sentences come from user reviews on Amazon, IMDB and Yelp. Each of these sentences are classified with a 0 or a 1. 0 means that the sentence is classified as negative, while 1 means that the sentence is classified as positive. There are 500 positive sentences and 500 negative sentences per file.

For each of these sentences we select the words that are most important and remove the less important words using Python's NLTK library, which is a very popular natural language processing library. Each of the files is read and changed into a format that is readable by a Naïve Bayes classifier that will be utilized to determine polarity of the tweets. The code parses the training data into an array that has multiple tuples of the type (sentence, polarity).

Once we have the training set, training of the classifier can begin. We pass as input the training set to the classifier and this returns a more qualified classifier that is more suitable for determining polarity of tweets. After the classifier is properly trained, we begin the process of classifying tweets. This step is like the process we did when generating the training set. First, we read all the tweets from tweets.txt, then we clean them (as described in 4.2.2), then we take the sentence field and extract the most important words from that sentence. After that we pass the words to our classifier, which then returns a classified tweet.

After we are done assigning a polarity to the tweet we then add a new field to our clean tweets. This field is called polarity; its value may be positive to signify that the tweet was classified as positive and negative to signify that the tweet was classified as negative. Tweets will now have the structure shown in figure 4.6. Finally, classified

tweets are stored in files based on polarity, positive tweets will be stored in positive.txt and negative tweets will be stored in negative.txt.

A challenge that we found while processing tweets was the occurrence of Spanish words. Since we have a combination of English and Spanish words for the sentiment analysis of our social data, a multi-lingual classifier needs to be implemented.

Another interesting issue is that our sentiment analysis is currently static. Online machine learning algorithms need to be explored so we can capture the dynamic nature of Twitter streaming data. It means a dynamic classifier can be used to learn new training sets as the Twitter streaming processing evolves.

```
{'polarity': 'positive', 'text': u'Deforestation is pushing endangered species to the brink. Tell @TIAA to cut ties w/ reckless palm oil producers! https://t.co/7WJviv6TYv', 'created_at': u'Mon Mar 13 13:25:44 +0000 2017', 'location': u'Puerto Rico '}
```

Fig 4.6: Example cleaned tweet

#### 4.2.4 Data Visualization

With the data now cleaned and classified we can proceed to use NVD3 to visualize it. We decided to use two types of charts for this twitter data, the first one is a pie chart and the second is a donut chart. The pie chart that is being created shows the percentage of positive and negative tweets related to smart grids in Puerto Rico. Two donuts charts represent the top 10 most repeated words in positive and negative tweets. To properly display these graphs, we had to convert the cleaned tweets into a format that the NVD3 could understand. Aside from that we are also using JavaScript to display a stream of tweets. Every 10 seconds, 10 new classified tweets appear in the stream. All tweets and data that has been classified as positive will appear as blue on the dashboard and all tweets that have been classified as negative will appear as red on the dashboard. As is the nature with NVD3 charts, the donut and pie charts are interactive. Section 3.2 shows figures of how the graphs utilized for the social data section.



## 4.3 Weather Data

In this section, we describe how we gather the weather data from the dark sky, then how we cleaned the data and analyzed it using Python. Finally, we explain how the data was visualized.

### 4.3.1 Data Gathering, cleaning and analysis

To access this API, the developer must first create an account in Dark Sky as a developer. The data is being gathered using Python and the forecastiopy library. Using this library, we created various functions for gathering weather data. The first of these functions is called `getHourlyWeather`; with this function, we gather weather information for the next 12 hours. Dark Sky provides many attributes but we are only interested in the date, description of the forecast, precipitation probability, humidity percentage, cloud coverage percentage and temperature in Fahrenheit or Celsius. The second of these functions is called `getTodaysWeather`; with this function we gather, the current forecast for today. From this function, we get the current forecast, the forecast for the rest of the day, the wind speed and direction, the precipitation probability, sunrise, sunset, min and max temperatures in Fahrenheit or Celsius. The third function that we created is called `getDailyWeather`; this function returns the forecast for the next 4 days. From this function, we are gathering the forecast for the day, the max and min temperatures and the precipitation probability. These three functions return the data as JSON files that look as shown in figure 4.7.

```
[{u'icon': u'partly-cloudy-day',
  u'precipProbability': '45%',
  u'summary': u'Partly cloudy throughout the day.',
  u'temperatureMax': '84',
  u'temperatureMin': '72',
  u'time': 'Thu 30, Mar 2017 12:00 AM'},
 {u'icon': u'rain',
  u'precipProbability': '23%',
  u'summary': u'Drizzle in the evening.',
  u'temperatureMax': '87',
  u'temperatureMin': '75',
  u'time': 'Fri 31, Mar 2017 12:00 AM'},
 {u'icon': u'rain',
  u'precipProbability': '73%',
  u'summary': u'Light rain starting in the afternoon, continuing until evening.',
  u'temperatureMax': '85',
  u'temperatureMin': '75',
  u'time': 'Sat 01, Apr 2017 12:00 AM'},
 {u'icon': u'rain',
  u'precipProbability': '54%',
  u'summary': u'Light rain starting in the afternoon, continuing until evening.',
  u'temperatureMax': '87',
  u'temperatureMin': '75',
  u'time': 'Sun 02, Apr 2017 12:00 AM'}]
```

Fig 4.7: JSON File of weather data

### 4.3.2 Data Visualization

Now that we have successfully gathered the weather data, the process of visualizing the data can begin. For the weather section, we are not actually generating any graphs using NVD3, instead we are using a combination of HTML, JavaScript, JQuery and CSS to display the data. NVD3 is not being utilized, because after speaking with potential users of the dashboard, they expressed that it would be better to have textual data and that the only visual data should be a map of Puerto Rico, describing factors like the current precipitation or wind speed. Section 3.3 shows figures of how this part of the dashboard looks like.

## 4.4 Market Data

This section focuses on how the market data, provided by the NYISO, was gathered, analyzed and visualized.

#### 4.4.1 Data Gathering

For this dashboard, we are interested in two datasets from the NYISO: load data and zonal LBMP data. This data is publicly available on the web and is provided as a csv file. To gather the load data, we create a function in Python that reads the csv and stores it in a variable. In this same function, we then convert that data from CSV format to JSON format as shown in figure 4.8. The load data is divided by year, month and day. For each hour of the day an array stores the load information. The process for zonal LBMP data is the same, except that we finish that the generated JSON looks different.

```
{'2017': {1: 1: [15665,
                14998,
                14438,
                14064,
                13932,
                14062,
                14366],
          2: [14352,
              13425,
              15325,
              15223,
              14992,
              14123,
              14234]}}
```

Fig 4.8: Example JSON of how load data looks like

#### 4.4.2 Data Modification

To properly visualize the data, we need to modify it so that it complies with the NVD3 chart. For example, to create a bar chart of load data, as shown in section 3.4, the data must be modified so that it looks as shown in figure 4.9. Therefore, the data was to be converted to an array with a JSON that has two attributes: the key and the values. The most important is the values field, which contains an array of JSONs in which the labels field is the hour and the value field is the load in that hour. This data is then made

accessible to the dashboard as a get request through our flask server.

```
[{'key': '20170331-loadData',
  'values': [{'label': '12:00 AM', 'value': 14431},
              {'label': '1:00 AM', 'value': 13947},
              {'label': '2:00 AM', 'value': 13662},
              {'label': '3:00 AM', 'value': 13635},
              {'label': '4:00 AM', 'value': 13921},
              {'label': '5:00 AM', 'value': 14926},
              {'label': '6:00 AM', 'value': 16759},
              {'label': '7:00 AM', 'value': 17877},
              {'label': '8:00 AM', 'value': 18172},
              {'label': '9:00 AM', 'value': 18260},
              {'label': '10:00 AM', 'value': 18191},
              {'label': '11:00 AM', 'value': 18007},
              {'label': '12:00 PM', 'value': 17745},
              {'label': '1:00 PM', 'value': 17675},
              {'label': '2:00 PM', 'value': 17668},
              {'label': '3:00 PM', 'value': 17830},
              {'label': '4:00 PM', 'value': 18112},
              {'label': '5:00 PM', 'value': 18450},
              {'label': '6:00 PM', 'value': 18682},
              {'label': '7:00 PM', 'value': 19036},
              {'label': '8:00 PM', 'value': 18771},
              {'label': '9:00 PM', 'value': 17960},
              {'label': '10:00 PM', 'value': 16750},
              {'label': '11:00 PM', 'value': 15465}]]}]
```

Fig 4.9: Modified JSON of load data

#### 4.4.3 Data Visualization

After converting the data, as explained in 4.4.2, we proceed to create three different types of charts for visualizing the market data. These charts are a bar chart, a line chart and a bar-line chart. The bar chart is utilized to represent the load data, the line chart represents the zonal LBMP data and the bar-line chart represents the comparison between one LBMP zone vs load data. These graphs are interactive. Section 3.4, has figures that demonstrate how the visualized market data looks like.

#### 4.5 Comparison between datasets

This section focuses on how the social, weather and market data was integrated together and visualized.

#### 4.5.1 Data Integration

To integrate the data, we have used Python. Several functions have been created which return the value of the dataset per hour of the data that we want to compare. To gather this data, we created 8 functions in Python that return either the (i) humidity data per hour for a given town, (ii) the precipitation data per hour for a given town, (iii) wind data per hour for a given town, (iv) temperature data per hour for a given town, (v) load data, (vi) LBMP data per zone, (vii) positive tweets per hour in a day and (viii) negative tweets per hour in a day. Basically, these functions call the functions that were described in the previous sections from this chapter and gather only the data that the user is interested in. We then created a class in Python that will handle the integration of this data. This class can store up to 4 different datasets for comparison and its main task is to merge those datasets into one dataset so that it can be visualized.

#### 4.5.2 Data Visualization

After integrating the data, we can start the process of visualization. For this section, we have decided to utilize a line chart because the type of data that we are comparing will be easier for the user to understand with this type of chart. The visualization of this chart works similarly to how it has worked in previous sections of the dashboard. As with the other graphs generated for this dashboard, we use NVD3 to visualize the data. Section 3.6, has examples of how this visualized data looks like.

## Chapter 5

# CONCLUSION AND FUTURE WORK

### 5.1 Conclusion

As stated previously in this document, this project has two main objectives. The first objective was to study and review the best practices in the collection, analysis and visualization of data from smart grids. The second objective was to implement a smart grid data workflow that includes data collection from diverse sources and interactive visualization. To achieve these objectives, we developed the OASIS dashboard to visualize data from smart grids. As a result of this project, we are able to highlight the following findings: (1) The underlying software used in this project facilitates the implementation of the dashboard. In particular the utilization of the Flask framework and Python libraries such as NLTK, Tweepy and Forecastio-py, made it viable the implementation because they are well-documented and easy to set up. Although other tools may be available, the architectural designs of the proposed visualization tool is easily replicable in other domains and using different tools such as the Java Play Framework and AngularJs. (2) The process of understanding the data sources and formats can be challenging. In particular,

selecting appropriate data sources and cleaning the data required considerable effort in the beginning of the project. Another issue is the availability of the data. For example, energy market data in Puerto Rico is not widely available. We used data from the NYISO to demonstrate how market data could be incorporated into visualizations as a proof of concept. (3) The utilization of NVD3 for visualization purposes greatly facilitates the implementation of interactive graphs.

We expect that by utilizing this unique product, users will gain a better understanding of data related to smart grids and potentially make conscious decisions regarding to energy consumption and/or production.

## 5.2 Future Work

The future work of this project is divided into two areas: development and implementation aspects and social aspects. Regarding the development and implementation aspect, the following important tasks are planned: (1) Introduce more streaming data coming from smart meters, community renewal energy sources and streaming weather data. (2) Integrate the dashboard with the marketplace built under the OASIS project. (3) Make the dashboard customizable for each user. This means that the graphs that are being shown for a consumer, may not be the same as the ones shown for a producer or administrator. (4) Deal with dashboard scalability issues. We have to carry out usability studies in terms of scalability to determine the requirements to set the number of simultaneous users. (5) Apply machine learning techniques and parallel data processing techniques to speed the processing processes. (6) Explore online machine learning techniques to improve sentiment analysis. We can develop mechanisms to visualize the trend of energy related tweets. (7) Combine the processed data to generate a recommendation system for users.

We believe its also important to analyze the following social aspects, as it will help us gain a better comprehension of the usefulness of the dashboard: (1) Identify the social

demographic characteristics of the users. For example, what is the family composition, socio-economic level, gender, language, and other socio demographic variables of interest. This type of study will provide insight on the design aspects of the dashboard.(2) Make the dashboard more accessible to all types of users, based on the study of their profile. Currently, the dashboard requires at least basic knowledge in concepts related to energy to be utilized. (3) Determine how communities of users utilize this dashboard. It is also of interest to study changes in the way people consume energy, based on the tools, such as this dashboard, and marketplace provided by the OASIS project.



# Bibliography

- [1] National Institute of Standards and Technology. NIST framework and roadmap for smart grid interoperability standards, release 1.0, [http://www.nist.gov/publicaffairs/releases/upload/smartgrid\\_interoperability\\_final.pdf](http://www.nist.gov/publicaffairs/releases/upload/smartgrid_interoperability_final.pdf). January, 2010.
- [2] M. He, S. Murugesan and J. Zhang, Multiple timescale dispatch and scheduling for stochastic reliability in smart grids with wind generation integration, 2011 Proceedings IEE INFOCOM, Shanghai, 2011, pp. 461-465.
- [3] D. Niyato, E. Hossain and A. Fallahi, Sleep and Wakeup Strategies in Solar-Powered Wireless Sensor/Mesh Networks: Performance Analysis and Optimization, in IEEE Transactions on Mobile Computing, vol. 6, no. 2, pp. 221-236, Feb. 2007.
- [4] X. Fang, D. Yang and G. Xue, "Online Strategizing Distributed Renewable Energy Resource Access in Islanded Microgrids," 2011 IEEE Global Telecommunications Conference - GLOBECOM 2011, Houston, TX, USA, 2011, pp. 1-6.
- [5] D. G. Hart, Using AMI to realize the Smart Grid, 2008 IEEE Power and Energy Society General Meeting - Conversion and Delivery of Electrical Energy in the 21st Century, Pittsburgh, PA, 2008, pp. 12.

- [6] V. C. Gungor, B. Lu, and G. P. Hancke. Opportunities and Challenges of Wireless Sensor Networks in Smart Grid. *IEEE Transactions on Industrial Electronics*, vol. 57, no. 10, pp. 3557-3564, Oct. 2010.
- [7] A. Armenia and J. H. Chow, A Flexible Phasor Data Concentrator Design Leveraging Existing Software Technologies, in *IEEE Transactions on Smart Grid*, vol. 1, no. 1, pp. 73-81, June 2010.
- [8] D. O'Neill, M. Levorato, A. Goldsmith and U. Mitra, Residential Demand Response Using Reinforcement Learning, 2010 First IEEE International Conference on Smart Grid Communications, Gaithersburg, MD, 2010, pp. 409-414.
- [9] P. D. Diamantoulakis, V. M. Kapinas and G. K. Karagiannidis, Big Data Analytics for Dynamic Energy Management in Smart Grids, *Big Data Research*, vol. 2, issue 3, September 2015, pp. 94-101.
- [10] A. Motamedi, H. Zareipour and W. D. Rosehart, Electricity Price and Demand Forecasting in Smart Grids, in *IEEE Transactions on Smart Grid*, vol. 3, no. 2, pp. 664-674, June 2012.
- [11] Leeds, D. J., 2012. The soft grid 2013-2020: Big Data and utilities analytics for smart grid. GTM Research, 13 December, <http://www.greentechmedia.com/research/report/the-soft-grid-2013>.
- [12] R. Agrawal, T. Imielinski, A. Swami, Mining association rules between sets of items in large databases. *Proc ACM SIGMOD Int Conf Manag Data*. 1993;22(2):207-16.
- [13] Kaya M, Alhajj R. Genetic algorithm based framework for mining fuzzy association rules. *Fuzzy Sets Syst*. 2005;152(3):587-601.

- [14] S. R. Safavian and D. Landgrebe, A survey of decision tree classifier methodology, in IEEE Transactions on Systems, Man, and Cybernetics, vol. 21, no. 3, pp. 660-674, May/Jun 1991.
- [15] A. McCallum and K. Nigam, A comparison of event models for naive bayes text classification. Proceedings of the National Conference on Artificial Intelligence, 1998. pp. 4148.
- [16] BE. Boser, IM. Guyon and VN. Vapnik, A training algorithm for optimal margin classifiers. Proceedings of the annual workshop on computational learning theory, 1992. pp. 144-152.
- [17] M. Ester, HP. Kriegel, J. Sander, M. Wimmer and X. Xu, Incremental clustering for mining in a data warehousing environment. Proceedings of the International Conference on Very Large Data Bases, 1998. pp 323-333.
- [18] C. Ordonez and E. Omiecinski, Efficient disk-based K-means clustering for relational databases, in IEEE Transactions on Knowledge and Data Engineering, vol. 16, no. 8, pp. 909-921, Aug. 2004.
- [19] J. Kogan, Introduction to clustering large and highdimensional data, Cambridge: Cambridge Univ Press, 2007.
- [20] R. Srikant and R. Agrawal, Mining sequential patterns: generalizations and performance improvements, Proceedings of the International Conference on Extending Database Technology: Advances in Database Technology, 1996. pp 317.
- [21] MJ. Zaki, Spade: an efficient algorithm for mining frequent sequences, Mach Learn, 2001;42(12):3160.

- [22] B. Pang, L. Lee and S. Vaithyanathan,. ”Thumbs up? Sentiment Classification using Machine Learning Techniques”. Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP). pp.7986.
- [23] Y. Simmhan, S. Aman, A. Kumbhare, L. Rongyang, S. Stevens, Q. Zhou and V. Prasanna, Cloud-Based Software Platform for Big Data Analytics in Smart Grids, Computing in Science Engineering, vol. 15, issue 4, July 2013, pp. 38-47.
- [24] N. Balac, T. Sipes, N. Wolter, K. Nunes, B. Sinkovits and H. Karimabadi, Large Scale predictive analytics for real-time energy management, 2013 IEEE International Conference on Big Data, Silicon Valley, CA, 2013, pp. 657-664.
- [25] R. Mallik, N. Sarda, H. Kargupta, S. Bandyopadhyay, Distributed data mining for sustainable smart grids, Proc. of ACM SustKDD11, 2011, pp. 16.
- [26] Z. Aung, M. Toukhy, J. Williams, A. Sanchez and S. Herrero, Towards accurate electricity load forecasting in smart grids, Proc. 4th International Conference on Advances in Databases, Knowledge, and Data Applications (DBKDA), 2012, pp. 5157.
- [27] K. le Zhou, S. lin Yang and C. Shen, A review of electric load classification in smart grid environment, Renew. Sustai. Energy, Rev. 24 (0) (2013) 103110.
- [28] S. V. Verdu, et al., Characterization and identification of electrical customers through the use of self-organizing maps and daily load parameters, IEEE PES Power Systems Conference and Exposition, 2004., 2004, pp. 899-906 vol.2.
- [29] E. Kyriakides, M. Polycarpou, Short term electric load forecasting: A tutorial, in: K. Chen, L. Wang (Eds.), Trends in Neural Computation, Vol. 35 of Studies in Computational Intelligence, Springer Berlin Heidelberg, 2007, pp. 391418.
- [30] X. S. Han, L. Han, H. B. Gooi and Z. Y. Pan, Ultra-short-term multi-node load forecasting-a composite approach, IET Gener., Transm. Distrib, 6 (5) (2012) 436444.

- [31] A. Sancho-Asensio, J. Navarro, I. Arrieta-Salinas, J. E. Armendriz-igo, V. Jimnez-Ruano, A. Zaballos and E. Golobardes, Improving data partition schemes in smart grids via clustering data streams, *Expert Syst. Appl.*, 41 (13), 2014, pp. 5832-5842.
- [32] H. Mori and E. Kurata, An efficient kernel machine technique for short-term load forecasting under smart grid environment, 2012 IEEE Power and Energy Society General Meeting, San Diego, CA, 2012, pp. 14.
- [33] C. Guan, P. B. Luh, L. D. Michel, Y. Wang and P. B. Friedland, Very Short-Term Load Forecasting: Wavelet Neural Networks With Data Pre-Filtering, in *IEEE Transactions on Power Systems*, vol. 28, no. 1, pp 3041, Feb. 2013.
- [34] Bei Li, S. Gangadhar, S. Cheng and P. K. Verma, Predicting user comfort level using machine learning for smart grid environments, *ISGT 2011*, Hilton Anaheim, CA, 2011, pp. 1-6.
- [35] M. Couceiro, R. Ferrando, D. Manzano, L. Lafuente, Stream analytics for utilities. Predicting power supply and demand in a smart grid, in: *Proc. International Workshop on Cognitive Information Processing (CIP)*, 2012, pp. 16.
- [36] Y. Murakami, Y. Takabayashi, Y. Noro, Photovoltaic power prediction and its application to smart grid, 2014 IEEE Innovative Smart Grid Technologies Asia (TSGT ASIA), Kuala Lumpur, 2014, pp. 47-50.
- [37] M. N. Q. Macedo, J. J. M. Galo, L. A. L. de Almeida, A. C. de C. Lima, Demand side management using artificial neural networks in a smart grid environment, *Renewable Sustain. Energy, Rev.* 41 (0) (2015) 128133.
- [38] P. Mack. Big data, data mining, and predictive analytics and high performance computing, in L. E. Jones (Ed.), *Renewable Energy Integration*, Academic Press, Boston, 2014, pp. 439 454.

- [39] J. Baek, Q. Vu, J. Liu, X. Huang and Y. Xiang, A Secure Cloud Computing Based Framework for Big Data Information Management of Smart Grid, in IEEE Transactions on Cloud Computing, vol. 3, no. 2, pp. 233-244, April-June 1 2015.
- [40] Z. Matei, C. Mosharaf, F. Michael J., S. Scott and S. Ion. Spark: Cluster Computing with Working Sets, USENIX Workshop on Hot Topics in Cloud Computing (HotCloud).
- [41] Z. Han and Y. Zhang, Spark: A Big Data Processing Platform Based On Memory Computing, 2015 Seventh International Symposium on Parallel Architectures, Algorithms and Programming (PAAP), December 2015, pp. 172-176.
- [42] W. Li, Q. Niu , W. Zhang and J. Pang, The Application of Spark in the Power Grid Intelligent Decision Analysis Platform, 2015 7th Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC), 2015, pp. 216-219.
- [43] G. Zhou, et al., The Static Security Analysis in Power System Based on Spark Cloud Computing Platform, 2015 IEEE Innovative Smart Grid Technologies Asia (TSGT ASIA), Bangkok, 2015, pp. 1-6.
- [44] M.B Galvez, et al., "First observations of the initial radar node in the Puerto Rico TropiNet X-band polarimetric Doppler weather testbed," 2013 IEEE International Geoscience and Remote Sensing Symposium IGARSS, Melbourne, VIC, 2013, pp. 2337-2340.
- [45] S. Ali and D. Kim, "Visualization methodology of power consumption in homes," 2013 International Conference on Open Source Systems and Technologies, Lahore, 2013, pp. 55-59.
- [46] M. Steiger, T. May, J. Davey and J. Kohlhammer, "Smart Grid monitoring through visual analysis," IEE PES ISGT Europe 2013, Lyngby, 2013, pp. 1-5.

- [47] J. Zhu, E. Zhuang, C. Ivanov and Z. Yao, "A data-driven approach to interactive visualization of power systems," 2012 IEEE Power and Energy Society General Meeting, San Diego, CA, 2012, pp. 1-1.
- [48] C. Marinovici, J. Hammerstrom, S. Widergren and G. Dayley, "Visualizations for Real-time Pricing Demonstration," ISGT 2014, Washington, DC, 2014, pp. 1-5.
- [49] I. Garcia, I. Diaz, PROTEUS Scalable online machine learning for predictive analytics and real-time interactive visualization
- [50] NIST Big Data Public Working Group, "NIST Big Data Interoperability Framework: Volume 1, Definitions", September 2015
- [51] D. Arora, K.F. Li, S. W. Nellie, "Consumers Sentiment Analysis of Popular Phone Brands and Operating System Preference Using Twitter Data: A Feasibility Study, IEEE 29th International Conference on Advanced Information Networking and Applications, 2015.