Analysis of Data in Wireless Networks

Ingale Ninad (nsi3177@rit.edu)
Department Of Computer Science
Rochester Institute Of Technology
Advisor: Dr. Carol Romanowski

Abstract—Are you happy with the quality of service you get from your wireless provider? How does quality relate to your geographical location or time of day? Does your service provider guarantee you certain quality of service? This paper aims to enable consumers to make an informed decision about which network service provider to choose by trying to answer such questions. We are trying to find patterns and correlations existing in between the various metrics in the wireless networks. The motivation behind this initiative was to be able to provide an environment where consumers as well as service providers will be able to analyze and compare the key performance indicators of wireless networks by area, time and by the service providers. With the help of big data analysis and machine learning algorithm we can provide some insights into the raw data collected by the android application developed by the Electronics and Telecommunication department at Rochester Institute Of Technology (RIT). These series of tests were conducted while driving around RIT. The paper provides an analytical approach on erroneous data (bad data) to derive some insight on how the data analytics would be helpful in debugging the problems or errors in the software development life cycle. This approach will not only be useful in reducing the debugging time required but also will help us to know which are the key performance indicators in the wireless networks.

I. INTRODUCTION

Mobile wireless service for calls and mobile data have become a necessity in our daily life. When we are depending so much on the smart phones we also should be able to know if we are getting satisfactory performance from the network service providers for what we are paying. One key aspect of avoiding paying more than what is required is knowing what to expect when we are using the services. Most of the service providers do not provide us with the insights about how their quality of service is based on geography, time or network type. Service providers might let us know about relative quality of service in comparison with one another which does not serve the purpose of knowing exactly what to expect.

Big data analysis and data mining are branches of science which allow us to analyze the historical data and come up with some information from the raw data available to us. It has its applications to the vast range of fields like health care, security, pattern matching etc. In simplicity, data mining refers to analysis of rapidly growing, huge amounts of heterogeneous data collected over the period of years. This technique can also be implemented to the telecommunications domain in order to find out correlation between various network metrics present in the the wireless networks to determine the quality of service provided by each individual network.

While each such key performance indicator for wireless network contains information about specific functionality of the network like signal strength, bandwidth, call drops, signal error etc, the collective dataset of all KPIs can prove to be a far more useful resource and a much stronger source for information. Collection of records for voice calls can determine possible patterns and trends which individual voice call record may not reveal. By analyzing this historical data we will not only be able to identify patterns between the different KPIs but help to identify possible features in the dataset to measure to determine the quality of service offered by any wireless network service provider.

As discussed above, wireless network metrics can be divided into two broad categories as voice call metrics and mobile data metrics. Some examples of the voice call metrics are signal strength and signal error. As one might imagine these metrics are mostly responsible to determine the call quality of quality of the network. An interesting correlation would be to find pattern of how signal strength and signal error will vary as we move away from the base station of one network service provider. Other category of the wireless network metrics is the mobile data metrics. These metrics are responsible to determine how the quality of mobile data service is based on time of day, location, distance from base station, data transfer direction (download or upload).

Another application of big data analysis and machine learning to the telecommunications domain is debugging the problems in the development phase of such projects. As we know, real world data is always full of noise and is full of missing or wrong values. When it comes to collecting the metrics from wireless networks, there might be events where the data is appropriate but there was a bug in the android application developed. Another case of erroneous data can be an instance actually containing a wrong value as compared to set of other instances. When it comes to determining whether an erroneous instance was a part of noise data or resulted from a bug in the android application can be a time consuming process and would require many iterations of testing android application and then collecting data all over again. This will exponentially increase the time required to implement such initiatives including the android development and analysis process. Machine learning can be a solution to this problem and can save a lot of development time by narrowing down the probable root cause of the
problem and hence speed up the development process.

II. BACKGROUND RESEARCH

There has been a lot of work done in analyzing the data in the wireless networks. This paper will discuss how big data analysis and machine learning techniques could be used to measure the quality of service provided by the network service providers based on time of day, geographical location and network type.

A. Federal Communications Commission (FCC)

Federal Communications Commission (FCC) has done analysis in similar area [2] though not with the use of big data analysis. In the paper presented by the FCC, the motivation was to study the competitive trend in the wireless networks industry amongst different service providers based on market shares, investment, financial indicators, network coverage based on geographical location (rural vs. urban area), importance of spectrum in the provision of the wireless services, pricing and last but not the least quality of service measured by various platforms.

According to [2], as of year 2015 there are four facilities based network service providers which are typically described as nationwide: AT&T, Sprint, Verizon wireless and T-Mobile. Although none of the providers cover the entire geographical area of the united states or the entire population of the country, each one of them have a coverage area which satisfy the needs of significant portion from both the categories i.e geographical area and population of the country.

Another aspect used for comparison between service providers is number of subscribers of the service. The [2] also focuses on comparison between the service providers based on the investment they have put in in order to provide the coverage.

When it comes to comparing service providers based on financial indicators, [2] relies on the factors like Revenue and average revenue per unit, Average revenue per unit by service provider and the consumer price index. Consumer price index (CPI) is a measure of average change over time in the prices paid by the consumers for a fixed quality of service provided [2].

[2] compares the quality of service provided by different providers on the basis of indicators like network coverage, connections and subscribers.

In summary the [2] compares the service providers on the basis of financial indicators, coverage indicators, popularity basis but it does not provide any information about what is to be considered as a key performance indicator before selecting a network service provider. This kind of analysis is necessary because it helps consumers to make an informed decision about the service they can expect.

III. DATA SOURCE

The data for this project consists of data collected using an android application developed as a part of the senior project [1] in order to collect different important metrics in the wireless networks. The data is collected in the form of JSON documents and is then stored in the PostgreSQL database as a relational data warehouse. This is an example of converting unstructured data into structured data. One advantage over here was that all the data collected for a particular entity e.g. data collected for a voice call was in the same format hence it was easier to convert all similar data into a relational table.

The data can be broadly categorized into data for voice calls and data for bandwidth tests. All the data related to voice calls can be considered to compare network service providers based on the quality of service they provide for calling. This data can be considered as collection of metrics such as signal strength, signal error, call drops and many other voice call related metrics. The other category which is bandwidth related data consists of the important metrics related to the bandwidth provided by each service provider. Important metrics in this case can be bandwidth, number of test intervals in a particular bandwidth test, distance from base station, warmup events etc.

Below is the diagram which represents the relational between various entities involved in the entire schema.

From the above diagram, we know that each call entity has more than one VisibleTowersEvent entities associated with it. This is because when we are making a call, at any point of time in any network we are always going to be associated with at least one tower (base station). The reason it might be related to more than one towers is, as we are on the call, we might move from the coverage range of one base station to the other base station. When we move out of range from one base station, the neighboring tower acts as the base station for the same call. We have a 1:2 relationship between the bandwidth test and VisibleTowersEvent, the reason behind that is each bandwidth test is configured in such a way that the data is collected at the start of the test and at the end of the test. In case we are in the range of the same base station, the starting visible tower and end visible tower for a bandwidth test would be one and the same. In case we move away from one base station into the range of neighboring base station we should be able to capture that information as well. Each bandwidth test is divided into number of test intervals based on some initial condition. We will discuss this initial condition further in the analysis part of this paper.

From the figure above, we can understand what constitutes an event during a bandwidth test. As discussed above in any bandwidth test the data is collected at the start of the test as well as at the end of the test in order to determine the towers visible during the entire test. Each bandwidth test is divided into number of intervals which are of the duration 200ms in order to study the pattern of the bandwidth over
Fig. 2. Bandwidth Test event hierarchy

a long connection and also to study the correlation between the warmup events and the regular intervals.

Fig. 3. Voice Call event hierarchy

This figure represents the event formation in a voice call. As we can see, a voice call might be related with more than one towers. This is because, during the call if a person moves out of range of one base station and into the range of neighboring base stations, the call has not changed but the neighboring conditions like the location, cell id and signal strength might have changed. Hence it constitutes a new event. In summary, according to the design considerations of the android application [1], we can say that following conditions constitute a new event:

- The android application will collect data every 15 minutes. This data will be just the set of towers visible to the app at that point in time.
- During a voice call test, in case the base station changes i.e person moves out of range of one base station and into another.
- During the bandwidth test, if at any point of time number of towers visible to the application change.

The figure below indicates important metrics in all the above discussed entities.

1) Event: An event in our dataset is recognized as the change in neighboring environment. Hence each time a new event is triggered, the key metrics to look at are all environment related metrics like GPS info, time stamps as they give us the most accurate location of the person when a new event triggered.
2) Tower: This entity is responsible for storing the information specific to the service provided as towers are uniquely identified by the type of network they support, signal strength and signal error at each tower. Towers might play an important role in analyzing voice call related data as they will be helpful for finding patterns between the type of technology a tower supports and the value for voice call KPIs at that particular tower.
3) Bandwidth Tests: As towers play a key role in analyzing the voice call data, similarly bandwidth tests play an important role in analyzing the bandwidth related data. Key metrics that matter are Direction of data transfer (Upload/Download), bandwidth in bytes per seconds.
4) Intervals: Each bandwidth test constitutes of number of intervals. Number of intervals and the size of the data to be transferred play an important role in deciding the quality of service provided by the service provider. Each bandwidth test contains one or more of warmup events which are the bandwidth at that instant. The size of the data to be sent over the entire bandwidth test is proportional to the initial bandwidth we get in the warmup event. In case of multiple warmup events we consider the average of all the bandwidth for warmup events.

IV. METHODOLOGY

The diagram below represents the flow of the entire process used to implement the solution discussed in this paper.

For the sake of understanding the process better, we will divide the entire flow diagram into three major parts:

- Original data set, Data cleaning and preprocessing
• Testing data set , training data set, iteratively building the model on trained data set.
• Applying the model best built on training and testing data set, validation data set and determining the accuracy of the final model built.

In the very first phase of the first stage of the entire data analysis process, we collect the raw data. In this particular case, raw data was collected by an android application already developed as a part of a senior project[4]. This data is collected in the form of JSON documents and later on is stored in the PostgreSQL database as a relational database following the entity relationship as mentioned in the figure 1.

The second phase of the first stage of data collection and preprocessing covers the data cleaning part. Data cleaning is further divided into parts as below

1) Treating bad data : This step takes care of the instances having bad values such as values out of range, all instances corresponding to a particular feature are having the same value when in reality they are expected to have different values. Such instances which are known to have bad data values are also known as anomalies in the data and need to be treated before beginning the analysis process or model building process. Bad data also might contain some outliers which are instances those are having values for attributes much greater than the average value for that particular feature. These kind of instances are responsible for skewing the data to the higher values and hence need to be treated.

2) Treating missing data : This step is responsible to fill in the missing values in the data set wherever the values are expected to be present. Missing data is one of the major problems faced in any real world data analysis task. These values need to be filled in with the most logical values based on the values present in other instances. During this phase we need help of features providing us knowledge about central tendency of the data like mean, median, mode and standard deviation. Although the most common and simple solution seems to fill in the missing values with the average of all the values for that feature, that does not work in all the scenarios. On the other hand, filling the missing values with the median of the the entire dataset provide us better results. The reason behind choosing median over arithmetic average is, median does take into consideration the standard deviation [3] and average does not. Due to this we can better understanding of the skewness of data using median.

3) Treating Imbalanced data: This step does take the bias in the data away. We build the model on the training data set which will give us the prediction about the new instances coming into the data set on the basis of new one. Now if we want the prediction of the values for new instances to be most correct and most close to real world, we will also want the data on which prediction model is built to be as close to real data.

The second step for the data analysis process is to build a model. This is actually two step process which includes understanding the problem at hand and deciding which kind of machine learning algorithm to be used to build the model. This selection can yield us two choices, one of them is supervised learning [4]. In supervised learning, we find the patterns in the data already present which is also referred to as training data set and then imply the knowledge gained from this learning to the new instances. This step requires presence of a target variable against which we are going to evaluate the instances to be in one class or the other.

Another category of machine learning algorithm is unsupervised learning [5] which involves analysis of data without any prior knowledge or pattern finding. This technique is more related to the finding the closest point which can be similar to a particular point, collect all such points in clusters and find the centroids of such clusters. This type of machine learning techniques can help us find how many different categories are actually present in the data set. One major drawback of this technique is we might need to run more number of iterations of algorithms before we find some definite pattern in the data. The reason for that being, at initial stage in the algorithm any point is considered as the centroid of the cluster and hence it might take more number of iterations to shift that centroid point from the corner to the point close to the central tendency of the data.

The last step in the analysis process is to check the accuracy of the model developed. In the previous step where
the model is being built recursively, at each step when we build a model, we test the model against the validation data set which is a part of training data set. This validation data set can also be considered absolutely different data set but in this case as we do not have any other source to collect data, we have considered some data from android application as the validation data. Once we decide which model gives us the best results based on criteria like false positive rate, F1-score, accuracy, precision we then select that model as the final model.

V. IMPLEMENTATION

In this section we will discuss about steps taken in order to implement the methodology described above.

A. Data Cleaning

As we all know data in real world contains a lot of noise in it. In this project as we were working on real and live data i.e. data was collected and then fed to the data set. This was the most time consuming process from all the steps mentioned in the methodology. The reason for the same being, data cleaning was not done all at once. After each iteration of cleaning the data, features giving the maximum information from the raw data were identified and then the problems like bad data and missing data for those features were fixed. Following were the methodologies used to fill in the missing data

- Median: Filing out median values in place of missing values is a valid imputing method. As median gives us the central tendency of the data as it takes into consideration the average mean as well as the standard deviation[6], we can say that all the points belonging to the same class represent the median value for that attribute.

- Multiple Imputation by chained equations: This technique is also called as MICE from the abbreviation of the term. In this technique, each of the feature in the data set is treated separately to fill in the missing values in. Let us consider there are k attributes in the data set $x_1, x_2, ..., x_k$. There are three steps in this algorithm which allow us to fill in the missing data for all attributes in the data set

  1) Posterior Predictive Distribution [7]: This method is a variation of the bayesian distribution. In this method an unobserved value for a attribute is calculated depending on the distribution of the observed variable. By doing this we will fill in the first attribute which as significant amount of values present as well as the amount of missing values is not too small to neglect. The reason behind this condition is we need a lot of different values of observed values in order to impute the missing values based on observed values distribution.

  2) Using regression algorithm: Now, as we have filled in the values in the first attribute using posterior predictive distribution, we will calculate the value for missing values for $x_2$ by regressing over all the other attributes $x_1, x_3, ..., x_k$. This imputation will be based on limited values where we have observed values for $x_2$.

3) Stabilize imputation results: Finding missing values for all the attributes as mentioned in the step2, we can have the imputed data set at the end of all iterations. One such complete iteration is called as 'cycle'. For the sake of stabilizing the results we repeated the entire process for around 10 cycles.

Processing JSON files: As discussed in section III, the data coming to us from the android application is in the form of JSON file and then it is stored into the relational data warehouse usingpostgresql. Whenever we come across any instance where some of the necessary attributes are not present in the data or are having wrong values for that particular instance, we are storing those JSON documents directly into the database as separate instances. Hence when we are analysing the reason for which the data is being sent to bad data table all we have is a relational table containing all the JSON documents. Hence it was necessary to parse those JSON documents and then retrieve the necessary attributes from the JSON. Below is the example for an individual bad data instance.

![Fig. 6. Bad Data in nested JSON format](image)

As we can see in the diagram above, a voice call has an instance of visible tower event which in turn has instances of towers associated with it. Although the nested events in the figure seem to have valid values for all the attributes, we are missing an important identity of the call which is CallID hence the record being in the bad data.
To parse this JSON we have used the JSON flattener package in python which will help us flatten this JSON which will make the structure of this data 2 dimensional hence making it possible to load the same in PostgreSQL database and perform analysis on top of that data. This was one of the most important tasks from the data cleaning step as we built a prediction model on top of the data processed from the JSON files to give a decision tree which helps us decide whether record coming from the android application is valid or not. This would help us save a lot of constraint checking efforts in the development lifecycle and hence reduce the time for the complete SDLC.

### B. Feature Selection

After we have the data cleaned, next important step to determine is which features give us the maximum information from the raw data available to us. this was done using algorithms mentioned below:

1) Principle Component Analysis (PCA): PCA gives us the best eigen value and eigen vectors around which we can say that the data is distributed most evenly. In this particular case, PCA was used for dimensionality reduction. As we have many dimensions associated with each and every entity, we need to know which ones to consider while building the prediction model. In order to do so, we selected only those features from the data set where the correlation amongst features was either strong positive or strong negative. Diagram below represents the correlation matrix given to us by PCA.

![Correlation Matrix](image)

As a result of PCA we came to conclusion that bytes per sec, total transfer bytes, signal strength, pci and asu are the most important attributes when analyzing the data.

2) Random Forest Ensamble Method:

In order to build an optimum prediction algorithm for classification, we need to use the attributes giving us the maximum information gain and we need to use the best rules to decide which instance belongs to which class. Random forest algorithm does this task for us. In summary, Random forest tries to build many trees (in the range of 1000-10000) with each attribute at the root of the node and then decides which attribute gives us maximum info gain i.e. which attribute occurred at the root of the tree maximum number of time. Doing this Random forest arranges the attributes in their descending value of feature importance. Below is the result we obtained from Random Forest algorithm.

![Random Forest Results](image)

Finally when we have the clean data, we know which features are the most important ones to be used to build the model, in this section we will discuss about the two algorithms used to build the decision tree.

a) REP Tree Classifier: Reduced Error Pruning (REP) Tree classifier worked on the concept of building the decision tree based on information gain and the entropy. It builds many decision trees and selects the best from them after all the trees are created. This decision is taken on the basis of information gain and variance in the data. The upside of this algorithm is it uses the back fitting model in order to implement the error pruning[11]. This means that if a particular branch of the decision tree is known to produce incorrect results from previous experiments in the same cycle, it simply rejects that branch and tries to go back to the previous root node and changes the decision rule.

Below are the results obtained from the REP-Tree algorithm implemented in Weka.

![REP Tree Results](image)
b) Random forest Classifier: As discussed in [8], Random Forest is an ensemble method for classification. It means that Random Forest tries to combine multiple weak solutions obtained and form a stronger one. In our case REPTree algorithm was providing erroneous results by classifying all the LTE instances as invalid data for Voice calls. Hence we turned up to Random forest, as discussed in the section for feature selection Random Forest algorithm builds large number of trees and then decides which attributes are responsible to provide maximum information gain and then builds the classification tree using only those attributes. This method is more suited in our case and also in most of the real world problems because it considers all the solutions by brute force method and then selects only those which are already strong or if not individually strong, it selects all weak models which can be combined into one strong model. Results obtained from Random forest algorithm implemented in weka are as below

![Fig. 10. Results obtained from Random Forest Ensemble Method](image)

From the above diagram, we can say that in order to check if the record is valid or not, we need to check the operating system from which the record is coming up and also the signal strength measured for that instance. This gives up a smaller and better tree as compared to REPTree and which helps us to come to the more accurate decision quickly.

From the results above, we can conclude that in our case RandomForest algorithm has proven to be a better algorithm. Although the accuracy of REPTree is not very much less than that of RandomForest, it still is a bad option as the precision value is very less. Which means that the prediction results are not even close to actual results.

VI. VISUALIZATIONS IN TABLEAU

![Fig. 11. Screen shot of the Tableau dashboard providing us with the summary of analysis](image)

![Fig. 12. Map showing the quality of service around Rochester Institute Of Technology](image)

VII. CONCLUSION

- Our results show that analysis of bad (erroneous) data can provide us a good pattern and helps us understand the problems in the software development cycle better.
- We were not only able to classify the good data from bad data but also provide the pattern about why the instances might be getting rejected.
- Understanding the data collection process helps us to tackle the problem at hand better.
- When doing data analysis, always make use of more than one methods to cross verify the results obtained from the previous methods.

VIII. FUTURE WORK

- This project lays a foundation of the system where consumers can make an informed decision about which service provider is better based on consumer needs.
- This can also be used to recognize and handle the problems with the software development lifecycle.
This can be useful to the service providers to keep a track of the quality of the service based on time, geographic area, technology

IX. ACKNOWLEDGMENTS

I would like to sincerely thank Prof. Dr. Carol Romanowski for her continuous guidance and motivation throughout this project. I would also like to thank Prof. Dr. Yossi Nygate for his continuous support and helping me understand the domain knowledge so effectively. His guidance and vision have made this project possible.

REFERENCES

[3] Xiang, W; et al Estimating the sample mean and standard deviation from the sample size, median, range and/or interquartile range. 14, 1, 1-24, Dec. 15, 2014. ISSN: 14712288