SIEMENS 2017 WIND ANALYTICS CONTEST

DEPARTMENT OF STATISTICS, UCF

Team Number: 14

Kanak Choudhury[1] Taha Mokfi

Mahsa Almaeenejad Md Jibanul Haque Jiban

Supervisor: Alexander Mantz1aris, Assistant Professor

Department of Statistics

University of Central Florida

[1] Correspondence: Kanak Choudhury,

Summary—In this research, we have used different statistical and data mining techniques (ANOVA, Markov Chain model, Path analysis, Clustering, Association Rules, Frequent pattern mining, Social Network Analysis) to explain and explore association and patterns in the dataset. Starting with descriptive statistics (Frequency table and graphical presentation) that gives us the insight of the data. From the exploratory data analysis, it is found that there is association between and among different variables. It is found that visit duration among different park were significantly different which indicates that the error found in different park might cause different pattern. It is also found that the code appearance among different park is also different. For that reason we have conducted cluster analysis and some other analysis that will explain clearly to support this statement. Again, since the data contains error codes that appear in three different time points (before the visit duration start, within visit duration and after the visit duration), by the exploratory analysis, it is found that the error pattern and occurrence is not same for all the time points as well as among different parks. That is why, we have conducted Conditional probability based on Markov Chain method to find the probability of event (i.e. code / code type) given that there is an event at present time point. The Markovian model also support the same conclusion that we found from exploratory data analysis.

We have extracted the time between successive visits for the same station so that we can investigate whether there is any regular time interval or any difference among the average of these times in each park. Using ANOVA, it is found that this average successive visit duration is not equal. There are significant differences for the average successive visit duration for the parks. We have found that there are 13 average successive visit duration pattern among the park (Table 6). Additionally, based on the visit duration, 18 different significant pattern were found for the average visit duration (Table 2).

Since, exploratory data analysis showed association and using ANOVA test we found different groups in categorical variables as well as investigating influential variables on visit duration. It is clear that there are significant differences between parks and even turbines in terms of visit duration or periods of error happenings.

Then using Markov Chain, we extract conditional probability of code happened in each station so that experts can use these tables to predict the next state of turbines. Next, using path analysis we have constructed statistical equations to find not only whether there is any relation among different factors but also what is that relation and how they are connected with each other. It is found that number of stop type of codes can be predicted by direct and indirect effect of event and warning type of codes (Figure 6). It is also found some specific path and connection among the paths for some other variables like StopUrgency (Figure 7 and 8).

We used clustering to find commonalties in the dataset. We clustered visits, error codes, and the parks. Resulted patterns are clearly supporting this idea that in each park and even each station we have different pattern of codes that happen before, after, or within the visit duration. But we can group these parks or turbines based on various factors as we discussed in the report. We could probably say that all of parks or turbines with the same cluster might suffer from the same symptom. For clustering we used different datasets and approaches to illustrate different patterns. Based on clustering visit IDs,

we have found that some certain visit pattern are more frequent in one park that the others (Figure 9 and 10). Similarly, based on the clustering codes, it was found 9 cluster and the distribution of codes among clusters are different by stop urgency. It is found that cluster 7 contains the most codes that are not very frequent codes and cluster 2, 3, 1 and 4 contains those codes most frequent in order (Table 19).

By analyzing the social network of the codes in the next section, we found communities of the codes which are mostly related to each other (Figure 20). Also, we identified those codes which are in the center of codes' network (Figure 21). By analyzing the community interaction, we can find causes and effects (Figure 22).

In order to find and analysis of sequence of the codes, we employed association rule mining and sequential pattern mining techniques (Table 24, 27 and 30). Although there are no few general rules for all the turbines, by keeping the frequent interesting sequence of codes which are extracted from this section we can help decision support systems to improve predictive capabilities (Figure 23).

Overall, the interesting part of our results is that most of methods that we used in this report can be limited to one turbine as well as generalized to all parks. It is important to mention that in this report we could not explain everything that we found from our analysis because of lack of knowledge about the code and process as well as time.

Finally, it is important to note that all the contest questions are overlapped with each other and sometime one can answer using simple cross tables which are very long or use some statistics which just give us a significance of the relationship not an abstract view of the pattern. Our group put and exhaustive effort to use all the well-known descriptive methods in data mining and provide simple but informative abstract visualization and description of relations in the dataset despite of limitations we mentioned at the end.

I. INTRODUCTION

In the sections of this document we provide a thorough analysis of the dataset regarding the wind generator parks provided in the spreadsheets. The methodology used varies according to the question being asked of the teams. In our investigations of the data we found a considerable set of useful conclusions regarding the significant park and code associations. The main point to note is that there is not a set of uniform sequence but a set of consistent transitions between different states which can be used to predict the following events to occur based on what is currently being observed. We also show an exhaustive set of clusters for the parks in figures which can be looked at for future reference.

II. EXPLORATORY DATA ANALYSIS (EDA)

For the EDA, we have conducted different tables and figures. However, since these tables are very large, we did not include it in this report. We have seen that based on park the code distribution is different and it is statistically different (p-value 0.0). About 47.6%, 35.57% and 16.84% codes were stop warning and event related codes and these are statistically different (chi-square 5060.03, p-value- <0.0001). Again, the maximum number of codes found in park008 (9.08%) and in park001 it is about 8.11%. On the other side, Park026 (0.40%) and Park036 (0.72%) contain least number of codes respectively.

It also found that the number of codes based on stop urgency by park is not evenly distributed that indicates that there might be some specific pattern and distribution of codes by park.

Table 1 shows that the successive visit time difference is different by park. For park007 the average time is about 62.3 days and the least time is 20.33 days for park033.

Table 1: Mean and SD for the successive visit time difference by park

Park Name	Mean	Standard Deviation
Park007	62.32323	71.40826
Park035	57.05696	77.11603
Park014	55.33333	64.88076
Park010	53.75	68.3466
Park029	52.73171	52.32756
Park011	47.87097	66.59407
Park006	44.50114	58.12382
⋮	⋮	⋮
Park032	24.71311	28.13196
Park033	20.32836	24.58919

III. MEAN COMPARISION FOR TIME DURATION

In this section ANOVA is performed to find the significant indicators of visit duration. Least Significant Difference (LSD) post hoc test is also applied which reveals which category labels in the categorical variables have more influence on visit duration than others. This helps to determine which parks or stations have longer visit duration than the others.

Then we generated a new variable called 'Time difference' and run ANOVA for all the categorical variables as well. Time difference is the difference of time (in days) between two consecutive codes in a turbine. For example, in turbine 1152 code 5104 occurred in 07.27.16 and the next code 3130 occurred in the same turbine in 07.28.16. So, the time difference is 1. This variable may help finding how

frequently a code occurs in a turbine and which code can be expected after a specific code.

A. Compare mean Visit duration by parks

Based on ANOVA test visit duration is significantly different among parks (F(36,7616) = 20.7, p = 2.2e-16). Average visit duration is shown in descending order in Table 2. Park036 has the highest length of visit duration and Park024 has the lowest in average. This may help the technicians to plan for the visits based on the parks. Using ANOVA test 18 different groups has been found based on visit duration.

Table 2: (LSD output): Comparison of average visit duration among parks

Parks	Mean Visit duration	Park Groups
Park036	15.62	1
Park015	13.71	2
Park003, Park006, Park014	13.27	3
Park007	12.72	4
Park025	12.56	5
Park030	12.55	6
Park026	12.06	7
Park035, Park011, Park029	11.63	8
Park004, Park005, Park009, Park022, Park016, Park019	11.08	9
Park020, Park023, Park021, Park010	10.69	10
Park013, Park027	10.25	11
Park034, Park017, Park002, Park033, Park018, Park012	9.79	12
Park008	9.07	13
Park031	8.31	14
Park037	7.91	15
Park032, Park028	7.17	16
Park001	6.40	17
Park024	3.81	18

The following boxplots is sorted according to the group order found from the previous LSD post hoc test. We can see from the figure that box-plot shows same pattern in parks as ANOVA.

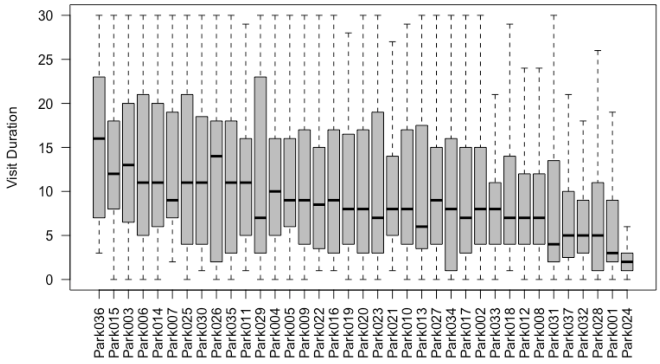


Figure 1: Boxplot of visit duration in each parks

B. Compare mean Visit duration by Station

Visit duration is significantly different among turbines (F(1,7651) = 13.087, p = 0.0002992). From out of 1562 turbines we have grouped these parks into 25 groups of turbines based on visit duration using LSD. (The LSD output is too large to show in this report)

C. Compare mean Visit duration by Factor-A,-C,-D

Visit duration is significantly different for different levels of Factor-A (F(9,7643) = 35.56, p = 2.2e-16), Factor-C (F(11,7641) = 18.59, p = 2.2e-16), and Factor-D (F(2,7650) = 7.77, p = 0.0004). Following tables shows the groups have been founded in each Factors. From these tables value 9 for Factor-A, 'DDD' for Factor-C, and A, C for

factor C influence visit duration more than any other levels in these variables.

Table 3 (LSD output): Comparison of average visit duration in Factor-A

Factor A	Mean visit duration	Groups
9	13.71	1
10	12.29	2
2	12.08	3
3	11.01	4
6	10.80	5
8	10.25	6
5, 7	9.90	7
4, 11	7.11	8

Table 4 (LSD output): Comparison of average visit duration in Factor-C

Factor C	Average visit duration	Groups
DDD	12.61	1
AAC	12.55	2
BBB, AAE, CCD, AAA, BBD, AAD, AAB	10.42	3
CCC, BBC, EEE	7.84	4

Table 5 (LSD output): Comparison of average visit duration in Factor-D

Factor D	Average visit duration	Groups
A, C	10.24	1
B	9.42	2

Table 6 (LSD output): Comparison of average visit duration in visit type

Visit Type	Average visit duration	Groups
SC	10.93	1
PL	10.42	2
CU, UN	7.77	3

D. Compare mean Visit duration by Visit Type

Visit duration is significantly different in visit type (F(3,7649) = 48.36, p = 2.2e-16). According to below table CU and UN causes same visit duration and less than other values influence the visit duration.

E. Compare mean successive visit time difference by parks

Time difference is significantly different among parks (F(36,7616) = 5.288, p = 2.2e-16). Average time difference is showed in descending order in the following table. Park007, 014, 035 have the highest average time difference. This means codes don't occur frequently in these parks; which is a good sign for not having a periodic symptom of turbines in the dataset.

Based on ANOVA test we grouped parks into 13 different groups. This is interesting fact because in clustering part in most of the times we got the 13 different clusters.

Time difference is significantly different among turbines (F(1,7651) = 3.021, p = 0.0822) at 10% level of significance. We grouped 1562 turbines into 73 groups based on LSD result.

Table 7 (LSD output): Comparison of average successive visit time difference by parks

Parks	Average Successive Time Difference	Groups
Park007, Park014, Park035	58.23	1
Park010, Park015	53.35	2
Park029	52.73	3
Park013	49.98	4
Park011	47.87	5
Park020, Park002, Park021, Park006, Park012	45.68	6
Park024	42.30	7

Park003, Park019, Park022, Park009, Park016	40.75	8
Park008, Park005, Park027, Park004	38.83	9
Park026, Park030, Park023, Park031, Park018	36.98	10
Park037, Park028	33.55	11
Park034, Park001, Park025, Park017, Park036	29.84	12
Park032, Park033	22.52	13

F. *Compare mean successive visit time difference by Factor-A*

Time difference is significantly different in Factor-A (F(9,7643) = 8.41, p = 1.365e-12) and Factor-C (F(11,7641) = 7.43, p = 8.085e-13). The following tables show resulted groups for each levels of these variables. Similar to results of ANOVA test on visit duration, here for factor A three levels of 10,9, and 2 and for factor C the value of "AAC" have more effect on time difference than the other levels.

Table 8 (LSD output): Comparison of average successive visit time difference by Factor A

Factor A	Average time difference	Groups
10	56.76	1
9	52.94	2
2	47.02	3
8	44.32	4
3,7	41.37	5
6	39.83	6
4, 5, 11	34.58	7

Table 9 (LSD output): Comparison of average successive visit time difference by Factor C

Factor C	Average time difference	Groups
AAC	50.15	1
AAE	46.8	2
AAB	43.71	3
BBD, CCD	40.91	4
AAA, BBB	39.74	5
BBC, EEE, AAD	35.44	6
CCC, DDD	26.23	7

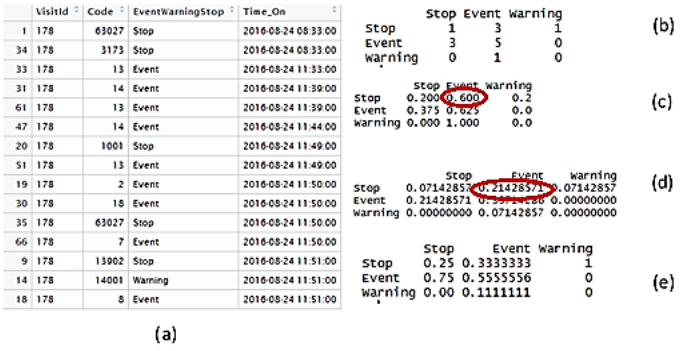
IV. Conditional Probability model and Path Analysis

In this section, we investigate the associated probabilities between the recorded events. First we look at the individual expected occurrence rates for VisitIDs and the different codes produced, and then look at the conditional transitions. The conditional transitions can then be taken from the pairwise associations in a transition matrix and put into a chained set of transitions for a probabilistic graph where complete sequences/paths of the codes can be traced. Using this information, we can then understand how different errors can appear directly and indirectly from associated transitions. With this knowledge, we put forward the hypothesis that this can assist the preparation for the repair of machinery by anticipating following errors associated with the ones currently considered.

A. *Construction of Transition Matrix based on Markov Chain*

A transition matrix is a simple yet powerful technique to create a table of future outcomes of pairs of variables. Given a set of variables we would like to depict succinctly how one set will evolve into another value or affect another variable over each time point. Our use of such a matrix here will indicate the probability of the events transitioning into another event code. It can be seen as a state transition, where the probabilities along the rows sum up to 1.

To construct the transition matrix for the variable "Code", the data has been sorted based on "Time_On" variable for each Visit ID since the data contains error Code that happed within 24 hours. For the sorted data, a square matrix (642 X 642) where row and column of the matrix represent the distinct error code in the data set and each cell represents the count of that combination, has been constructed. For example, for the variable "EventWarningStop" (Event, Warning, Stop) we have the following table. The value 5 means that 5 times 'Event' and then 'Event' appear in the data. In other way, the state 'Event' appears 5 times given that the present state is 'Event'. So, for constructing this data matrix we have considered the time of the code appearance. We have used this data construction method for several different analyses.



	VisitId	Code	EventWarningStop	Time_On
1	178	63027	Stop	2016-08-24 08:33:00
34	178	3173	Stop	2016-08-24 08:33:00
33	178	13	Event	2016-08-24 11:33:00
31	178	14	Event	2016-08-24 11:39:00
61	178	13	Event	2016-08-24 11:39:00
47	178	14	Event	2016-08-24 11:44:00
20	178	1001	Stop	2016-08-24 11:49:00
51	178	13	Event	2016-08-24 11:49:00
19	178	2	Event	2016-08-24 11:50:00
30	178	18	Event	2016-08-24 11:50:00
35	178	63027	Stop	2016-08-24 11:50:00
66	178	7	Event	2016-08-24 11:50:00
9	178	13902	Stop	2016-08-24 11:51:00
14	178	14001	Warning	2016-08-24 11:51:00
18	178	8	Event	2016-08-24 11:51:00

(a)

(b)

	Stop	Event	Warning
Stop	1	3	1
Event	3	5	0
Warning	0	1	0

(c)

	Stop	Event	Warning
Stop	0.200	0.600	0.2
Event	0.375	0.625	0.0
Warning	0.000	1.000	0.0

(d)

	Stop	Event	Warning
Stop	0.07142857	0.21428571	0.07142857
Event	0.21428571	0.35714286	0.00000000
Warning	0.00000000	0.07142857	0.00000000

(e)

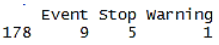
	Stop	Event	Warning
Stop	0.25	0.3333333	1
Event	0.75	0.5555556	0
Warning	0.00	0.1111111	0

Figure 2: (a) Subset of data, (b) Transition matrix based on the data (a), (c) Conditional probability matrix (For Example Probability of appearance of error code that is Event given that the present state is Stop is 0.60), (d) Probability matrix (for example, probability of appearance of consecutive error codes that are Stop and Event is 0.214), (e) Conditional Probability with respect to column factor.

B. Using the frequency estimates to derive connections between variables:

Another way that has been used to construct data based on Frequency. For example, each Visit ID the number times the different labels of a variable appears gives us a distribution by averaging the associated event space with it. This can give a raw indication for the prior expectation of events with a particular Visit ID. For the data (Figure 1 (a)), the following table can be constructed. Since the data contains 7653 unique Visit ID, each data set will contain 7653 observations with counts of each labels of the variable. This concept of data construction has been used for different variables (Code, EventWarningStop, Time of Code Appearance (Before, Within, and After Visit) etc.).

Some new variables have been generated based on concatenation of the variables. For example, new variable "EWS_B_A_concate" created by concatenating variables "EventWarningStop" and "Code_B_A" in the following figure.



	Event	Stop	Warning
178	9	5	1

Figure 3: Data table based on frequency count.

	EventWarningStop	Code_B_A	EWS_B_A_concate
208	Warning	2In	Warning.2In
209	Stop	1Before	Stop.1Before
212	Stop	1Before	Stop.1Before
211	Warning	1Before	Warning.1Before
213	Warning	2In	Warning.2In
218	Warning	2In	Warning.2In
216	Stop	2In	Stop.2In
217	Stop	2In	Stop.2In
210	Stop	3After	Stop.3After
214	Stop	3After	Stop.3After
215	Warning	3After	Warning.3After
219	Stop	1Before	Stop.1Before
221	Stop	1Before	Stop.1Before
220	Warning	2In	Warning.2In
222	Warning	2In	Warning.2In

Figure 4: Variable "EWS_B_A_concate" created by concatenating variables "EventWarningStop" and "Code_B_A"

C. *Probability Model Based on Markov Chain:*

To find the conditional probability based on a Markov chain model, a transition Matrix based on a Markov Chain method (Section 3.1) has been used to construct the data. After getting the transition count matrix, it is possible to find conditional probability explained in Figure 2 (c). With this framework, we are able to see the events as a state space evolution, where states are characterized by the different codes and the probability that the code changes from one to another.

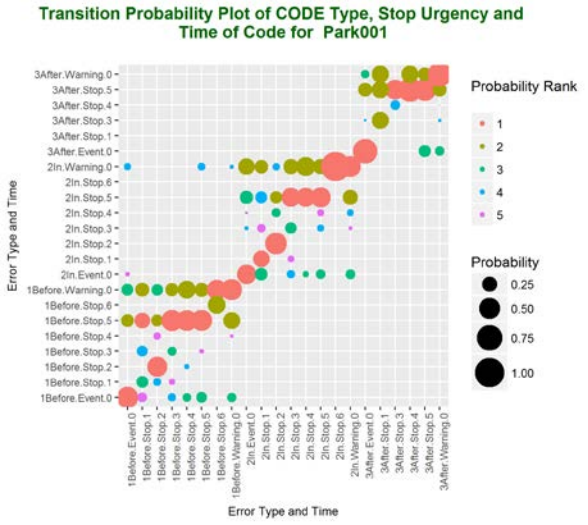
Figure 5 shows conditional probability for some Park. The x-axis shows the present state and y-axis shows the next state. The color represents the rank of the probability and the size of the bubble represents the value of the probability. This probability indicates that if there is a warning code and it appears before the visit start, the largest probability that the next code that may appear before the visit is warning code. In other words, the highest probability of getting warning code given that there is a warning code before the visit start at park 1. The second largest probability is related to stop type of code with urgency 5 given that there is a warning code at present before the visit start (Figure 4 (a)). For the park 1 (Figure 4 (a)), most of the times the largest (Rank 1) probability is stop type of code with urgency 5 given that the present type of code is stop with urgency 1, 3, 4, 5 before the visit start. And the second largest probability is warning type of code irrespective of present state of code before the visit start. It also indicates that whenever some stop type of code with urgency 1 to 5 appear, the next appearance of code will be stop type with urgency 5 is the highest. The fourth largest (Rank 4) is warning type of code happening within visit time given that the present code is either event or stop with urgency 5 happening before the visit start. The pattern of appearance of code is almost similar within the visit duration as it is seen before the visit start for park 1. However, after the visit the appearance of code is different compare to before and within visit. Mostly, it appears as stop with urgency 5 or a warning related code irrespective of the present state of code.

Different parks show different patterns based on the conditional probability (Figure 4 (a), (b), (c), (d), (e), (f)). For the park 3 (Figure 4 (b)), the pattern of the conditional probability is different than in park 1. In park 3, if there is a stop code with urgency 3, the next code will be the warning before the visit start. However, within the visit duration, there is not enough variation of type of code appearance. The highest rank of type of codes are event or warning or stop with urgency of 5 types of codes. Therefore, from the ranking we can assess the degree of belief for event transitions.

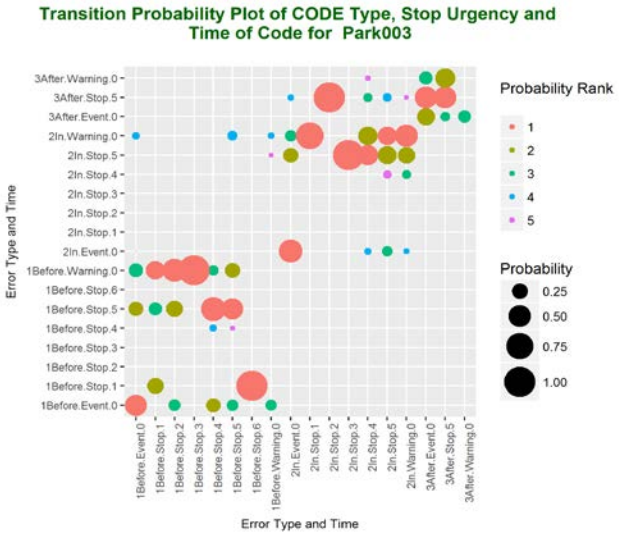
In park 11 (Figure 4 (c)), the probability of getting a warning type of code given that at present when

there is a code irrespective of type is almost zero. This allows us to minimize that problematic contingency of a warning after the error release is a low probability event. It mostly shows the same type of code successively (i.e. appearance of stop type code with urgency 3 given there is a code stop type code with urgency 3). A similar pattern was found within the visit duration as it was before the visit start. On the other hand, for park 36 (Figure 4 (f)), the appearance of stop type code with urgency 5 is largest for most cases irrespective of the present state of code for before the visit start and within visit duration.

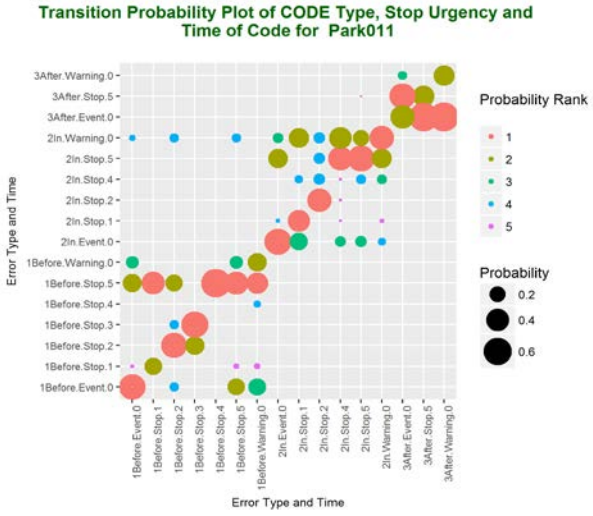
It is seen that those parks with small number of assets show different pattern than those parks with larger number of assets. For example, in park 1, park 20, park 21 contain 87, 112 and 66 assets respectively and the pattern of type of code with urgency before the visit start and within the visit time are almost similar though the pattern after the visit duration is different (Figure 4 (a), (d), (e)). On the other hand, park 3, park 11 and park 36 contain 21, 14 and 30 assets respectively and all pattern based on the conditional probability is almost different with each other. This also indicates that it would be better to study each park individually to find the unique different patterns each one produces, from the cluster of codes and the path (sequence) of code appearance. All the charts for each station are provided in appendix 1.



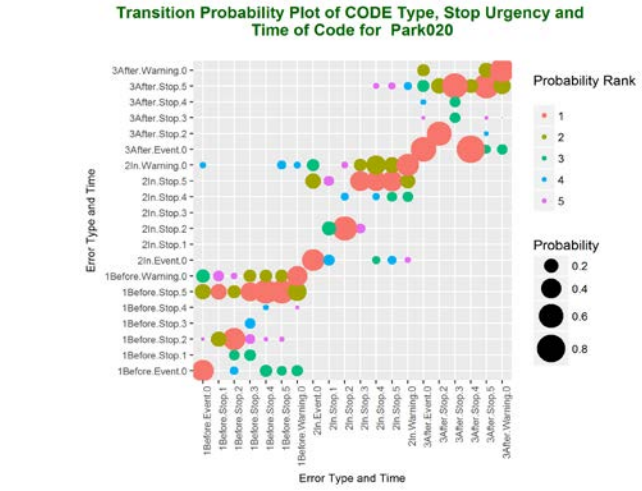
(a)



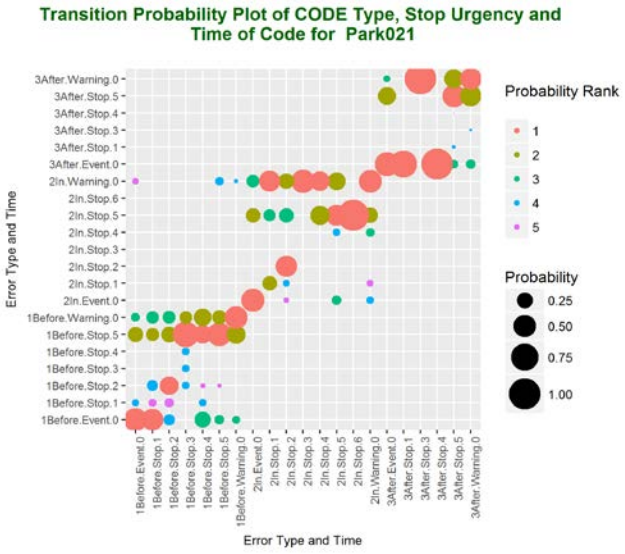
(b)



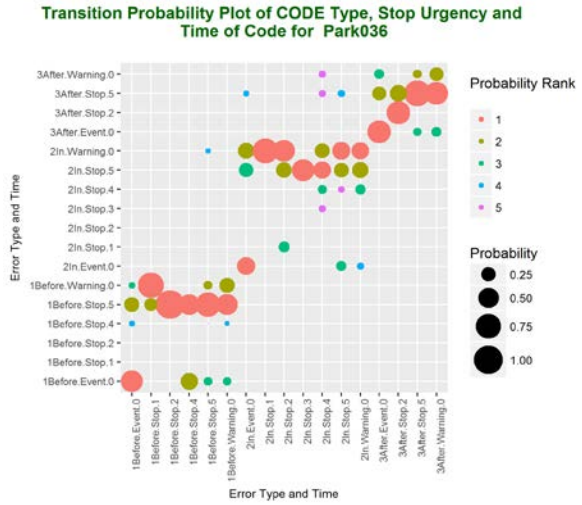
(c)



(d)



(e)



(f)

Figure 5: Conditional Probability Plot. X-axis represents the present state and y-axis represents the future state. Color of the graph represents the rank of the probability (Rank 1 (largest probability) – Tank 5) and size of bubble represents the value of the probability of the state transitions. This plot represents the probability of appearance of type of code with urgency given the present appearance of the type of code with another urgency.

D. Path Analysis of events in sequential orderings:

Our main objective for the path analysis (structural equation model) is that the factors are correlated and that they are dependent on each other. For example, the stop type of code might be affected by some warning type of code. On the other hand, warning type of code might be affected by stop type of code. So, we want to quantify the direction and weight that are the caused for a factor. These may or may not be symmetric and the probability of the event transitions will tell us whether this is so or not.

For the path analysis, we have constructed data based on frequency (Section 3.2) and we have used the correlation matrix for the structural equation model (Path Analysis) so estimated parameters will be standardized and independent of the quantities of the events so that we focus on the conditional state space transitions. This type of modelling will assist in a visual representation.

For the model A, we have created a new variable which is combination of “EventWarningStop” and “ManualStop” variables. This variable contains four labels Stop.FALSE, Stop.TRUE, Warning.FALSE, and Event.FALSE. Example of data matrix is given

below. The value of the variables represents the number of times each type of codes appears in each visit ID.

Table 10: Sample Data Table for the Model A

Visit ID	Event.FALSE	Stop.FALSE	Stop.TRUE	Warning.FALSE
178	21	41	3	12
252	5	16	8	6
450	40	21	7	25
534	0	1	0	2

The main objectives for the model A are:

- Does the number of Stop.FALSE type of codes caused by number of Event.FALSE and Warning.FALSE type of codes?
- Again, whether Stop.FALSE is the reason for Stop.TRUE or not?
- Similarly, does the number of Warning.FALSE type of code depends on Event.FALSE and Stop.TRUE type of codes?
- Also, is there any indirect effect on Stop.FALSE that is caused by Event.FALSE through Warning.FALSE? Additionally, is there any indirect effect on Stop.TRUE by Event.FALSE through Stop.FALSE?

Model A:

$$Stop.FALSE = a_1 Event.FALSE + a_2 Warning.FALSE + \varepsilon_1$$

$$Stop.TRUE = b_1 Stop.FALSE + \varepsilon_2$$

$$Warning.FALSE = c_1 Event.FALSE + c_2 Stop.TRUE + \varepsilon_3$$

Indirect Effect: $\text{IndEff1} := a_2 * c_1$

$\text{IndEff2} := b_1 * a_1$

Total Effect: $Total1 := a_1 + c_1 * a_2$

Based on the fitted model, it is found that the model A is statistically significant (p-value: 0.00). Model diagnostics also support the conclusion. It is also found that number of Stop.FALSE type of codes caused by number of Event.FALSE (0.171) and Warning.FALSE type of codes (0.586) and these are positive. Event.FALSE and Warning.FALSE are positively associated with Stop.FALSE. In other words, if the number of Warning.FALSE type of error increases, the number of Stop.FALSE type of codes will increase by 0.586 times. Similarly, number of Warning.FALSE type of codes can be predicted by Event.False (0.24) and Stop.TRUE (-0.069). The negative coefficient indicates that as the number of Stop.TRUE codes increases the number of Warning code decreases. Table 11 also showed the variance and indirect effect of the model A. The path of the model can be shown by the Figure 5. So, model A explained all the objective and those are statistically significant. For the Stop.FALSE, there is a direct path from Event.FALSE as well as indirect path through Warning.FALSE.

Table 11: Parameter estimate, p-value and standardized estimate for the path model A.

Regressions						
	Estimate	**Std.Err**	**z-value**	**P(>|z|)**	**Std.lv**	**Std.all**
Stop.FALSE ~						
Evn.FALSE (a1)	0.171	0.009	18.591	0	0.171	0.171
Wrn.FALSE (a2)	0.586	0.010	56.579	0	0.586	0.586
Stop.TRUE ~						
Stop.FALSE (b1)	0.469	0.013	36.803	0	0.469	0.47
Warning.FALSE ~						
Evn.FALSE (c1)	0.24	0.012	20.754	0	0.24	0.239
Stop.TRUE (c2)	-0.069	0.016	-4.361	0	-0.069	-0.069

Variances:

	Estimate	Std.Err	z-value	P(>|z|)	Std.lv	Std.all
.Stop.FALSE	0.604	0.01	61.807	0	0.604	0.604
.Stop.TRUE	0.81	0.013	61.737	0	0.81	0.81
.Warning.FALSE	0.981	0.018	54.659	0	0.981	0.979

Indirect Effect

	Estimate	Std.Err	z-value	P(>|z|)	Std.lv	Std.all

IndEff1	0.14	0.007	18.924	0	0.14	0.14
IndEff2	0.08	0.005	17.161	0	0.08	0.08
IndEff3	0.10	0.010	10.129	0	0.100	0.10
Total1	0.311	0.011	27.871	0	0.311	0.311

For the model B, we have created a new variable which is combination of "EventWarningStop" and "Urgency" variables. This variable contains eight labels Stop.1, Stop.2, Stop.3, Stop.4, Stop.5, Stop.6, Warning.0, and Event.0. For data construction, we have used the similar method explained in Model A.

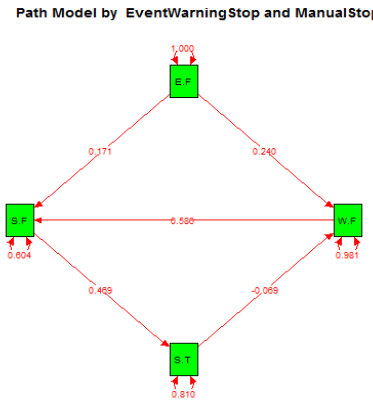


Figure 6: Path model for the model A. Arrow indicates the direction of the variable and the value indicates coefficient of the parameter. S.F, E.F, W.F, and S.T indicate Stop.FALSE, Event.FALSE, Warning.False, and Stop.True respectively.

The main objectives for the model B are:

- Does the number of Stop.5 type of codes caused by number of Stop.3, Stop.4, and Warning.0 type of codes?
- Again, whether Warning.0 depends on Event.0 or not?
- Does the number of Stop.1 type of code depends on Warning.0 type of codes?
- Is there any simultaneous effect on Stop.1, Stop.2, Stop.3 and Stop.4?
- Also, is there any indirect effect on Stop.5 from Warning through Stop.4? Additionally, is there any indirect effect on Stop.5 by Stop.1 and Stop.2 through Stop.3?
- What is the indirect effect on Stop.5 by Event.0 through Warning.0

Model B:

$$Stop.5 = a_1 Stop.3 + a_2 Stop.4 + a_3 Warning.0 + \epsilon_1$$

$$Warning.0 = b_1 Event.0 + \epsilon_2$$

$$Stop.1 = c_1 Warning.0 + \epsilon_3$$

$$Stop.2 = d_1 Stop.1 + \epsilon_4$$

$$Stop.3 = e_1 Stop.2 + \epsilon_5$$

$$Stop.4 = f_1 Stop.3 + f_2 Warning.0 + \epsilon_6$$

$$Event.0 = g_1 Stop.5 + \epsilon_7$$

$$Warning.0 = h_1 Stop.6 + \epsilon_8$$

Correlation Model: $Warning.0\ Event.0$

Indirect Effect:

$$\text{IndEff1} := a_3 * b_1$$

$$\text{IndEff2} := c_1 * d_1$$

$$\text{IndEff3} := e_1 * a_1 * d_1$$

$$Total := \text{IndEff3} + a_3 + a_2$$

$$\text{IndEff4} := g_1 * a_3$$

$$\text{IndEff5} := g_1 * b_1$$

Based on the fitted model, it is found that the model B is statistically significant (p-value: 0.00). Model diagnostics also support the conclusion. It is also found that number of Stop.5 type of codes can be determined by number of Stop.3 (0.234), Stop.4 (0.202), and Warning.0 (0.279) type of codes and these are positive. All these labels are positively associated with Stop.5. In other words, if the number of Warning.0 type of error increases, the number of

Stop.5 type of codes will increase by 0.279 times. Similarly, number of Warning.0 type of codes can be predicted by Event.0 (1.378). Table 12 also showed the variance and indirect effect of the model B.

The path of the model can be shown by the Figure 6. So, model B explained all the objective and those are statistically significant. For the Stop.5, there is a direct path from Warning.0, Stop.3, and Stop.4 as well as indirect path from Event.0 through Warning.0. It is important to mention that Stop.6 cannot determine by any other factors. Instead, Stop.6 type of codes are the cause for Warning.0 type of codes. Since Stop.6 are the most serious type of codes and might happen independently before getting any warning previously. Also, all the stop codes are connected based on the urgency type. It is also found the Event.0 and Warning.0 has negative relation which indicates as the number of warning increases the number of event decreases. Additionally, all the indirect paths that we consider in the objectives are statistically significant.

Table 12: Parameter estimate, p-value and standardized estimate for the path model B.

Regressions:

	Estimate	Std.Err	z-value	P(>|z|)	Std.lv	Std.all
Stop.5 ~						
Stop.3 (a1)	0.234	0.009	24.987	0	0.234	0.235
Stop.4 (a2)	0.202	0.011	18.675	0	0.202	0.203
Warning.0 (a3)	0.279	0.028	9.825	0	0.279	0.279
Warning.0 ~						
Event.0 (b1)	1.378	0.115	12.003	0	1.378	1.382
Stop.1 ~						
Warning.0 (c1)	0.262	0.011	23.437	0	0.262	0.262
Stop.2 ~						
Stop.1 (d1)	0.532	0.01	55.156	0	0.532	0.532
Stop.3 ~						
Stop.2 (e1)	0.36	0.011	33.615	0	0.36	0.36
Stop.4 ~						
Stop.3 (f1)	0.218	0.011	20.046	0	0.218	0.219
Warning.0 (f2)	0.184	0.014	13.144	0	0.184	0.184
Event.0 ~						
Stop.5 (g1)	0.271	0.012	23.111	0	0.271	0.27
Warning.0 ~						
Stop.6 (h1)	0.058	0.009	6.203	0	0.058	0.058

Covariances:

	Estimate	Std.Err	z-value	P(>|z|)	Std.lv	Std.all
.Warning.0 ~~						
.Event.0	-1.215	0.107	-11.344	0	-1.215	-0.842

Variances:

	Estimate	Std.Err	z-value	P(>|z|)	Std.lv	Std.all
.Stop.5	0.58	0.016	35.257	0	0.58	0.583
.Warning.0	2.272	0.267	8.495	0	2.272	2.286
.Stop.1	0.92	0.015	61.828	0	0.92	0.921
.Stop.2	0.708	0.011	61.853	0	0.708	0.708
.Stop.3	0.86	0.014	61.844	0	0.86	0.86
.Stop.4	0.874	0.014	61.192	0	0.874	0.876
.Event.0	0.917	0.015	61.833	0	0.917	0.917

Defined Parameters :

	Estimate	Std.Err	z-value	P(>|z|)	Std.lv	Std.all
IndEff1	0.385	0.023	16.676	0	0.385	0.386
IndEff2	0.14	0.006	21.732	0	0.14	0.139
IndEff3	0.045	0.002	19.619	0	0.045	0.045
Total	0.527	0.025	21.111	0	0.527	0.527
IndEff4	0.076	0.008	9.661	0	0.076	0.075
IndEff5	0.373	0.03	12.568	0	0.373	0.373

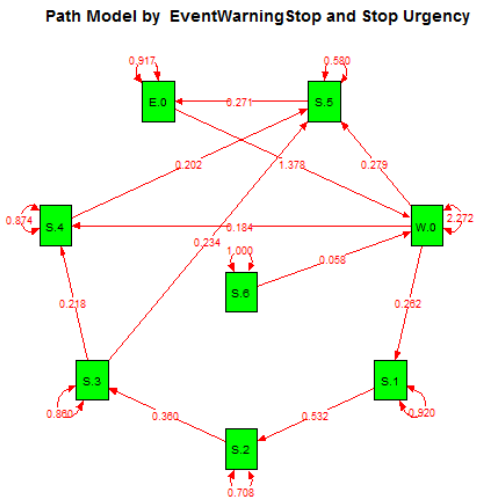


Figure 7: Path model for the model B. Arrow indicates the direction of the variable and the value indicates coefficient of the parameter. E.0 and W.0 indicate Event and Warning with urgency. Similarly, S.1 to S.6 indicate Stop with urgency.

For the model C, we have created a new variable which is combination of "EventWarningStop" and time of occurrence (Before, Within, and After the visit start). This variable contains nine labels Event, Warning and Stop for each three-time point. For data construction, we have used the similar method explained in Model A.

The main objectives for the model C are:

- Relation and path among Stop, Warning and Event type of code for the three different time points (before, within and after the visit).
- How is it connected with different time interval?
- What is the direct and indirect path among them?
- Is it possible to predict the number of Stop, Warning and Event type of codes based on other factors simultaneously?

Model C:

$$Stop.1Before = a_1 Event.1Before + a_2 Warning.1Before + \epsilon_1$$

$$Warning.1Before = b_1 Event.1Before + \epsilon_2$$

$$Event.2 \in c_1 Stop.2 \in +c_2 Warning.2 \in +\epsilon_3$$

$$Warning.2 \in d_1 Stop.2 \in +d_2 Warning.1Before + \epsilon_4$$

$$Warning.3After = e_1 Stop.3After + \epsilon_5$$

$$Stop.2 \in f_1 Stop.1Before + \epsilon_6$$

Indirect Effect:

$$\text{IndEff1} := a_2 * b_1$$

$$\text{IndEff2} := d_1 * c_2$$

$$\text{IndEff3} := b_1 * d_2$$

$$\text{IndEff4} := f_1 * a_2$$

Based on the fitted model, it is found that the model C is statistically significant (p-value: 0.00). Model diagnostics also support the conclusion. It is also found that number of Stop type of codes can be determined by number of Warning (0.431) and Event (0.403) type of codes and these are positive before the visit start. We found the similar path for model A. All these labels are positively associated with Stop. In other words, if the number of Warning type of error increases, the number of Stop type of codes will increase by 0.431 times before the visit. Similarly, number of Warning type of codes can be predicted by Event (0.429). Table 13 also showed the variance and indirect effect of the model C. The path of the model can be shown by the Figure 7.

Table 13: Parameter estimate, p-value and standardized estimate for the path model C.

Regressions:						
	Estimate	Std.Err	z-value	P(>|z|)	Std.lv	Std.all
Stop.1Before ~						
Evnt.1Bfr (a1)	0.403	0.009	44.952	0	0.403	0.403
Wrnng.1Bf (a2)	0.431	0.009	47.977	0	0.431	0.431
Warning.1Before ~						
Evnt.1Bfr (b1)	0.429	0.01	41.489	0	0.429	0.429
Event.2In ~						
Stop.2In (c1)	0.456	0.012	38.895	0	0.456	0.456
Wrnng.2In (c2)	0.083	0.012	7.072	0	0.083	0.083
Warning.2In ~						
Stop.2In (d1)	0.537	0.01	56.323	0	0.537	0.537
Wrnng.1Bf (d2)	0.107	0.01	11.188	0	0.107	0.107
Warning.3After ~						
Stp.3Aftr (e1)	0.611	0.009	67.583	0	0.611	0.611
Stop.2In ~						
Stop.1Bfr (f1)	*0.11*	*0.011*	*9.643*	*0*	*0.11*	*0.11*

Covariances:						
	Estimate	Std.Err	z-value	P(>|z|)	Std.lv	Std.all
.Event.2In ~~						
.Warning.3After	0.034	0.008	4.317	0	0.034	0.049

Variances:						
	Estimate	Std.Err	z-value	P(>|z|)	Std.lv	Std.all
.Stop.1Before	0.503	0.008	61.859	0	0.503	0.503
.Warning.1Befor	0.816	0.013	61.859	0	0.816	0.816
.Event.2In	0.743	0.012	61.859	0	0.743	0.744
.Warning.2In	0.692	0.011	61.859	0	0.692	0.693
.Warning.3After	0.628	0.01	61.859	0	0.628	0.627
.Stop.2In	0.988	0.016	61.859	0	0.988	0.988

Defined Parameters :						
	Estimate	Std.Err	z-value	P(>|z|)	Std.lv	Std.all
IndEff1	0.184	0.006	31.382	0	0.184	0.184
IndEff2	0.045	0.006	7.017	0	0.045	0.045
IndEff3	0.046	0.004	10.802	0	0.046	0.046
IndEff4	0.047	0.005	9.454	0	0.047	0.047

However, within the visit duration, the path is inverse. As the number of stop increases the number of warning and event type of codes increases and it is moving towards event type of codes. It is because when the technician fixing the problem they are getting more event type of codes. And the warning and stop type of codes within the visit duration are connected with the warning and stop type of codes before the visit starts and the relations are positive.

On the other hand, warning and stop are connected after the visit duration and warning after the visit connected with event within the visit duration and stop type of codes after the visit connected with the event type of codes before the visit start. It is not possible to explain the reason for that without further analysis and knowledge about different codes.

So, model C explained all the objective and those are statistically significant. Additionally, all the indirect paths that we consider in the model are statistically significant.

What we can conclude from these probabilistic state space models is that there are clusters of the events which have a higher transition rate amongst them. The probability to move between clusters is not as likely and shows how different labels are associated. There can be multiple explanations for this result and should be interpreted by the technicians/engineers. This can assist to create a meta even label for a type of even encompassing a set of associated events. It can also provide an expected outcome for a set of events to occur due to an initial error being raised and how other ones can be more or less likely to occur in sequence.

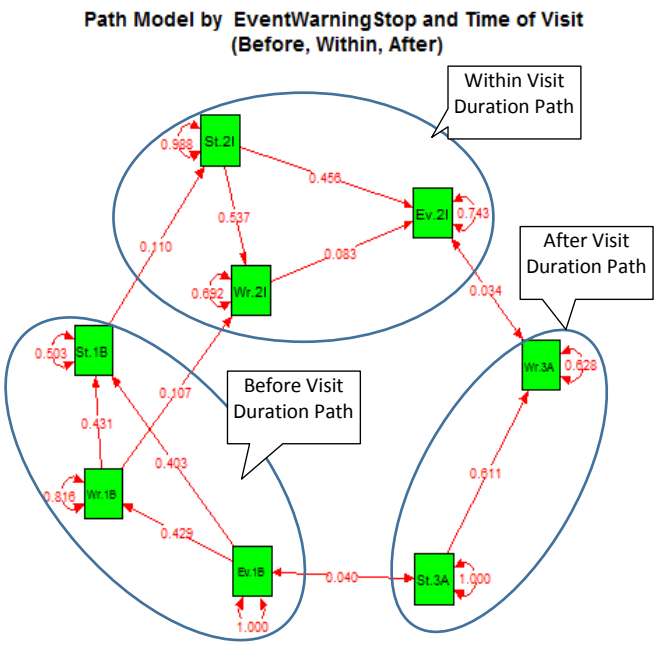


Figure 8: Path model for the model C. Arrow indicates the direction of the variable and the value indicates coefficient of the parameter. Ev, St, and Wr stand for Event, Stop and Warning respectfully and 1B, 2I and 3A stand for Before, Within and After visit duration.

V. Clustering Codes and Visits

Here we provide an analysis of the data with some non-classical statistics, such as cluster analysis and community detection. Cluster analysis is widely used and the results allow us to differentiate different groups by simplifying the problem greatly, and in a similar goal we apply community detection to the dataset. Since we have a network of association we can view the data as a community of edges and not only as data points. This connected view of the labels will provide a visual assistance to see how the connectivity affects the results.

A. *Clustering*

We used clustering methods for investigating the commonalities in the dataset. In this regard, we constructed three different datasets. First one is to consider each VisitId as the entity and then aggregate the occurrence of the codes in the different columns. The second strategy is to set the Code as entity and calculate the frequency of the code in each VisitIds in different columns and the last one is aggregate codes based on parks. Following we discuss our findings and procedures.

B. *Cluster Visit ID*

In this dataset VisitIds are entity and codes frequency are aggregated into each column. After creating this data, different standardization techniques (normalization, quintile, and binary variables) were applied. For this project we have used nested clustering technique because distribution of frequency of most of the codes are very high. To find the best number of clusters we used Davies-Bouldin[2] (DB) index that uses average similarity between each cluster.

After an exhaustive search on the whole dataset and with different subset with respect to time of code appearance (before, within, and after the visit duration) we conclude that quantile approach is the best one on this dataset and K-means can better catch the dissimilarities among different Visit IDs.

[2] The original paper: Davies, D. L., & Bouldin, D. W. (1979). A cluster separation measure. IEEE transactions on pattern analysis and machine intelligence, (2), 224-227.

There were 9 cluster found for the visit IDs. The distribution of visit IDs with respect to cluster ID (Tbale 13) showed that cluster 4 contains the most visits IDs.

Table 14: Distribution of visit Ids in clusters

Cluster	1	2	3	4	5	6	7	8	9
Number	350	327	962	2717	457	1338	844	284	374

Interpreting of each cluster and giving a label to each cluster would be hard without knowing about the relationship of these clusters with other variables. For more illustrations, we used some plots to show the abstract view of the outputs.

Figure 9 is the bubble plot of clusters verses each park normalized by Park. In the following plot, each point indicates the frequency of VisitIDs in each Park in each cluster. This frequency then is normalized based on all VisitIds happened in that Park. Based on this graph we could conclude that most of Visits happened in Park024 are in cluster 4. In contrast, for Park026, the most Visits are in cluster 6.

Next plot is similar with the last one but here the values are normalized by clusters rather than parks so that we can investigate frequency of codes that happened in each cluster with respect to each cluster.

Here we can see that out of all visits which are in cluster 9 most of them are happened in Park008 than any other parks. This is the same for cluster 2 and cluster 1 in which visits happened mostly in Park001 and Park034 respectively. The interpretation of these charts from either perspective will gives good understanding of patterns in Visits with respect to Parks.

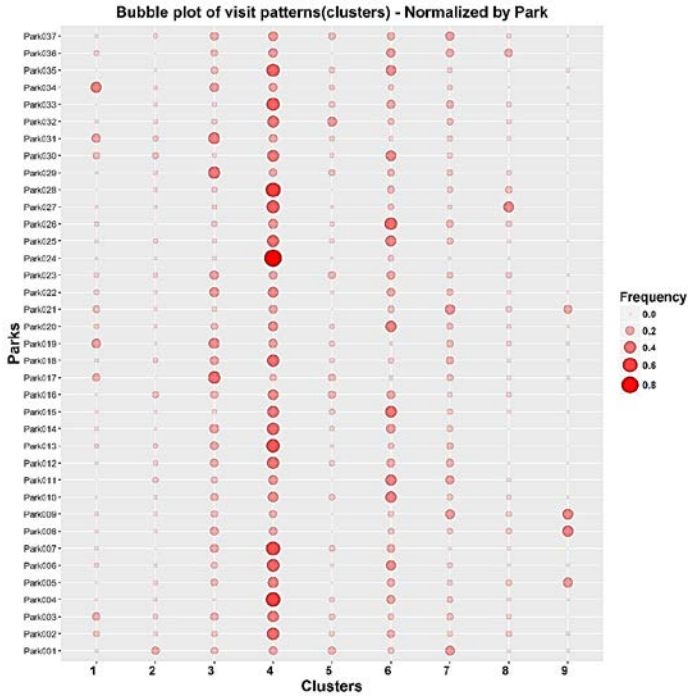


Figure 9: Park versa clusters

Moreover, sometimes maybe frequency of Event, Warning, and Stop in each visit might be more interesting. In this regard, the two following chart is the same as the chart in figure 9 but here each observation indicates the frequency of codes which happened in each Visits instead of frequency of visits. In figure 11, we can say that most of codes which are happened during the visits for Park028 and Park027 are from the same pattern of cluster 8. Furthermore, for these two parks no visits happened with the pattern in cluster 9.

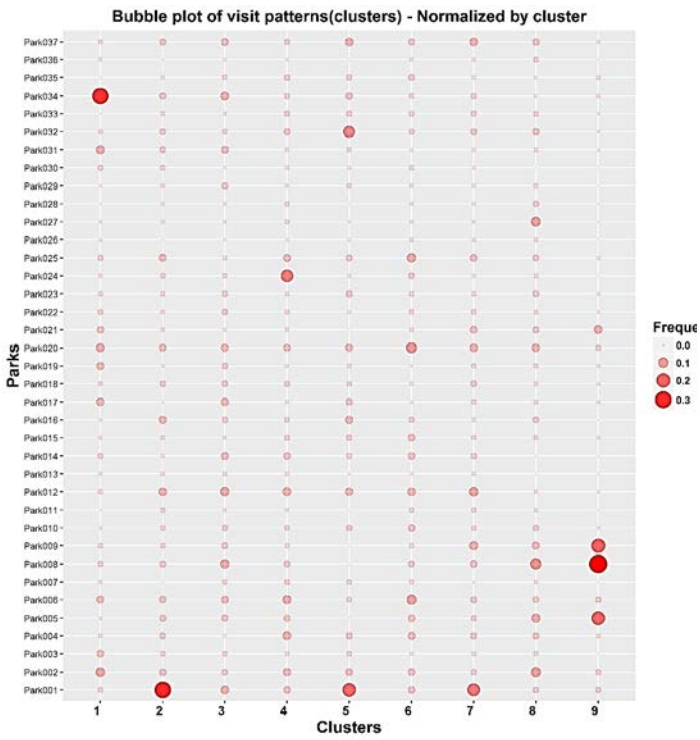


Figure 10: Park versa clusters

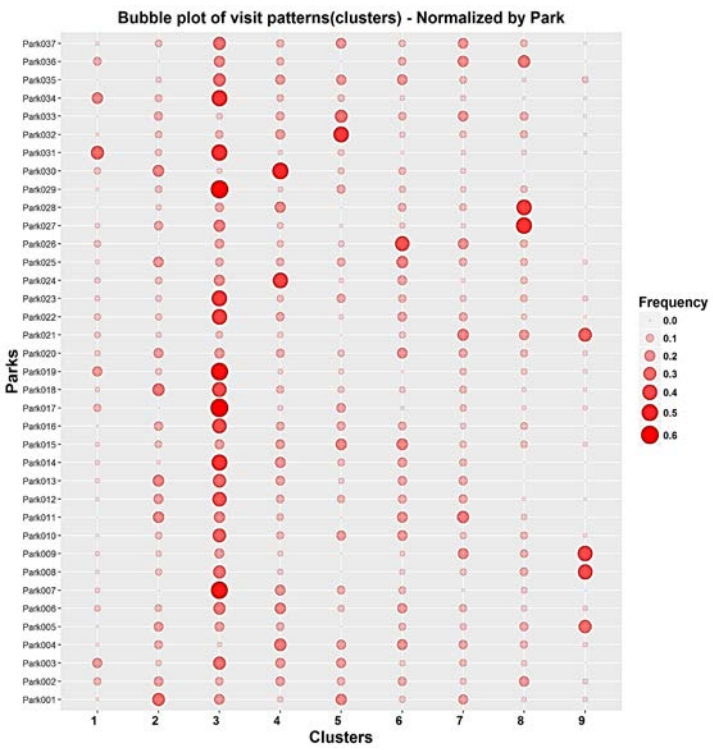


Figure 11: Park versa clusters based on code frequency

Next plot clusters Parks based on Visit Clusters (clusters of visits that we applied before) and Occurrence of visit in each cluster. This is the clustered version of figure 9. Here we just clustered and reordered park so that each color in this plot indicates similar parks according to Visit pattern. This can be interpreted as cluster into cluster. Because basically this plot was the output of clustering and we did another clustering (based on parks) on this plot so that we can see how different parks are similar to each other in terms of Visit patterns if we can call each cluster as a special pattern.

Also, one could investigate the relationship of these VisitId clusters with other variables in the dataset. The next chart shows the frequency of codes occurred in different Visits based on clusters and VisitType which is normalized by clusters. We can say there are more codes occurred in Visits in cluster 3 which have Visit Type=”CU” than any other types. For visits in cluster 2, type is mostly “UN”.

In order to better find the patterns, we divided our dataset into three subsets based on the codes that happened Before, After, or within the visit and redo all mentioned steps (which we ran on whole dataset in previous part) with tuning parameter. We will have three datasets before dataset, after dataset, and within dataset.

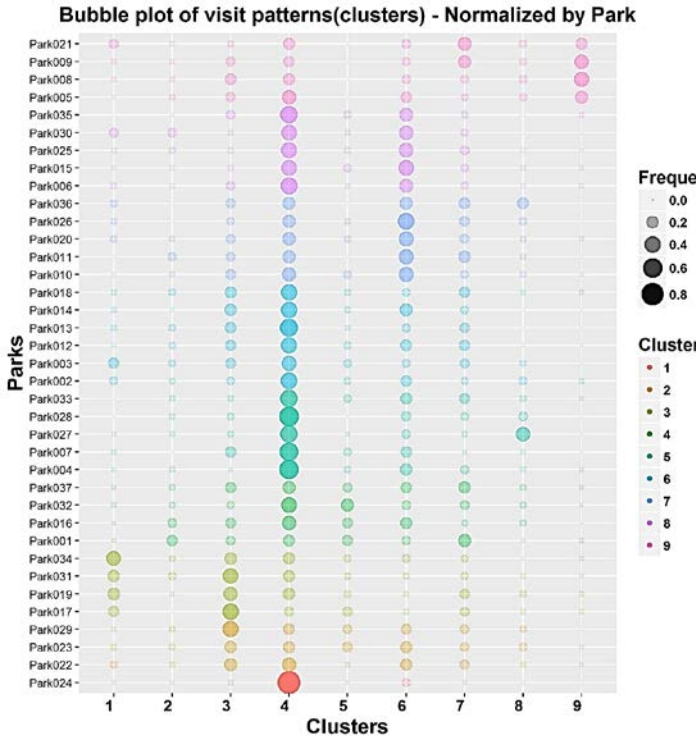


Figure 12: Park cluster on visit cluster

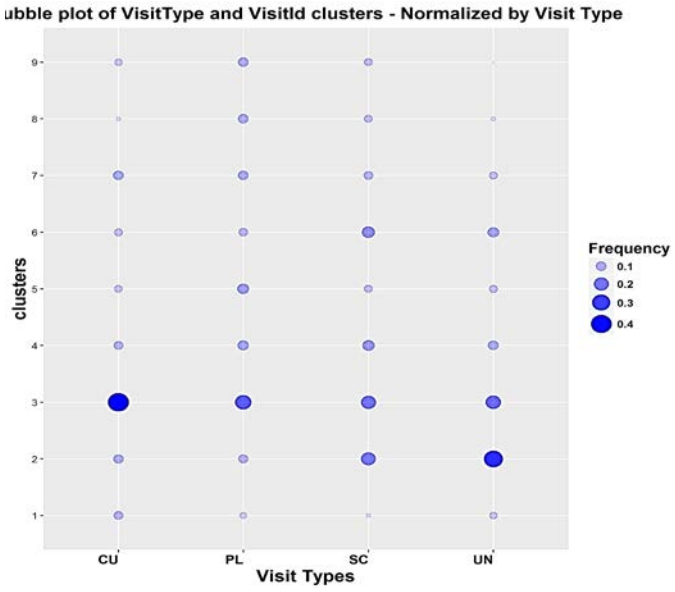


Figure 13: Visit types and VisitId clusters

Then again we used similar plots from previous section to dig into patterns. First, we selected all the codes which occurred before visits and keep those records only. In the plot 14 the observation is frequency of visits which happened in each park based on different clusters (Pattern). For example, we can say most of visits in Park031 and Park034 are from cluster 7 and inversely most of visits for Park022 are from cluster 3.

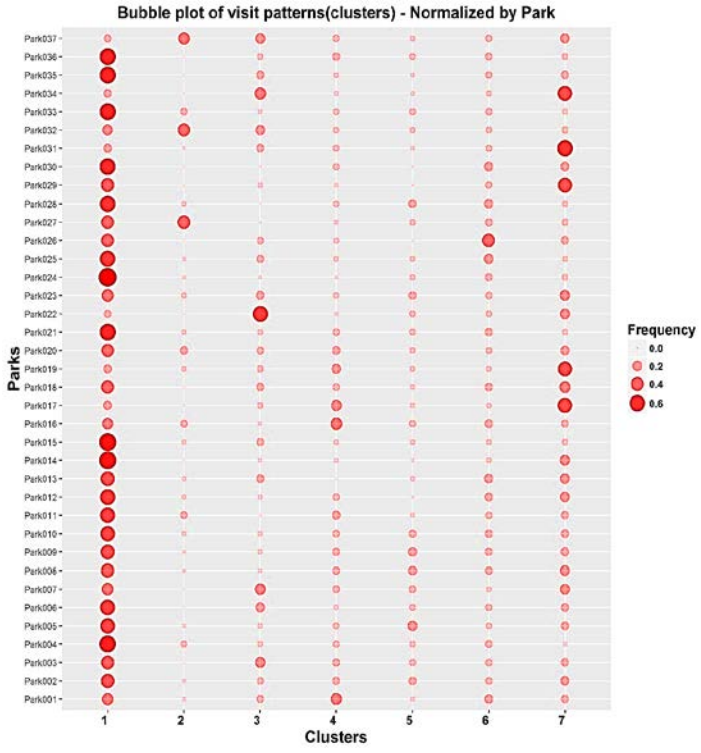


Figure 14: Park and clusters based on before dataset

In the appendix 3, we put all the other plots for before, after, and In datasets. These plots are based on both frequency of visits and frequency of codes which happened in each visit.

C. Parks are entity and columns are Codes

Here, we considered each park as entity and aggregated occurrence of codes in each park as our variables. So, we will have 37 rows and 642 variables

(one variable for each code). Using the same strategies for clustering, using quantile method for discretizing columns, we got 7 different clusters. One can extend this approach just for the codes occurred Before, After, and within each visit. Table 14 shows the cluster memberships. Now we can conclude that in terms of frequency of code occurrence in whole dataset Park002 and Park012 are similar to each other than to any other park. In is important to mention that here because number of observations are less than number of variables it is not possible to use PCA for dimensionality reduction directly.

Table 15: Distribution of visit Ids in clusters

Cluster Number	Parks
Cluster 1	Park001, Park006, Park020, Park034
Cluster 2	Park002, Park012
Cluster 3	Park003, Park007, Park014, Park017, Park019, Park022, Park023, Park031
Cluster 4	Park004, Park016, Park018, Park024, Park025, Park032, Park037
Cluster 5	Park005, Park021
Cluster 6	Park008, Park009
Cluster 7	Park010, Park011, Park013, Park015, Park026, Park027, Park028, Park029, Park030, Park033, Park035, Park036

D. Codes are entity and VisitId are Codes

In this step, we aggregated data based on the codes. Here, codes are observation and columns are Visit IDs. Because we wanted to avoid getting same pattern in this step, we first discretized data based on occurrence of codes in all visits and then transpose data so that codes are as observations. The data was so skewed and the only method which gave us the best result was the Ward method in hierarchical clustering. To avoid having skewed clusters, we did a hierarchical clustering once and then find the cluster with the largest members and again re run Ward method on those observations. This is the two-step clustering. Finally, the distribution of codes in each cluster is as follow:

Table 16: Distribution of visit Ids in clusters

Cluster Number	1	2	3	4	5	6	7	8	9
Members	148	65	16	13	19	47	295	21	18

If we calculate the occurrence of each code based on the whole dataset we will have table 16. Although most of code Ids are placed in cluster 7 (with 295 code members), in terms of frequency these 295 different codes happened just 9920 times in whole dataset.

Table 17: Distribution of visit Ids in clusters

cluster	2	3	4	5	1	6	7	8	9
Frequency	51935	38992	17538	13218	29032	4974	9920	4250	2070

Out of all error codes there are 9 codes which cause manual stop. The following tables shows this codes together with the assigned clusters. Most of these codes are placed in cluster 1 and 3 and code number “1002” is in cluster 6.

Table 18: Codes cause the stops and cluster membership

Code	CodeCluster
1007	3
1014	3
1001	1
1002	6
1003	1
1008	1
1015	1
1017	1
1021	1

Furthermore, we can investigate the frequency of each of 642 codes with respect to Stop Urgency variable. We can see that all the codes which are in cluster 4,5, and 9 are either have urgency of 0 or 5. There are only two clusters of codes in which the urgency is 6. These are clusters 1 and 7.

In chart 15, the observations are frequency of codes in whole dataset which are shown in crosstabulation of Urgency and Code cluster and normalized by Urgency. We can say that most of codes which have Urgency of 2 and 1 are in cluster 1. Also, most of codes from urgency 4 are placed in cluster 1. Appendix 4 contains list of codes and associated clusters.

Table 19: Cross tabulation of urgency and code clusters

	Clsuter 2	**Clsuter 3**	**Clsuter 4**	**Clsuter 5**	**Clsuter 1**	**Clsuter 6**	**Clsuter 7**	**Clsuter 8**	**Clsuter 9**
Urgency 0	26	8	8	14	57	21	191	11	14
Urgency 1	5	3	0	0	8	1	0	0	0
Urgency 2	6	1	0	0	19	4	9	1	0
Urgency 3	2	0	0	0	4	3	6	0	0
Urgency 4	0	0	0	0	2	0	2	0	0
Urgency 5	26	4	5	5	57	18	85	9	4
Urgency 6	0	0	0	0	1	0	2	0	0

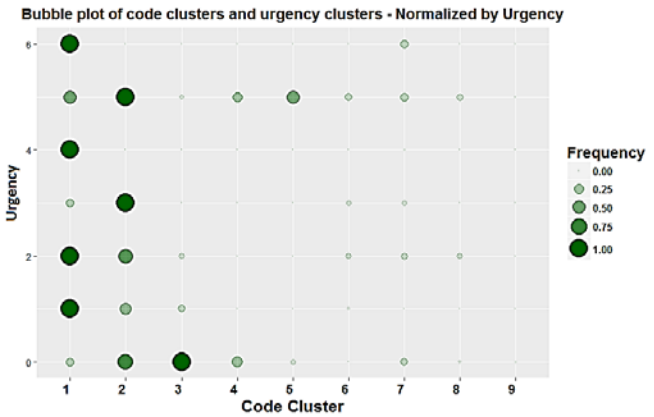


Figure 15: Park and clusters based on before dataset.

E. Analyze the stations based on Visit clusters

In this section, we visualized stations based on the clusters (which here are output of clustering for Visit clustering). For this purpose, we used three different datasets with respect to occurrence of the code Before, After, or within the visit.

Figure 16 is for Park019 and frequency here is number of visits which are in the same cluster for codes happened after the visits. But it is important to note that these are the output of clustering visits IDs based on the code happened after the visit. Here for example, for station number 1177 we can say that most of visits in this station followed the same code patterns that is why most of visits are in just one cluster (cluster 4). For station number 1172, most of visits are similar and they are in cluster 1 of visit clusters but some of visits in this station are from cluster 8 as well. We could conclude that stations number 1182, 1178, 1177, and somehow 1173 in this station have similar patterns in visits based on the codes occurred after the visit.

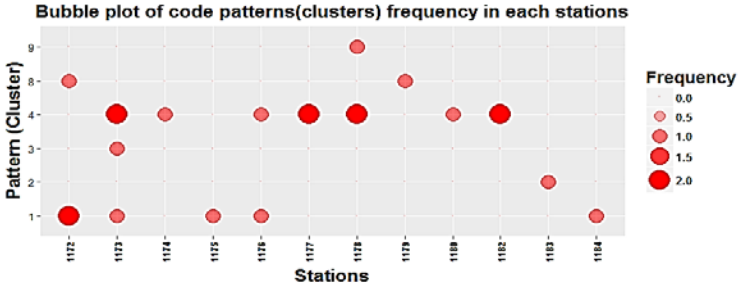


Figure 16: Bubble plot of Stations in Park019 (After data)

The next plot shows the frequency of visits for visits pattern before visit time for each station in Park019. One can see that stations number 1172, 1176, and 1177 have the same pattern. In these stations, most of visits are from cluster 7. So, we can say they are same in terms of visits in which same codes happened before the visit.

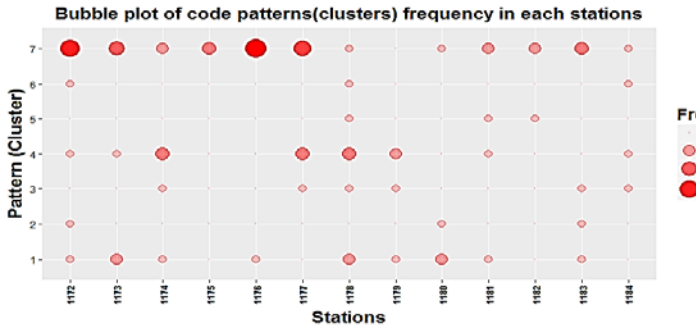


Figure 17: Bubble plot of Stations in Park019 (Before data)

Next chart is based on codes happened within the visit. Again, y variable indicates the clusters of visits based on codes happened within the visit and x is the stations. As it is clear, in this chart compared to previous charts most of cells are filled. This means that patterns of these stations in terms of codes which happened within visits, are different form each other.

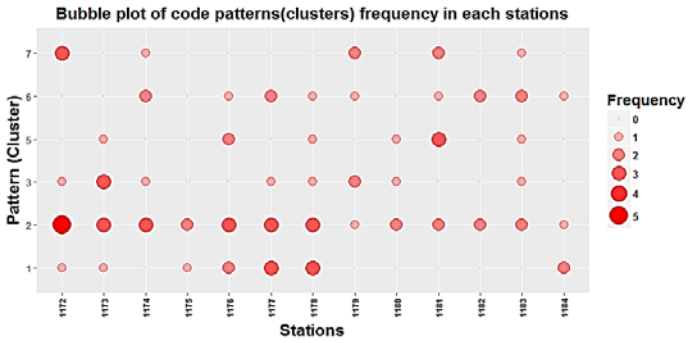


Figure 18: Bubble plot of Stations in Park019 (within visit data)

F. Social Network Analysis

If we consider the transition matrix, this matrix here can be treated as adjacency matrix which we can create network of codes which are interconnected. The adjacency matrix for this problem is directed because if A and B be two codes, for some of the cases first A happens then B and Vis versa with unequal magnitudes. Also, this matrix is weighted matrix and weights between the edges is either frequency or time. In this regard, we created 6 different adjacency matrices.

Table 20: most central codes in code networks

Data	Most Central Codes
Transition matrix for frequency of codes Before visit	3130, 13902, 1001, 13, 14, 8, 13900, 5122, 59, 10105
Transition matrix for average time between codes Before visit	13902, 5112, 5111, 7, 9303, 2, 1018, 14001 ,64101, 18
Transition matrix for frequency of codes After visit	3130, 13902, 13, 14, 1001, 5122, 14302, 13900, 10105, 18
Transition matrix for average time between codes After visit	13902, 8, 13, 1001, 18, 9303, 7, 2, 3130, 17027
Transition matrix for frequency of codes Within visit	1020, 1001, 1005, 13902, 1023, 7, 18, 8, 13900, 2
Transition matrix for average time between codes Within visit	8, 13902, 9, 1023, 1001, 1020, 2, 7, 18, 14001

The first three matrices are based on transition matrix that we used in Markov Chain modeling. We created one matrix for codes which happened before visit, one for after visits, and the last one for within visit codes. The next three matrices are based on adjacency matrix of codes but this time instead of

frequency, the values in matrix are average time between two codes.

First we computed a betweenness centrality measure for all the six matrices. The betweenness is (roughly) defined by the number of geodesics (shortest paths) going through a vertex or an edge, which gives us an estimate for the question 'when going from a random state to another random state, is there a state which is frequently visited?'. Here we can state that a node (which are codes here) with higher betweenness centrality would have more control over the network of codes, in other words most of codes passes through this code.

Following table shows the most central codes for each of these six networks. It can be seen that except for transition matrix for frequency of codes Before and After visit most of central nodes are varies in each dataset.

Based on our findings in the project we figured out there might be some codes which have strong relations which other than with other groups of code. To investigate these interactions, it is good practice to plot the graph of networks and look for communities. Here, nodes are codes and edges are either the frequency or mean time between happening of two codes.

Table 21: Detected communities' members

Data	**# Communities**	**# Communities with one member**
Transition matrix for frequency of codes Before visit	57	3
Transition matrix for average time between codes Before visit	59	40
Transition matrix for frequency of codes After visit	64	21
Transition matrix for average time between codes After visit	119	80
Transition matrix for frequency of codes Within visit	54	14
Transition matrix for average time between codes Within visit	105	75

Because all of datasets are directed and weighted we used Infomap[3] community finding which works good on this dataset. This technique find community structure that minimizes the expected description length of a random walker trajectory. Number of communities of codes are different in each dataset. Table 21 shows the number of communities detected in each dataset.

The above table indicates number of communities with just one node in each community. Next we visualized some these communities which have highly central nodes in them. First graph shows one of the central communities for the dataset of transition matrix for frequency of codes Before visit.

Codes 13 and 14 are most central nodes in this graph and one can see all the interactions between different nodes in this community of codes (We just added and X to each codes name). Based on this graph we can conclude that code 14 causes code

[3] The original paper: M. Rosvall and C. T. Bergstrom, Maps of information flow reveal community structure in complex networks, PNAS 105, 1118 (2008)

12110 and then 13 in a loop. Also, there is another loop that starts with code 14 then 2122 and then to 2123. We can depict the graph of transition matrix for average time between codes Before visit as follow.

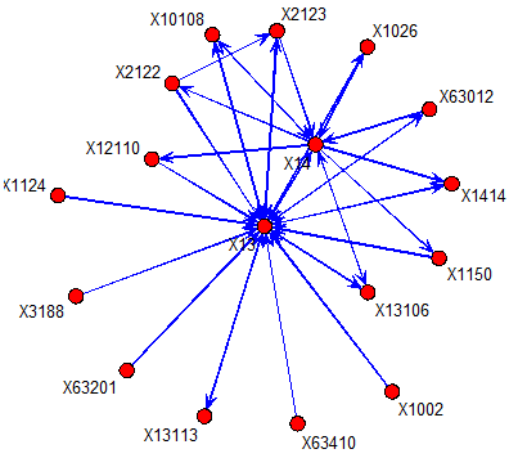


Figure 19: A community of codes for before visit codes

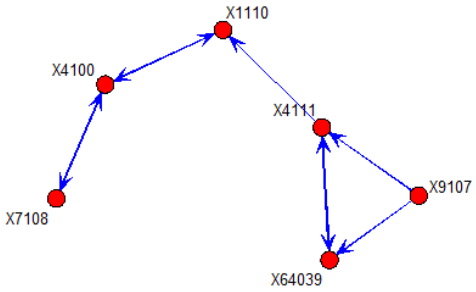


Figure 20: A community of codes for after visit codes

In this community, there is strong relation between code number 4111 and 64039 and they interact with each other in both ways. Next graphs are for transition matrix of frequency and time for codes happened after visits.

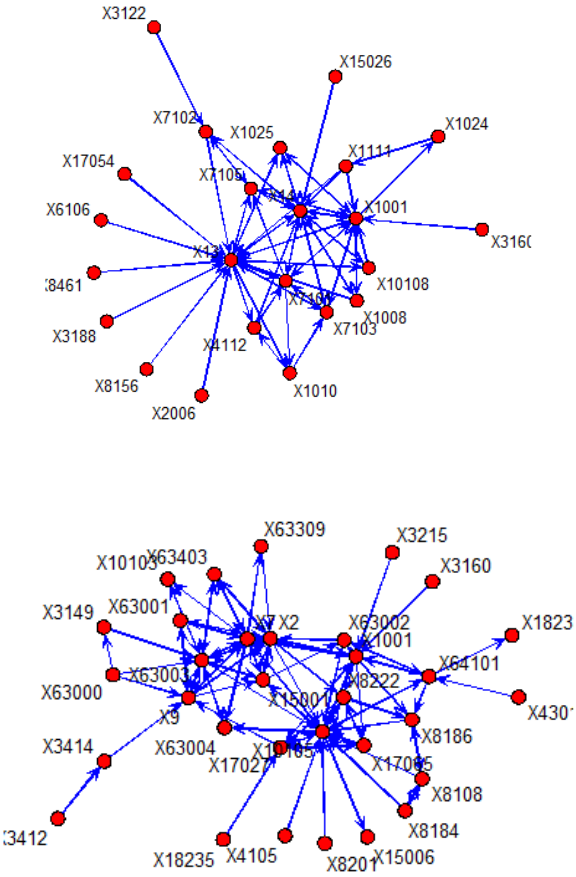


Figure 21: A community of codes for after visit codes

Next these graphs are for transition matrix of frequency and time for codes happened within visits.

G. Pattern mining

In this section Association Rule Mining and Sequential Pattern Mining techniques have been applied. In both approaches we used classified visits based on the occurrence of the code before, after and during the visits. We call these datasets 'before dataset', 'after dataset', and 'within dataset'.

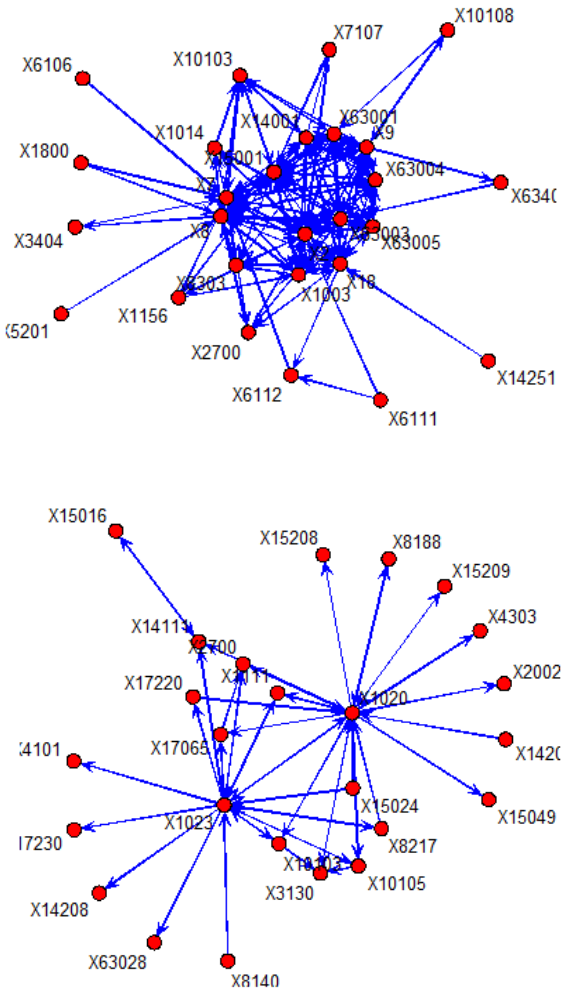


Figure 22: A community of codes for within visit codes

H. Association rule mining

To get the association rules, the dataset first divided into three different parts by having codes happening before, after and during a visit and then association rules were extracted based on occurrences. The visits are rows and codes are columns in these three datasets. Values in tables are frequency of the specific code in specific visit.

a) Before Visit

First dataset contains codes which have happened before visits. The minimum values for support and confidence are respectively 0.15 and 0.99 and the minimum length of rules is 2. In this category, a total of 448 rules were extracted. Rule length distribution for this category is as follows.

Table 22: Rule length distribution for errors that happen before the visit

Number of error code in each rule	2	3	4	5	6	7
Number of rules	28	124	158	100	33	5

Most of the patterns (158 rules) contain 4 codes in the left-hand side (LHS) and right hand side (RHS) of the rules. According to next table, the mean of confidence for all the rules is 0.99 which is approximately 1. Mean and Median of rules supports is close together at around 0.16. More importantly mean of rules lift is about 4.4 which indicates that rules are not happening in random pattern.

Table 23: summary of quality measures for the rules that happen before the visit

support	**confidence**	**lift**
Min. :0.1500	Min. :0.9900	Min. :2.804
1st Qu.:0.1597	1st Qu.:0.9970	1st Qu.:3.424
Median :0.1626	Median :1.0000	Median :5.485
Mean :0.1696	Mean :0.9982	Mean :4.453
3rd Qu.:0.1736	3rd Qu.:1.0000	3rd Qu.:5.549
Max. :0.2919	Max. :1.0000	Max. :5.608

Table 23 contains some of the extracted rules with high confidence and support. For the first rule we can say that if code 15001 happens, code 14001 will happen with almost %100 probability. Rule six prove that if codes 1018 and 5112 happens together then rule 5104 should occur with almost %100 probability. If one create a rule based database these rules can be used to predict possible patterns. For example,

although the support for the last rule is about %2.7 but it gives an important pattern about the code 1001. Code 1001 is one the codes which causes the stop in a station. This rule induces that if error codes 1022 and 7111 happens then code 1001 will appear.

Table 24: Extracted rules sorted on lift

antecedent => consequent	support	confidence	lift
{15001} => {14001}	0.172152	0.990714	5.55696
{1018,5104,5110,5112,5122} => {5111}	0.168764	0.996190	5.60798
{15001} => {14001}	0.172152	0.990714	5.55696
{13902,15001} => {14001}	0.167957	0.990485	5.55568
{1018,5110,5112} => {5104}	0.169409	0.991501	5.52637
{1018,5112} => {5104}	0.169409	0.990566	5.52115
{1022,7111}=>{1001}	0.027423	1.000000	5.77188

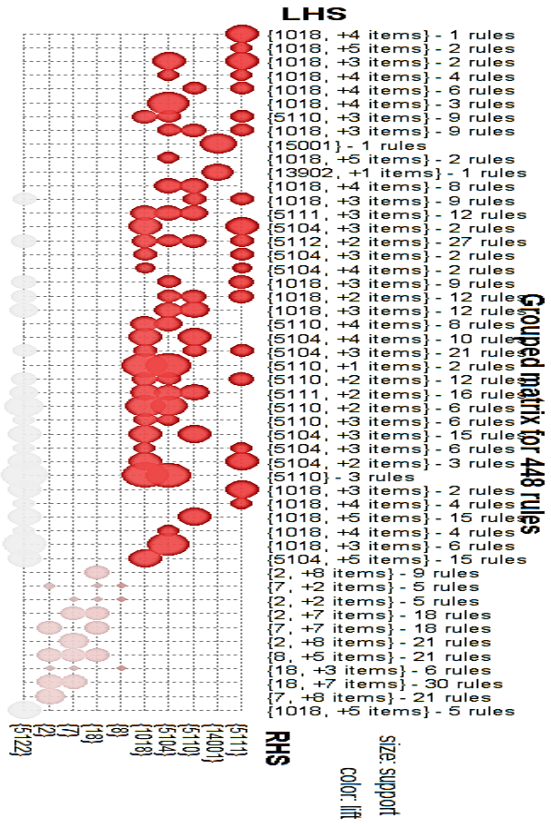


Figure 23: Grouped matrix for errors before the visit

Antecedents that are statistically dependent on the same consequents are similar and thus can be grouped together. The default interest measure used is lift to visualize the grouped a balloon plot with antecedent groups as columns and consequents as rows was used. The color of the balloons represents the aggregated interest measure in the group with a certain consequent and the size of the balloon shows the aggregated support. The default aggregation function is the median value in the group. The number of antecedents and the most important (frequent) items in the group are displayed as the labels for the columns. Furthermore, the columns and rows in the plot are reordered such that the aggregated interest measure is decreasing from top down and from left to right, placing the most interesting group in the top left corner. Although, it is important to just focus on the rules which are interesting based on knowledge domain, we used all the rules for this visualization. From the plot, we can conclude that code 1018 as an important code together with some other codes in left hand side of rules causes different codes such as 5111, 14001, and 5110.

Figure 23 visualize association rules using vertices and edges where vertices represent codes and edges indicate relationship in rules. Size of the bubble in this graph is related to support and color shows the lift. According to this graph code number 5111 is in the center and this indicates that this code appeared in lots of left hand sides and right hand sides of the rules. Also, code 5113 occurred mostly in left hand side which means the patterns in which this code is caused other codes are more frequent than the patterns in which Code 5113 is the consequence.

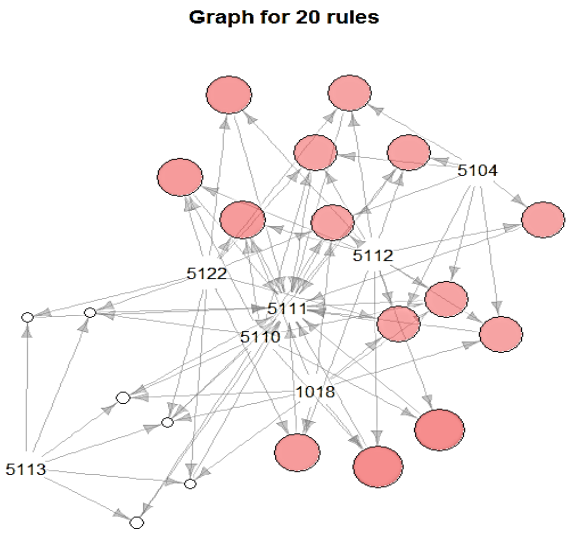


Figure 24: Network of association rules for before dataset

b) ***After Visit***

In this category, we extracted rules related to codes that happen after visits. The values for support and confidence are respectively 0.03 and 0.09 and the minimum length of rules is 3. In total, 284 rules were extracted. Rule length distribution for this category is as follows. In this subset of data, rules with 3 LHS and RHS are more than others. This is in contrast with before dataset in which number of codes in each rule for rules with 4 codes was the highest among the other. This caused by the fact that distribution of codes in after dataset is more condensed than before code.

Table 25: Rule length distribution for errors that happen after the visit

Number of error code in each rule	3	4	5	6
Number of rules	93	80	35	6

Min of the lift for these rules is about %5 and the median is %27.8 which is higher than min. This means we have some patterns with less possibility of happening than random event and some rules which are rarely happens by random chance. By changing the support and confidence in before dataset we could not get any rule with this high lift according to reasonable support. We could say that in after dataset rules happens rarely than before dataset.

Table 26 summary of quality measures for the rules that happen after the visit

support	**confidence**	**lift**
Min. :0.03023	Min. :0.5725	Min. : 4.97
1st Qu.:0.03253	1st Qu.:1.0000	1st Qu.:18.93
Median :0.03253	Median :1.0000	Median :27.80
Mean :0.03402	Mean :0.9714	Mean :22.77
3rd Qu.:0.03406	3rd Qu.:1.0000	3rd Qu.:29.36
Max. :0.05281	Max. :1.0000	Max. :29.36

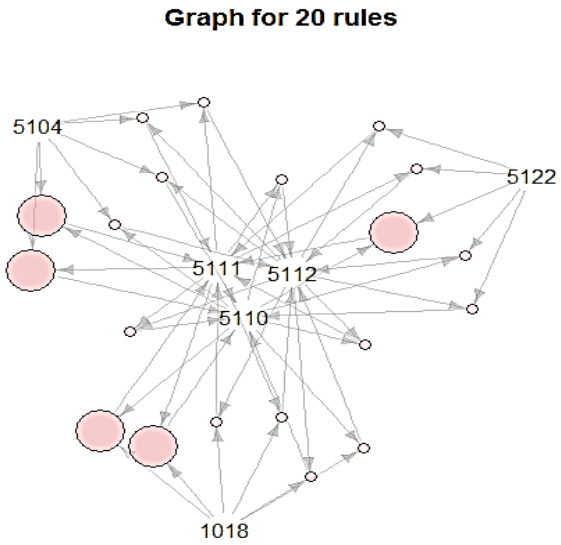


Figure 25: Network of association rules for after dataset

In contrast with before dataset here we should not look for those codes which cause stops because usually after visits no stop will happen. Here the

second rules stats that if code 2 and 18 happens then code 8 will appear with probability of %100.

Next there is network chart of association rules based on the codes for before dataset. It seems again here codes which starts with 5 are almost at the centers of graph and they cause pattern with highest lift.

Table 27 summary of quality measures for the rules that happen after the visit

antecedent => consequent	support	confidence	lift
{14038,15038} => {1390}	0.032146	1	8.6810
{8,18} => {7}	0.052812	1	18.9347
{2,18} => {8}	0.052812	1	18.9347
{18,13902} => {8}	0.030233	1	18.9347
{7,18,13902} => {8}	0.030233	1	18.9347
{8,18,13902} => {2}	0.030233	1	18.9347

c) ***Within Visit***

In this category, we have extract rules related to codes that happen during visits. The values for support and confidence are respectively 0.03 and 0.09 and the minimum length of rules is set to 3. Total of 448 rules were extracted. Based on rule distribution we could say rules with five codes are dominant.

Table 28 Rule length distribution for errors that happen within the visit

Number of error code in each rule	2	3	4	5	6	7	8
Number of rules	31	108	213	230	140	47	7

Although the confidence for these rules are almost in average 1 but the lift is less than two previous datasets which is indication of rules might not be interesting too much.

Table 29 summary of quality measures for the rules that happen within the visit

support	confidence	lift
Min. :0.09212	Min. :0.9028	Min. : 1.000
1st Qu.:0.10545	1st Qu.:0.9888	1st Qu.: 1.701
Median :0.13838	Median :1.0000	Median : 4.411
Mean :0.13374	Mean :0.9847	Mean : 3.736
3rd Qu.:0.14922	3rd Qu.:1.0000	3rd Qu.: 5.035
Max. :0.58487	Max. :1.0000	Max. :10.453

Table 30 summary of quality measures for the rules that happen within the visit

antecedent => consequent	support	confidence	lift
{14038,15038} => {1390}	0.14099	1.0000	5.0952
{8,18} => {7}	0.11511	1.0000	5.0952
{2,18} => {8}	0.19613	1.0000	5.0918
{18,13902} => {8}	0.11511	1.0000	5.0918
{7,18,13902} => {8}	0.10544	1.0000	5.0952
{8,18,13902} => {2}	0.15327	1.0000	4.8807
{1020} => {1015}	0.10910	0.1091	1.0000

Based on first rule it can be infer that if codes 14038 and 15038 happened then code 1390 will happens with probability of %100. There were some rules which have one of the stops codes in their RHS. Last rule stats that if code 1020 happens then 1015 (which causes manual stops) will occur.

This visualization graph is mostly based on codes 8, 7, 2 and 18. It seems in within dataset, codes number 7 and 2 are most central in patterns.

I. Sequential Pattern Mining

To find the sequential patterns, we used cSPADE4 algorithm. We used the three datasets from the

[4] The original paper: M. J. Zaki. (2001). SPADE: An Efficient Algorithm for Mining Frequent Sequences. Machine Learning Journal, 42, 31–60

previous section. First we prepared the dataset based on the time sequence of code occurrence in each visit. We combined all the codes which happed in the sequence of N minutes (like a chain of 15 minutes). For example, if N would be 5 minutes and the data is as follows:

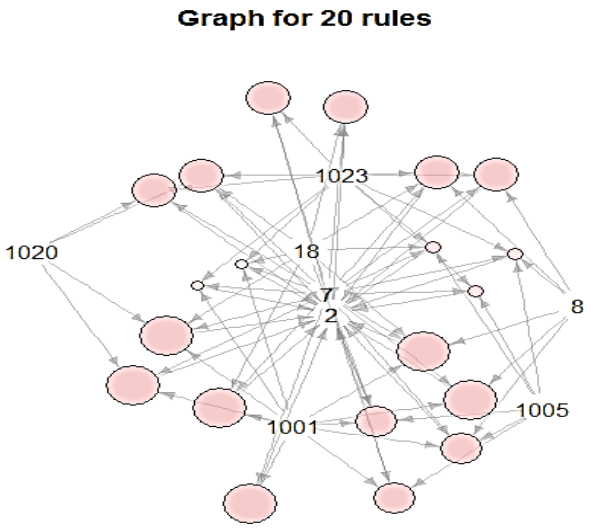


Figure 26: Network of association rules for within dataset

Table 31 example of dataset

VisitId	TimeOn	Code
178	8/24/2016 8:33	1001
178	8/24/2016 8:36	1002
178	8/24/2016 8:39	1003
178	8/24/2016 9:39	1005

We transformed this data to the below table:

Table 32 prepared data

VisitId	Code	Time Id
178	100,110,021,003	1
178	1005	2

In this data, all the codes which happened in the sequence of 5 minutes consequently are combined. Here, codes 100,110,021, and 003 happened within five minutes of each other that is why they are combined. But, 1005 happened one hour after the last code of 1003, and that is why it should be in the second sequence. In this regard, we combined codes based on sequence of 1, 5, 15, 30. Each combination sometimes has more than 10 codes which causes lots of combination if code sequence within N minutes does matter. That is why we sort codes in each combination of codes.

We got to many rules and here in the below table we put some of the rules as sample. These rules are based on sequences with different durations. We note that these rules are with respect to three datasets before, after, and within.

Table 33 Table of sample sequential rules

#	Dataset	Rule	Support	Combined by N min
1	Before	{1007}, {9,63003}	0.03	N=5
2	After	{10105}, {3130}	0.02	N=5
3	In	{1001,1020,1022}, {7111}	0.003	N=5
4	Before	{10105}, {3130}	0.03	N=15
5	After	{59},{59}	0.02	N=15
6	In	{ 1001,1020,1022}	0.03	N=15
7	Before	{3130},{10105}	0.02	N=30
8	After	{ 1111},{ 13,14}	0.01	N=30
9	In	{ 1020,1023}	0.1	N=30
10	Before	{ 1007},{ 5113}	0.08	N=1
12	After	{ 13},{14}	0.06	N=1
13	In	{1020,1023}	0.12	N=1
14	Before	{1007},{5113}	0.09	N<1
15	After	{13},{13},{14}	0.02	N<1
16	In	{1020,1023},{1001}	0.03	N<1

Based on above table we can interpret some of these rules as follows:

- Rule 1: If code 1007 happens then code 9 and 63003 will happens after this code within 6

minutes or more (note that this sequential rule is for the visits which happened before visit)

- Rule 3: If codes 1001 and 1020 and 1022 happens in a sequence (order is not important) of less than five minutes of each other, then code 7111 will happen in next sequence which is in next 5 minutes of more (for within visit dataset)
- Rule 9: Codes 1020 and 1023 happens within 30 minutes of each other. Support %10 means this rule happens in %10 of visits (for within visit events)
- Rule 15: If code 13 happens then within less than 1 minute code 13 again happens and then within less tha 1minute code 14 will happen in %2 of visits.

One good approach to analyze these huge amounts of sequential rules is to compare the support of the same rules which happened before and after or within and before (or any other combination) of the visits. In table below 10 rules are extracted which were similar in before and after dataset. In both datasets, codes are combined if they happened in a sequence of 15 minutes of each other.

For example, rule number 7 which is, if codes 9 and 63003 happens in an interval of 15 minutes, then codes 2,7,8,18,13902,14001, and 15001 happens in a next sequence which is 16 minutes (or more) after the first sequence. This rule has %2 support on before dataset and just %0.6 support in after dataset. This means this sequence is more probable in before the visits incidents that after the visit.

Table 34 Compare sequential rules for before and after datastes

rules	Rules	After (support)	Before (Support)
1	{13,14}	0.062	0.045
2	{13309}	0.008	0.042
3	{59}, {59}	0.024	0.041
4	{2,7,8,18,13902,14001,15001}	0.007	0.036
5	{10105}, {3130}	0.013	0.025
6	{3130},{10105}	0.008	0.024
7	{9,63003}, {2,7,8,18,13902,14001,15001}	0.006	0.02
8	{5122,13140}, {5122}	0.008	0.012
9	{5122}, {5122,13140}	0.005	0.01
10	{10118}, {10128}	0.007	0.008

VI. LIMITATIONS

The major limitation in this project is that, as this is more descriptive task than predictive, nobody can make sure that whether outputs are useable and interesting except domain experts who have good knowledge about the dataset. Although competition task was divided into five tasks but some of the tasks like “Consider breaking the analysis into codes and patterns occurring prior to the Visit code, separately from those occurring on or after the Visit code starts” can be accomplished using many different approaches. For example, for this specific task (task 3) we have used sequential pattern mining, network analysis, Markov Chain and some others.

Another limitation is that there is no exact and detail information about the error codes. If there would be more general classification of each error code so that categorizing similar codes and making better interpretations would be easier for analysts.

Lack of domain knowledge expertise in our group prevent use to interpret models very well. Most of the times analyst need more information about the outputs so that they can dig into the results and make that informative and more clear.

Due to highly overlapped questions we prefer not to mention exactly which analysis is for which question but one can simply find all of the answers in different sections.

VII. RESULT AND DISCUSSION

We have shown in the analysis here a significant amount of information is gathered which can be analyzed to provide associations of the visit durations, parks and errors. This information can potentially be used to predict the situations which can arise from a current state and prepare for associated situations. The prepared and anticipation can further reduce the variability of the durations and minimize the overall off times. The networks provided here as well show us how different groups of events have associations between each other which can provide information on the error reporting to represent a class of codes as a different label.