

# **CS4961-CS4962 Senior Design**

## **Machine Learning for Signal Quality Monitoring**

### **Project Documentation**

Document Prepared by

**Dave Bravo-Ruelas  
Andrew Garcia  
Yao Tsai  
Jose Virgen  
Alex Perez  
George Arias**

5 May 2017

CALIFORNIA STATE UNIVERSITY LOS ANGELES



Los Angeles, California

## **Table of Contents**

1. Executive Summary
2. Introduction
3. User Guide
  - a. Installation Manual
  - b. User Manual
4. Architecture Diagram
  - a. Mobile Application Overview
  - b. Machine Learning to Application Overview
5. Conclusions

## 1.0 Executive Summary

The purpose of this project is to develop a web service in conjunction with a mobile application to handle push notifications regarding anomalies pertaining to uplink data. Users will be able to register, login, and receive notifications upon alarm detection. Alarm Detection is done on the AT&T side using their own Monitor and Control Unit. The web service shall be able to receive and parse alarms by sending the alarm directly to the phones of technicians and superiors instead of having someone call the technicians directly. Furthermore, the project also includes a predictive service in which the system shall be able to predict alarms based on trends in alarms regarding but not limited to location, vendors, and weather.

The first semester was spent on setting up the different databases associated with maintaining different types of data that our mobile application would require. This included the database for alarms themselves, surveys that detail how these alarms were handled, and a separate database to handle users and user authentication for privacy purposes. We were given the technical context in which the mobile application and web service would be used. We created the web service prepared to receive the JSON payload that would be created by AT&T's monitor and control system (M&C); our software would then use this alarm to expand different functionalities.

Our objectives are clearly listed as: 1. Archiving Alarms 2. Notifying the responsible Technicians at specific ground-stations 3. Construct reports to aid technicians when constructing new ground-stations/antennas.

The first objective is archiving the alarms. Simply, our web service archives the alarm received into the alarm database. The second objective is satisfied through our use of push notifications. When an alarm is received

from AT&T's M&C, our web service automatically takes the location of the alarm, and sends a notification to the mobile phones of technicians/engineers subscribed to that specific location with information about the alarm received. The responsible employee would be expected to login to their account, and select an alarm that they wish to be accountable for. This enables technicians to address repairs without competing with other technicians for that same alarm.

The third objective is satisfied using predefined queries to enable technicians the construction of reports. Employees can fill out the different parameters to receive a collection of surveys, alarms, and vendors that pertain to their search. This hopes to allow technicians to more easily access prior alarms and surveys, and to choose vendors with a better track-record.

The final part of the project serves as a template for the implementation of machine learning algorithms. Currently, the machine learning algorithms apply two unsupervised methods, but future projects can implement these (and others) methods similarly if AT&T wishes to provide the data for anomalous/non-anomalous uplink traffic. The intent is that the predictive algorithms will notify technicians of possible alarms, so that they can address the alarms before they occur. This will enable preventative action against possible outages, and possible further degradation of equipment reliant on the equipment that experiences the malfunction.

## 2.0 Introduction

The sponsor for this project is AT&T/DirecTV. To showcase the importance of our project, it is imperative that we explain how customers of AT&T receive their T.V. signal. Channel providers send their signal to AT&T, at which point AT&T beams the signal to satellites in space, and finally the satellites disperse this signal to the antennas of AT&T customers.

Our project focuses on the uplink part of the traffic--notably, the antennas at ground-stations. There are currently 13 ground-stations spread across the United States, and a total of over 100 antennas in these ground-stations. Each ground-station contains a Monitor and Control center (M&C), which enables AT&T employees to view current statuses of various pieces of equipment through the use of the equipment's monitoring system. When an alarm is detected by an equipment's monitoring system, the M&C receives a JSON payload from the monitoring system indicating various parameters for the detected alarm. An employee that notices this alarm at the M&C subsequently calls a technician, so that the technician can then go to the affected antenna, and complete the repair.

This system creates a notable amount of overhead. Since the M&C employees must focus on various different screens at any given time, an alarm produced by the equipment's monitoring system can go unnoticed. These employees can also simply not be in the room when the alarm is produced. This does not include the time it takes for the technician to drive out to the antenna, and complete the repair. Furthermore, more experienced technicians will have a significant advantage when repairing equipment, as opposed to newer technicians.

The aforementioned problems are resolved by the following software needs. Our software essentially cuts the “middleman” in the transfer of information. Instead of transferring information from equipment monitoring system to M&C to technician, the equipment monitoring system sends alarm information directly to the technicians. This reduces the overhead produced by the M&C needing to physically see the alarm, and physically needing to call the technician responsible. Furthermore, a new technician can use the “Similar Past Anomalies” page to view previously resolved alarms as a reference. Uplink alarms are not archived, so our web service performs this function as well.

### **3.0 User Guide**

#### **a. Installation Manual**

##### **Required Software**

- **Java Runtime Environment (JRE)**
  - The server-side application (API) is made with Java Jersey Framework
  - Jersey Framework files are provided in project files
  - Download link for JRE  
<https://java.com/en/>
  
- **Apache Tomcat 8.0+**
  - Utilizes Tomcat’s ability to deploy Java web-apps
  - Download link for Apache Tomcat  
<https://tomcat.apache.org/download-80.cgi>
  
- **MongoDB**
  - Database to store JSON payloads
  - Download link for MongoDB  
<https://www.mongodb.com/download-center#community>

## Required 3<sup>rd</sup> Party Services

- **Google Firebase**
  - Handles / monitors mobile application's real-time push notifications
  - Google account needed to use their API and tools

## Set up

### Starting Application Server

- Follow STEP 1 – STEP 9 on this tutorial to start your tomcat server
  - <http://www.tutorialspoint.com/articles/run-your-first-jsp-program-in-a-pache-tomcat-server>
- Mirror the folder structure of WEB-INF folder from our project into your local server
- Reminder: Make sure all the library files are in your libs folder
- Open browser and open <your url>/MongoService/version to check if the server configured correctly
- For local server the URL will look like  
localhost:8080/<project folder name>/MongoService/version

### Starting MongoDB

- Follow startup guide
  - [Installation for Windows](#)
- Import Database and Collections at innermost directory at MongoDB
  - use command “mongorestore --archive=<path of .archive file>”
  - This will set up the appropriate database and collections for the web service to work appropriately
- Ensure the database and collections were imported correctly by using the mongo interpreter

## Mobile Application Installation (Android)

- Download the .apk file in an Android phone and install.

## Android Source Code

- Pull from <https://github.com/agarcia94/SignalMonitoring>

b. User Manual

Receiving a Notification

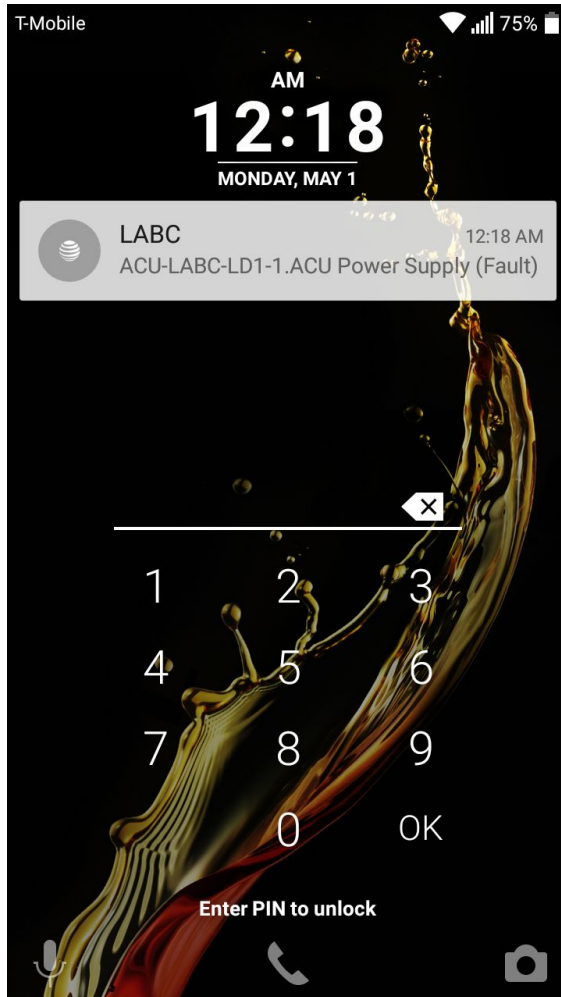
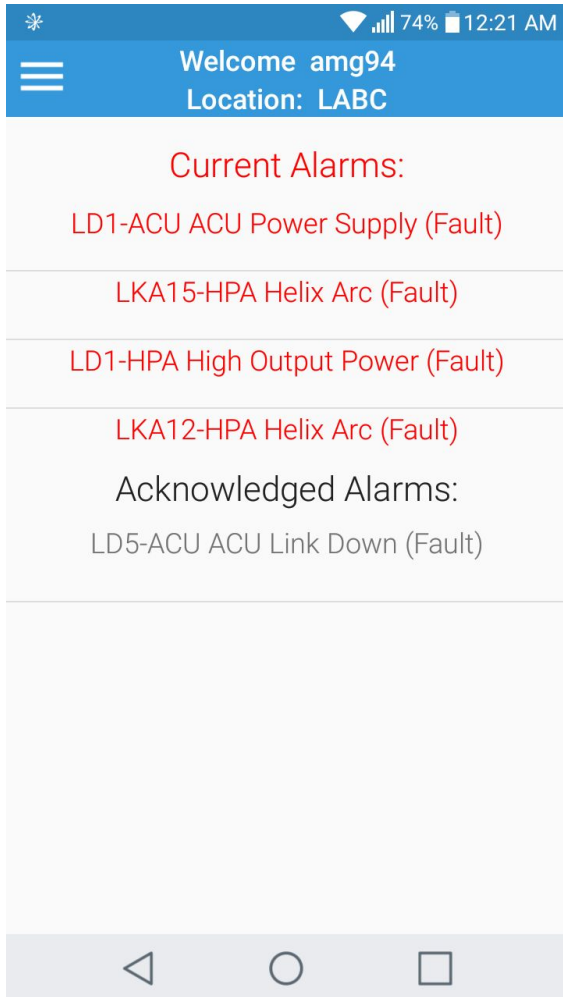


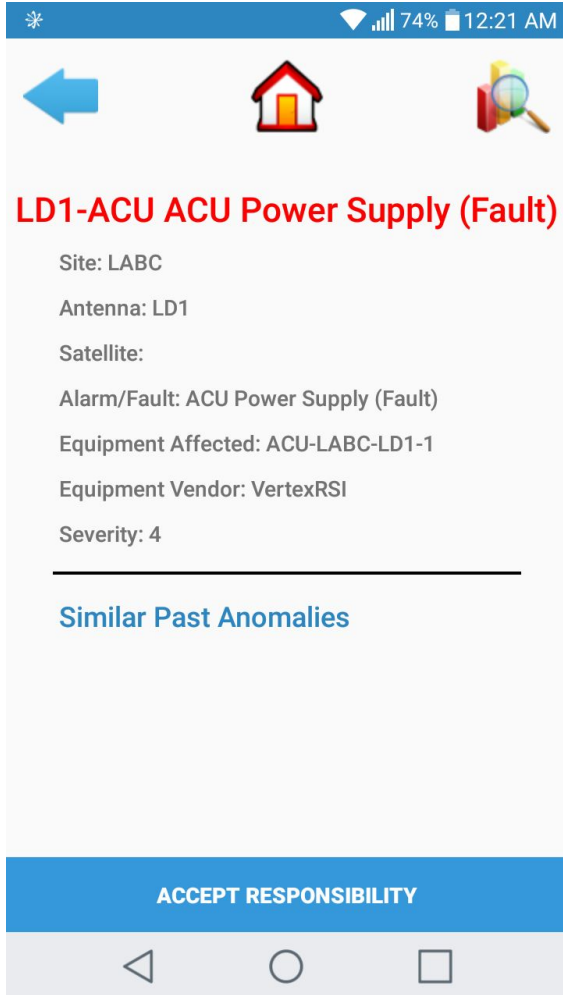
Figure 3.1. LABC Notification

**Figure 3.1** illustrates how a notification would be sent to an LABC technician. Once the notification is clicked, the user will be taken to a home screen, stating alarms that are pending for resolution (note “Current Alarms” in **Figure 3.2**) and alarms that have already been resolved (note “Acknowledged Alarms” in **Figure 3.2**).



**Figure 3.2. Home Page**

The user can then click on any of the alarms, resolved or not, and be able to see a descriptive explanation of the alarm (reference **Figure 3.3**).



**Figure 3.3. Detailed Description of Alarm**

If the current technician cannot immediately address the current anomaly, he/she can either press the back button or home button to go back to the home page and resolve a different anomaly. If the current technician can become the designated individual to resolve the current anomaly, then the technician shall press the “Accept Responsibility” button and be transferred to the survey page (reference **Figure 3.4.a** and **Figure 3.4.b**).

Thank You For Accepting Alarm

Date/Time: Mon Apr 1 20:09:20 GMT+00:00 2016

Name: amg94

Site: LABC

Antenna: LD1

Satellite:

Alarm/Fault: ACU Power Supply

Equipment Affected: ACU-LABC-LD1-1

Equipment Vendor: VertexRSI

Severity: 4

Description of Issue/Anomaly (Please be as detailed as Possible):

electrical problem

How long did it take you to resolve this issue/anomaly?

In Progress

Figure 3.4.a. Survey for Technician

Thank You For Accepting Alarm

How long did it take you to resolve this issue/anomaly?

In Progress

Resolution?

Transfer To Backup Antenna

Was This a Moratorium?

No

Was There An Outage?

Yes

If YES, How Long Was The Outage?

20+ Minutes

SUBMIT

Figure 3.4.b. Survey for Technician

Figure 3.4.a and Figure 3.4.b illustrate what occurs after the “Accept Responsibility” button is pressed in Figure 3.3. The user answers the questions in the survey, presses the submit button, and now the survey is completed.

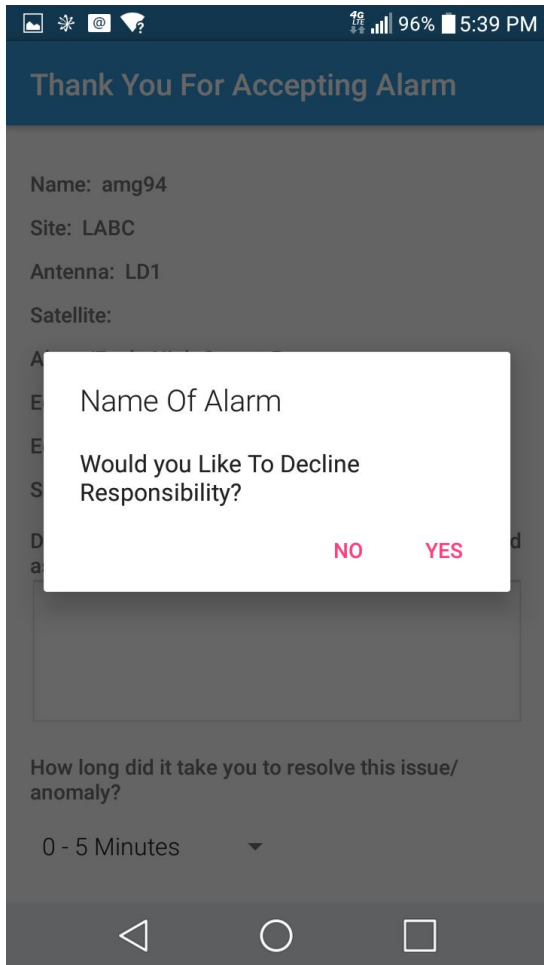


Figure 3.5. Decline Anomaly Prompt

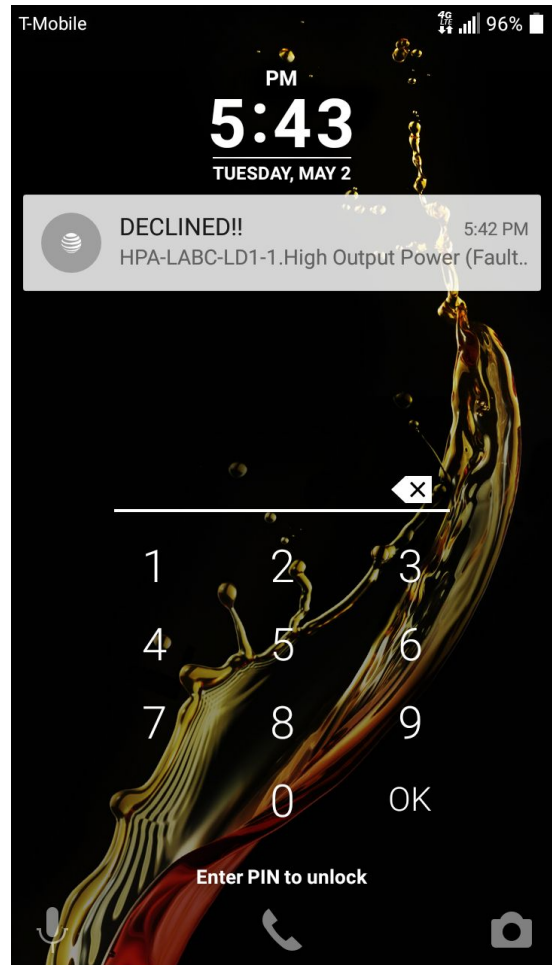


Figure 3.6. Notification of Declined Anomaly

If the current technician changes his/her mind after initially acknowledging responsibility, the technician can decline the anomaly. This is done by pressing the back button and as a response, a prompt will be displayed (shown in **Figure 3.5**). If the technician presses “YES”, then the technician is transferred to the home page and a decline notification is sent to all the technicians at the respective ground station where the anomaly occurred (shown in **Figure 3.6**). If the user selects “NO,” then the technician will remain on the survey page.

## Receive Notifications From Multiple Ground Stations

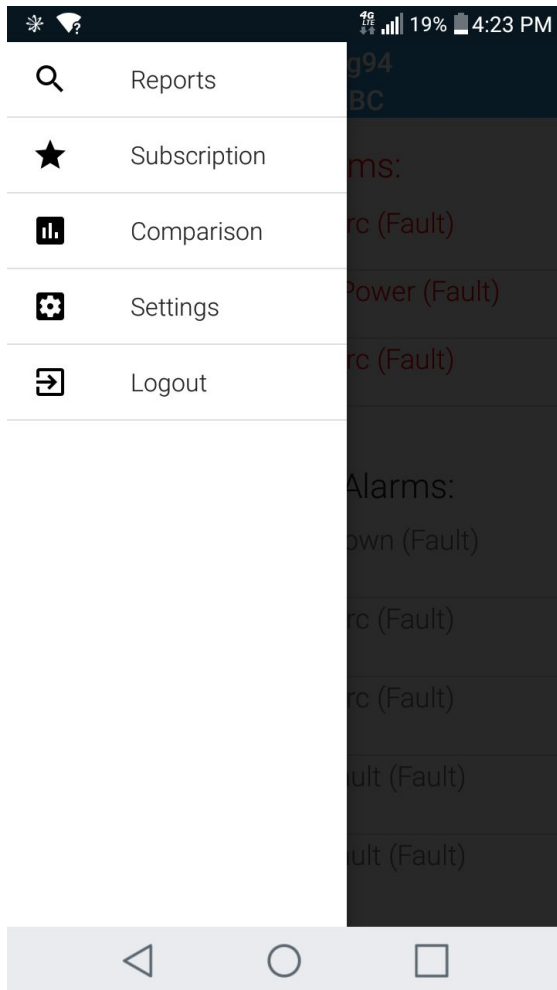


Figure 3.7 App Drawer

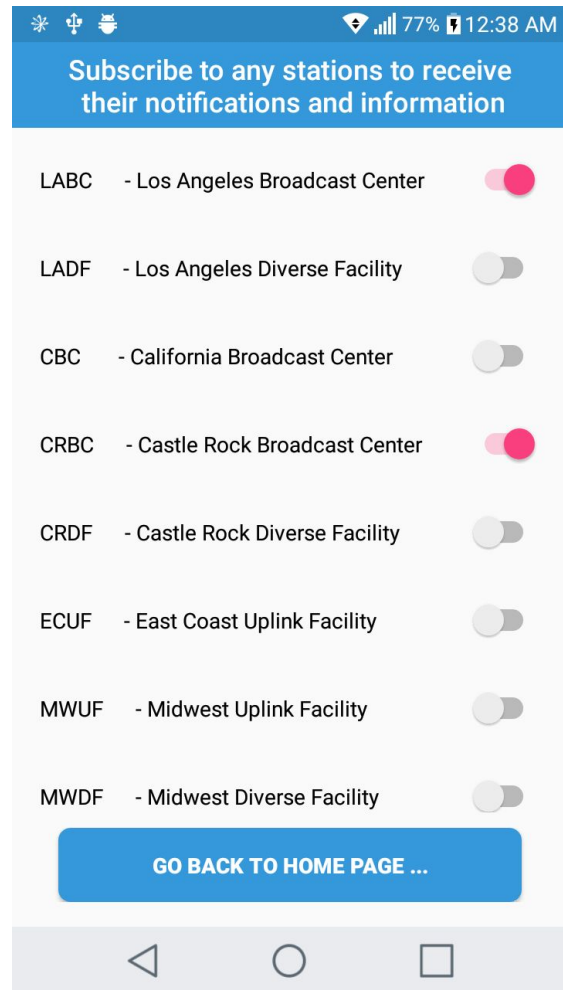


Figure 3.8 Subscriptions Page

If a technician wants to receive notifications from multiple ground stations, the first step is to swipe right on the app and the app drawer will be shown (note **Figure 3.7**). Click on the “Subscription” tab and the list of ground stations available to subscribe to will be shown (refer to **Figure 3.8**).

## Search for Archived Surveys

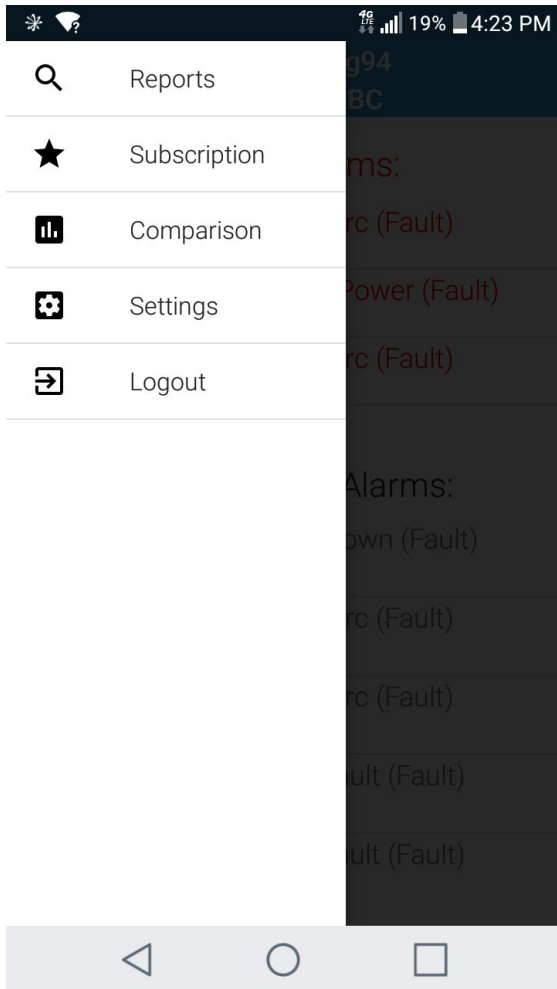


Figure 3.9 App Drawer

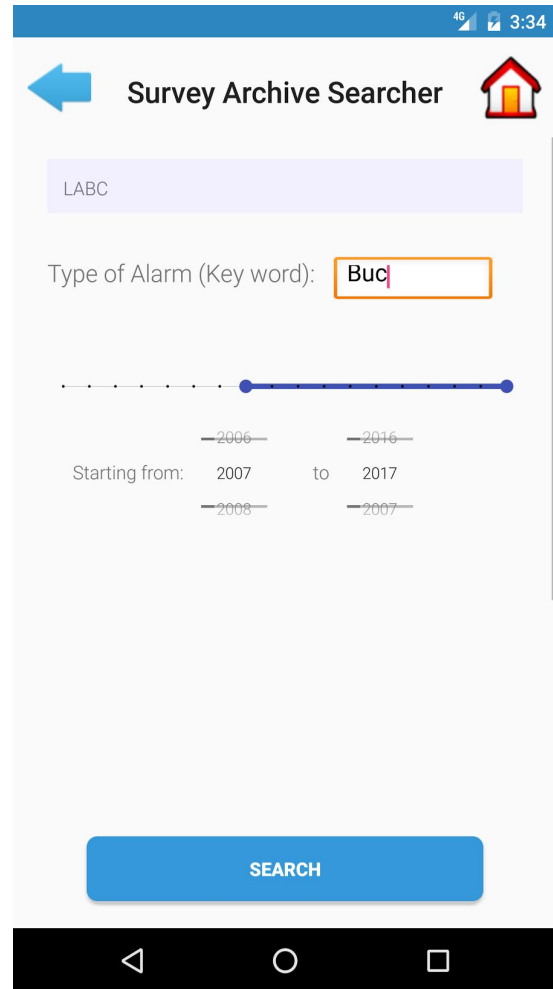
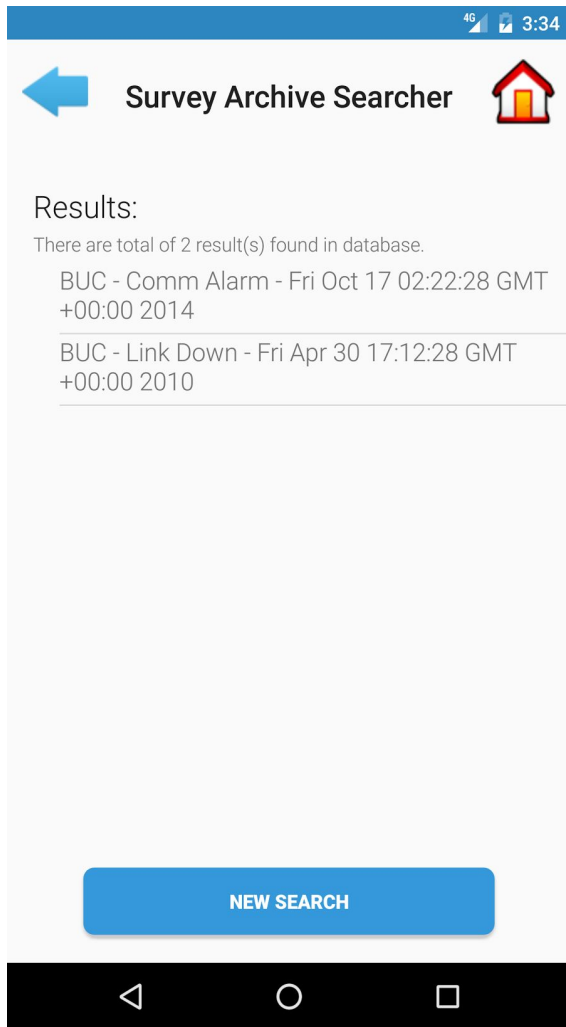
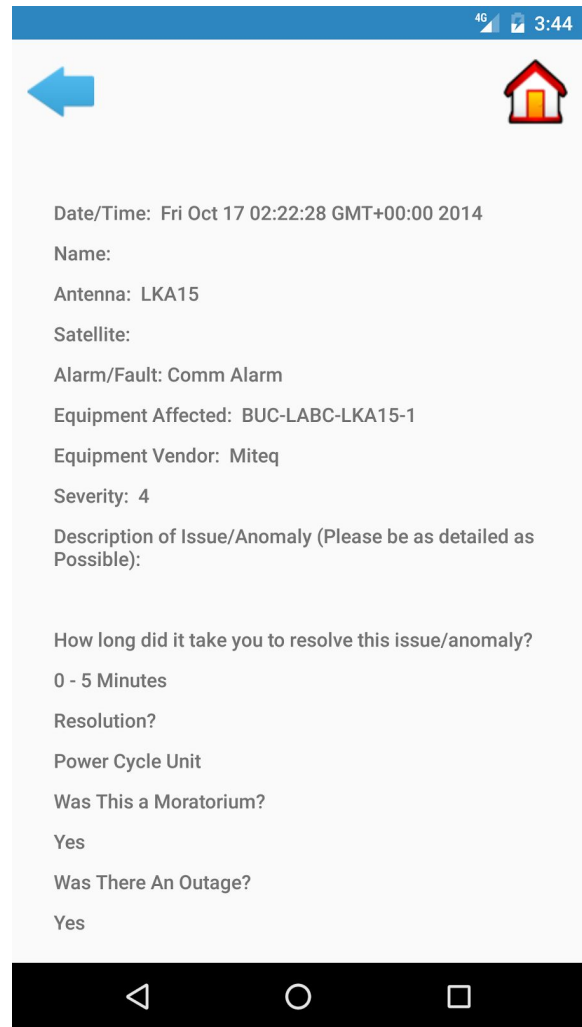


Figure 3.10 Search Criteria

To search for archived surveys, the first step is to open the app drawer by swiping right (note **Figure 3.9**). Click on the “Reports” tab and then a screen with placeholders for search criteria will be displayed (note **Figure 3.10**). In the faint blue rectangular box in **Figure 3.10**, select the ground station of interest. Then provide the alarm type of interest in the text field that’s next to the message “Type of Alarm”. Lastly, a date range must be provided. Press the search button and then a listing of relevant surveys will be displayed (reference **Figure 3.11**).



**Figure 3.11 Matching Surveys**



**Figure 3.12 Archived Survey Responses**

**Figure 3.11** showcases surveys that pertain to the search criteria provided in **Figure 3.10**. If the user then clicks on one of the results, the user will then be shown the contents of the selected survey along with its corresponding responses (refer to **Figure 3.12**).

## Comparison Reports

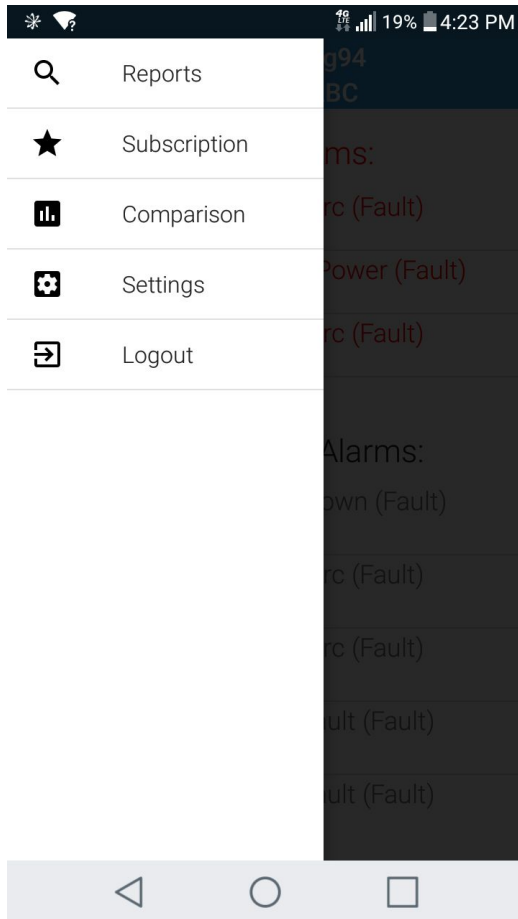


Figure 3.13 App Drawer

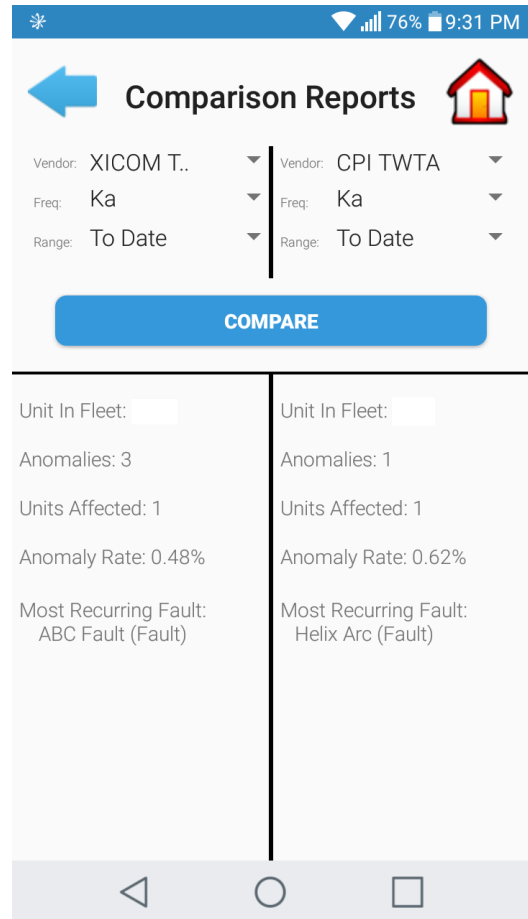


Figure 3.14 Comparison Reports Page

If the user wants to compare two hardware vendors side-to-side, the process is started by swiping right on the phone and opening the app drawer (note **Figure 3.13**). Click on the “Comparison” tab and then the user will then be transferred to the Comparison Reports page (refer to **Figure 3.14**). The user will then select a vendor, frequency, and date range. Once the user has provided all the values of his/her interest, the user will then press the “Compare” button and the appropriate results will be shown.

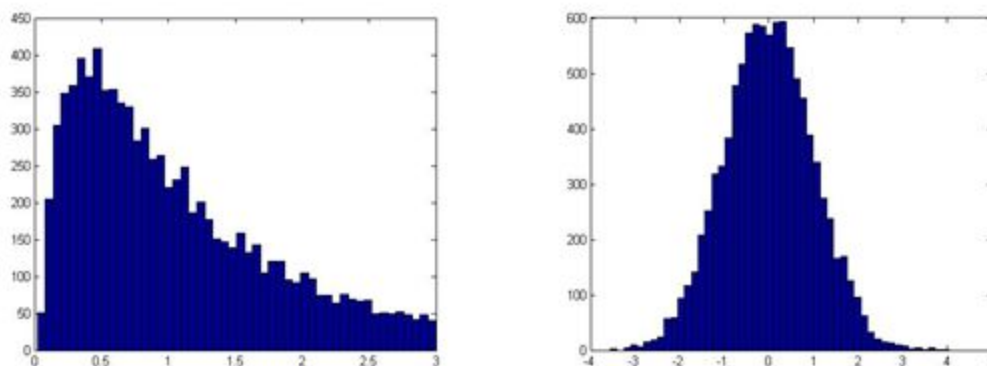
## Machine Learning

The Machine Learning algorithms implemented used Unsupervised Learning through Probabilistic Anomaly Detection and Clustering. The Probabilistic Method compares the density of combinations of features to calculate a probability of a sample to be anomalous. The Clustering Method creates clusters of samples based on similarities in features of the samples to differentiate groups of samples. This way, we can distinguish which groups containing which characteristics are more likely to be anomalous, versus groups containing other characteristics that are less likely. Our dataset consisted mostly of weather data for the year 2016 in Los Angeles.

### Probabilistic Method

We begin with selecting features that we think are most likely to indicate the probability of anomalous samples. In our example, we use Rain and SNR (Signal-Noise Ratio). We can assume that heavy Rain may have an impact of anomalous activity, and we can further assume that a low signal indicates an anomaly somewhere in the system.

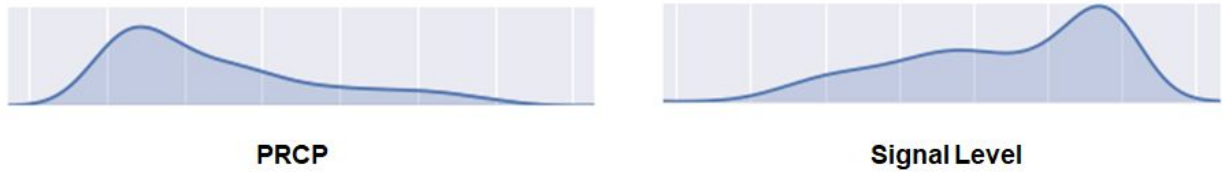
After selecting these features, we attempt to transform the samples into a more normal-distributed dataset. For this, we experimented with various statistical transformations along with some randomness to vary the samples enough that they would be normal. **Figure 3.15** shows an example of the unaltered dataset versus the transformed normal dataset.



**Figure 3.15**

After we have adequately transformed the dataset, we can begin constructing the normal probability distribution for each of the features we plan on using. For our

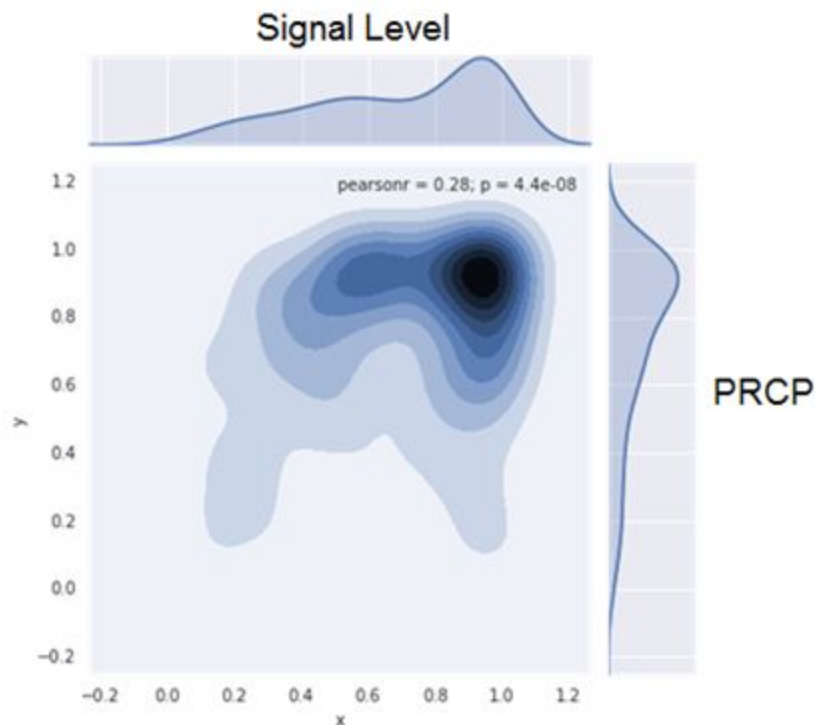
example, we created two (one for each feature). **Figure 3.16** shows a visual representation of the normal distribution of our respective features.



**Figure 3.16**

Visually we can see that most days experienced little rain, and rarely experienced high rain. Similarly, we can see that most days experienced high signal, and less experienced low signal.

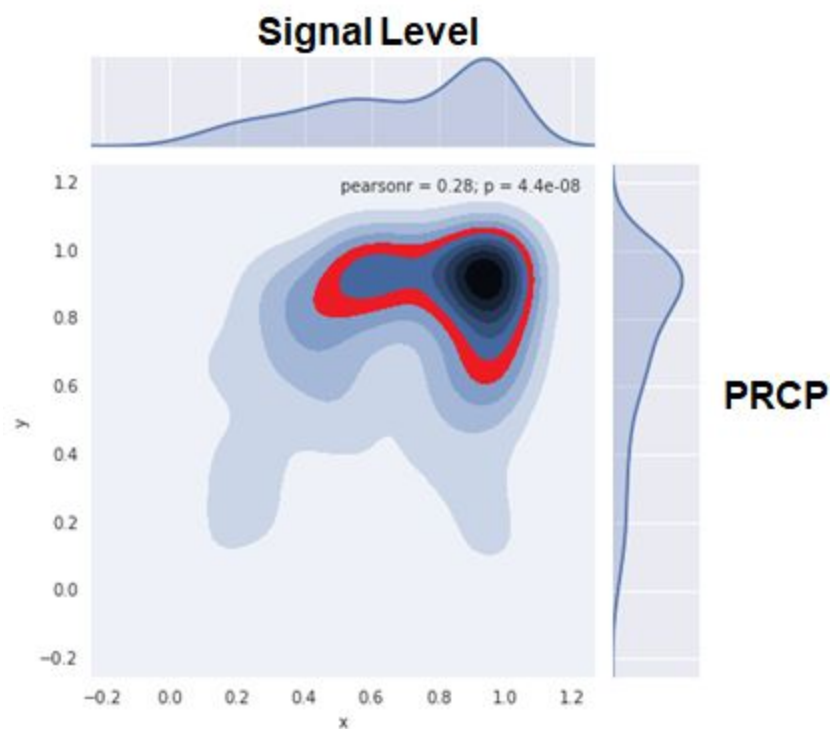
The next step is to create a multivariate normal distribution for the two features. Essentially, each feature becomes an axis, with probability being the third axis--creating a 3D shape. If we transpose this shape onto a flat surface, the resulting shape is similar to **Figure 3.17**. However, keep in mind that this visualization is not possible when more features are used, as this will create hundred-dimensional graphs when hundreds of features are used. It is common to use hundreds of features, which means our use of two features is simply for simplicity and visualization.



**Figure 3.17**

In the figure, the darker spots are more dense than lighter spots. This means that the darker spots indicate more samples that pertain to that specific value of Rain and Signal Level. We can confirm this by aligning the peaks in the normal distribution graphs for the features, and matching them on the graph. We see that the peaks indeed meet at the darker regions of the graph.

The final step is creating a threshold for what we consider to be anomalous. This threshold can be manually defined, and thus all samples that lie outside of this threshold will be predicted as anomalous by this algorithm. An example of a threshold for **Figure 3.17** can be seen in **Figure 3.18**.



**Figure 3.18**

The threshold essentially *becomes* the model. Using the probability density formula (explained further later), we can compare any future sample with Signal Level and Rain to the threshold to create a prediction indicating anomalous/normal.

Into the more technical aspect of this part, we use the Probability Density Formula to calculate a sample's probability, and created the multivariate normal distribution by multiplying across a sample's features. This means that we calculate the probability density for a given sample for a given feature (Probability Density finds the probability of this sample in a given dataset), and then calculate it again for the other

features. Multiplying these probabilities gives us that sample's probability density, i.e. how likely it was for this sample to have this combination of features. Since anomalies (by definition) occur much more infrequently than normal samples, we can deduce that an abnormal combination of features is the most likely to be an anomaly. Again, we assert that when using two features, the calculations may seem simple, but in general we can have hundreds of features and this lengthens the computations, and showcases the future plans that can build upon this project.

## Clustering

The Clustering method is arguably more visually intuitive than the probabilistic method. The clustering algorithm groups samples together that share similar values for features. **Figure 3.19** shows how these samples may be grouped using KMeans (a popular clustering algorithm) using the same features we used above.

Clustering is a way of defining groups within the overall dataset so that the groups share similarities with other samples of the same group, while remaining different to the samples of another group. Thus, after finding the cluster for any future data, we can see if it is considered anomalous. In **Figure 3.19**, for example, if a future data point is plotted into the green cluster (representing high rain and low signal level), we can deduce this sample will have the highest probability of being an anomaly. As stated in the Probabilistic Method, we note that this clustering can span several hundred features, in which case we cannot visually represent the clusters, but instead represent the data points as vectors of a multi-dimensional graph. Then, finding the distance between the vectors to decide their cluster, and thus having a more complex model. **Figure 4.2** shows how these algorithms relate to the mobile application.

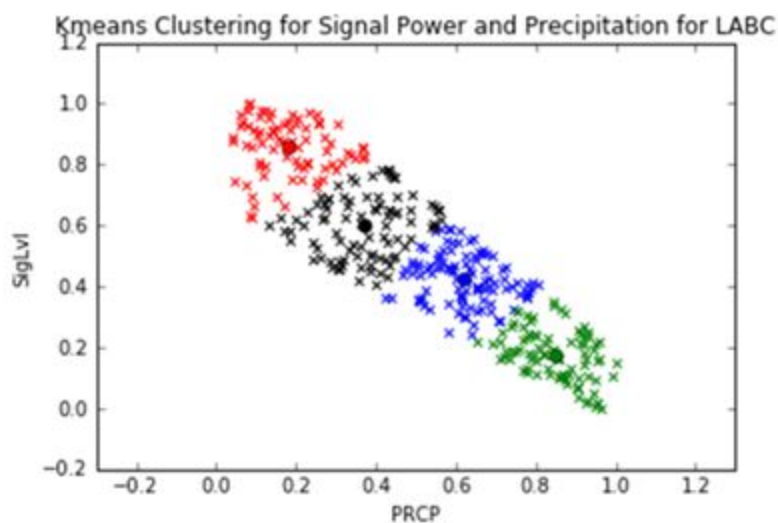


Figure 3.19

4.0 **Architecture Diagram**

a. **Mobile Application Overview**

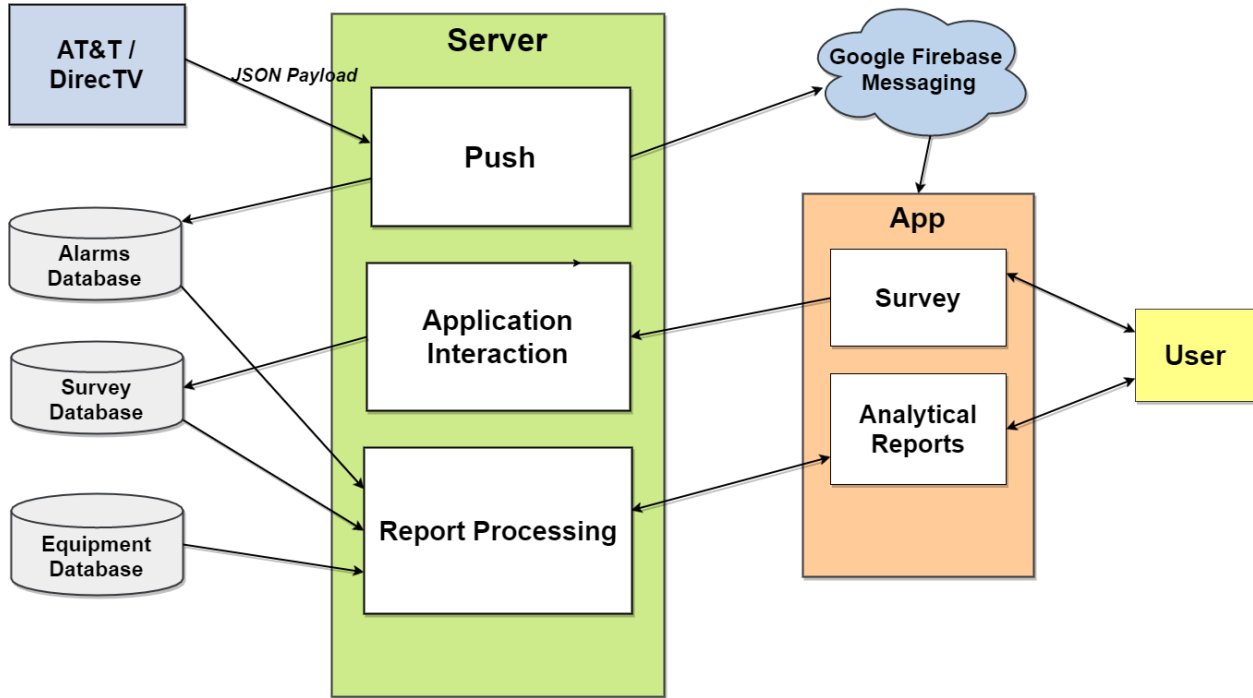


Figure 4.1

**Figure 4.1** showcases the data-flow, starting from the AT&T/DirecTV JSON payload to our web server. This begins the push notification to the technicians, directing them to the survey when they accept responsibility for an alarm. The alarms and survey responses are archived in the appropriate databases. The report module constructs the desired report from parameters given by the user, and showcases this report graphically on the mobile application.

## b. Machine Learning to Application Overview

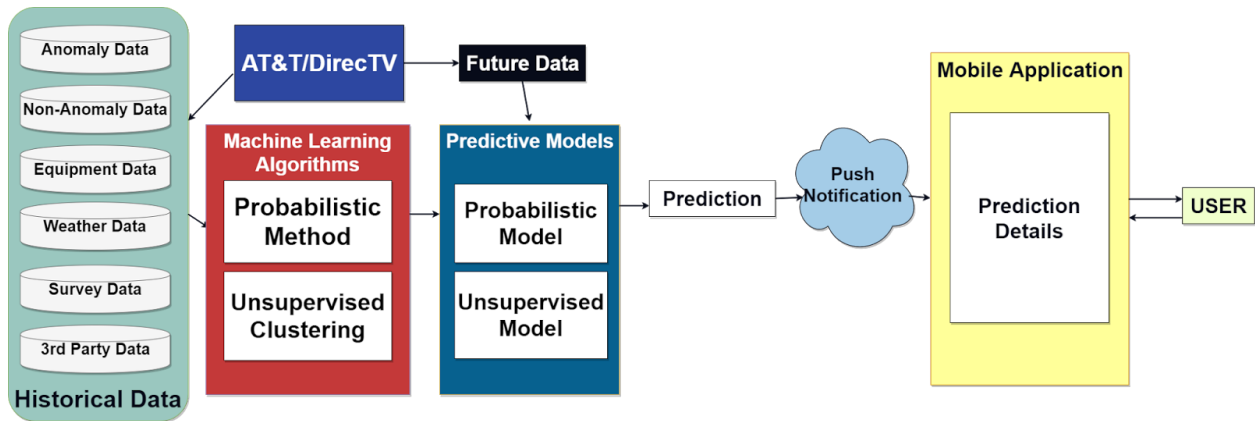


Figure 4.2

**Figure 4.2** shows how the machine learning algorithms and resulting predictive models relate to the mobile application. After funneling in our data from various sources into the implemented machine learning algorithms, we construct a model that can produce predictions. When AT&T wants to check if a sample is likely to be anomalous, the model can take in the sample and produce a prediction. The prediction can be sent to the mobile application to alert technicians of the anomaly prediction, so that they can take the appropriate steps in preventing, or reducing the likelihood.

## 5.0 Conclusion

Overall, we successfully integrated the requirements into our project. We provided a functionality for technicians to respond to anomalies without the use of cellphones and the M&C. Now, technicians are enabled to act autonomously via the application. We've also enabled the support of coworkers through the use of "Similar Past Anomalies," creating a support system for responding to alarms.

However, we were not given uplink data from AT&T. This notably hindered the approach for the machine learning aspect of the project. This meant that it was difficult to differentiate an anomalous and a non-anomalous sample (if all you have are rainy days, how do you know what constitutes a sunny day?). Our methods were unsupervised, meaning they specialized in this exact differentiation without knowing which sample was anomalous, and which was not. In the future, if AT&T is able to provide anomalous AND non-anomalous data for uplink traffic, we can use different methods that may prove to be more accurate. Furthermore, we can use data from other sources that may be related to the likelihood of an anomaly so that all indicators of anomalies are taken into account. After establishing a sufficiently accurate model, an additional mobile page can be added to view prediction details.

The notable lesson we learned for this project was the omittance of complacency. From the beginning, we started the project immediately. Beginning with requirements and project design, we spent multiple days meeting deadlines and milestones we set for ourselves, and fulfilled them. By the end of the first semester, we were farther ahead than other teams, and it became tempting to become complacent. However, we still had other work to do, and there was no time to become complacent. Down to the expo day, we rehearsed several times with both advisors, and it became increasingly apparent that any complacency during the previous semester would have only hindered our project status. All in all, we are thankful for the mentorship and guidance, and appreciate the lasting impact of this project.