Towards Developing Secure Software Interfaces for Autonomous Grid Decision Models

Yigit Alparslan[1], Dongen Bradley[2] Baris Taskin[3]
[1,2,3]Department of Electrical and Computer Engineering, Drexel University, Drexel University
Philadelphia, PA
Email: {ya332[1], dz78[2], bt62[3]}@drexel.edu

***Abstract*—This study explores two aspects of software systems that can be used for smart drid IoT devices. First aspect is the visualisation of data points collected from IoT sensors in a household. Second aspect is the study of a software system to create a Wireless connection between an Arduino in a smart IoT hub and a remote file-server, also known as a database.**

***Index Terms*—Internet-Of-Things,Grid Security, Smart Sensors, Arduino, MongoDB Database, Integration,**

I. Introduction

Smart IoT systems are everywhere and creating a software system that would do three main objectives are of vital importance:

1) Collect data points from smart sensors
2) Learn from the data points
3) Predict the best time use the grid and best time to use the batter or solar panels
4) Persist the data in a remote database
5) Visualize the data so that user can have a user experience that can be commercialized later

We will focus on the last two points on this study.

II. Related Work

There is a huge push towards building predictive models in IoT applications that work at grid level. We can see the push in the industry. Commercialized IoT softwares that tell users when to use the grid, get off the grid and start using the solar panels can be seen here[1] and here[2]:

Researchers have focused on studying both commercialized and non-commercialized systems. We don't have access to the proprietary information regarding the commercialized systems but we will report the landscape for the smart IoT software and hardware integration in the design and results section.

III. Experiment Results, Analysis and Performance Evaluation

We break up our design choices by database, databse integration, arduino integration and finally sensor data visualizations.

A. Software for Database

MongoDB is used to persist the data.

Algorithm 1 MongoDB integration

Initialize a NoSQL Database
Create an EXPRESS REST endpoint so that the this can be used to connect to other devices on the internet.
Create a GET and POST API endpoints
Connect the DB to the Ardunio

Algorithm 2 Arduino Integration

Initialize the Wifi module
Scan for the WiFis
Provide credentials
Connect to internet
Make a POST request to the Database in order to persist the data

B. WiFi Integration

Here we can see the code in action.

```
#include <SPI.h>
#include <WiFi.h>

void setup() {
  // initialize serial and wait for the port to open:
  Serial.begin(9600);
  while(!Serial) ;

  // attempt to connect using WEP encryption:
  Serial.println("Initializing Wifi...");
  printMacAddress();

  // scan for existing networks:
  Serial.println("Scanning available networks...");
  listNetworks();
}

void loop() {
  delay(10000);
  // scan for existing networks:
  Serial.println("Scanning available networks...");
  listNetworks();
}

void printMacAddress() {
```

```
  // the MAC address of your Wifi shield
  byte mac[6];

  // print your MAC address:
  WiFi.macAddress(mac);
  Serial.print("MAC: ");
  Serial.print(mac[5],HEX);
  Serial.print(":");
  Serial.print(mac[4],HEX);
  Serial.print(":");
  Serial.print(mac[3],HEX);
  Serial.print(":");
  Serial.print(mac[2],HEX);
  Serial.print(":");
  Serial.print(mac[1],HEX);
  Serial.print(":");
  Serial.println(mac[0],HEX);
}

void listNetworks() {
  // scan for nearby networks:
  Serial.println("** Scan Networks **");
  byte numSsid = WiFi.scanNetworks();

  // print the list of networks seen:
  Serial.print("number of available networks:");
  Serial.println(numSsid);

  // print the network number and name
  // for each network found:
  for (int thisNet = 0; thisNet<numSsid; thisNet++) {
    Serial.print(thisNet);
    Serial.print(") ");
    Serial.print(WiFi.SSID(thisNet));
    Serial.print("\tSignal: ");
    Serial.print(WiFi.RSSI(thisNet));
    Serial.print(" dBm");
    Serial.print("\tEncryption: ");
    Serial.println(WiFi.encryptionType(thisNet));
  }
}
```

C. Visualisation

We also looked at different cloud models to visualize the data that we collected from the smart sensors. We looked at the three visualization tools offered on the market.

1) SAP Analytics Cloud
2) Power BI
3) Tableau

Microsoft Power BI:

1) Release Date 2014
2) Imported csv is automatically scanned to find data types.
3) 99$/user/month (Power BI Cloud and Desktop) (Direct contact for enterprise edition)
4) Limited responsive design.
5) Not supposed to look on phone.
6) Embedding the dashboard to a third party website is easy with embed code.
7) No Smart Predicts and Smart Discovery

Tableau:

1) Release Date 2003
2) After importing a csv, datatypes need to be manually selected.
3) 70$/user/month (Tableau On-premise or Cloud)
4) Limited responsive design.
5) Not supposed to look on phone
6) Embedding the dashboard to a third party website is easy with embed code in iFrame
7) No Smart Predicts and Smart Discovery

SAP Analytics Cloud:

1) Release Date 2015
2) Imported csv is automatically scanned to find data types.
3) 22$/user/month (Cloud)
4) Responsive Design to support mobile devices
5) Embedding the dashboard to a third party website is possible through SAC API.
6) Smart Predicts and Smart Discovery(Run on raw data to get insights)

Based on the comparisons, we decided to move forward with SAP Analytics Cloud because of Smart Insight, and Smart Discovery Features

All the models were done on the cloud, and none of the systems are on premise applications.

Fig. 1. SAP Analytics Cloud was chosen to visualize sensor data and grid data

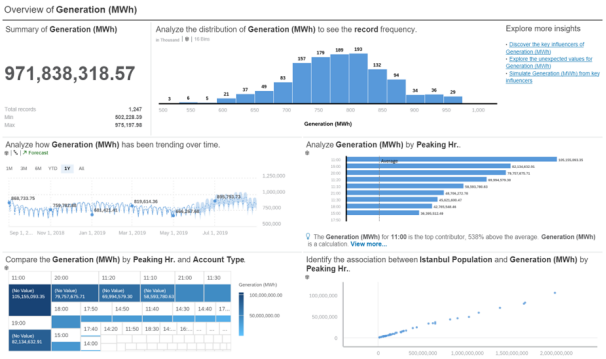

Fig. 2. SAP Analytics Cloud can be used to predict the peak load at the grid level

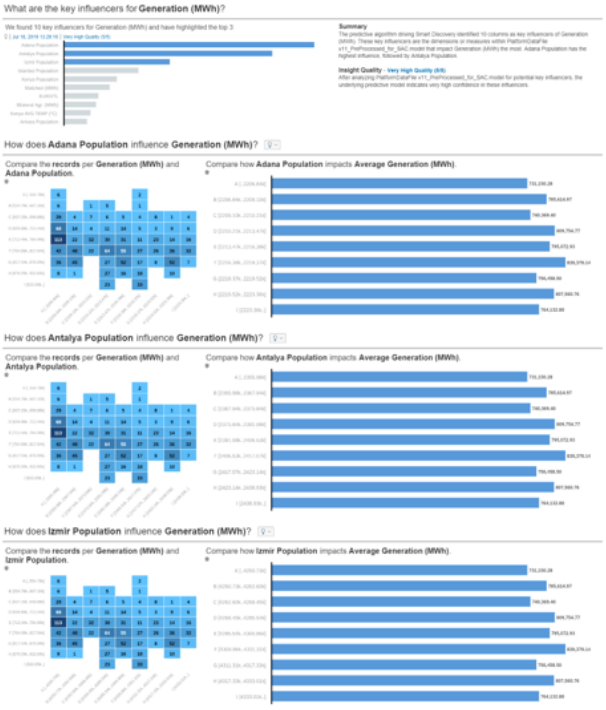

Fig. 3. SAP Analytics Cloud shows here the relationship between the load and the currency rate

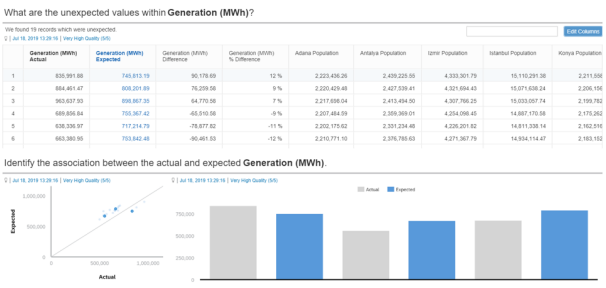

Fig. 4. SAP Analytics Cloud can compare energy consumption trends at different cities.

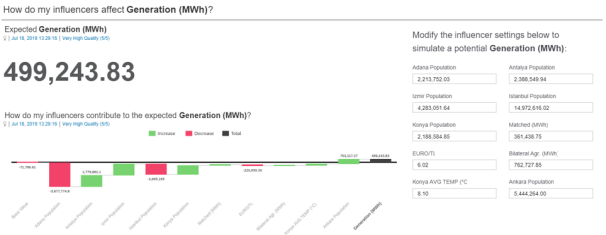

Fig. 5. SAP Analytics Cloud can be used to predict the power trends based on the sensor data by looking at different features such as weather

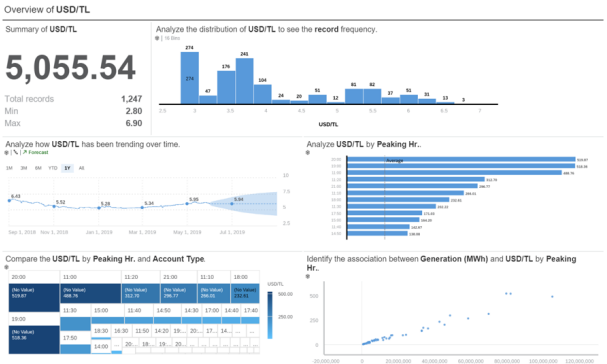

Fig. 6. SAP Analytics Cloud Peak Load vs Time

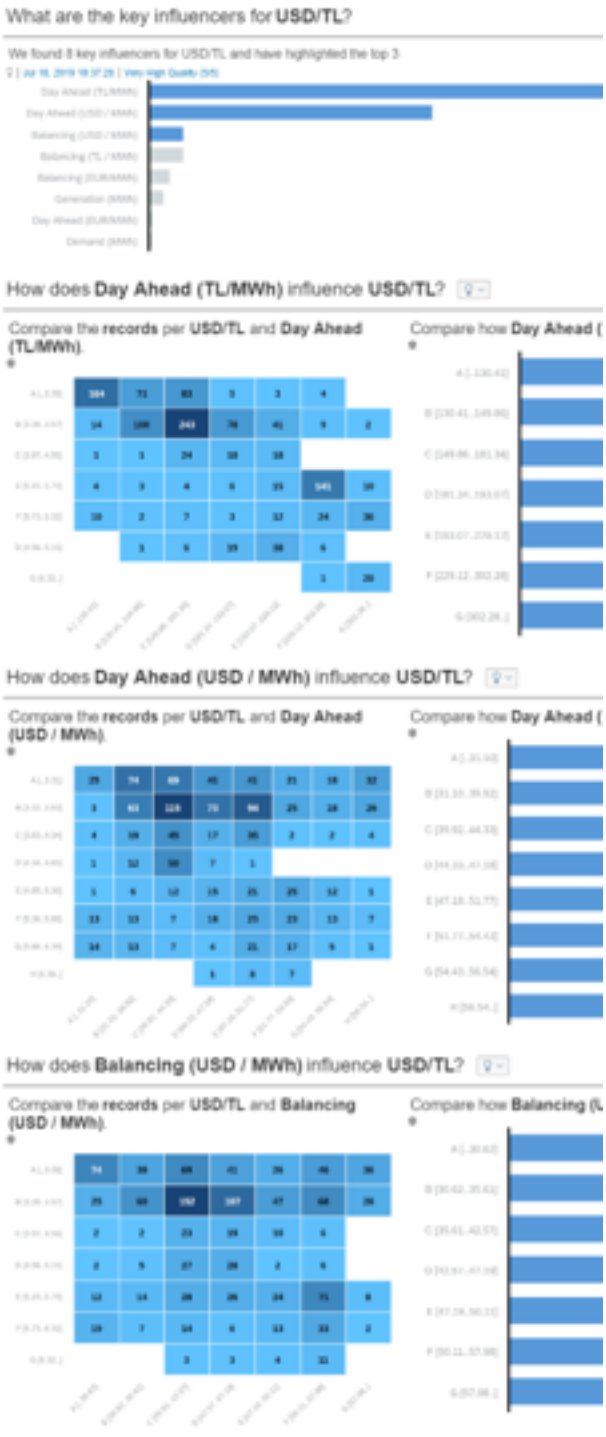

Fig. 7. SAP Analytics Cloud DAy ahead and Influence on USD/TL

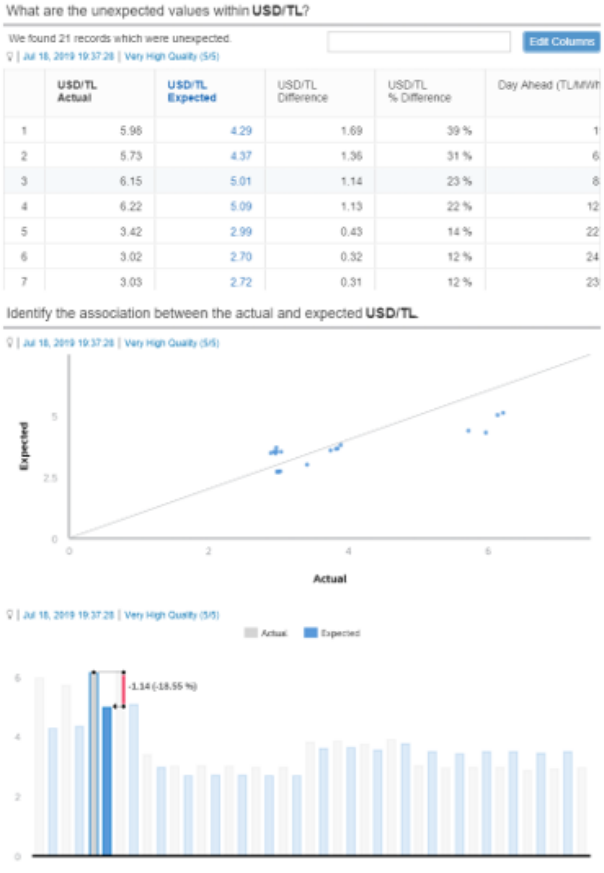

Fig. 8. Unexpected values at USD/TL based on peak load by using SAP Analytics Cloud

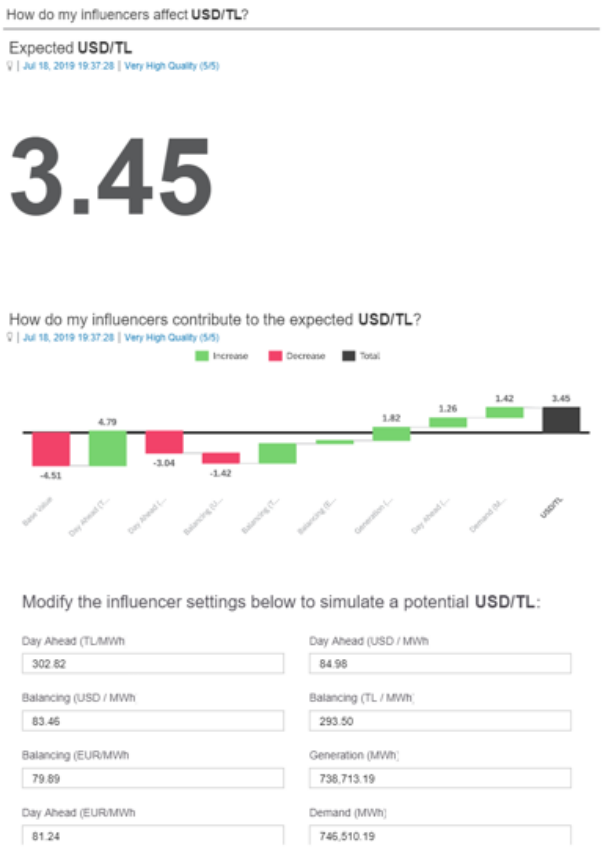

Fig. 9. Different influences affecting the currency rate

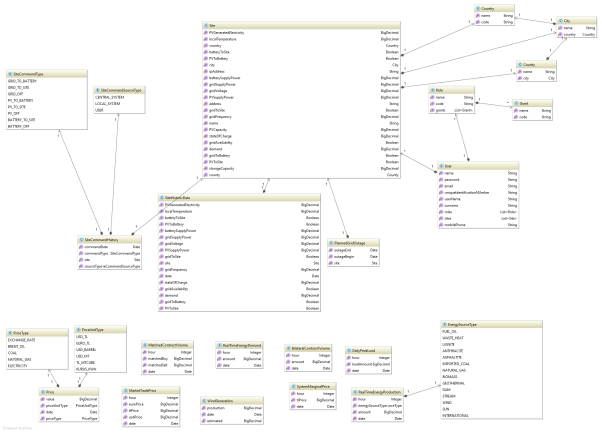

Fig. 10. Class Diagram for one of the modules used in a smart Hub

IV. Conclusion

In this paper we have proposed a novel software visualiation specifically SAP Analytics Cloud. We tried to build preditice models and

V. Acknowledgements

We would like to acknowledge Dr. Baris Taskin for his support from Department of Electrical and Computer Engineering and Bradley Zongen for his mentoring

VI. Future Work

We have used NoSQL databases. We would like to study the scalability and cost-performances analysis of MySQL databases compared to NoSQL databases.

VII. Refences

[1] SAP SE Entreprise Software, URL: https://www.sap.com/products/cloud-analytics/features.html

MongoDB https://mongodb.com/

DFA Engineering, URL: https://www.dfamuhendislik.com/