POLITECNICO DI TORINO

Master's degree course in Communications and Computer Network Engineering

Master's Degree Thesis

Unsupervised Machine Learning for Mining Alarm Logs of a Large Telecommunication Network

Supervisors:
Prof. Marco Mellia

Correlatori:
Prof. Maurizio M. Munafo
Dr. Luca Vassio

Candidate:
Golnazsadat Zargarian
matricola: S224955

Academic Year 2017 – 2018

Contribution

This thesis is in collabaration with TIM (formely Telecom Italia) under a research contract with Polythecninc university of Turin.

Summary

Communication technologies such as WiFi or cellular networks are designed to be extremely efficient. While this is the original purpose, having fault tolerance and reliable networks seems to be the ultimate goal. One step towards reaching reliability in systems is anomaly detection and identifying sources of failures.

Alarm log plays a crucial role in the network area since it has been mostly known as a valuable source of information for anomaly detection and reporting network failures. An alarm is a message produced by a network element, typically when a problem happens unexpectedly. Given the major issues of syslog processing, failures in a telecommunication network are reported to management centers in the form of alarms but unfortunately this report is a limited view from the network element which seem to add unreliability.

Furthermore, one fault can result in a number of different alarms from several network elements. The information contents of alarm messages are very diverse. Some alarms could issue the problems in logical concepts, such as virtual paths, whereas others are concerned with physical devices, e.g., power supplies or cable failures. Some alarms acknowledge a distinct failure, such as the incoming signal is missing, whereas some only report a high error rate without any sufficient information for the cause. All these different types of messages should be handled in alarm correlation systems.

Manual analysis of such logs is time-consuming and costly because they involve an extensive amount of data. On the other hand, the automatic detection of useful information can be also quite challenging. As a result, finding suitable methods to process these logs in a proper way is a well-established problem in the network analysis area.

In this perspective, it is possible to use approaches based on supervised classification systems, provided that we have access to labeled data that allows training of the system. However, the available data as will be discussed is not always sufficient and correctly labeled to allow a supervised classification approach. To solve this problem, one can alternatively utilize unsupervised approaches, in which data is used without labels, and then it is an adequate baseline for exploration techniques. With unsupervised machine learning techniques, we will mine data logs and thus provide meaningful information about possible causes and cascade effect in a network failure. Since we are using such methods, we will review the main steps of machine learning used in our thesis to build a predictive model. Data Gathering, preparation, cleaning and manipulation are parts of these steps. Then choosing an ML model is required which is then followed by evaluation and parameter tuning.

The available data is extracted from TIM Network Operations Center (NOC) which includes the list of whole alarms during specific months for different provinces of Italy. In the second chapter, we will discuss the characteristics of used data to recognize its behavior. As we will see, network devices produce thousands of alarms daily but only about ten percent was presented to the operator (They were reported on the same period of months mentioned previously). Each alarm has some features that indicate the time and geographical place in which the alarm was recorded together with extra information about the cause, severity and etc. To reduce the computational complexity, we will only focus on the most important features that are domain-driven by static rules.

Since the percentage of presented/reported alarms are small, it could imply what we will find as a rule may not be generated by the reported alarms and therefore the interpretation of such rules would rely too heavily on the network domain expert. Set of available data can be fully described and illustrated by using probability distribution function of alarm inter-arrival time of aggregations for features of interest such as province, region and etc.

Since most of our interesting features in the dataset are categorical, it is difficult to define a measure of distance for clustering algorithms so we will choose association rule mining on frequent items as an alternative approach. In order to use such methods, we will recall preliminaries of market basket analysis problem, in which the main objective is to extract actionable knowledge and co-occurrences from the vast features of transactional databases in order to gain competitive advantage. Throughout the third chapter we will address the required steps to produce frequent items, association rules and finding correlations.

The next two chapters will discuss unsupervised machine learning methodologies in which separated network devices and device types is investigated. The first step is to look for items that we are interested to study and then define the transactions for them. For using any pattern mining algorithm, we are required to transform the data from its frame format into transactions such that each row corresponds to a transaction whereas each column indicates an item. Defining these matrices require experiments since each method has different results and its own advantages. For choosing items we will focus on each network device to extract specific correlations. We then identify which devices were raising alarms at the same time bin more frequently. We will focus on Turin province to reduce complexity and study two datasets reported in two different month of May and September 2017.

We will then apply frequent pattern mining methods on this matrix to extract frequent items that are later used to find temporal and spatial co-occurrences. Some temporal correlations among power plants and events are evident. Nevertheless, finding the direct spatial correlation is harder to accomplish because we are not informed about the topological connectivity among plants.

We outline the most significant mutual rules which hold true with a high probability in two selected provinces located close to each other (Turin and Milan). We consider the significance of a rule with its measures of interestingness such as lift, support, and confidence. Visualization of these rules together with the knowledge from domain expert is another measure of importance.

We will show how observed frequent patterns help us to recognize possible future anomalies as situations appeared in the past and avoid them from happening again. Our work is to aid the experts in recalling and formulating correlation patterns in an efficient way. Given obtained rules derived from an alarm database, domain expert is able to verify whether the rules are useful or not. Some of the rules may reflect causal correlations and give new insights into the behaviour of the network elements whereas others may be irrelevant.

While most of the thesis is devoted to rules obtained from suitably defined transaction matrices, in the last chapter we broaden our scope of investigation to another changes that could be done in order to optimize this definition. As we will observe, parameterization is needed when searching for proper methods in order to find the required information from the data. In our approach, we apply this with different thresholds and data selections. As a result, the method reveals a set of selected informative rules. Then experts can learn quite a lot from the data and find the answer to questions such as: "What are the distributions of alarms types and their causes?", "What are the most common combinations of devices that generated alarms?", "Is there any correlation among the alarms coming from different sources?", and so on. By logic, this kind of information and knowledge about the network could be even more valuable than the rules found in the data because such information can relatively easily be interpreted.

we will conclude the feedback from TIM network maintenance team in Rome who confirmed that rules similar to what we found were already presented in their system. So our automatic rules are useful for their systems. Moreover, TIM will use the rules we extracted as an input of machine learning algorithms to "detect patterns". These rules are stored in the systems as a list of "situations" presented together with meta-data (location, resolution and etc).

Acknowledgements

Foremost, I would like to thank my advisor, Prof.Marco Mellia for his adept supervision, continuous support, patience, and productive ideas during this thesis. I was honored to have this great opportunity to work with his team that motivated me every day to improve in the research area and gave me a platform to express my own ideas.

My special thanks also goes to Prof.Maurizio Munafo and Luca Vassio for exceptional mentoring. Their feedbacks and solid pieces of advice have been invaluable at various stages of my thesis.

I would like to express my sincere gratitude to the research group in TIM Joint Open Lab in the polytechnic university of Turin and their operation/engineer colleagues in Rome who made this research possible and for constantly helping me with their suggestions to explore new sides of the project.

Finally, I can not put into the words of how grateful I am to my parents for everything, the true meaning of heroes in my life, as well as my brother for giving me strength, hope, and faith.

Contents

List of Tables

Chapter 1

Introduction

Communication technologies such as WiFi or cellular networks are designed to be extremely efficient. While this is the original purpose, having fault tolerance and reliable networks seems to be the ultimate goal. One step towards reaching reliability in systems is anomaly detection and identifying sources of failures. Various types of telecommunication networks for mobile devices such as 2G/GSM, 3G/UMTS, and 4G/LTE have similar architectures and their engineering is usually evolving into single-point failures which make them more vulnerable by design. As depicted in Figure 1.1, the key components of the mentioned technologies are Mobile Switching Center (MSC), Base Station Controller (BSC), Base Transceiver Station (BTS).

These elements function hierarchically with MSC as their root. Therefore, failure of MSC leads to the failure of the entire network. Moreover, a base station controller meditates the communication between base transceiver station and the mobile switching center. As a result, if BSC reports an error, its existing controlled BTSs will also report an error. Figure 1.2 shows possible situations of failures in a cellular network and their effect on the rest of network.

As we observed, It is important to accurately detect abnormalities in order to avoid generating a high number of false positives in the network. It is also vital to avoid false negatives that would contribute to a network malfunction.

System alarm log has an extremely important part to achieve this aim because it reports network failures. Looking in a telecommunication alarm log, the goal is to find relationships among alarms. These can then be used in the on-line analysis of the incoming alarm stream, for instance to better explain the problems that cause alarms, suppress redundant alarms, predict a set of critical situations and severe faults, and finally suggest actions by learning from experiences of the past. Nevertheless, it is not an easy task to analyze syslog manually, since it can be often time consuming and costly. For this reason, usage of automatic techniques such as machine learning algorithms is fundamental.

In this perspective, it is possible to use approaches based on supervised classification systems, provided that we have access to labeled data that allows training of the system. However, the available data as will be discussed in 2.2 is not always sufficient and correctly

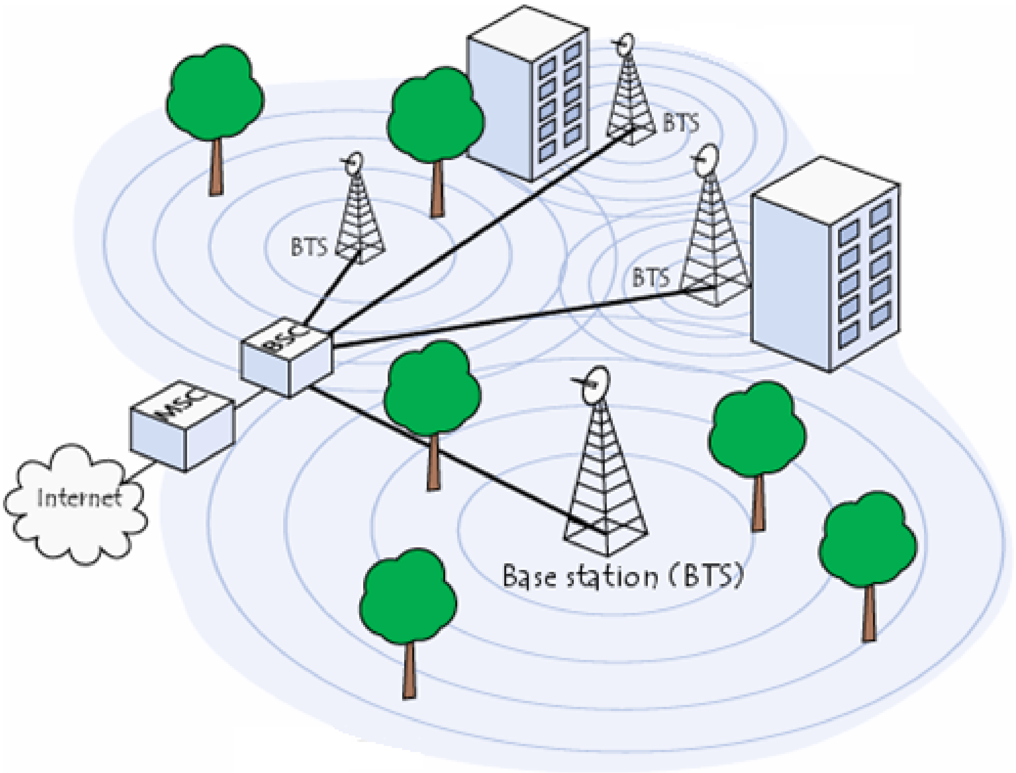


Figure 1.1: A sample schematic of Cellular Network [1]

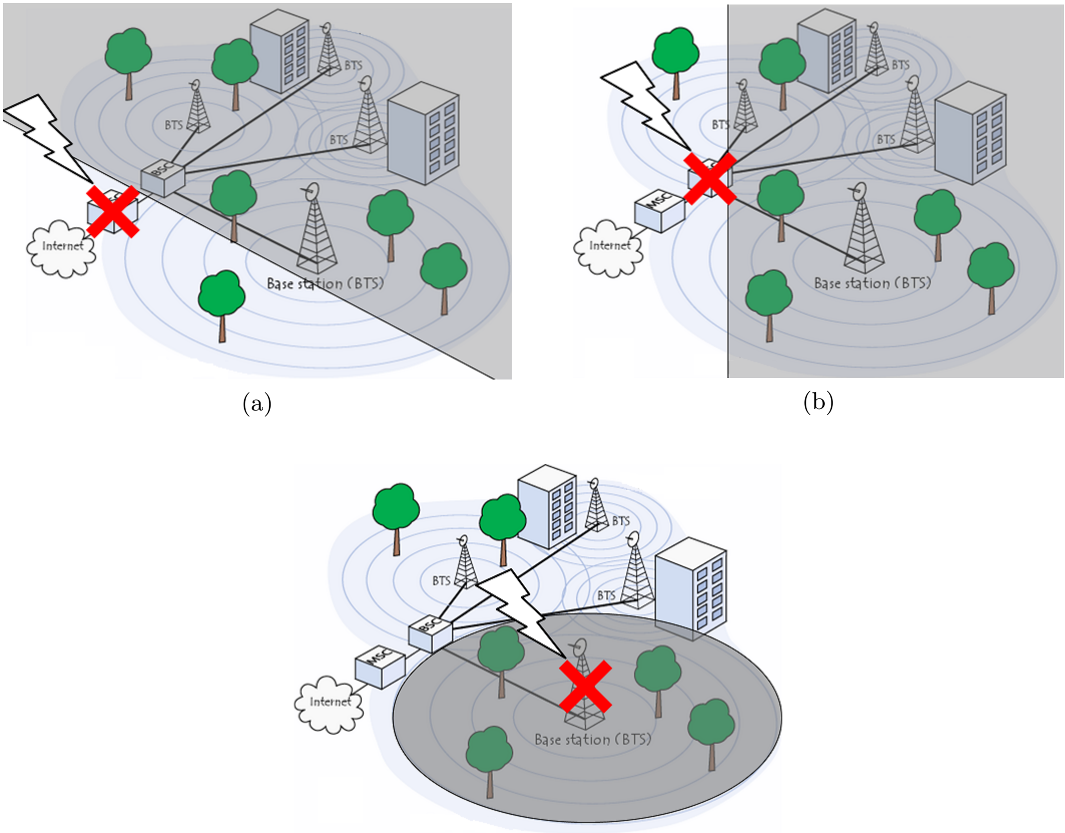
'labeled' to allow a supervised classification approach. To solve this problem, one can alternatively utilize unsupervised approaches, in which data is used without labels, and then it is an adequate baseline for exploration techniques.

Our research will focus on unsupervised machine learning and analyzes the available data to highlight characteristics, correlations, dependencies and any anomalies that reflect a change in the system behaviour even in the absence of labeled data. We will use a method to mine big datasets of alarms by using a matrix of transactions and items. Afterwards, we show patterns produced by association rules among these items. The idea of using pattern mining as a particular apparatus is due to using unsupervised approaches to find frequent patterns which can eventually lead to synthesize a **meta-alarm** or to recognize frequent critical situations.

Moreover, the data we have is in fact characterized by many categorical features, so it is well suited to this approach, whereas use of clustering algorithms would be ineffective due to the major issue of defining **distances** between features for the categorical note.

We undertake the study by introducing a well-documented developed modeling technique called **market basket analysis** (see 3.1) to find correlations in a data set (e.g., purchases in store receipts). For example one question in market basket analysis is:

- Given all the receipts, what are the groups of items that are bought together with a



(a)

(b)

(c)

Figure 1.2: Possible failures of cellular network: (a) - MSC failure, (b) - BSC failure, (c) - BTS failure

high probability?

Turning now to our case, the question is about defining the transactions(receipts) as the group of alarms occurred together in the same time interval, and geographical area. Evidently, here the question is:

- Given the transactions seen, what are the elements that appear together frequently?

In the following, we will explore the term **syslog** more widely and explain its processing method. Next, we will have a look on previously related works.

1.1 A Brief Journey into Syslog Processing

1.1.1 What Is Syslog?

Syslogs are messages sent by networking devices such as routers, switches, wifi access points and etc widely used to help monitor a network and issue its anomalies. Syslog messages have different types which are mostly dependent on device kinds. Even so, by using time stamp and actual log message and severity of the field, we can identify when, where and to what degree that specific log was important to be stored.

1.1.2 Challenges of Syslog Processing

While syslogs offer various utilities tailored to our problem in the area of network mining, its processing involves a lot of issues. They are often free-form with little structure[1] and different formats which may not be human readable and high-level [7]. In general, important logs based on the system failures are hidden in the majority of those that are reporting the daily routine processes. This means, not every syslog is an indication of a failure that could impact on the performance of the network.

The message relationship models are forced to be constantly updated to keep up with network changes. This introduces a huge amount of data related to syslog which makes the procedure of analyzing much more complex. To analyze log files separately and manually link each related log afterwards is seldom practical and highly time-consuming.[10]

In the interest of simplicity, we talk about alarms in the next chapters of this thesis. Nonetheless, the previous preliminaries should be understood to cover understanding of further definitions and findings.

1.1.3 Alarms

Whenever, certain patterns of syslogs are observed, alarms will be raised. An alarm is a message produced by a network element, typically when a problem happens unexpectedly. Given the major issues of syslog processing, failures in a telecommunication network are reported to management centers in the form of alarms but unfortunately this report is a limited view from the network element which seem to add unreliability. Furthermore, one fault can result in a number of different alarms from several network elements. Table 1.1 shows an example format of alarms with selected fields in our available data set. Each row in the table has features such as NeID which shows ID of the network device that has generated the alarm along with timestamps, type, severity and cause of that alarm. Although each row in the data set contains over 100 fields/columns, in order to reduce complexity of working with a large data set, we can only consider an alarm as a multiple fields of the most important features selected by experiments and domain expert.

[1] Structured data is significantly known for being easily parseable

1.1.4 What Do Alarms Report?

The time field reported in each alarm is recorded by the sender, typically at a granularity of one second. Sender of an alarm can be identified at the level of a network element [15]. Description field reports the problem accompanied with information that is available to the sender.

The information contents of alarm messages are very diverse. Some alarms could issue the problems in logical concepts, such as virtual paths, whereas others are concerned with physical devices, e.g., power supplies or cable failures. Some alarms acknowledge a distinct failure, such as the incoming signal is missing, whereas some only report a high error rate without any sufficient information for the cause. All these different types of messages should be handled in alarm correlation systems.

1.1.5 Alarm Correlation Technique

Alarm correlation is a central technique for processing the flow of alarms arriving in a management center into a smaller but more useful set of reports by looking at the active alarms within a time window, and interpreting them as a group.

1.2 Research Questions and Challenges

As stated before, this research was conducted in order to effectively filter the redundant alarms, identify the faults, and suggest corrective actions. We try to reduce the workload of network managers by processing the large alarm data set. Building a correlation model, however, suffers from the complexity and diversity of network elements and the large variation in the patterns of alarm occurrences. In this thesis, we present methods for semi-automatic discovery of patterns in data base.

1.3 Related Literature

One of the comparative works on pattern mining is argued in [23], [24], [25]. These studies are based on the first knowledge discovery system ever built known as Telecommunication Alarm Sequence Analyzer or TASA. This system is aimed at analyzing alarm set collected from GSM networks by using data mining approaches. These analyzes are based on finding alarm correlations and filtering the related ones and predicting a combination of forthcoming malfunctions. TASA discovers patterns in form of associations and episode rules. This application is one of the starting points in knowledge discovery and analyzing network log data with frequent pattern mining. It finds a large rule collection and gives the user criteria for including or excluding certain rules.

Despite the effectiveness of TASA, this software suffers from a downside in providing an overwhelming amount of rules given from the alarms that occur often together in a time period. These rules could point out to associations of items such A, B and C and present a subset of every possible combination of these items which clearly creates large data sets of rules. An additional problem is when episode rules are mined, alarms from simultaneous but with separate cause of faults appear correlated to each other. This correlation is statistically true because they do occur during the same time period. However, due to network structure, the faults causing these alarms probably have nothing to do with each other.

Furthermore, researches from [25], [26] and [27], that were studied earlier, deal with episode and association rules in a form of small pieces of local correlations in the network. These are semi-automatic approaches to acquire meaningful information from alarms in order to collect the required knowledge for knowledge-based systems like alarm correlators. Unfortunately, when there are too many pieces of uncleared relations, the big picture of the network remains vague or corrupted.

literature in [3] and [6] drew our attention to sequential pattern mining to better discover the failures and temporal correlations. However, we discovered later from the visualizations of our data set that categories of itemsets do not follow a sequential pattern and is random. For instance the cause of generated alarms from associated devices are not always in the same order.

A review on the work of [5], reveals that Varandi investigated the issue of failure detection by using a clustering technique. Primely, the proposed model constructs clusters by grouping event logs on the basis of their message characters and afterwards detects failures by tracking anomalous events which do not belong to any existing cluster. The shortcomings of this method is clear because it is not applicable for categorical features and defining a measure of distance for each cluster would not be easily possible.

Several studies for instance [15] have been performed on historical event log mining, periodic patterns/ similarities and their visualization. A recent review on this area including those suggested by TIM have been dedicated to analysis of structured logs such as syslogs. Literature from [8] and [9] deals with root cause analysis in a graph that models failures in networks. Work in [7] focuses on analysis of syslogs with a much more structured data format.

1.4 Organization of the Thesis

The rest of this thesis is organized as follows:

Chapter 2 will list all the main features in the dataset. It will then discuss characteristics of the used data and plot distribution functions to recognize its behaviour.

Chapter 3 will be devoted to basic definitions and preliminaries of market basket analysis, one of the main methods in practice, used by large retailers to uncover meaningful associations among customers purchase data as well as pattern mining concepts and association rules. It later describes an overview of argued methodologies for finding a matrix of transaction and itemsets which is well-defined for our problem. Throughout this chapter, we will address the required steps to produce frequent items, association rules and finding correlations.

Chapter 4 will study an unsupervised machine learning methodology in which separated network devices will be investigated. This approach extracts the specific temporal correlations among network devices.

Chapter 5 will study an unsupervised machine learning methodology in which the types of each network device will be investigated. Unlike the previous chapter, this approach extracts more general correlations among network devices by focusing on their types. It also tries to investigate geographical correlations alongside temporal associations.

Chapter 6 will conclude the research and explains the further possible studies.

NeId	FirstOccurrence	Original Severity	AlarmType	EM	Probable–Cause	Latitude	Longitude
UBTSMI469	2017-05-01T00:00:00	Major	equipmentAlarm	JERAMI001	equipmentMalfunction	45.47021667	8.868333333
UBTSMI118	2017-05-01T00:00:00	Warning	qualityOfServiceAlarm	JERAMI001	performanceDegraded	45.42155833	9.253363889
UBTSMI144	2017-05-01T00:03:00	Critical	processingErrorAlarm	JERAMI001	softwareError	45.45599722	9.195111111
UBTSMI6A0	2017-05-01T00:11:00	Major	equipmentAlarm	JERAMI001	equipmentMalfunction	45.54019722	9.101825

Table 1.1: A sample of actual alarms in the data set with selected fields in table format

Chapter 2

Characterization of Alarms

2.1 Source of Data

In this thesis, several data sets from TIM network operation center have been used for verifying the derivative rules and patterns. Datasets have been extracted from TIM Network Operations Center (NOC) and Service Operations Center (SOC)(see Figure 2.1). NOC controls the state of the telecommunications network and manages anomalous situations that must be corrected to ensure accurate operations. SOC, similar to NOC, monitors the operation of services offered by the company. Data collected is provided by TIM which includes a list of alarms during months of May and September of 2017. Network devices produce thousands of alarms daily but only about ten percent was presented to the operator (They were reported on the same period of months mentioned previously). Since the percentage of presented/reported alarms are small, it could imply what we will find as a rule may not be generated by the reported alarms and therefore the interpretation of such rules would rely too heavily on the network domain expert.

2.2 Main Features

As mentioned earlier, each alarm has some features that indicate the time and geographical place in which the alarm was recorded together with extra information about the cause, severity and etc. To reduce the computational complexity, we are only focusing on the most important features that are domain-driven by static rules. In the next part, we will outline the definitions of these features. Here are the main features of alarms:

1. **Central**: Identifies the location of antenna center which has generated the alarm.

2. **NeID**: It identifies a 9-letter id of the network device which has sent the alarm and specifies technology, equipment, province and plant id of the device. Technology is shown by the first letter and it is according the table 2.1. The equipment could be a base station controller, radio network controller, base transceiver station or etc.

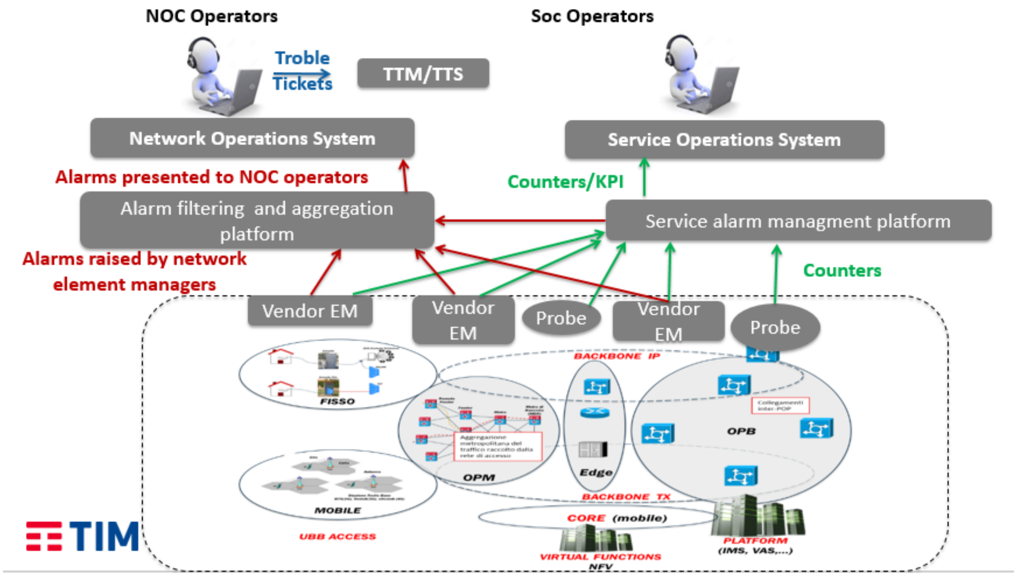

NOC Operators
Soc Operators
Troble Tickets
TTM/TTS
Network Operations System
Service Operations System
Alarms presented to NOC operators
Counters/KPI
Alarm filtering and aggregation platform
Service alarm managment platform
Alarms raised by network element managers
Counters
Vendor EM
Vendor EM
Probe
Vendor EM
Probe
BACKBONE IP
FISSO
Collegamenti inter-POP
Aggregazione metropolitana del traffico raccolto dalla rete di accesso
OPM
Edge
OPB
MOBILE
BACKBONE TX
CORE (mobile)
UBB ACCESS
VIRTUAL FUNCTIONS
NFV
PLATFORM
(IMS, VAS,...)
TIM

Figure 2.1: Operations Center Diagram

Different devices might belong to the same center, for instance, let us consider NeID of two below devices:

- GBSCTO050
- GBSCTO052

These two devices are base station controllers which are using GSM technology and both belong to the same center located in Turin province. Detailed characteristics of the first device is reported in Table 2.2.

Technology (1st Letter)	
G	GSM
U	UMTS
9	UMTS 900
1	LTE 1800
2	LTE 2600
8	LTE 800

Table 2.1: Possible abbreviations for technologies used in network ID

1st Letter		2nd–3rd –4th Letters	5th –6th Letters		7th –8th –9th Letters
G	GSM	BSC	TO	Torino	050

Table 2.2: Example: characteristics of the device GBSCTO050

3. **Std ProbableCause No**: Probable cause of the alarm which has three cases:

  - The alarm is from Element Manager[1].
  - The alarm is result of an ABAM rule, which creates a parent alarm that has alarms from the vendors as children. It is reliable but often announced as "*indeterminate*". This occurs when Manager is equal to IMPACT and Agent is CIC.
  - Alarm is from SOC manager. In this case, the alarm is generated by rules on network counters, not by machines(we do not consider it).

4. **OriginalNeId**:It is composed of network type and Element Manager id.
5. **Last Occurrence**: Last instant in ABAM encountered alarm occurred. The last notification by the Element Manager is highlighted in the *NeLastAlarm* field.
6. **First Occurrence**: The first instant ABAM encountered this alarm. The first notification by the Element Manager is highlighted in the *NeStartAlarm* field.
7. **Alarm Count**: Number of alarms between NeStartAlarm and NeLastAlarm. There is no information about timing distribution of alarms in that period. For simplicity, we consider these simple alarms uniformly distributed over time in the following analyses.
8. **Original Severity**: It is assigned directly by the Network Element during and it never changes. There are four different types of severity including: Major, Minor, Critical and Warning.
9. **Summary**: This field describes the problem generated by specific vendors. This feature is an expanded explanation of probable cause, therefore their statistics are similar.
10. **Alarm Type**: There are five different types of alarms: equipment, communication, environmental, processing error and quality of service.
11. **Duration ABAM**: Minutes between the first instant in which the alarm occurred in ABAM (FirstOccurrence) and the last instant (LastOccurrence).
12. **Specific Problem**: This field describes the cause of generated alarm in specific.

2.3 General Statistics of the Collected Data

In the first attempt, we try to identify the macroscopic characteristics of the data that could be used as a feature. Since there are quite different types of vendors and systems in the data set, the available data is very heterogeneous and granular. The temporal granularity of the alarms, which are aggregated at the precision of minutes, creates a large number of contemporary events. The progressive filtering of data by only focusing on

[1] OSS system from which the alarm is generated

specific subsets(same region or geographical area) can significantly improve the analysis. In the next sections, we will investigate the distribution of alarms by using plots such as probability distribution functions and bar graphs.

2.3.1 Inter Arrival Time

Set of available data can be fully described and illustrated by using probability distribution function of alarm inter-arrival time of aggregations for features of interest such as province, region and etc. Figure 2.2 indicates probability distribution function of raised alarms in the whole Italy based on their inter-arrival time in less than one hour, which are categorized by province. Based on the plot, data set is very granular and heterogeneous and more than 60 percent of events reported in Milan and Bologna are "*simultaneous*". Even by analyzing different regions, the heterogeneous behavior is observable as shown in Figure 2.3. The plot represents, probability distribution function of raised alarms in the whole of Italy based on their inter-arrival time in less than one hour, which are categorized by location.

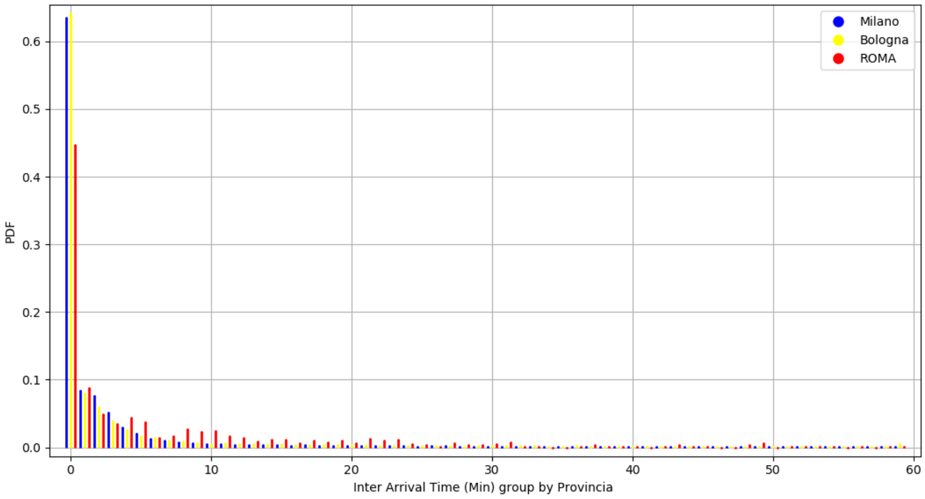


Figure 2.2: Distribution of raised alarms based on the inter-arrival time grouped by province

2.3.2 Features Value Distributions

It is also useful to display characteristics of data with the values. Plot 2.4 gives a general view for frequencies of the raised alarms categorized by alarm types in the north of Italy in May. Most of the alarms are basically from communications, quality of service. Figure 2.5 reports frequencies of the raised alarms categorized by original severity in the north of Italy in May. Given the plot, the most common severity of alarms is major.

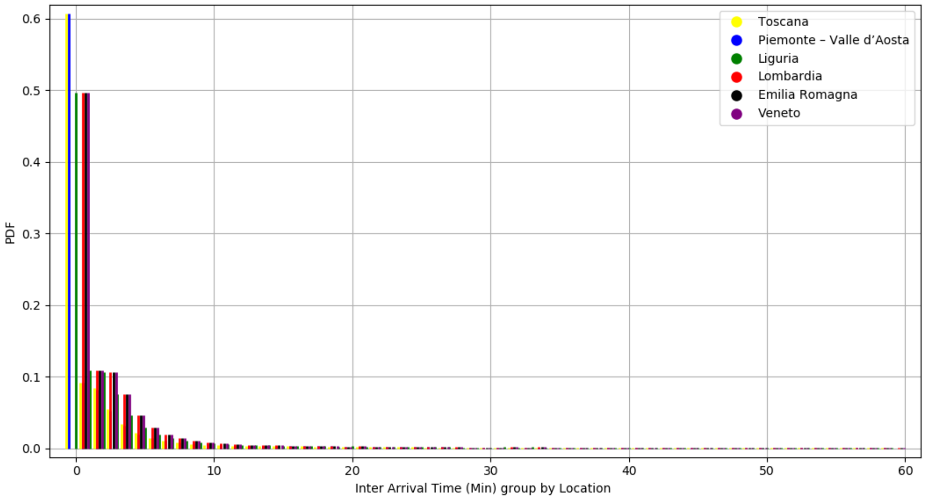


Figure 2.3: Distribution of raised alarms based on the inter-arrival time grouped by location

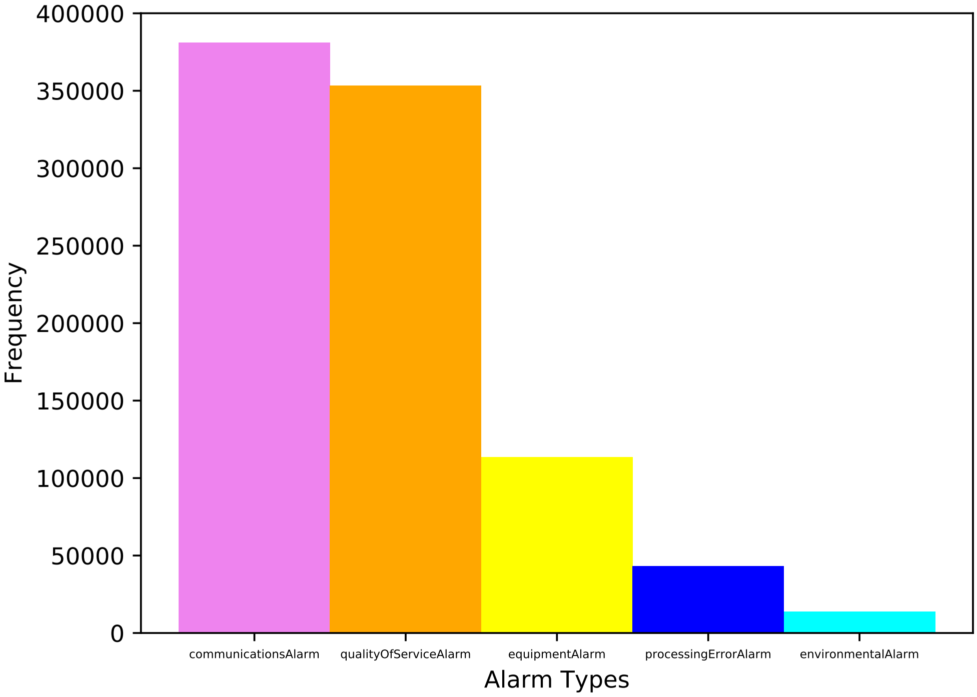


Figure 2.4: Bar Graph: Raised alarms categorized by alarm types in north of Italy in May

Figure 2.6 depicts frequencies of the raised alarms categorized by probable cause feature in the north of Italy in May. The frequency is reported on a logarithmic scale and has a

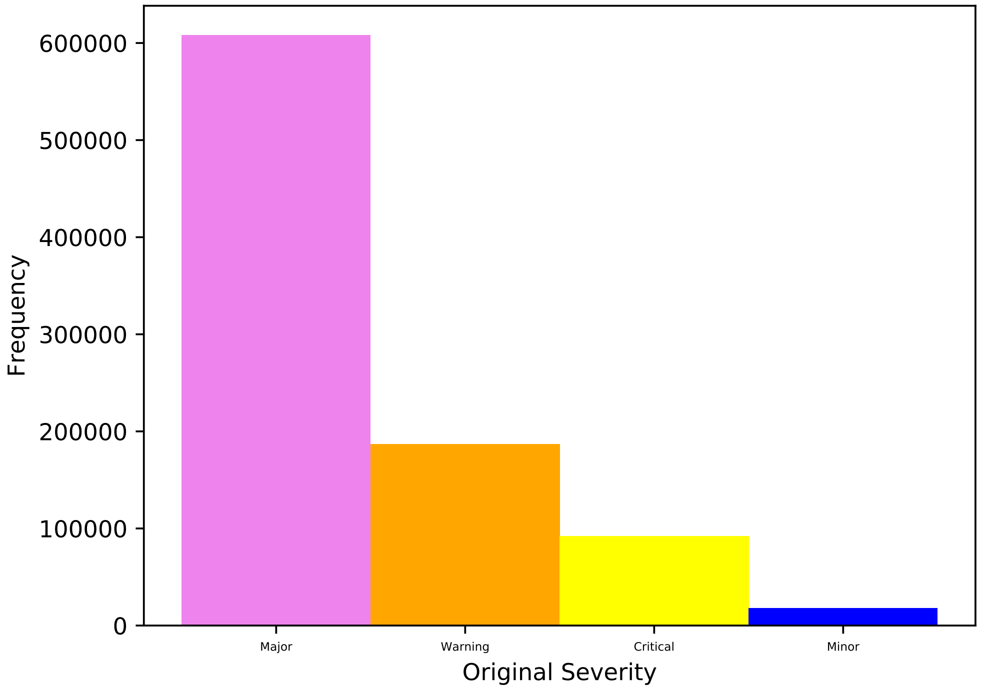


Figure 2.5: Bar Graph: Raised alarms categorized by original severity in north of Italy in May

large variety of values. Indeterminate and unavailable are respectively the most popular types of probable cause.

Table 2.3 shows 10 most probable causes reported in north of Italy in May. Table 2.4 adds the most popular specific reason for these probable causes along with recorded frequencies.

Probable–Cause	**Frequency**
Indeterminate	442328
Unavailable	114390
RemoteNodeTransmissionError	95650
aIS	91368
EquipmentMalfunction	57105
UnderlyingResourceUnavailable	29213
PerformanceDegraded	22490
CallEstablishmentError	17693
M3100–synchronizationSourceMismatch	5855
SoftwareError	4487

Table 2.3: Frequencies of probable cause for raised alarms in north of Italy in May

The dataset in September is for the most part similar to May. As a result, the dominant

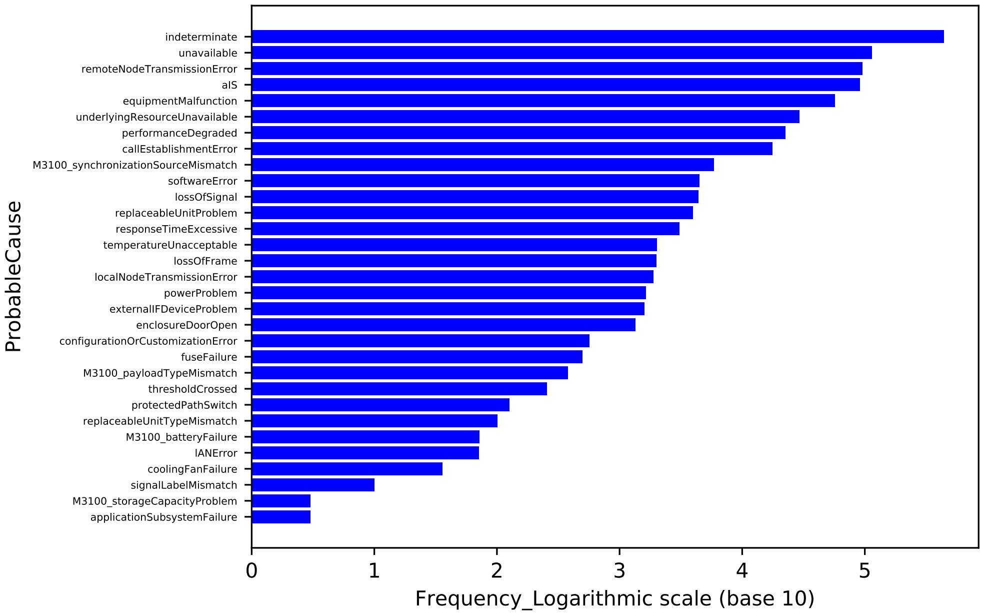


Figure 2.6: Bar Graph: Raised alarms categorized by probable cause in north of Italy calculated in a logarithmic scale with base 10

Probable–Cause	**Specific Problem**	**Frequency**
indeterminate	BTS with No Transactions	164627
unavailable	Heartbeat Failure	26071
remoteNodeTransmissionError	IMA Link Reception Unusable at Far End	95623
aIS	PDH Alarm Indication Signal	91368
equipmentMalfunction	DigitalCable–CableFailure	10179
underlyingResourceUnavailable	Service Unavailable	29207
performanceDegraded	Carrier–RxDiversityLost	22220
callEstablishmentError	UtranCell –NbapMessageFailure	10945
M3100–synchronizationSourceMismatch	Synch Reference Path HW Fault	4414

Table 2.4: Frequencies of most popular specific problems for most common probable causes in north of Italy in May

factors in the distribution of alarm type and severity which were illustrated previously in the bar graphs are held true as well in September. Table 2.5 shows 10 most probable causes reported in the whole Italy in September. Table 2.6 adds the most popular specific reason for these probable causes along with recorded frequencies.

2.3.3 Temporal Evolution

In the following, statistical significance is analyzed by using temporal evolution for alarms generated by devices in different areas. Particularly, we choose to focus on single provinces with a remarkable geographical size such as Turin, which is characterized by different

Probable–Cause	**Frequency**
Indeterminate	870251
Link Failure	383132
Unavailable	272660
RemoteNodeTransmissionError	144259
aIS	125352
Out Of Service	92753
Equipment Malfunction	63469
Invalid Message Received	58601
UnderlyingResourceUnavailable	48387
Unspecified Reason	45183

Table 2.5: Frequencies of probable cause for raised alarms in whole of Italy in September

Probable–Cause	**Specific Problem**	**Frequency**
indeterminate	BTS with No Transactions	271047
Link Failure	SCTP Link Fault	361693
unavailable	PLMN Service Unavailable	97856
remoteNodeTransmissionError	IMA Link Reception Unusable at Far End	107094
aIS	PDH Alarm Indication Signal	125352
Out Of Service	UMTS Cell Unavailable	57104
equipmentMalfunction	DigitalCable–CableFailure	10413
Invalid Message Received	Inter-System Communication Failure	42769
underlyingResourceUnavailable	Service Unavailable	48387
Unspecified Reason	NE Is Disconnected	22487

Table 2.6: Frequencies of most popular specific problems for most common probable causes in whole of Italy in September

operators. However, the large number of alarms still makes the system very verbose and chatty. As shown in Figure 2.7, the plot represents a temporal evolution comparison of raised and reported alarms generated by different centers(almost 700) located in Turin province in May. On the x-axis, the start time of each alarm registered at ABAM is reported. On the y-axis, there is a progressive identifier for central stations of the province, assigned arbitrarily according to the appearance. This figure indicates the involved each center by a unique identifier. Temporal correlations among the devices in the same center are shown by vertical lines, whereas there are no spatial correlations observed. Moreover, vertical lines are the sign of simultaneous events while horizontal lines stand for continuous events over time. As can be seen, exceptional activity has been recorded on 2$^{\text{nd}}$ and 11$^{\text{th}}$ of May.

Figure 2.8 represents temporal evolution of raised alarms generated by different centers(almost 550) located in Turin province in September.

As observed, in this month there are fewer devices involved and raised alarms are less in comparison to May. Despite this fact, heterogeneity of data is still remarkable.

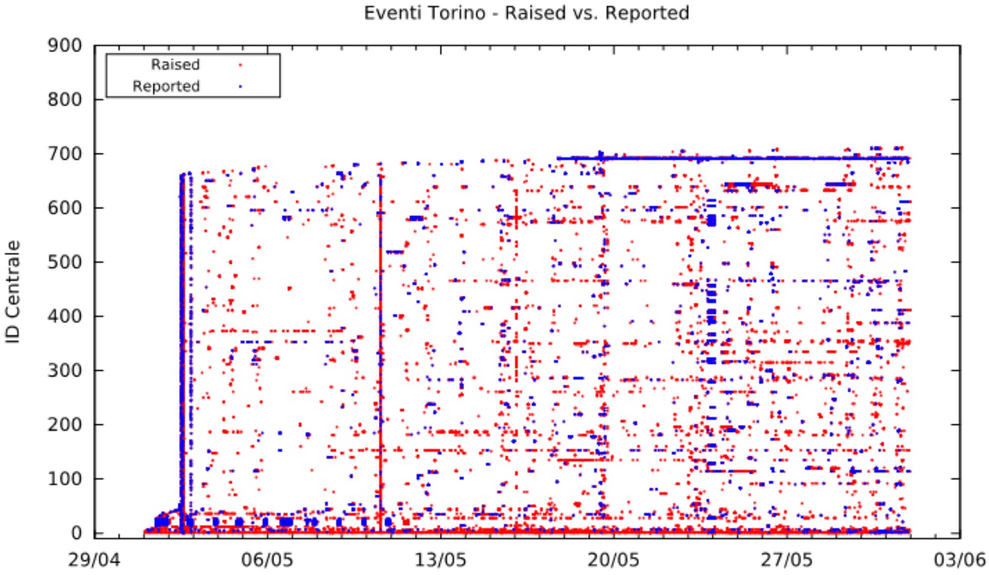


Figure 2.7: Temporal evolution of raised and reported alarms for Turin province in May

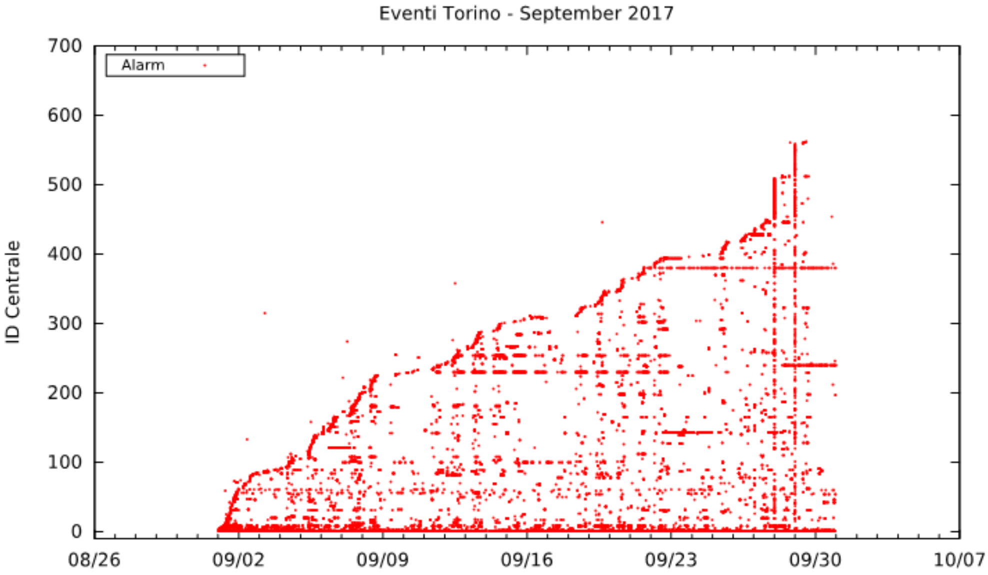


Figure 2.8: Temporal evolution of raised alarms for Turin province in September

It is also useful to perform the analysis of data by different features such as severity and alarm type. Figure 2.9 highlights the temporal evolution of events aggregated by the severity of alarms for Turin province in May. It is obvious from the plot, that major is the dominant factor in original severity.

Figure 2.10 highlights the temporal evolution of events aggregated by type of alarms for

Turin province in May. As we can observe, there are no dominant factors for alarm types in Turin province. This could be due to the major event which has occurred in 2nd of May that results in the growth of communications alarm frequency. On the other hand, there seem to be some specific center identifiers which are always generating equipment alarms and increases heterogeneity.

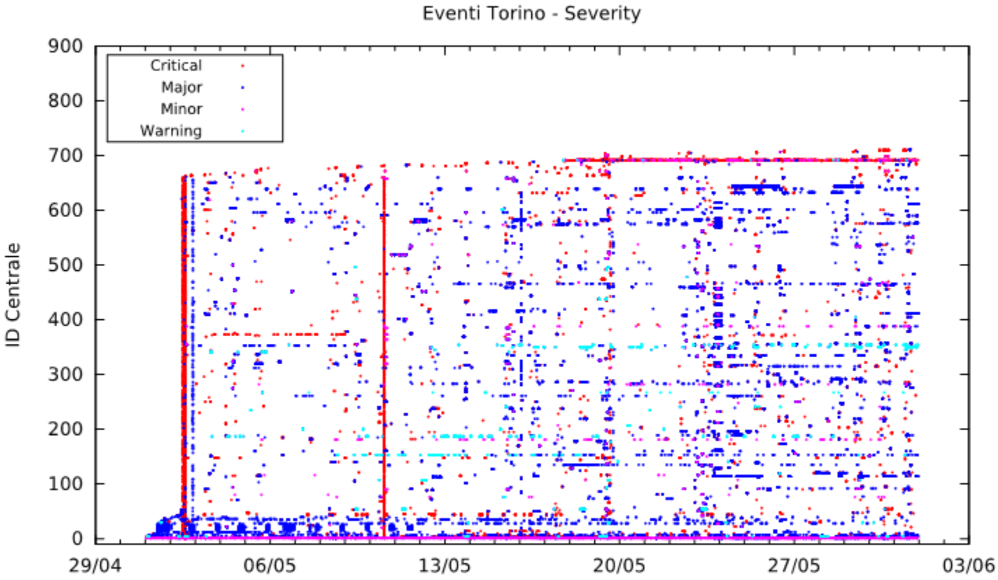


Figure 2.9: Temporal evolution of events aggregated by severity of alarms for Turin province in May

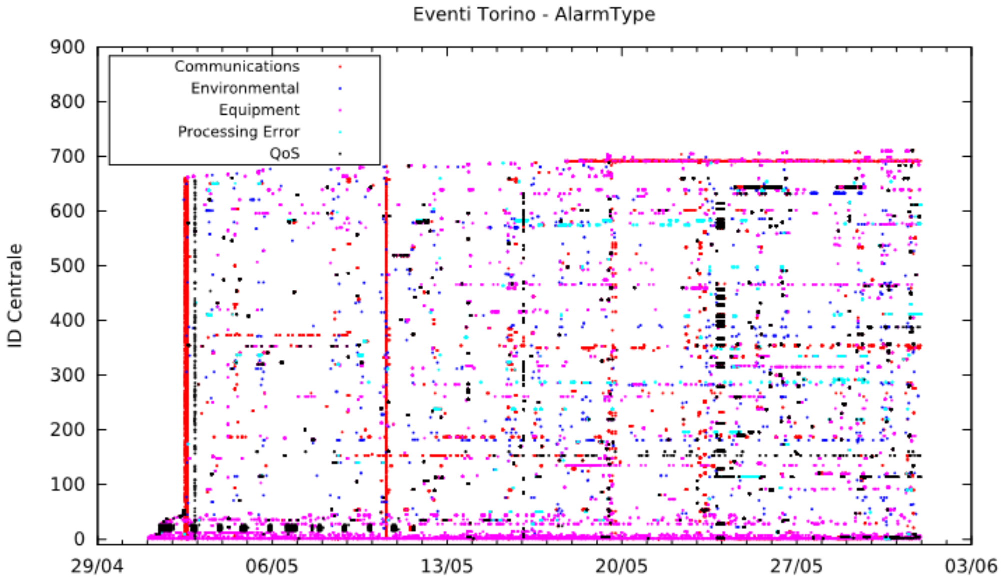


Figure 2.10: Temporal evolution of events aggregated by type of alarms for Turin province in May

Chapter 3

Steps to Pattern Discovery

In the previous chapter, we outlined characteristics of the available data. We indicated the most important features that were domain driven in the data set and showed an overview of its heterogeneity. Additional studies to understand more completely the key features are required and will be given in this chapter.

In the following, we will analyze the data by means of *Market Basket Analysis*[2], one of the key techniques in practice used by large retailers to uncover meaningful associations among customers purchase data. Within the framework of this criteria, we will then develop different definitions for finding frequent items and afterward deploy association rules on these items.

3.1 A Gentle Introduction on Market Basket Analysis

There has been numerous studies to investigate the challenging problem of market basket analysis, in which the main objective is to extract actionable knowledge and co-occurrences from the vast features of transactional databases in order to gain competitive advantage. In the cutting edge paper [2], authors introduced a methodology for mining association rules and then broaden their algorithm with the rule discovery in AI area. The particular introduction for basket data type transactions cited here, served as essential data for our studies.

To formulate the problem, assume a standard retail store sells a large set of products P. We define each transaction as below:

Definition 3.1. [**Transaction**] A transaction $p \subseteq P$ is the set of products an individual customer buys in a single trip to the store. Transaction database T = p is the set of all transactions the store has processed within a given time period(see [22] and [16]).

Example 3.1. Let us illustrate the concept by a basic sample of such data known as **market basket transactions** which is depicted at table 3.1. In this data, each row shows a transaction of items bought together by customers, identified by a unique ID.

TID	Items
1	Bread, Milk
2	Bread, Pizza, Beer, Eggs
3	Milk, Pizza, Beer, Cola
4	Bread, Milk, Beer, Eggs, Pizza
5	Bread, Milk, Pizza, Cola

Table 3.1: A sample of market basket transactions

Hidden associations in the large data set can be extracted by **association analysis** to reveal interesting relationships among items. In our example, the following rule can be discovered from table 3.1:

$$\{\text{Pizza}\} \rightarrow \{\text{Beer}\}$$

This rule implies there is a strong relationship between pizza and beer and it is likely the customers who buy pizza also buy beer.

Binary representation is one of the terminologies used when discovering associations. We illustrate each market basket data in a binary format such as detailed matrix in table 3.2. Therefore, we indicate each transaction T_i as a sparse binary vector, or as a set of discrete values showing identifiers of binary attributes that are instantiated to the value of 1. The binary value is equal to 1 if item is presented in the transaction and 0 if otherwise. This representation is used since it is very simple to understand but it ignores some certain details such as the frequency of items bought or their quantitative value(in above example the value is the price of each item).

TID	Bread	Milk	Beer	Eggs	Pizza	Cola
1	1	1	0	0	0	0
2	1	0	1	1	1	0
3	0	1	1	0	1	1
4	1	1	1	1	1	0
5	1	1	0	0	1	1

Table 3.2: Binary representation of market basket data in example 3.1

In this section we introduce definition of an itemset and one of its important properties known as support.

Definition 3.2. [**Itemset**] Let $I = \{i_1, i_2,...,i_d\}$ be the set of all items in a market basket data and $T = \{T_1, T_2,...,T_N\}$ be the set of all transactions. Each transaction t_i contains a subset of items chosen from *I*. In association analysis, a collection of zero or more items is called an itemset. In case an itemset contains k items then it is termed as a k-itemset. By this definition, a null itemset contains no elements. A transaction t_j is said to contain an itemset X if X is a subset of t_j.

Definition 3.3. [**Support**] A support value is an important propert of an itemset which indicates the fraction of transactions containing that particular itemset.

Observation 3.1. In the example 3.1, Milk, Pizza, Beer, Cola is 4-itemset. In addition, the second transaction in table 3.1 contains itemset of {Pizza, Beer} but it does not contain the itemset of {Pizza, Cola}. The support count for {Bread, Pizza, Beer} is equal to 2 because there are only two transactions that contain all three items together.

We use definition 3.1 to represent an important component of finding patters, *the frequent itemsets*. As we will discuss in the following sections, the frequent itemsets are mined from the market basket database. The computational cost for this process is often more than the rule generation itself. For this reason, efficient algorithms, for producing frequent itemsets, such as Fp-Growth3.4.2 and Apriori3.4.1 are applied.

3.2 Frequent Pattern Mining

Frequent pattern mining is generally described by market basket analysis, a typical data mining task for which it is well-documented. As we observed in 3.1, market basket analysis attempts to identify associations, or patterns, among the majority of items that have been chosen by a particular shopper and placed in their market basket, be it real or virtual, and assigns measures for comparison.

Mining frequent itemsets to extract common patters is one the backbones of research in data mining[17] area. Pattern mining which is a generalization of market basket analysis, sets the stage to work on unordered sets of simple objects (e.g., strings) and to find common itemsets, across multiple transactions and producing subsets of items that occur together more often in transactions on a database.

Please note that any attribute, or combination of attributes could be predicted in association. As association does not require pre labeling and it is a form of unsupervised learning which fits perfectly into our solution.

Now that we are familiar with the concept, let us state some essential definitions for a given items in the transactional data base as following:

Definition 3.4. [**Frequent Itemset**] An itemset is frequent if its support is greater than or equal to minimum support. Itemsets with a number of items smaller than minimum length could be discarded.

Definition 3.5. [**Closed Itemset**] For a given support value, the itemset presenting the highest number of items is known to be closed. The closed attribute implies that there is no other itemset made by more items with the same support.

Definition 3.6. [**Pattern**] Frequent closed itemsets for simplicity are called patterns.

The problem of pattern mining can be represented as below(see [4]):

Proposition 3.1. Given a database D with transactions T_1 ... T_N , determine all patterns P that are present in at least a fraction s of the transactions (The fraction s is referred to as the minimum support).

Looking for all itemsets is a #P-hard problem [18], but well-known algorithms can efficiently compute patterns. One proposed method is to calculate association rules which is tied together with frequent itemsets.

3.3 Association Rules

Association Rules are strongly linked to frequent patterns since they are count as "second-stage" outputs derived from these patterns. In data mining area, association rules are widely used to analyze retail basket or transaction data intended to identify frequent patterns, associations, correlations and rules which are discovered in the data set based on concepts obtained from measures of interestingness (see[20]).

In order to select interesting rules from the set of all possible rules, constraints on various measures of significance and interest are used. The best-known constraints are minimum thresholds on support and confidence. The full description of these measures can be explained as below:

Let X be an itemset, $X \Rightarrow Y$ an association rule and T a set of transactions of a given database.

Definition 3.7. [**Support**] is an indication of how frequently the itemset appears in the dataset. Support of X with respect to T is defined as the proportion of transactions t in the dataset which contains the itemset X.

$$\text{supp(X)} = \frac{|\{t \in T; X \subseteq t\}|}{|T|} \tag{3.1}$$

Definition 3.8. [**Confidence**] is an indication of how often the rule has been found to be true. The confidence value of a rule, $X \Rightarrow Y$, with respect to a set of transactions T, is the proportion of the transactions that contains X which also contains Y. Confidence is defined as:

$$\text{conf}(X \Rightarrow Y) = \frac{supp(X \cup Y)}{supp(X)} \tag{3.2}$$

Observation 3.2. Note that $supp(X \cup Y)$ means support of union of the items in X and Y. This is somewhat confusing since we normally think in terms of probabilities of events and not sets of items. We can rewrite $supp(X \cup Y)$ as the probability $P(E_X \cap E_Y)$, where E_X and E_Y are the events that a transaction contains itemset X and Y, respectively. Thus confidence can be interpreted as an estimate of the conditional probability $P(E_Y|E_X)$, the probability of finding the RHS of the rule in transactions under the condition that these transactions also contain the LHS.[12]

Definition 3.9. [**Lift**] Lift interprets the importance of a rule which can be defined as below:

$$\text{lift}(X \Rightarrow Y) = \frac{supp(X \cup Y)}{supp(X) \times supp(Y)} \tag{3.3}$$

If s(body) is support of the rule body and s(head) is support of the rule head, we can have an alternative definition for lift as below:

$$\text{lift} = \frac{\text{confidence}}{\text{expected confidence}} = \frac{\text{confidence}}{\frac{\text{s(body)}\times\text{s(head)}}{\text{s(body)}}} = \frac{\text{confidence}}{\text{s(head)}} \tag{3.4}$$

It is assumed that there is no statistic relation between the rule body and the rule head . This indicates that the occurrence of the rule body does not have an effect on probability for the occurrence of the rule head and vice versa [34].

If a rule has a lift equal or close to 1, we can imply the rule body and the rule head appear almost as often together as expected and that the occurrence of the rule body has almost no effect on the occurrence of the rule head. When two events are independent of each other, no rule can be drawn involving those two events.

If the lift is larger than 1, this lets us know the degree to which those two occurrences are dependent on one another, and makes those rules potentially useful for predicting the consequent in future data sets. The larger this degree of dependency gets, the more positive effect the rule body has on the occurrence of the rule head.

The value of lift is that it considers both the confidence of the rule and the overall data set.[13]

In addition to confidence, other measures of interestingness for rules have been proposed. Some popular measures are: All-confidence, Collective strength, Conviction, Leverage and Lift.

Consider the sets of items U and V. An association rule is then defined as below:

Definition 3.10. [Association Rules]

The rule $X \Rightarrow Y$ is considered an association rule with a minimum support s and a minimum confidence c, when the following two conditions hold true:

1. The set $X \cup Y$ is a frequent pattern.
2. The ratio of the support of $X \cup Y$ to that of U is at least c.

- The confidence of the rule is equal to the ratio of the support of $X \cup Y$ to that of the support of X.

Observation 3.3. A set of association rules $R(\mathbf{T}$, s, c) is defined by a transaction database $\mathbf{T}$, a minimum support parameter s and a minimum confidence parameter c.

3.3.1 Research Questions and Challenges

Users are often interested in finding association rules involving only some specified categories rather than all. Hence, constraints on measures such as support and confidence can be specified. This is in order to set the number of discovered rules to reasonable amount [21] and eliminate the uninteresting rules. Support is an important measure, since a rule which has a very low support may occur by chance rather than causality. The impact of confidence is noticeable as well. For a given rule $X \Rightarrow Y$, the higher the confidence, the more probable it is for Y to be present in the same transactions with X.

However, by setting a high value for measures of interestingness, we may lose some correlations. Challenging questions that will be arisen are:

- How to **efficiently** generate rules from frequent itemsets?
- Are all the strong association rules discovered are interesting enough to present to the user?

There are key issues that need to be addressed when using association analysis for market basket data. Discovering patterns from a large transaction data set can be computationally expensive. Moreover, some of the patterns may be potentially spurious since they might be happening by chance. This is because association rules do not always suggest causality among items but they specify strong co-occurrences.

Let us discuss these questions in the next sections.

Interestingness Measurements

The concepts of both interestingness and redundancy are somewhat subjective.

Despite the fact that interestingness of a rule depends heavily on the choice of user, there exists a principle for making this decision. A rule (pattern) is interesting if:

- It is unexpected (surprising to the user); and/or
- Actionable (the user can do something with it)

Selecting the best rules demand a thorough research on all measures and a variety of datasets. Another key factor in choosing wisely to consider the improvement expectancy from the output. Will it be improved in terms of time/space complexity, number of operations taken or number of steps?

3.4 A Review on Frequent Pattern Mining Algorithms

3.4.1 Apriori

Apriori is one of the earliest frequent pattern mining algorithms for discovering association rules. It is one of the most well-known algorithms for discovering frequent patterns which is a level-wise, breadth-first algorithm that counts transactions. Apriori algorithm uses prior knowledge of frequent itemset properties. Apriori uses an iterative approach known as a level-wise search, in which n-itemsets are used to explore (n+1)- itemsets.

To illustrate the idea, lattice structure is used to enumerate the list of all possible itemsets. As you can observe in 3.1 the graph shows an itemset lattice for I={a, b, c, d, e}. Generally, a data set that contains k items can potentially generate up to $2^k - 1$ frequent itemsets without the null set. Because k can be very large in many practical applications, the search space of itemsets that is required to be explored is exponentially large.

Candidate Itemsets Generation and Pruning

To generate candidate itemsets, the following are requirements for an effective candidate generation procedure:

- It should avoid generating too many unnecessary candidates.
- It must ensure that the candidate set is complete.
- It should not generate the same candidate itemset more than once.

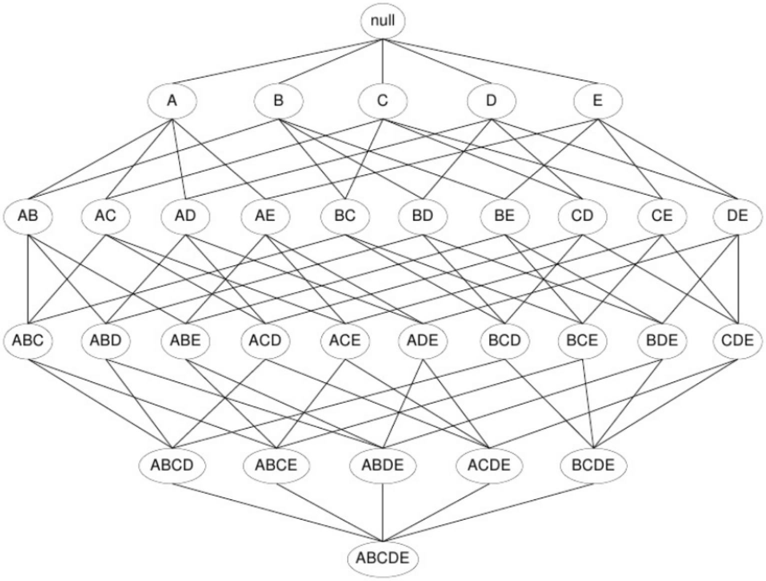


Figure 3.1: An itemset lattice

A method to discover frequent itemsets is to determine the support count for every candidate itemset in the lattice structure. It means if the candidate is contained in a transaction, its support count will be increased. This type of approach could be very costly since it needs $O(NM\omega)$ comparisons where N is the number of transactions, $M = 2^k - 1$ is the number of candidate itemsets and ω is the maximum transaction width.

Reducing the number of candidate itemsets ***(M)*** is one way to reduce the computational complexity of frequent itemset generation.

Proposition 3.2. Using the Apriori property introduced in the next section is an efficient method to eliminate some of the candidate itemsets without counting their support values.

The use of support for pruning candidate itemsets follows the below property:

Definition 3.11. [**Apriori Property**] The property insists If an itemset is frequent, then all of its non-empty subsets must also be frequent. Then if an itemset is infrequent, then all of its supersets must also be infrequent.

For instance, consider the lattice in 3.2. Suppose $\{c, d, e\}$ is a frequent itemsets, then all of its subsets, namely, $\{c, d\}$, $\{c, e\}$, $\{d, e\}$, $\{c\}$, $\{d\}$ and $\{e\}$ should be also frequent.

The idea of exponential search based on support measure is known as **support-based pruning**. This strategy is made possible because of a property named as **anti-monotone** of support measure which suggests the support for an itemset should never exceed the support for its subsets. A two-step process consists of join and prune actions are done iteratively.

Definition 3.12. [**Monotonicity Property**] Let I be a set of items, and $J = 2^I$ be the power set of I. A measure f is monotone if:

$$\forall X, Y \in J : (X \subseteq Y) \rightarrow f(X) \leq f(Y)$$

which means if X is subset of Y, then $f(X)$ must not exceed $f(Y)$.

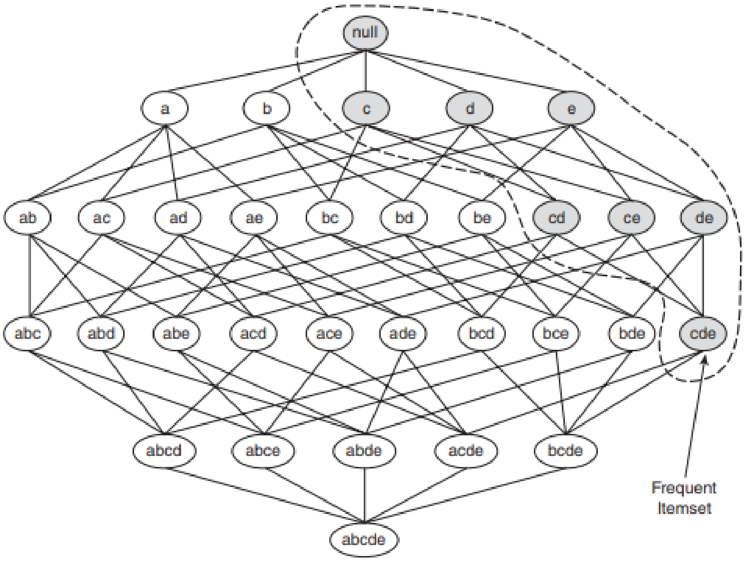


Figure 3.2

Frequent Itemset Generation in the *Apriori* Algorithm

As preivously mentioned, Apriori is a breadth first exploration of a structured arrangement of the itemsets. The pseudo code for frequent itemset generation part of Apriori algorithm is indicated in Algorithm 3.1 .

This algorithm has two important characteristics. It is a *level-wise* algorithm. For example it traverses the itemset lattice on level at a time. Moreover, it develops a *generate and test* strategy for finding frequent items. At each iteration, new candidate itemsets are generated from the frequent itemsets found in the previous iteration. The support for each candidate is the counted and tested against the minsup threshold [14].

Algorithm 3.1 Frequent itemset generation of the *Apriori* algorithm

```
$k = 1$.
$F_k = \{i | i \in I \wedge \sigma(\{i\}) \geq N \times minsup\}$.        ▷ Find all frequent 1-itemsets
repeat
    $k = k + 1$.
    $C_k = apriori\text{-}gen(F_{k-1})$.        ▷ Generate candidate itemsets
    for each transaction $t \in T$ do
        $C_t = subset(C_k, t)$.        ▷ Identify all candidates that belong to t
        for each candidate itemset $c \in C_t$ do
            $\sigma(c) = \sigma(c) + 1$        ▷ Increment support count
        end for
    end for
    $F_k = \{c | c \in C_k \wedge \sigma(c) \geq N \times minsup\}$.        ▷ Extract frequent k-itemsets
until $F_k = \emptyset$
Result= $\bigcup F_k$.
```

K-Apriori

The most influential algorithm for efficient association rule discovery from market databases is K-Apriori, which uses the previous mentioned Apriori property. This algorithm shows good performance with sparse datasets hence it is considered. The K-Apriori algorithm extracts a set of frequent itemsets from the data, and then pulls out the rules with the highest information content for different groups of customers by dividing the customers in different clusters.[22]

3.4.2 Frequent Pattern (FP) Growth

An effective alternative approach called *Fp-growth* encodes the dataset using a compact data structure as *FP-tree* and extracts frequent itemset directly from this structure. In the first pass, the algorithm counts occurrence of items (attribute-value pairs) in the dataset, and stores them to 'header table'. In the second pass, it builds the FP-tree structure which is a representation of the input data.

FP-tree is constructed by reading the data set one transaction at a time and inserting them onto a path in the tree. Items in each instance have to be sorted by descending order of their frequency in the dataset, so that the tree can be processed quickly. Items in each instance that do not meet minimum coverage threshold are discarded. If many instances share most frequent items, FP-tree provides high compression close to tree root.

Recursive processing of this compressed version of main dataset grows large itemsets directly, instead of generating candidate items and testing them against the entire database. Growth starts from the bottom of the header table (having longest branches), by finding all instances matching given condition. New tree is created, with counts projected from the original tree corresponding to the set of instances that are conditional on the attribute, with each node getting sum of its children counts. Recursive growth ends when no individual items conditional on the attribute meet minimum support threshold, and processing continues on the remaining header items of the original FP-tree.

Once the recursive process has completed, all large itemsets with minimum coverage have been found, and association rule creation begins.[14]

Notes on FP-Tree

A FP-tree is a compressed representation of the input. It is constructed by reading the dataset one transaction at a time and mapping each transaction onto a path in the FP-tree. The more the paths overlap with one another, the greater the compression that can be achieved. An FP-tree is typically smaller than the size of the uncompressed data, because many transactions in market basket data often share a items in common. However, the physical storage requirement for the FPtree is higher than the original data, because it requires additional space to store pointers between nodes and counters for each item.

3.5 Tools

In this project, we use several tools to analyze data and visualize the results.

3.5.1 Python

Python [30] is an excellent interpreted high-level programming tool for data analysis and general-purpose programming since it is user friendly and pragmatic. Moreover, it is complemented by practical third part packages that were designed to deal with large amounts of data. We put our knowledge of Python data containers into the project since containers set the model for more powerful data objects of NumPy. NumPy package extends Python with a fast and efficient numerical array object.

3.5.2 R

R [31] is a free software environment for statistical computing and graphics. It compiles and runs on a wide variety of UNIX platforms, Windows and MacOS. The R language is widely used among statisticians and data miners for developing statistical software and data analysis. It is an interpreted language in which users typically access it through a command-line interpreter. The capabilities of R are extended through user-created packages developed primarily in R, which allow specialized statistical techniques, graphical devices, import/export capabilities, reporting tools and etc.

3.5.3 Rapidminer

Through this thesis we use RapidMiner Studio 8.2 [32], an open core model, for generating frequent items and association rules. Rapidminer is a data science software platform that provides an integrated environment for data preparation, machine learning, deep learning and etc. It is used for business and commercial applications as well as for research, education, training, and application development plus it supports all steps of the machine learning process including data preparation, results visualization, model validation and optimization. In this software, data mining processes/routines are viewed as sequential operators. RapidMiner functionality can be extended with additional plugins which are made available via RapidMiner Marketplace such as Weka Extension.

3.5.4 Weka

Weka[1] is an open source software and a collection of machine learning algorithms for data mining tasks. Weka contains tools for data pre-processing, classification, regression, clustering, association rules, and visualization. It is also well-suited for developing new machine learning schemes.

The extension of Weka in RapidMiner[33] combines two of the most widely used open source data mining solutions. By installing it, we can extend RapidMiner to everything that is possible with Weka while keeping the full analysis, preprocessing, and visualization power of RapidMiner. All modeling methods and attribute evaluations from the Weka machine learning library are then available within RapidMiner. we will get access to additional modeling schemes, rule learners and other materials.

Primely, for producing association rules, we use an operator from Weka extension. One downside factor is ordering the rules which is not very convenient by using this operator.

[1] Waikato Environment for Knowledge Analysis

3.6 Pattern Discovery

Given these preliminaries for finding patterns, let us now turn to our original problem and the methodology used in our research. Since the goal is to find frequent patterns, we are going to follow the exact previous steps to achieve this goal. Since our problem is based on the market basket analysis, the first step is to look for items that we are interested to study and then define the transactions for them. Defining these matrices requires experiments since each method has different results and its own advantages. In this chapter, we briefly introduce each method because after this is done, we will proceed with the next steps to select and visualize rules obtained from each method. we will explain in detail each of their individual results in the next chapters.

3.6.1 First Step: Matrix of Transactions and Itemsets

For using any rule mining algorithm, we are required to transform the data from its frame format into transactions such that each row corresponds to a transaction whereas each column indicates an item.

In this part, we present the matrix definitions that lead us to interesting findings. At the first approach, we only consider temporal correlation and search for any geographical correlations after obtaining the rules.

Two possible approaches for choosing the transaction definition for time window exists:

- Fixed non-overlapping windows
- Partially overlapping sliding windows

Initially, we take fixed non overlapping windows of 2 hours. The rational behind this assumption is based on the granularity of alarms. As we observed in section 2.3.1, more than 50% of alarms happen simultaneously. This value is of course arbitrary and as we see through the chapter, changing it has an impact on the level of specificity of the rules. However, by this value, windows will not include alarms

When choosing an item since features are categorical, we should look at the most important ones. For our case, the major features are *Original Severity*, *Alarm Type*, *Probable Cause* that are selected by domain expert based on their importance.

Each transaction contains the set of items observed in an interval of 2 hours. So, after selecting the features, we should count all events reported in each interval as numerical "features" for each item. For effective processing, matrix is then mapped into binary data(0 if not present, 1 if> 0). As a result, for a specific transaction i, if an item j is observed then the matrix position (i,j) converts to 1. If the item j is not seen in the transaction i then the matrix position (i,j) will be remained 0. Since the goal is to find the frequent items which occur together so transactions with more number of items will provide useful information about the network behaviour. Furthermore, because presence of an item in a transaction is more considerable rather than its absence, an item is a **asymmetric** binary value.

In order to find the most frequent items, we apply FP-Growth Operator in Rapidminer. The FP-growth algorithm is an efficient algorithm for calculating all frequently-occurring itemsets in a transaction database, using a novel data structure known as FP-tree, divide and conquer method in nature. For choosing a frequent pattern mining algorithm, we opted Fp-growth due to its efficiency when working with our data set. FPGrowth utilizes a depth-first search instead of a breadth first search and uses a pattern-growth approach (this means that unlike Apriori, it only considers patterns actually existing in the database). Whereas, Apriori utilize a level-wise approach where it will generate patterns containing 1 items, then 2, 3 and etc. Moreover, it will repeatedly scan the database to count the support of each pattern. As the dimensionality of the database increases with numbers of items, Apriori needs more search space and consecutively the I/O cost will increase. As a result, due to compact structure of Fp-tree and candidate generation, Fp-growth requires less memory and execution time (see [28] and [29]).

However, given the matrix, caution must be exercised since it is not yet in the suitable input format for FP-Growth algorithm. As we can see from figure 3.3, it is necessary to convert the market basket data type into binary values, since the algorithm works only with this type of values. For this reason, the Numerical to Binominal operator is applied to change these numerical attributes to binominal ones.

Observation 3.4. The restricted use of binary type entails a loss of information. It limits the event counter in a window to the simple presence of a device alarm in the considered interval. This apparent lack of information can be justified by a posteriori study of rules.

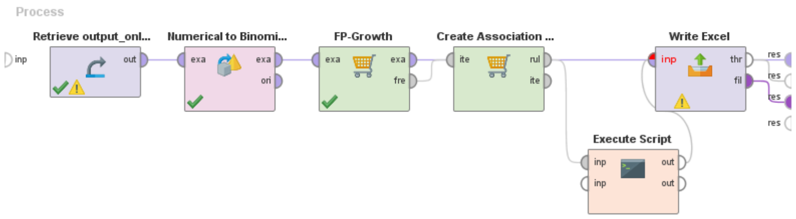


Figure 3.3: A scheme for creation of association rules

As previously mentioned, for finding frequent items, we focus on two approaches. First we consider temporal correlation of specific devices then we proceed the study with a more generic definition by considering each device type.

- **Seprated Devices:** we will consider temporal correlations among specific devices.

- **Seprated Device Types:** we will consider a more generic definition and find temporal correlation among different types of devices. The study is then proceeded to also investigate spatial correlations.

3.6.2 Second Step:Rule Generation

Now we have our transaction dataset, and it shows the matrix of items being observed together. We do not actually see how often they are seen together, and we do not have the rules either. But we are going to calculate it in the next following part. The output of FP-Growth operator is frequent items which are the suitable input for creating association rules. We have to set the parameters such as measures of interests for the rules. These are set arbitrary and it is dependent on the choice of the user.

The output of Create Association Rules Operator gives a summary of rules and the information on total items mined, and the minimum parameters we set earlier. We select the rules by order of lift and length of items.

3.6.3 Third Step: Visualization

After obtaining the rules, we should visualize them based on measures of interestingness. Having done this, we can then easily choose interesting rules and visualize them over time.

Chapter 4

Analysis of Separated Devices

4.1 Overview of the Methodology with Separate Devices

To find correlation among different devices, a simplified approach of transactions is defined to consider only the network device IDs and extract specific relations. Given this definition, an example of the obtained matrix is shown at table 4.1. The table indicates a sample transaction considered in a certain arbitrary time bin where device 0 is raising at least 4 alarms whereas device 1400 is raising no alarms.

Device0	Device1	...	Device1405	Device1406
4	92	...	0	205

Table 4.1: Matrix of transactions and itemsets: considering each network device ID

We are eager to see which devices were raising alarms at the same time bin more frequently. We will focus on Turin province to reduce complexity and study two datasets reported in two different month of May and September. First, let us consider the data set for the all the raised alarms in Turin province in month of May 2017. A brief statistics of this dataset is as below:

- Alarms raised: 38563
- Centers involved: 700
- Devices involved: 1400
- Time bin: 2 hours

Number of transactions is calculated for time bins of 2 hours through this month with 31 days which leads to:

$$\text{Number of Transactions} = \frac{31 \times 24}{2} = 372$$

This means the input matrix has 1400 distinct items(devices) which are the columns and 372 transactions/baskets which are the rows.

We proceed with the similar approach to obtain interesting rules (if any) in September 2017. Based on the temporal evolution of raised alarms in September as we observed in 2.8, the previous chapter, there are less devices that generate alarms in this month since there was no "global failure" event involving most of the network and devices.

- Alarms raised: 25109
- Centers involved: 560
- Devices involved: 930
- Time bin: 2 hours

As previous the number of transactions is calculated for time bins of 2 hours through this month with 30 days which leads to:

$$\text{Number of Transactions} = \frac{30 \times 24}{2} = 360$$

Now we have our transaction dataset, and it shows the matrix of items being observed together. We do not actually see how often they are seen together, and we do not have the rules either. But we are going to calculate it in the next following part.

The obtained matrix can be used in order to extract the frequent item sets by Fp-growth Algorithm. An example of Fp-tree in our studied case is shown at 4.2. This table shows the most frequent item in the data set is GBSCTO033 with the support equal to 86.1%.

Support	item 1	item 2	item 3
0.861	GBSCTO033	...	...
0.861	GBSCTO034	...	...
0.755	GBSCTO033	GBSCTO034	...
0.666	UBTSTO109	...	...
0.579	GBSCTO032	GBSCTO033	GBSCTO034
0.503	GBSCTO033	UBTSTO109	...

Table 4.2: A sample of *FP-tree* obtained by RapidMiner

4.2 Rule Generation

The output of FP-Growth operator is frequent items which are the suitable input for creating association rules. We have to set the parameters such as measures of interests for the rules. These are set arbitrary and depends on the user's choice.

The below thresholds are set for the datasets of May and September respectively. These values are lower for September, simply, since if we set the threshold too high, we would only obtain few rules due to having less devices generating the alarms in compared to May.

- Dataset: Turin, May
- Minimum confidence: 0.95
- Lower bound for minimum support: 0.085
- Other parameters are set to the default of Rapidminer

.

- Dataset: Turin, September
- Minimum confidence: 0.3
- Lower bound for minimum support: 0.03
- Other parameters are set to the default of Rapidminer

The output of Create Association Rules Operator gives a summary of rules and the information on total items mined, and the minimum parameters we set earlier. The number of rules obtained for May is 6995 whereas this is 1999 for September. These are obviously large numbers so it is not reasonable to go through them without visualizing the results based on measures of interestingness.

4.3 Rule Selection and Visualization

For visualizing the rules, we use R to plot all of them with the our previously set threshold. We select the rules by order of lift and length of items. Figure 4.1 shows rules in May based on their support and lift with the shading of confidence. As we can observe most of the rules have a support close to the minimum threshold 8.5% and 10%. Furthermore, these are mainly the rules with highest lift and confidence. This actually highlights the inverse correlation of support and lift described by formula 3.4 meaning rules that are held true in fewer time bins are presumably more reliable and vice versa.

Figure 4.2 shows two-key plot of rules in May considering the same definition for matrix. Rules are ordered by support and confidence and the colors show number of items (devices) involved in the rule. It is noticeable from the figure that common rules have less devices in compared to rare ones. Rules with less support such as 10% are very interesting since they show a possible correlation among large number of devices equal to 10.

Using the same approach, we can plot all of the rules obtained in September with our previously set threshold. Figure 4.3 shows these rules based on their support and lift with the shading of confidence. As we can observe the majority of rules have support between the minimum 3% and 20%. However, the rules with highest lift equal to 24 have a support close to 3% which is as previously mentioned the result of an inverse relation between support and lift.

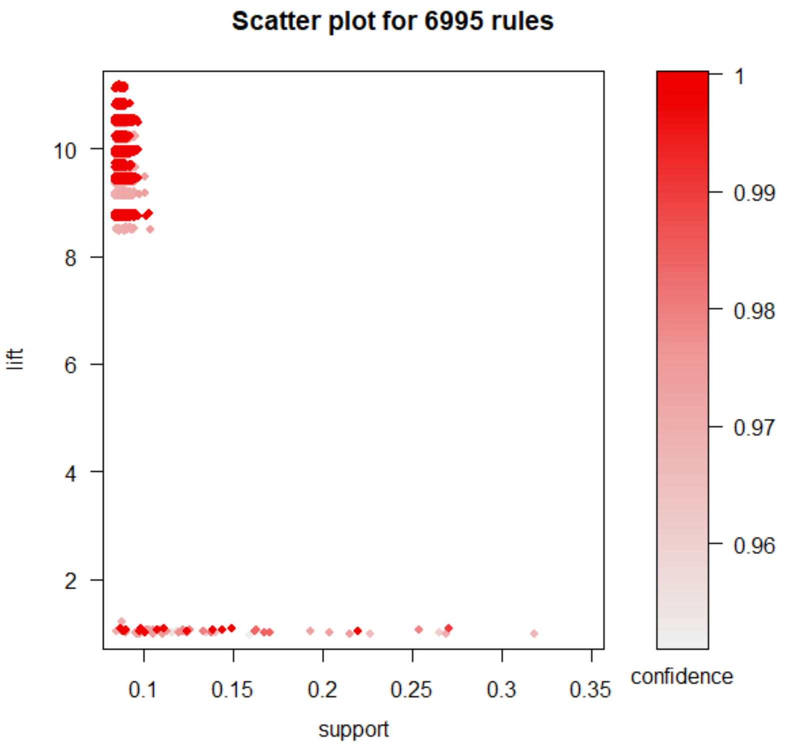


Figure 4.1: Scatter plot of rules in May considering separate devices ordered by support and lift with confidence shading

Figure 4.2 shows two-key plot of rules in September considering the same definition for matrix. Rules with order of 4,5 and 6 usually have a support less than 5% which makes them rarely observed in the dataset.

Now that we have an overview of rules in May and September, we will respectively select two examples from each dataset that are considered interesting due to their fairly high lift, long sequence and inevitably less support.

4.3.1 First Example

This rule as observed below, is involving 10 different devices over Turin province on May. The rule is interesting because as shown in map 4.5 devices are located in close centers with respect to each other and we can even see some devices are in the same center.

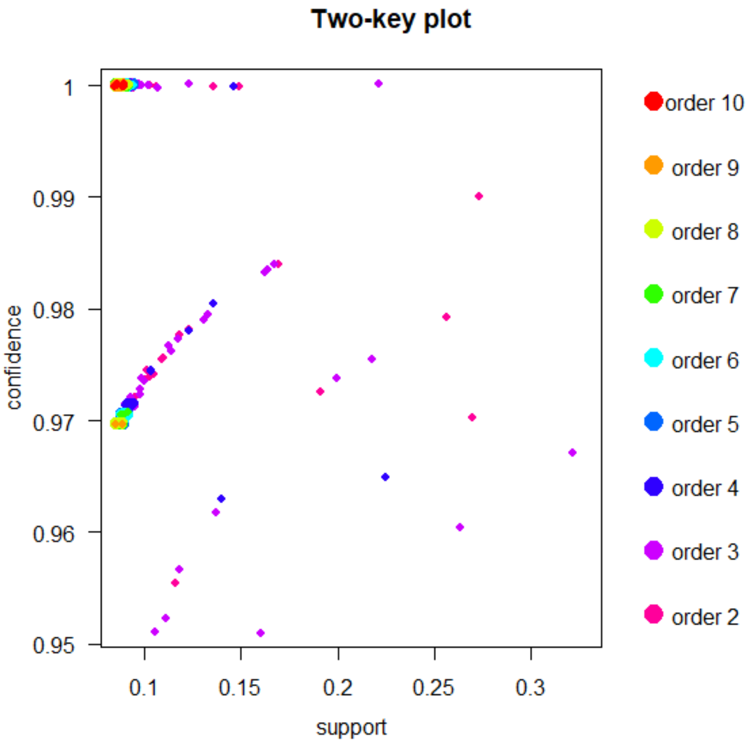


Figure 4.2: Two-key plot of rules in May considering separate devices by support and confidence ordered by number of items in the rule

Antecedent	UBTSTO27F UBTSTO08E UBTSTO384
Consequent	UBTSTO0B7 UBTSTO14A 8BTSTO384 1BTSTO0B7 8BTSTO0B6 1BTSTO156 1BTSTO00D

As we can observe in the below table information on the interestingness of a the rule is reported as an output.

Confidence	Lift	Leverage	Conviction
0.97	11.15	0.08	15.07

The antecedent holds true for 33 time bins whereas this number for consequent is 32 times and this leads to a confidence of 97%.

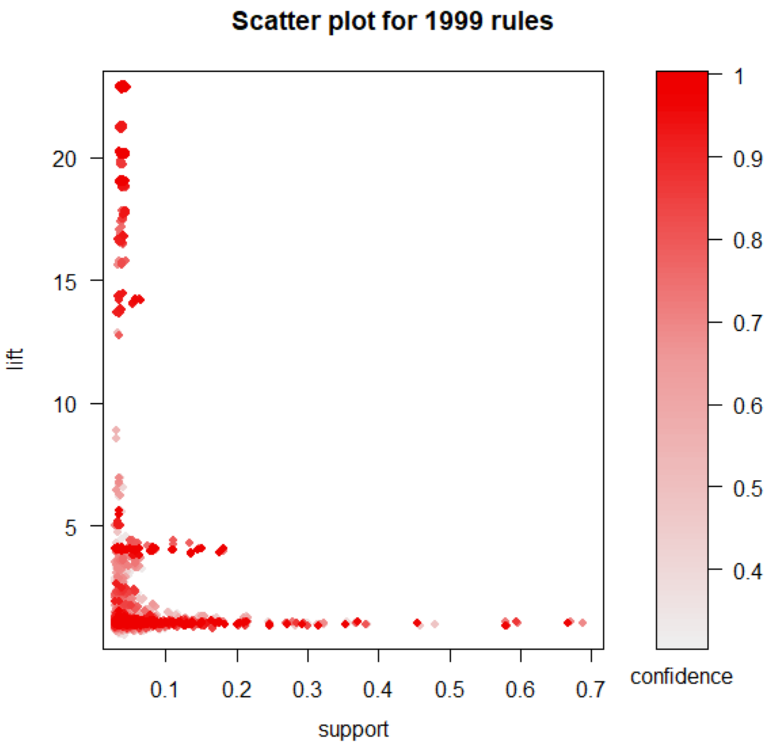


Figure 4.3: Scatter plot of rules in September considering separate devices ordered by support and lift with confidence shading

After exploiting the raw data for additional information, we find out that some devices are in the same centers, or in a very closed region. This may be an indication of a factor for correlation. All of the devices involved are BTS working either with UMTS/LTE technology. Almost every error raised had a probable cause of indeterminate with type of quality service alarm. A single problem "sync reference PDV problem" too often reported in the raw data. Most of the alarms have been reported to the network operations center which seem to make the rule already interesting on TIM previous findings.

Visualization

Despite the fact that there are measures for interpreting a rule as an interesting one, we need more infomation such as distribution of alarms over time.

Figure 4.6 shows the distribution of alarms which are generated by each device of interest. This is reported for every transaction so we do not see alarms to appear more than once in this graph. It is clear from the bar graph that almost all devices raise a similar number of alarms.

As shown in 4.7 and 4.8, figures report the scatter plot of alarms categorized by two of most important features, alarm types and probable cause. We apply an arbitrary value of

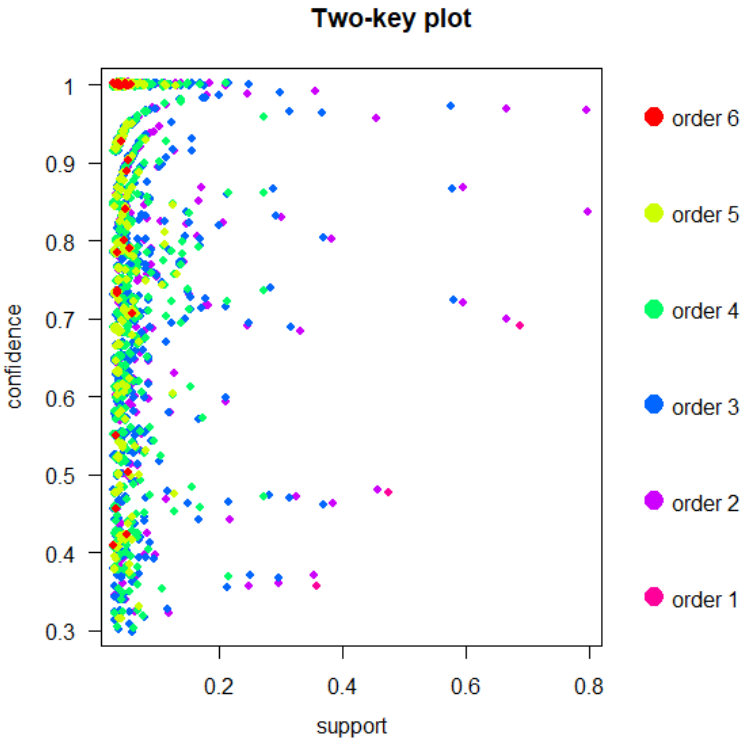


Figure 4.4: Two-key plot of rules in September considering separate devices by support and confidence ordered by number of items in the rule

jitter on the y-axis to better observe the distribution of alarms. For this means, we assign each alarm type a unique numerical value (same is done for probable cause field) because jitter can not be applied on categorical features. Label zero in figure 4.7 shows quality of service alarm type and label one stands for communications alarm. Label zero in 4.8 shows indeterminate while label seven is unavailable. This correlation is obsevred in 32 time bins and temporal correlation is indeed present.

Although correlation is observed based on the scatter plots (see4.7, 4.8), we can validate its level of strength by zooming in the time bins of 2 hours(intervals of 20 minutes). Figure 4.9 indicates a significant correlation of interested devices but no particular synchronization is visible.

Observation 4.1. This rule highlights that the results so far have been promising. Even though, this definition for the transactional matrix does not provide any general overview of device types and instead focuses on each device itself.

4.3.2 Second Example

We select another rule that has a high lift and support to investigate. The geographical coordinates of devices involved in this rule is shown by map 4.10.

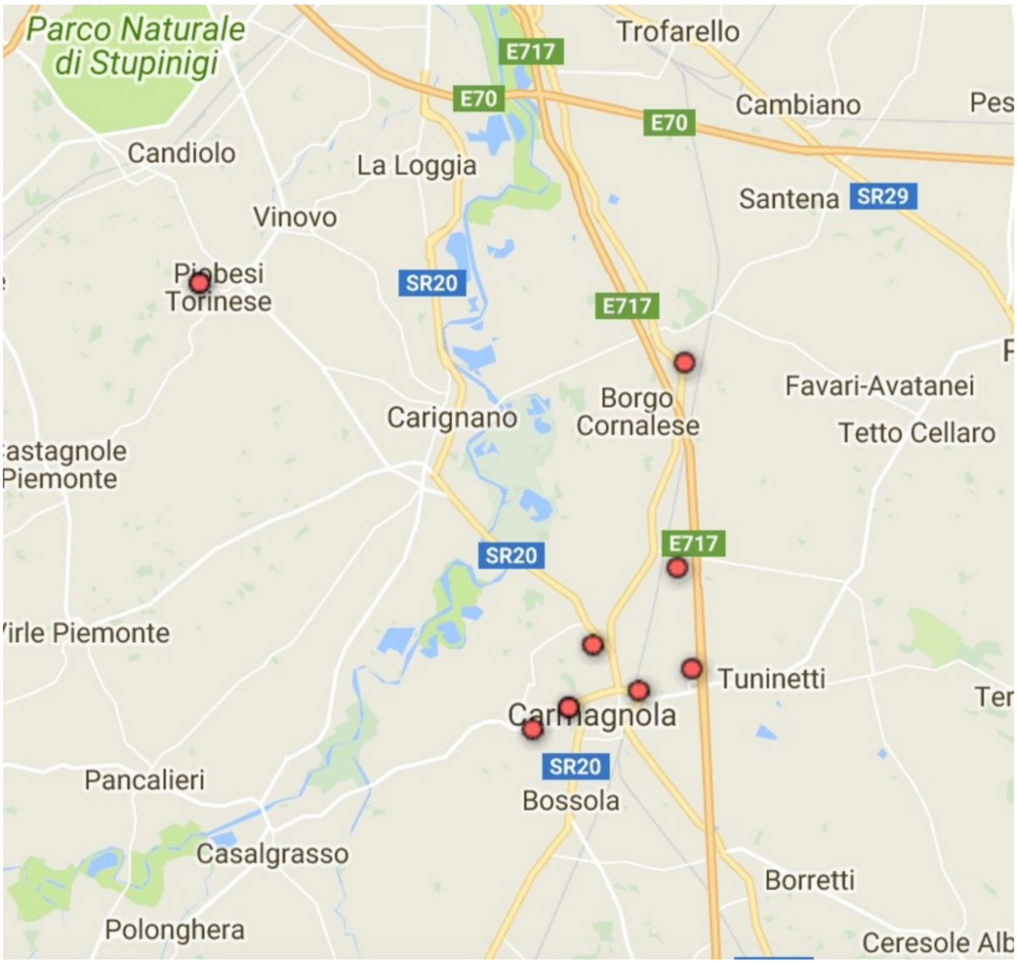


Figure 4.5: Geographical location of devices involved in the first case

Antecedent	URNCTO030 8BTSTO26E
Consequent	UBTSTO26E

As we can observe in the below table, information on the interestingness of a the rule is reported as an output.

Confidence	Lift	Leverage	Conviction
1	8.98	0.07	25.77

The antecedent and consequent hold true for 29 time bins and this leads to a confidence of 100%. Given the geographical coordinates of devices, this rule suggests that there is a correlation between two centers that are almost 22 kilometers apart.

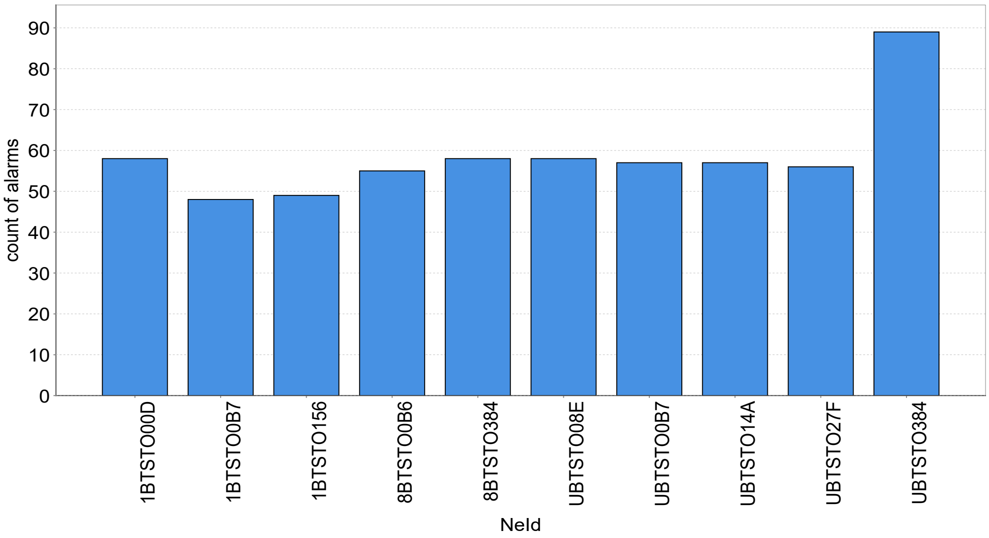


Figure 4.6: Bar Graph: Distribution of alarms generated by each device of interest in the first case

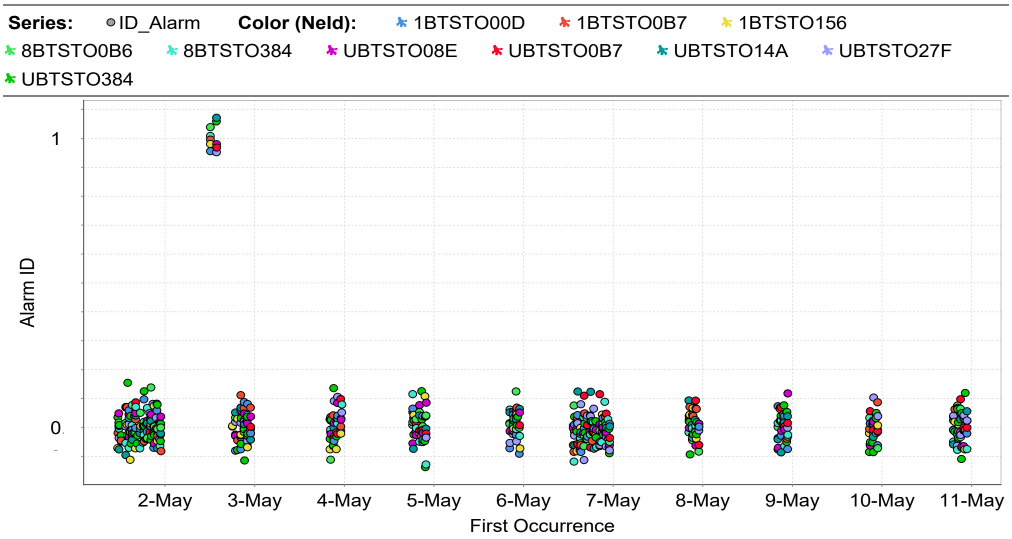


Figure 4.7: Scatter plot: Alarms categorized by alarm types over time in the first case

Visualization

By visualizing the rule, we can validate to some extent whether the correlation is by chance or not. Figure 4.11 shows the distribution of alarms which are generated by each device of interest. As previous, this is reported for every transaction. One device is seem to generate the majority of alarms and is more chatty. This can be an issue if alarms are scattered over time.

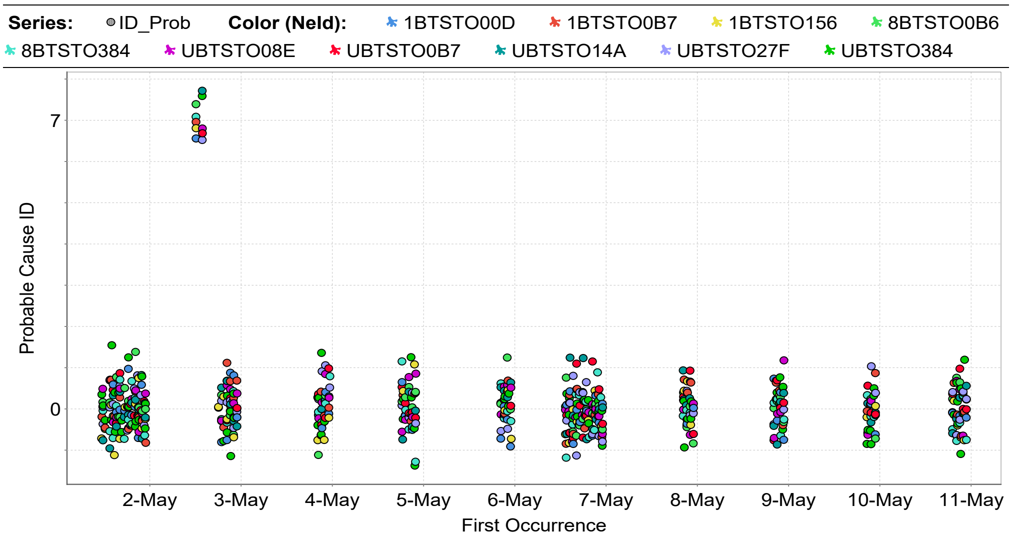


Figure 4.8: Scatter plot: Alarms categorized by probable cause over time in the first case

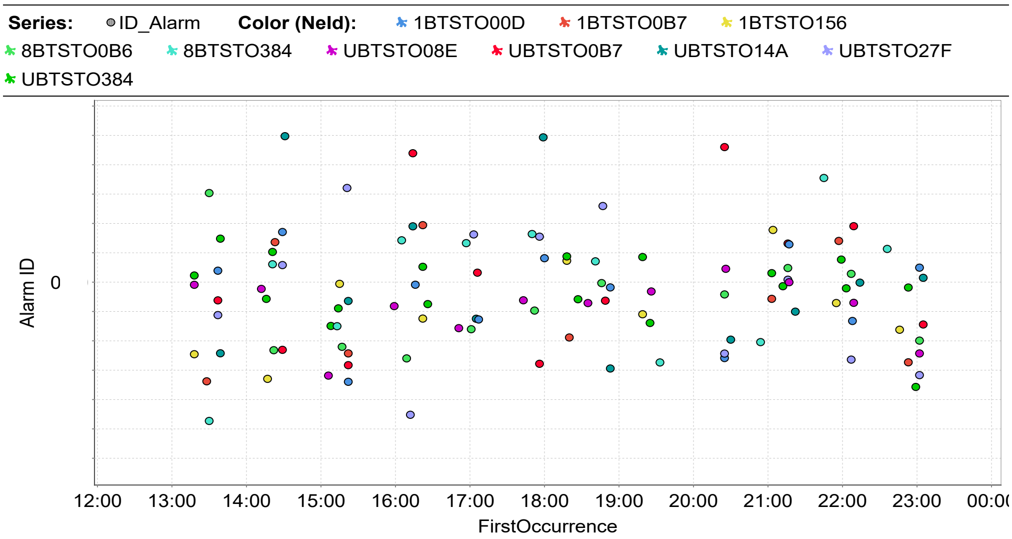


Figure 4.9: Zoomed Scatter plot: Alarms categorized by alarm type over time in the first case in 6th of May

Observation 4.2. There seems to be correlation between one of the devices,UBTSTO26E (located in Almese Est) and URNCTO030(located in Lancia), even if they are far away from each other(about 22km).

Let us look at the scatter plot in Figure 4.12 which is reporting the alarms categorized by alarm types. As we can observe devices are raising a large number of alarms which make the task of locating the time correlations, much more challenging. It seems from the figure that most of events are happening at the same time plus the same bin. Label 0, 1

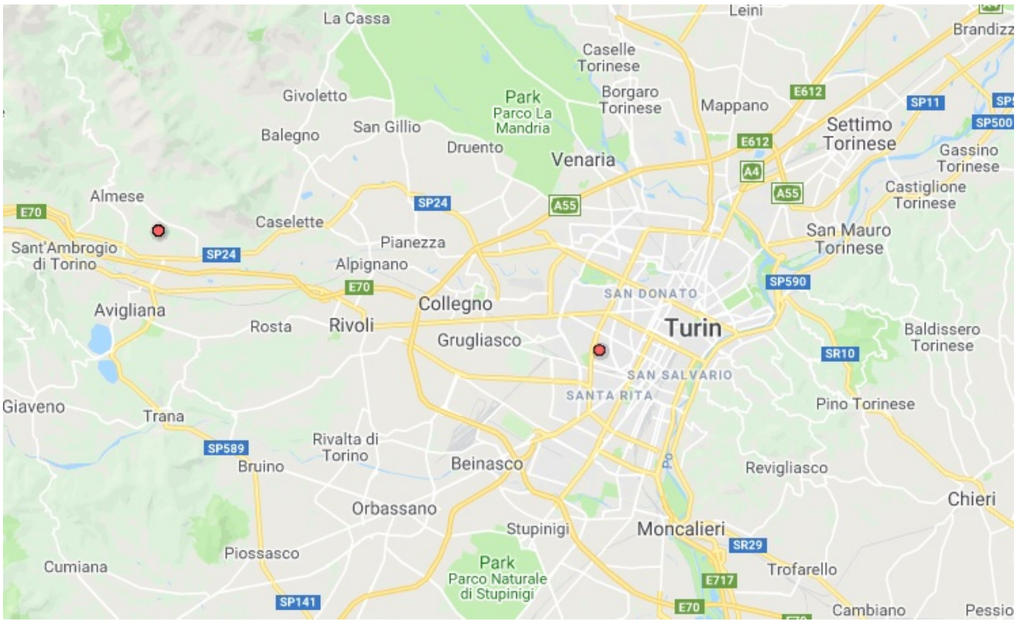


Figure 4.10: Geographical location of devices involved in the second case

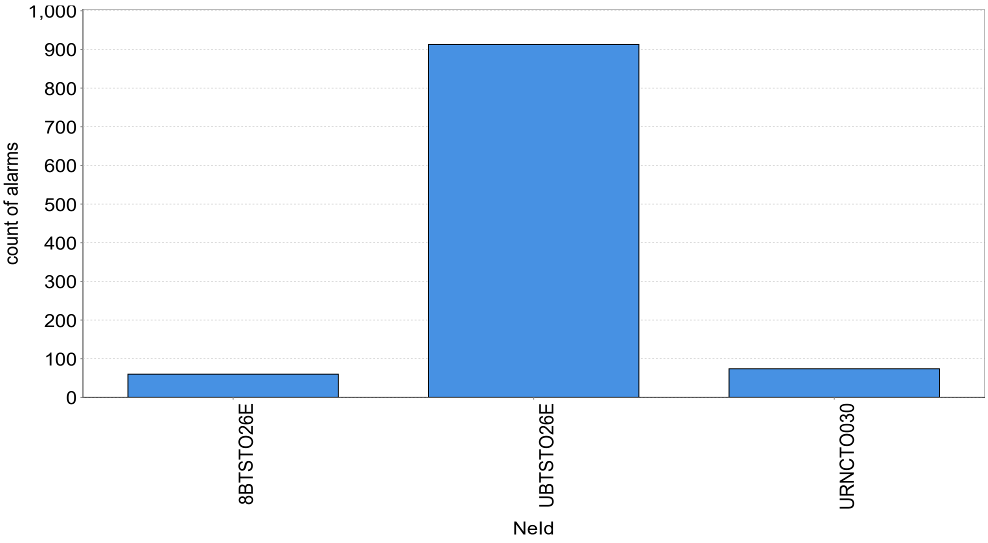


Figure 4.11: Bar Graph: Distribution of alarms generated by each device of interest in the second case

and 2 stand for communications, quality of service and equipment alarms respectively.

Figure shown in 4.13 could be a useful representation since it reported the types of probable cause of each alarm involved in this case. Label 0, 4, 7, 9 and 10 stand for indeterminate, synchronization source mismatch, unavailable, call establishment error and loss of signal probable causes, respectively. As we can see, the majority of alarms generated by device URNCTO030 is due to loss of signal. Part of alarms generated by device UBTSTO26E are indeterminate. This is an issue cause at this point, we do not have any more information about this probable cause.

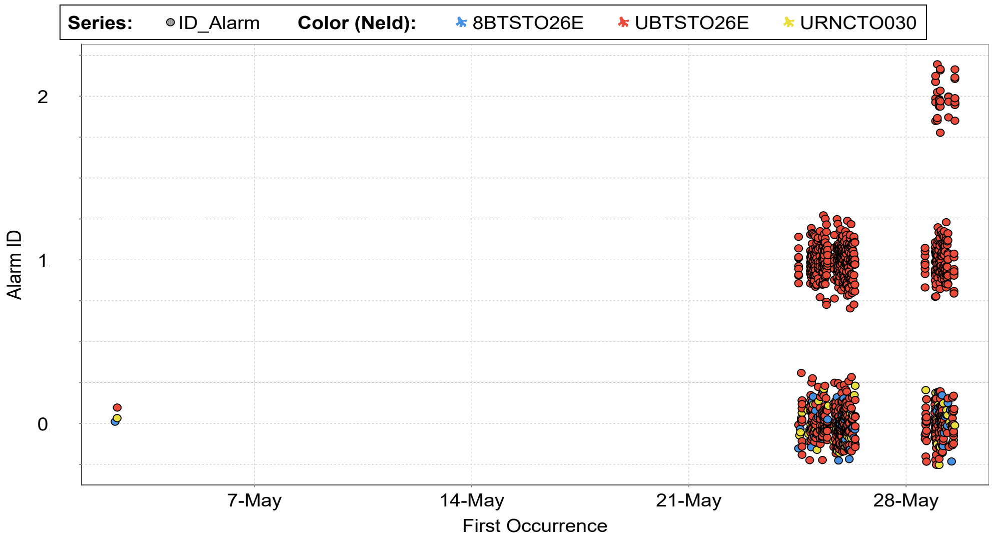


Figure 4.12: Scatter plot: Alarms categorized by alarm types over time in the second case

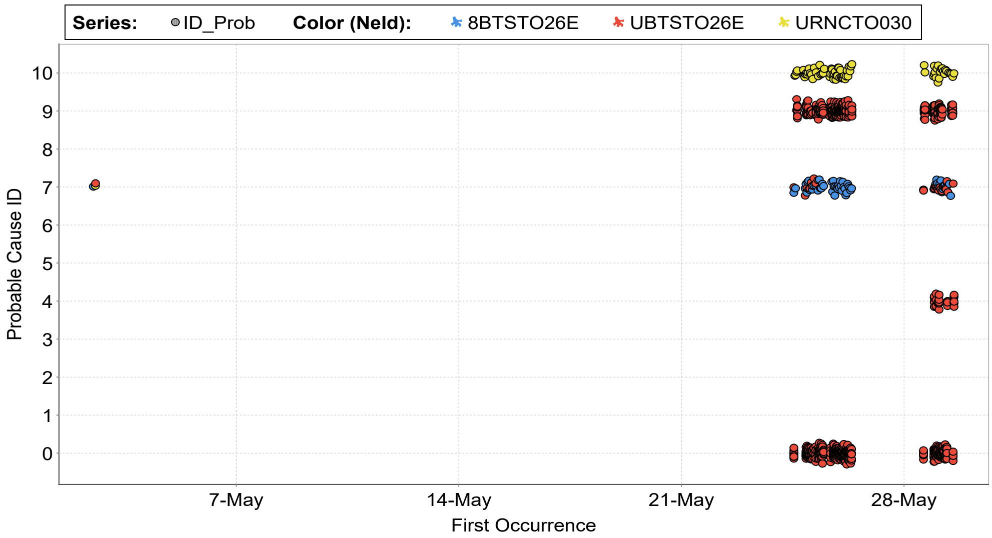


Figure 4.13: Scatter plot: Alarms categorized by probable cause over time in the second case

This correlation can be validated by zooming on an optional day such as 25 of May and looking for time bins of 2 hours as Figure 4.14 suggests. There are a lot of alarms generated in each time bin by the device URNCTO030; although the figure confirms a correlation between this device and the another, UBTSTO26E. Types of alarms are mainly from the two most famous types, communication alarms and quality of service. This plot helps us to conveniently observe the correlation and identify the distribution of alarms generated by each device in the slots.

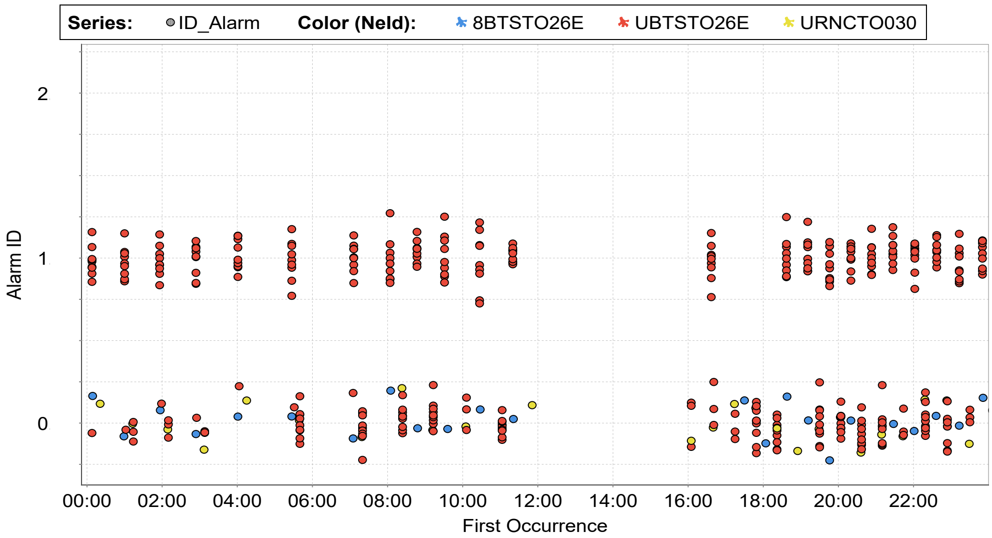


Figure 4.14: Zoomed Scatter plot: Alarms categorized by alarm type over time in the second case in 25th of May

Observation 4.3. As a conclusion, we observed from the scatter plots that, there seems to be correlation between one of the devices in Almese Est and the one is Lancia, even though they are not close in distance and almost 22km apart. Searching the specific cause of alarms through the other fields of data set, we find out that:

- UBTSTO26E is raising a lot of errors of UtranCell–Service Unavailable.
- 8BTSTO26E is raising a lot of errors known as HeartBit–Failure.
- URNCTO30 is raising a lot of NodeSync–Phase Difference measurement failed

A possible scenario for this case could be a failure of LTE network in Almese (device 8BTSTO26E) in which it creates a lot of issues on UMTS network (UBTSTO26E) and conseqeutly, it is reflecting on RNC and the device URNCTO30 located in Lancia or vice-versa.

4.3.3 Third Example

We select an interesting rule from September dataset that has almost the highest lift among rules with a reasonable support to investigate.

Antecedent	2BTSTO187 1BTSTO247
Consequent	1BTSTO398 2BTSTO3B1 1BTSTO002

As we can observe in the below table, information on the interestingness of a the rule is reported as an output.

Confidence	Lift	Leverage	Conviction
1	24.5	0.04	12.47

The antecedent and consequent hold true for 13 time bins and this leads to a confidence of 100% and support equal to 3.6%. This means 13 times out of 13, these permutation of devices were reporting alarms together in the same time bins. The geographical coordinates of devices of interest is reported in figure 4.15.

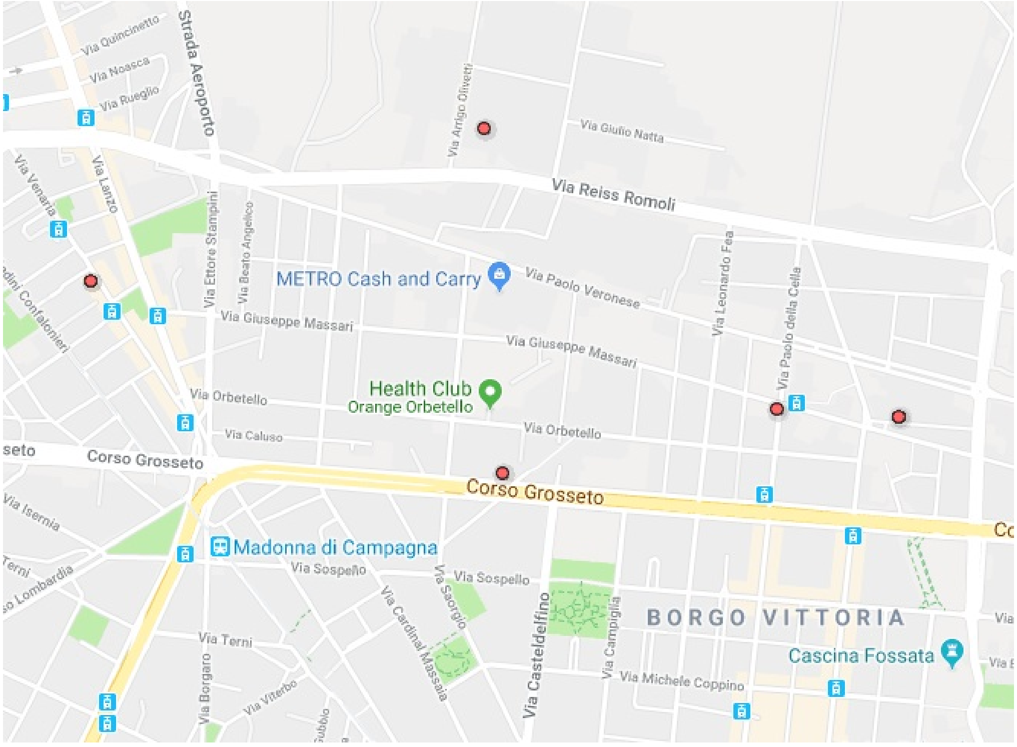


Figure 4.15: Geographical location of devices involved in the third case

Visualization

By visualizing the rule, we can validate to some extent whether the correlation is by chance or not. Figure 4.16 shows the distribution of alarms which are generated by each device of interest. The devices seem to raise an almost equal number of alarms.

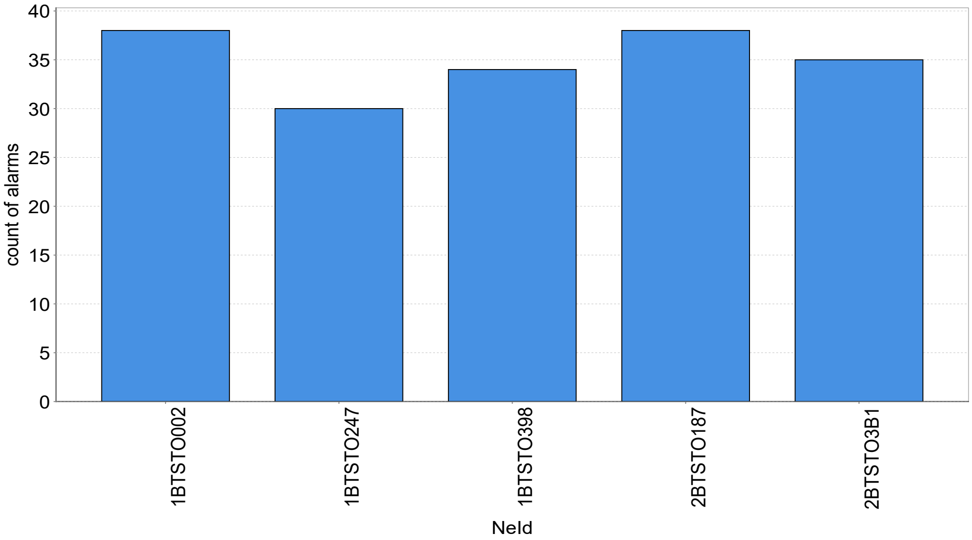


Figure 4.16: Bar Graph: Distribution of alarms generated by each device of interest in the third case

The scatter plot in figure 4.17 is reporting the raised alarms through September generated by devices of interest at the same time bins and categorized by types of alarms. Label 0 and 1 stand for communication and quality of service alarms respectively.

The vertical lines show co-occurrent alarms which could be a reason of strong correlation among devices and the large value of lift in this rule in comparison to the other ones.

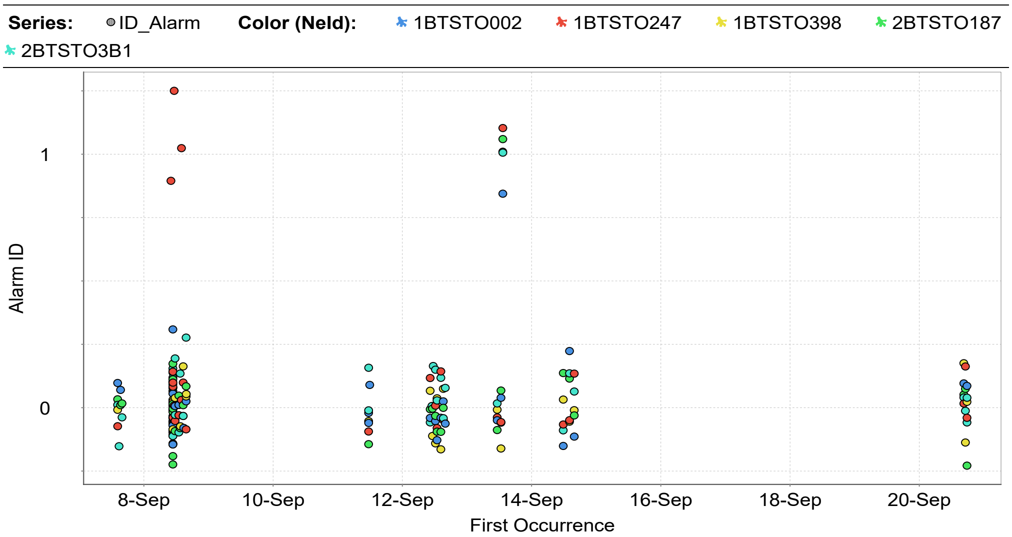


Figure 4.17: Scatter plot: Alarms categorized by alarm types over time in the third case

Figure shown in 4.18 represents the types of probable causes of each alarm involved in the rule. Label 0,9 and 10 stand for indeterminate, unavailable and loss of signal respectively. As shown in the figure, majority of alarms have the a probable cause of unavailable.

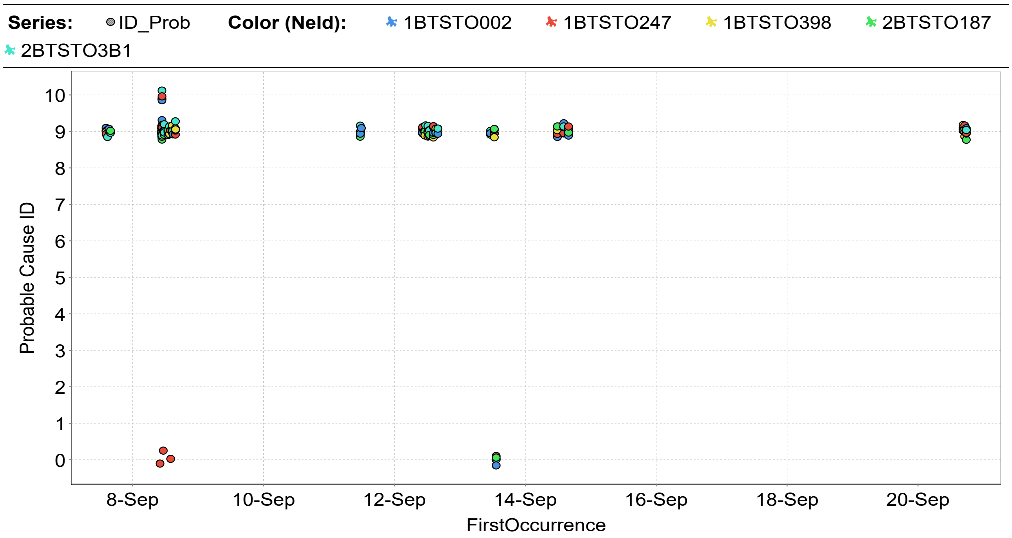


Figure 4.18: Scatter plot: Alarms categorized by probable cause over time in the third case

The correlation is validated by zooming on an optional day in 8th of September and looking for time bins of 2 hours as figure 4.19 suggests.

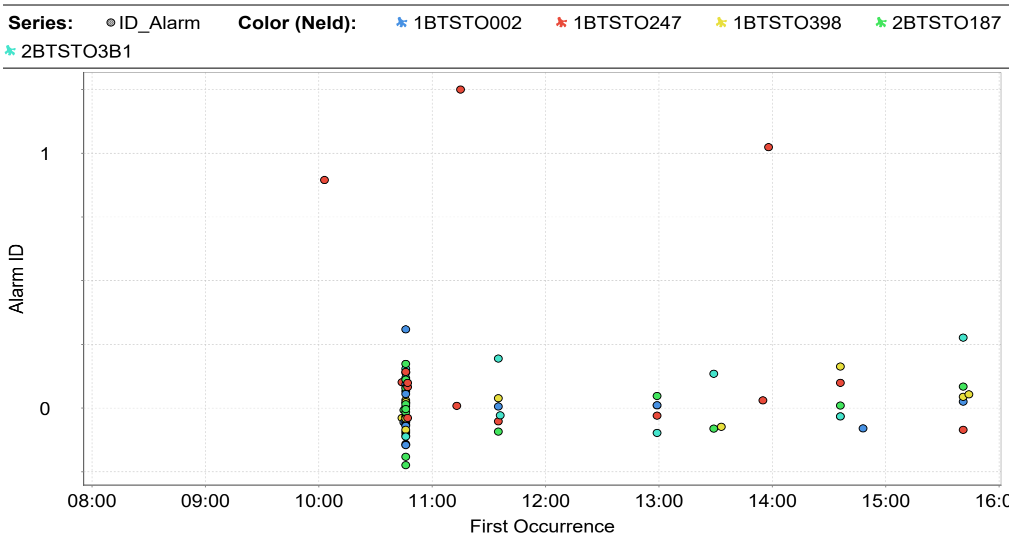


Figure 4.19: Zoomed Scatter plot: Alarms categorized by alarm type over time in the third case in 8th of September

4.3.4 Fourth Example

In this case we choose to focus on a rule which does not have a very long sequence but still stands out because of its high lift and confidence. Here, we want to show whether all the rules including good measures are reliable or not.

Antecedent	UBTSTO218 1BTSTO392
Consequent	UBTSTO392

As we can observe in the below table, information on the interestingness of a the rule is reported as an output.

Confidence	Lift	Leverage	Conviction
0.92	14.29	0.03	5.62

The antecedent and consequent hold true for 13 time bins and this leads to a confidence of 100% and support equal to 3.6%. This means 13 times out of 13, these permutation of devices were reporting alarms together in the same time bins. The geographical coordinates of devices of interest is reported in figure 4.20.

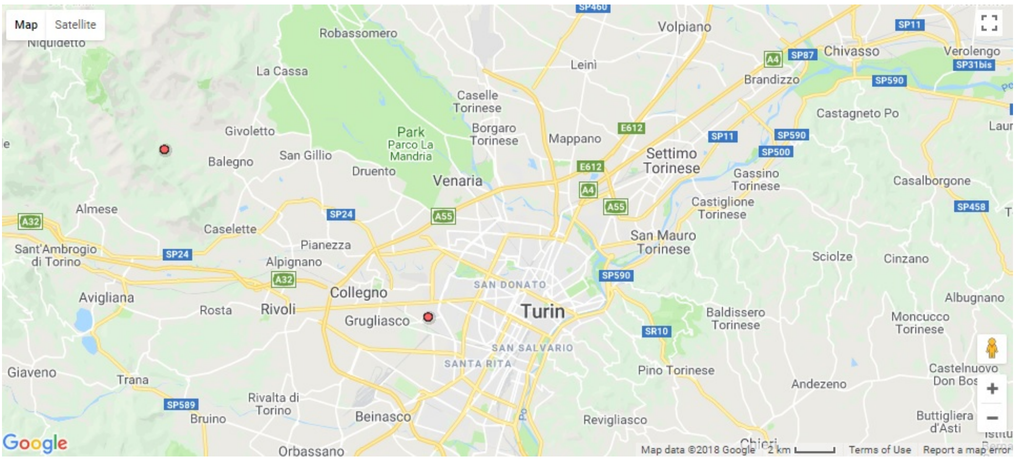


Figure 4.20: Geographical location of devices involved in the fourth case

Visualization

By visualizing the rule, we can validate to some extent whether the correlation is by chance or not. Figure 4.21 shows the distribution of alarms which are generated by each device of interest. Total number of alarms raised by these devices show that this correlation might be not correct since most of the alarms are from a device located in one center. Let us see if they are correlated over time by looking at the scatter plots of alarms.

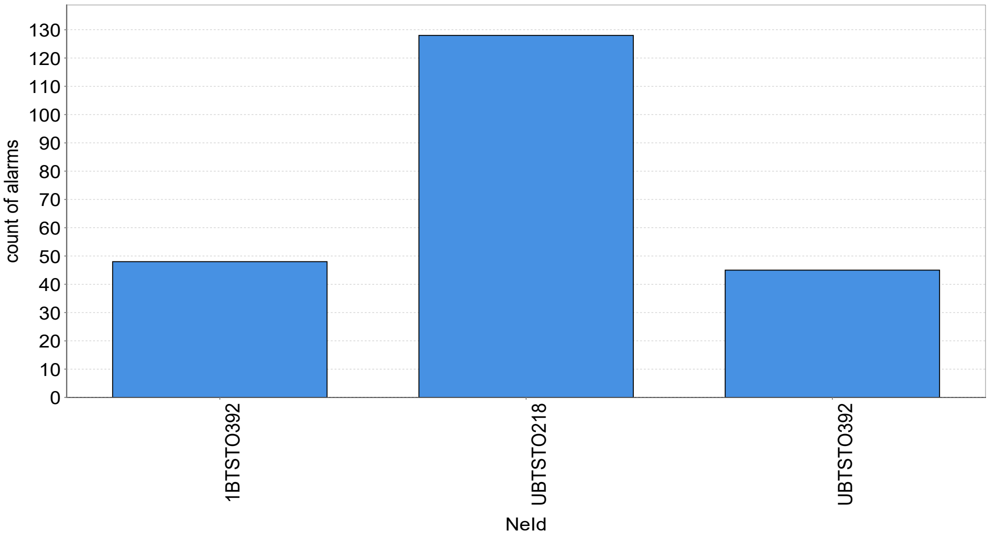


Figure 4.21: Bar Graph: Distribution of alarms generated by each device of interest in the fourth case

The scatter plot in figure 4.22 is reporting the raised alarms through September generated by devices of interest at the same time bins and categorized by types of alarms. Label 0,1 and 2 stand for equipment malfaunction, processing error and communication alarm types.

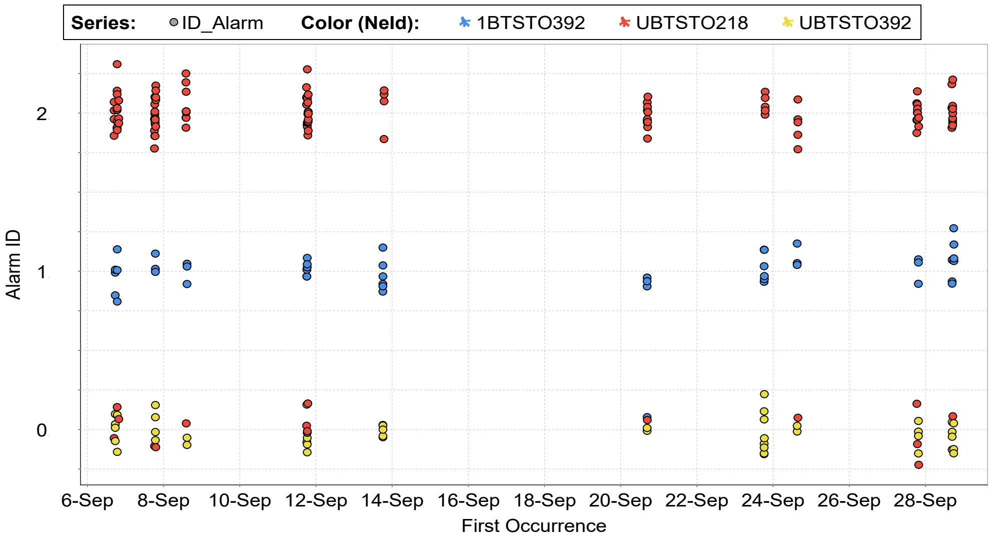


Figure 4.22: Scatter plot: Alarms categorized by alarm types over time in the fourth case

Figure shown in 4.23 represents the types of probable causes of each alarm involved in the rule. Label 1, 3, 4, 5, 6, 7, 8 and 23 stand for equipment malfunction, underlying resource unavailable, aIS, loss of frame, remote node transmission error, local node transmission error, synchronization mismatch, power problem probable causes respectively.

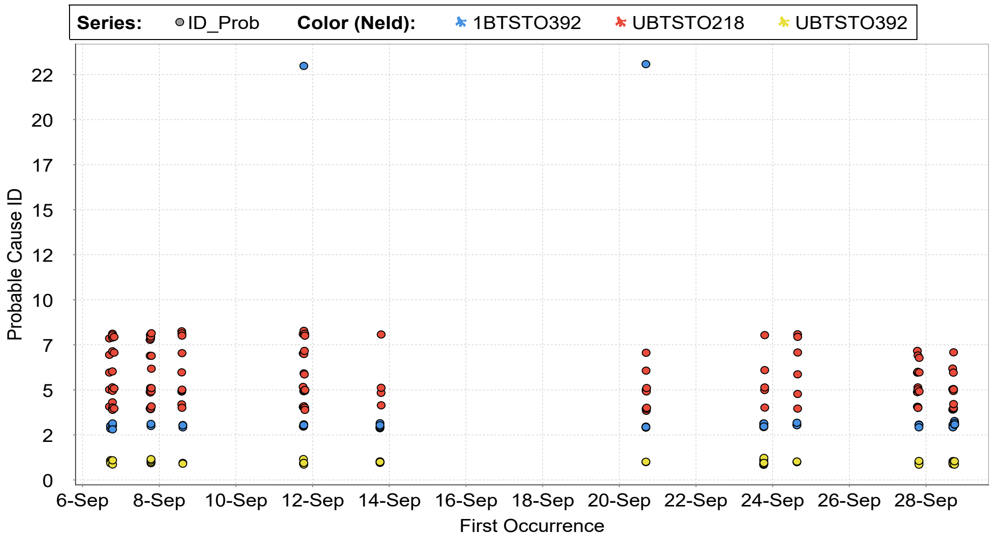


Figure 4.23: Scatter plot: Alarms categorized by probable cause over time in the fourth case

The correlation is validated by zooming on an optional day in 8th of September and looking for time bins of 2 hours as figure 4.24 suggests. Having said that, this rules is not reporting an unexpected interesting correlation among devices since one device is always generating alarms and the other two happen to raise alarms at the same time slots together with the first device.

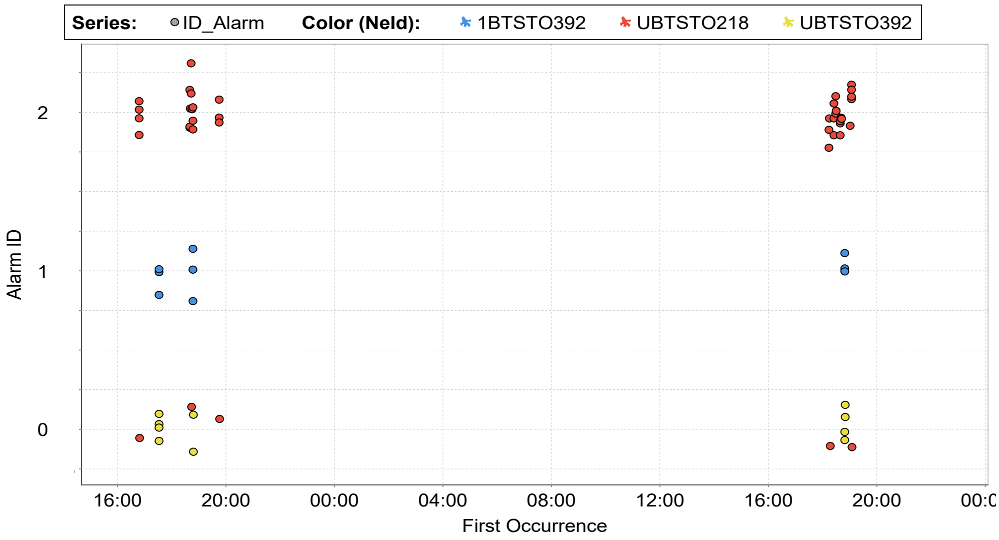


Figure 4.24: Zoomed Scatter plot: Alarms categorized by alarm type over time in the fourth case in 7th of September

Observation 4.4. With a comparison between both set of the rule sets we realize there are some new rules in September that did not appear in May. Moreover, we see no correlation in September among the devices of interest appeared in May (first and second case).

Chapter 5

Analysis of Separated Device Types

This chapter focuses on a more generic approach to find correlations among devices. Here, instead of focusing on each device itself, we view each device by its type. This allows us to drive methodologies with a general view of devices. Type of each device is identified by the first four letters of network device ID field. Table 5.1 describes possible combinations for device types.

Type	Technology and Device
GBSC	GSM, BSC
UBTS	UMTS, BTS
GBTS	GSM, BTS
9BTS	UMTS900, BTS
1BTS	LTE1800, BTS
8BTS	LTE800, BTS
URNC	UMTS, RNC
2BTS	LTE2600, BTS

Table 5.1: Possible combinations of device types

5.1 Overview of the Methodology with Probable Cause and Alarm Type for each Device Type

As a first pass over considering types of devices, we choose to also add features such as types of alarm and probable cause for each type to the matrix definition. By doing this, we will find more specific correlations among devices since the methodology contains more details. As we recall in the table , the first four letters of device network ID shows the type.

By considering a province, we assigned a unique identifier to each probable cause. Given these definitions, an example of the obtained matrix is shown at table 5.2. This indicates a sample transaction considered in a certain arbitrary time bin where number of alarms raised by device type GBSC which have a probable cause of type 0 are equal to 12 and number of alarms raised by device type GBSC which have an alarm type of type 0 are equal to 196.

GBSC_PC0	...	GBSC_PC26	...	GBSC_Alarmtype0	...	GBSC_Alarmtype4	UBTS_PC0	...
12	...	3	...	196	...	38	6	...

Table 5.2: Matrix of transactions and itemsets: considering each network device ID

5.1.1 Advantages and Disadvantages of methodology

Focusing on each device in the data, enabled us to exploit more specific co-occurrences which eventually allows us to take patterns obtained in a specific province and look for those exact recurring patterns in other provinces. However, there are number of potential weak points in the method that need to be considered. After obtaining the rules by this definition, we consider below example:

Antecedent	UBTS_unavailable 1BTS_communicationsAlarm UBTS_callEstablishmentError 8BTS_communicationsAlarm URNC_communicationsAlarm
Consequent	1BTS_unavailable URNC_lossOfSignal

We can observe a sort of **inflation** from the rules (e.g. the selected rule) which shows correlation due to one probable cause results in one alarm type or similar. This happens because in our matrix we have two ones for each alarm type and probable cause of the same device type. Second, based on 2.4 the distribution of alarms are not in the same range. Communications and quality of service alarms are the most popular so they are presented in most of the obtained rules with a high probability. Therefore, since alarm type has only few types, it would not be a suitable feature to be considered into the matrix because it does not really provide us with extra information.

5.2 Overview of the Methodology with Probable Cause for each Device Type

This approach only assumes probable cause feature for device types. To build the matrix of transactions and item sets, we assigned a unique identifier to each probable cause in different provinces. Applying the same procedure as before, an example of the obtained matrix is shown at table 5.3.

GBSC_PC0	...	GBSC_PC26	...	UBTS_PC0	...	UBTS_PC26	...
12	...	3	...	196	...	38	...

Table 5.3: Matrix of transactions and itemsets: considering each network device ID

By this definition, we will find out which device types were raising alarms at the same time bin more frequently. In this section, we will focus on Turin province to reduce complexity and study two datasets reported in two different month of May and September.

5.2.1 Rule Generation

The below thresholds are set for the datasets of May with two different time bins and also a time bin of 2 hours in September respectively. Other parameters are set to the default of Rapidminer.

- Dataset: Turin, May
- Minimum confidence: 0.7
- Lower bound for minimum support: 0.05
- Time bin: 2 hours

- Dataset: Turin, May
- Minimum confidence: 0.7
- Lower bound for minimum support: 0.05
- Time bin: 1 hours

.

- Dataset: Turin, September
- Minimum confidence: 0.9
- Lower bound for minimum support: 0.08
- Time bin: 2 hours

The output of Create Association Rules Operator gives a summary of rules and the information on total items mined, and the minimum parameters we set earlier. The number of rules obtained for May is 1910 in time bin of 2 hours whereas this is 160 in time bin of 1 hour (as time bin interval gets shorter, the rules become more specific). The total obtained rules for September is 827.

5.2.2 Rule selection and visualization

As before we select the most interesting rules to investigate. we will respectively choose examples from each dataset of May and September that are considered interesting due to their fairly high lift, long sequence and inevitably less support.

First Example

This rule suggest there is an association between device types of URNC and UBTS in time interval of two hours in May.

Antecedent	URNC_lossOfSignal UBTS_aIS
Consequent	UBTS_remoteNodeTransmissionError

As we can observe in the below table information on the interestingness of a the rule is reported as an output.

Confidence	Lift	Leverage	Conviction
0.98	2.03	0.05	10.38

The antecedent holds true for 40 time bins whereas this number for consequent is 39 times and this leads to a confidence of 98%. Temporal correlation is observable in figure 5.1 and 5.2. As observed in figure 5.1 the arbitrary unique ID of network devices with type UBTS (from 0-16) is separated from URNC (from 17-21).

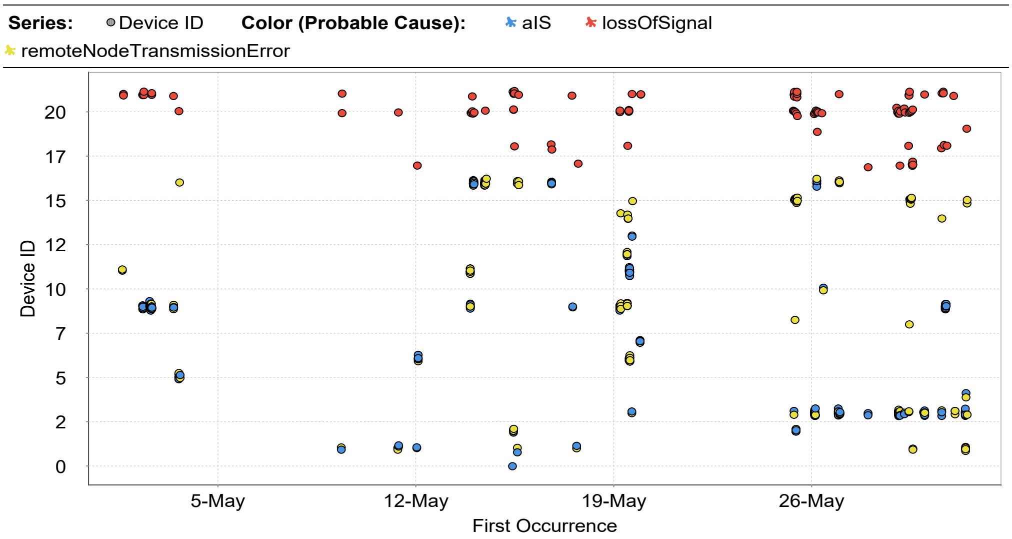


Figure 5.1: Scatter Plot: Device ID sorted by DeviceTypes vs First Occurrence

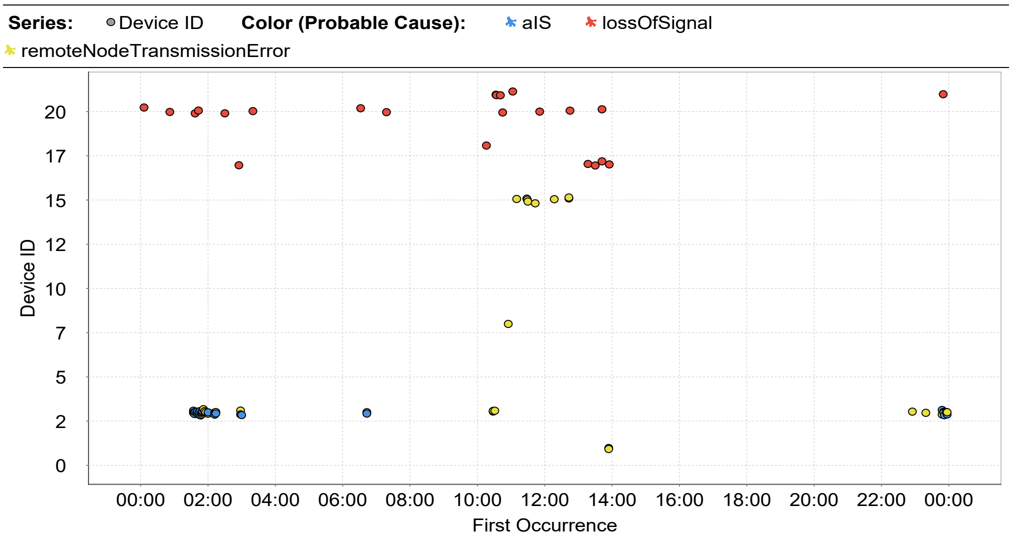


Figure 5.2: Zoomed Scatter Plot: Device ID sorted by DeviceTypes vs First Occurrence in 29th of May

Second Example

This rule suggest there is an association between device types of URNC and UBTS and 1BTS in time interval of two hours in May.

Antecedent	1BTS_equipmentMalfunction URNC_lossOfSignal 1BTS_indeterminate
Consequent	UBTS_unavailable

As we can observe in the below table information on the interestingness of a the rule is reported as an output.

Confidence	Lift	Leverage	Conviction
1	2.06	0.03	10.79

The antecedent holds true for 21 time bins whereas this number for consequent is 21 times and this leads to a confidence of 100%. Temporal correlation is observable in figure 5.3 and 5.4.

Third Example

As mentioned before, the matrix implementation is considered on time bins of 2 hours. Now let us see the rules obtained when time bin is changed to 1 hour. (740 time slots in total). Number of rules are decreased significantly to 160 rules in compared to previous assumption(2 hours) with the same parameters set for FP-growth operator.

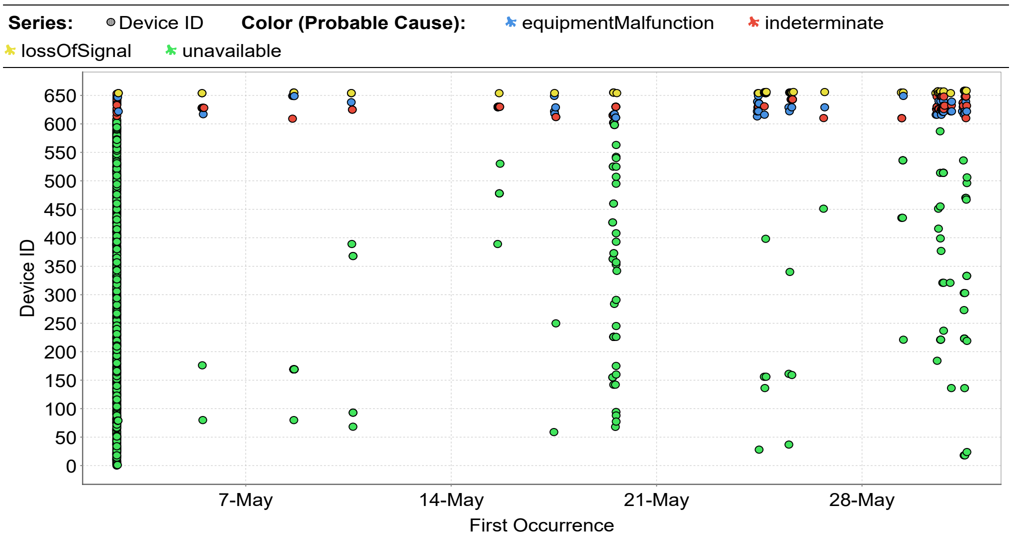


Figure 5.3: Scatter Plot: Device ID sorted by DeviceTypes vs First Occurrence

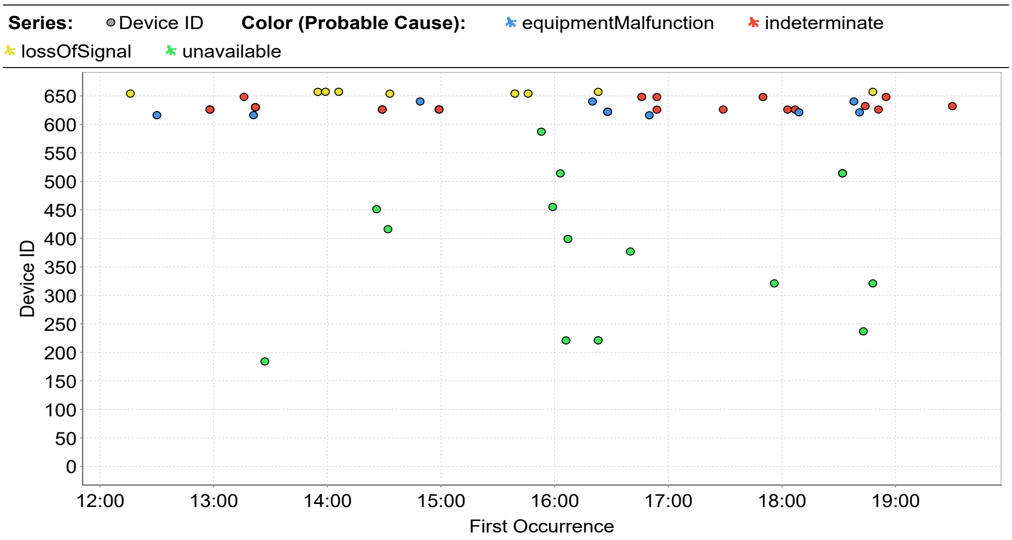


Figure 5.4: Zoomed Scatter Plot: Device ID sorted by DeviceTypes vs First Occurrence in 30^{th} of May

This rule focuses on association in the same device type with different probable causes time interval of one hour in May.

Antecedent	UBTS_M3100_synchronizationSourceMismatch UBTS_aIS
Consequent	UBTS_remoteNodeTransmissionError

As we can observe in the below table information on the interestingness of a the rule is reported as an output.

Confidence	Lift	Leverage	Conviction
0.98	2.92	0.05	17.91

The antecedent holds true for 54 time bins whereas this number for consequent is 53 times and this leads to a confidence of 98%. Temporal correlation is observable in figure 5.5 and 5.6.

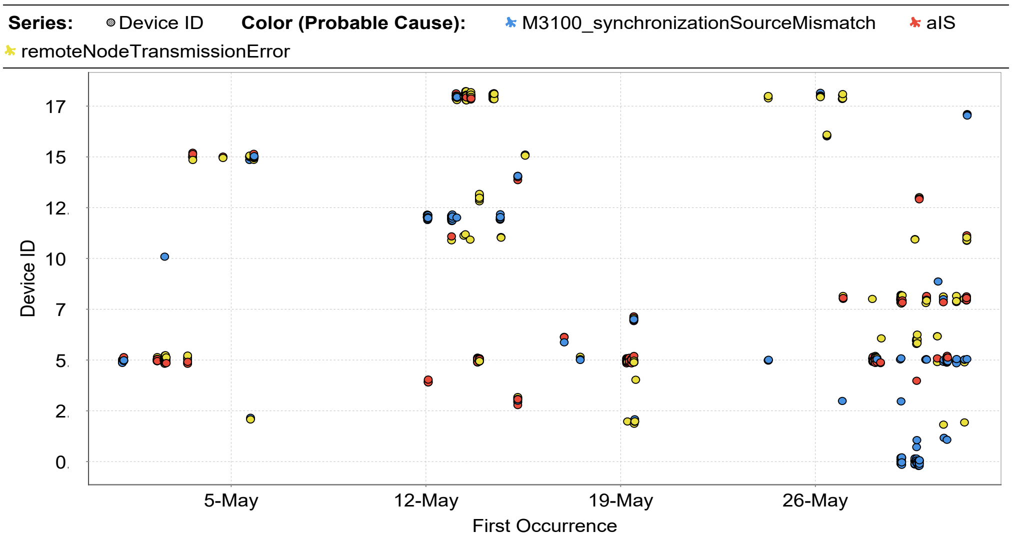


Figure 5.5: Scatter Plot: Device ID vs First Occurrence

Fourth Example

This rule focuses on association between UBTS and 9BTS device types in September.

Antecedent	UBTS_aIS UBTS_lossOfFrame 9BTS_equipmentMalfunction
Consequent	UBTS_remoteNodeTransmissionError

As we can observe in the below table information on the interestingness of a the rule is reported as an output.

Confidence	Lift	Leverage	Conviction
1	2.32	0.02	11.18

The antecedent holds true for 32 time bins whereas this number for consequent is 32 times and this leads to a confidence of 100%. Temporal correlation is observable in figure

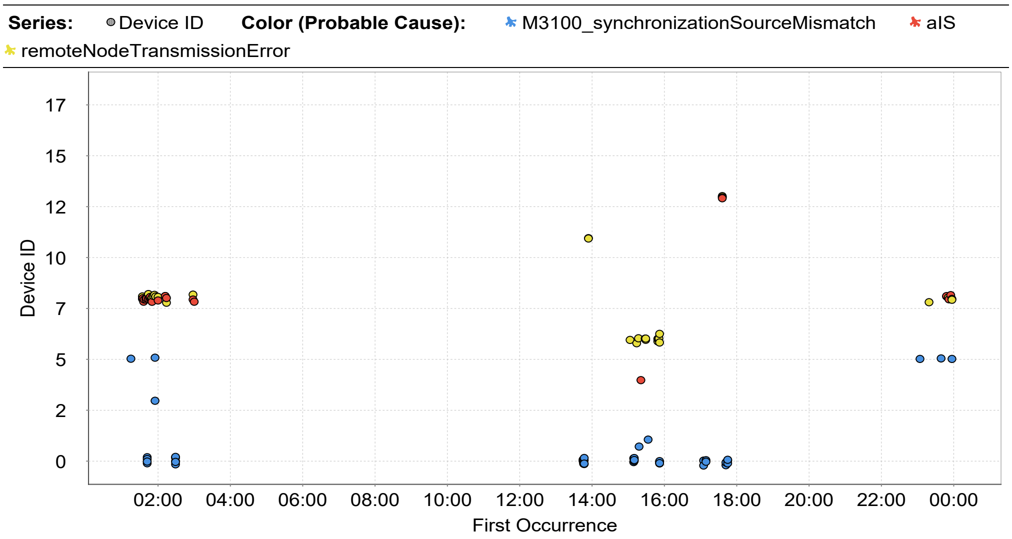


Figure 5.6: Zoomed Scatter Plot: Device ID vs First Occurrence in 29th of May

5.7 and 5.8.

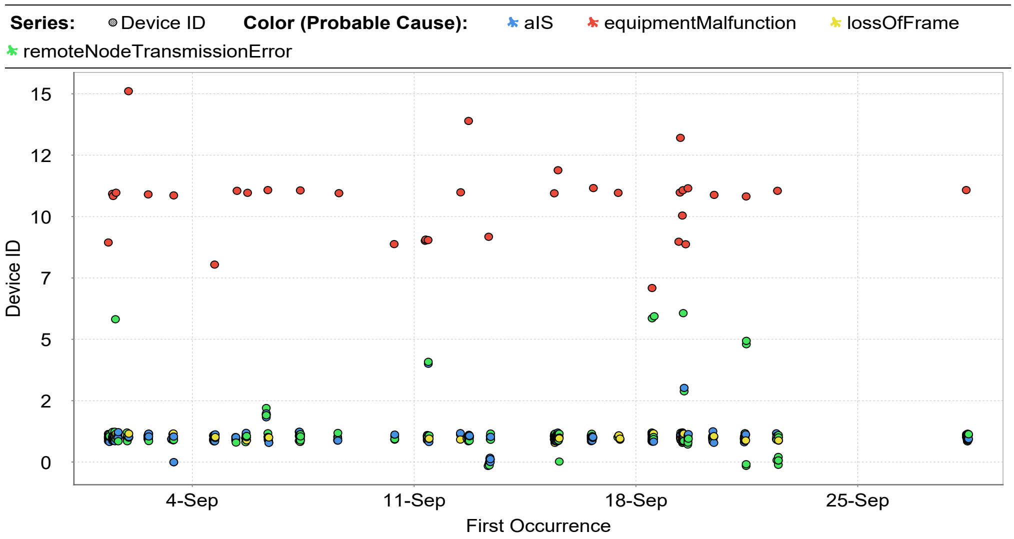


Figure 5.7: Scatter Plot: Device ID sorted by DeviceTypes vs First Occurrence

5.3 Overview of the Methodology with Geographical Space Division

In the next approach, we tried to generalize the definition of matrix by concentrating on spatial correlation among device types as well as the temporal one.

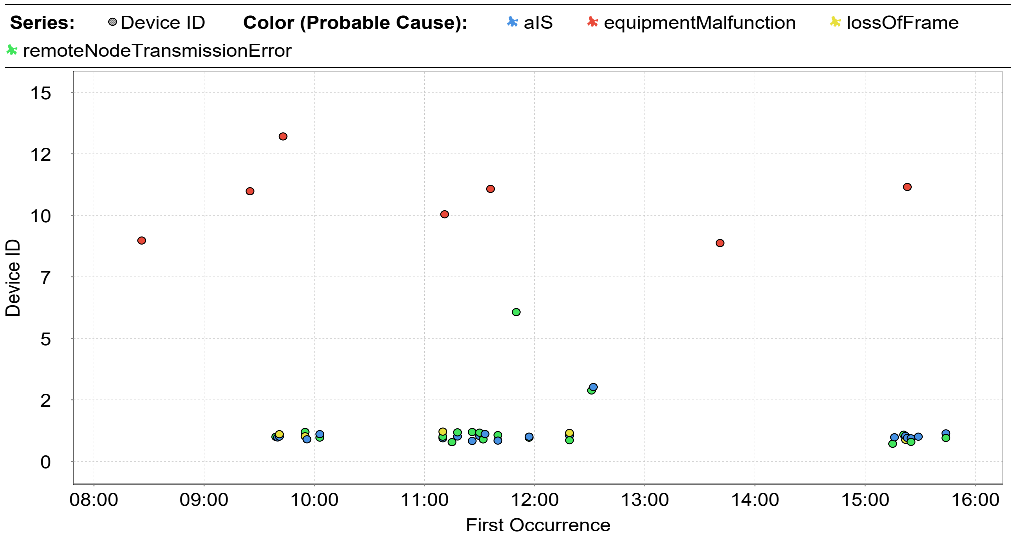


Figure 5.8: Zoomed Scatter Plot: Device ID sorted by DeviceTypes vs First Occurrence in 19th of September

In the interest of spatial correlation, we should first divide the province into separate zones which can be done by different clustering algorithms. However, one possible question is how to define the boundaries? We used DB-scan and K-means algorithms to cluster a specific province such as Turin and then experimented both algorithms with different parameters. We first considered DB-scan since it does not require number of clusters beforehand. Let us see the results which DB-scan suggests in figure 5.9.

The biggest cluster contains 35959 points out of 38062 even though there are 30 clusters found. So, unfortunately, this algorithm does not help much in adding space limitations. Figure 5.10, instead, shows us the network devices in Turin province clustered by k-means algorithm. This algorithm requires the number of clusters before performing. By choosing k=20, the algorithm creates boundaries where there is no need. We tried other values for this parameter and eventually as shown in figure 5.11 k=5 seems to be a good fit.

We then applied k-means for Milan province, too. Clustering is again done by some experiments on the k value and eventually we realized k=7 as figure 5.12 shows, is a good choice.

We have also choose to consider a column called "specific problem" to our transactional data. This is because the most two frequent types of probable cause features based on 2.3 are "indeterminate" and "unavailable". Specific problem is a more detailed field of probable cause. For example, there are a lot of specific problem types for "indeterminate" probable cause. Therefore, we should insert a threshold as shown in the tables 5.4 and 5.5 : (>5000) for indeterminate and (>2000) for unavailable.

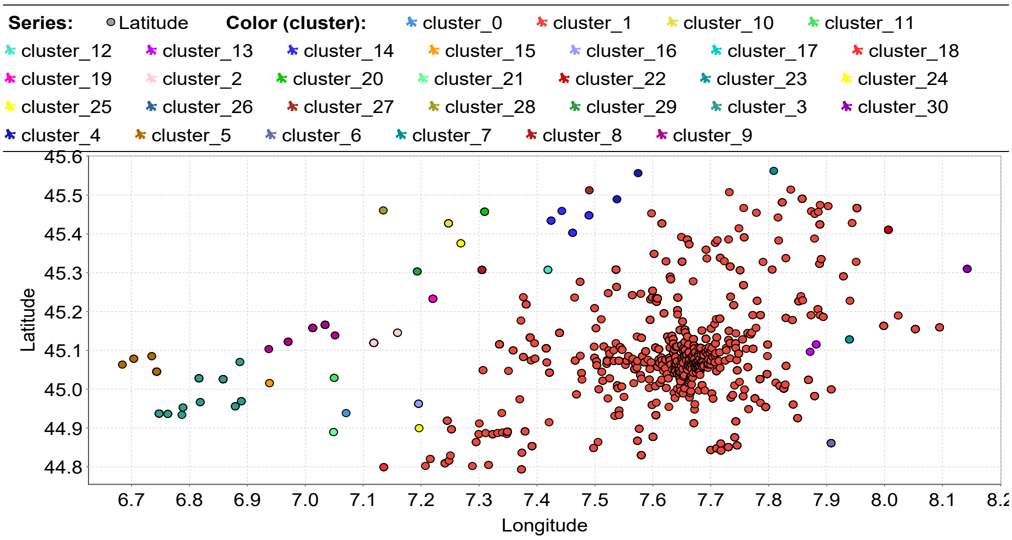


Figure 5.9: Turin province clustered by DB-scan algorithm

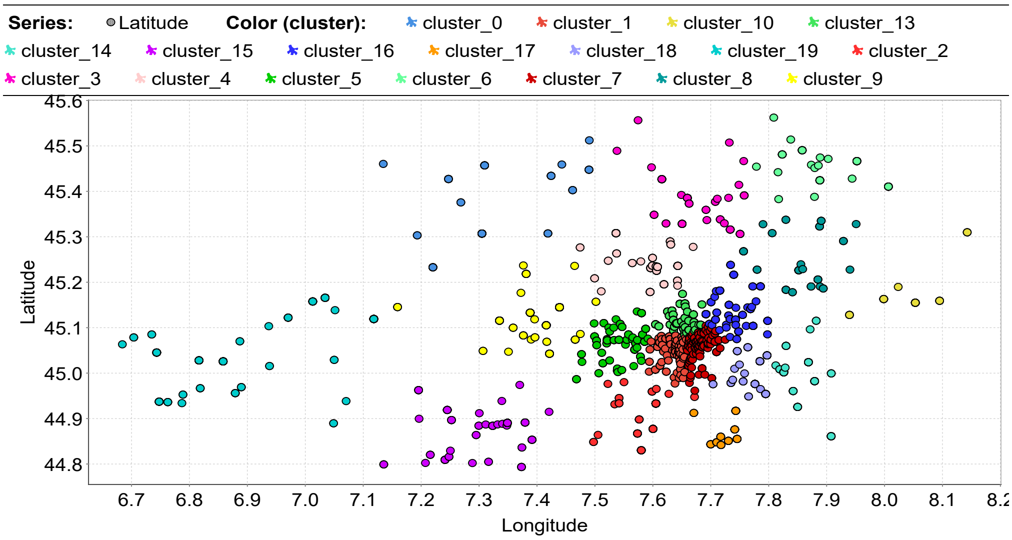


Figure 5.10: Turin province clustered by k-means algorithm with k=20

5.3.1 Rule Generation

The below thresholds are set for the datasets of May and a time bin of 2 hours. Other parameters are set to the default of Rapidminer.

- Dataset: Turin, May
- Minimum confidence: 0.7
- Lower bound for minimum support: 0.01

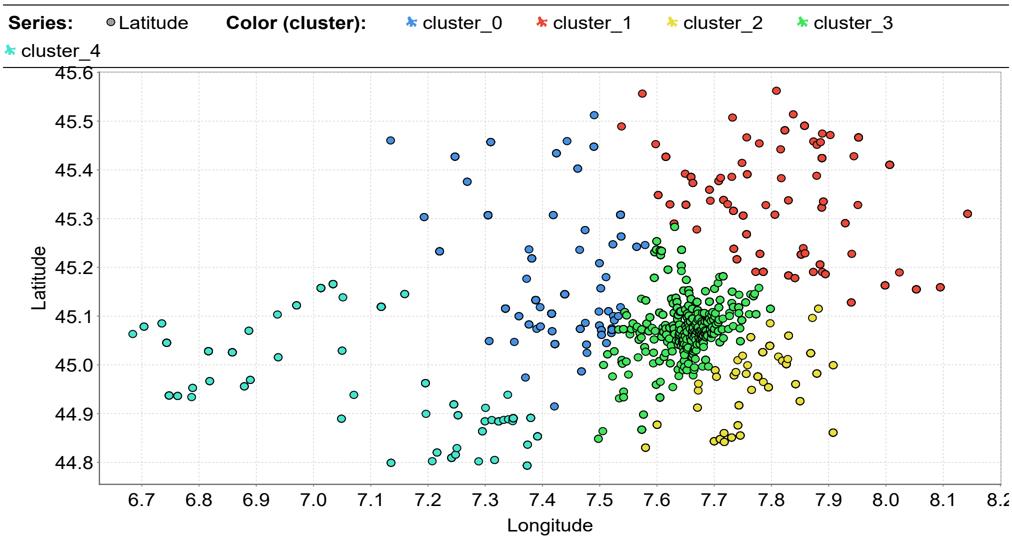


Figure 5.11: Turin province clustered by k-means algorithm with k=5

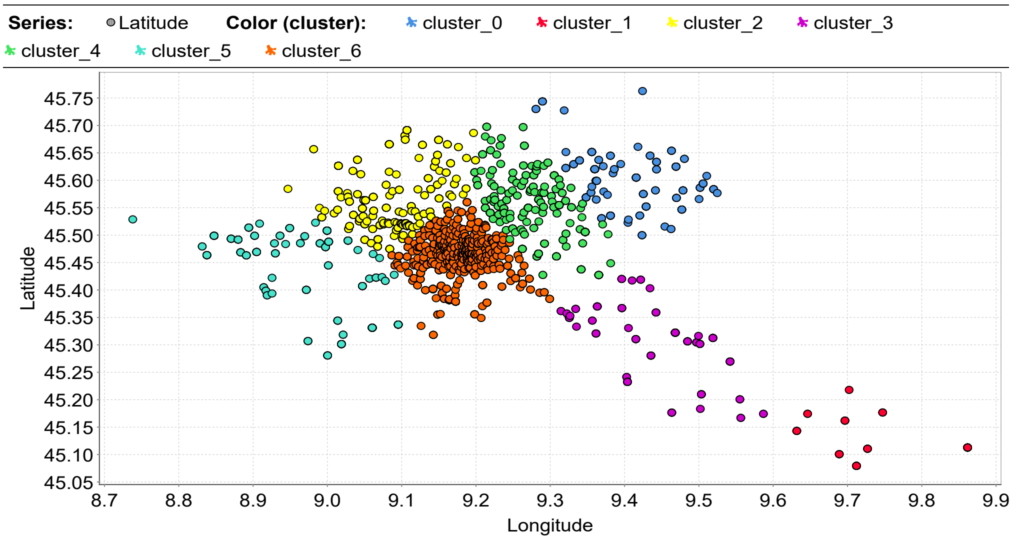


Figure 5.12: Milan province clustered by k-means algorithm with k=7

- Number of clusters: 5

The output of Create Association Rules Operator gives a summary of rules and the information on total items mined, and the minimum parameters we set earlier.

Probable–Cause	**Frequency**
BTS with no transactions	164627
Cell operation degraded	74399
UtranCell_ServiceUnavailable	27820
Cell logical channel availability supervision	15880
BCCH missing	15384
Cell faulty	8946
Data output, Ap transmission fault	7927
Wcdma cell out of use	6760
Synchronization lost	6055
CH congestion in cell above defined threshold	5992
Pcm failure	5543
Fault rate monitoring	5376

Table 5.4: Total Types and frequencies of "indeterminate" based on specific_problem field

Probable–Cause	**Frequency**
Heartbeat Failure	26071
NTP Server Reachability Fault	24830
Contact to Default Router 1 Lost	24355
Contact to Default Router 0 Lost	22569
PLMN Service Unavailable	12274
Remote IP Address Unreachable	2893

Table 5.5: Total Types and frequencies of "unavailable" based on specific_problem field

5.3.2 Rule selection and visualization

As before we select the most interesting rules to investigate. we will respectively choose examples from the dataset that are considered interesting due to their fairly high lift, long sequence and inevitably less support.

First Example in Turin

Antecedent	UBTS_equipmentMalfunction
	GBSC_indeterminate(Data output, Ap transmission fault)
	UBTS_indeterminate(UtranCell_ServiceUnavailable)
	8BTS_unavailable(Heartbeat Failure)

Consequent	GBTS_indeterminate(Cell logical channel availability supervision) UBTS_unavailable(Heartbeat Failure) URNC_lossOfSignal

As we can observe in the below table information on the interestingness of a the rule is reported as an output.

Confidence	Lift	Leverage	Conviction
0.8	35.03	0.01	4.07

The antecedent holds true for 25 time bins whereas this number for consequent is 20 times and this leads to a confidence of 80%.

In this example all the devices are located in the same cluster (center of Turin). Geographical visualization of these devices is shown in figure 5.15. Temporal correlation is observable in figure 5.13 and 5.14.

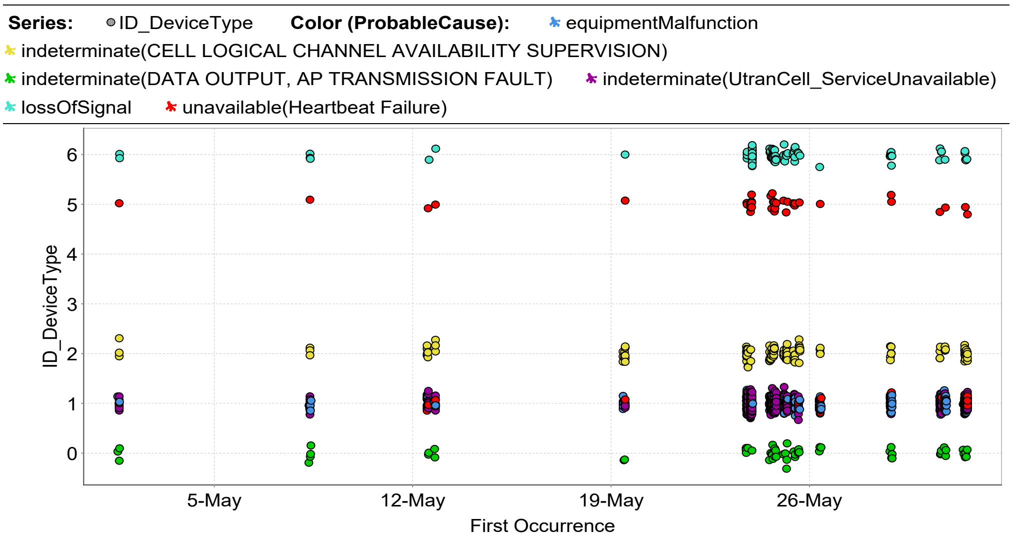


Figure 5.13: Scatter Plot: DeviceTypes vs First Occurrence in the First Example in Turin

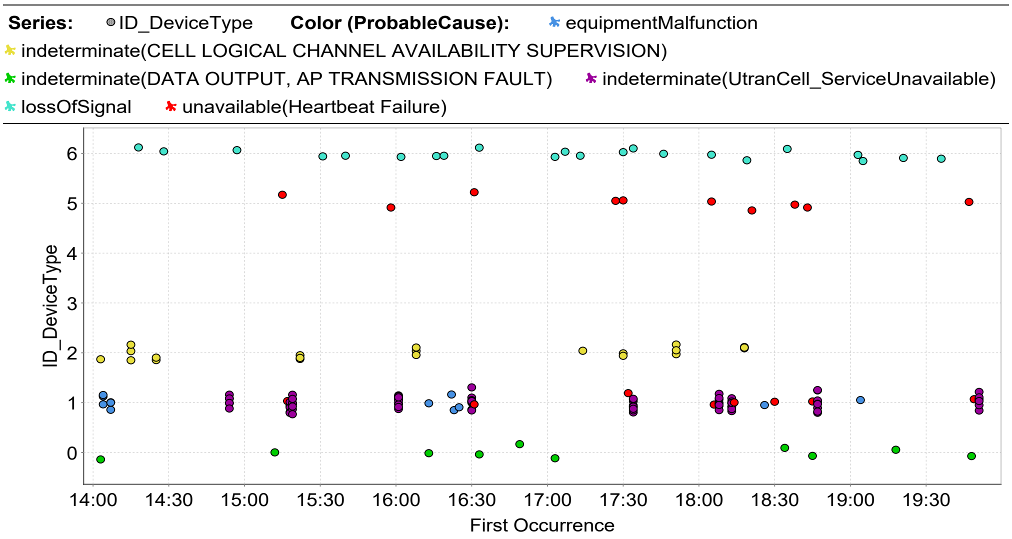


Figure 5.14: Zoomed Scatter Plot: DeviceTypes vs First Occurrence in 24th of May in the First Example in Turin

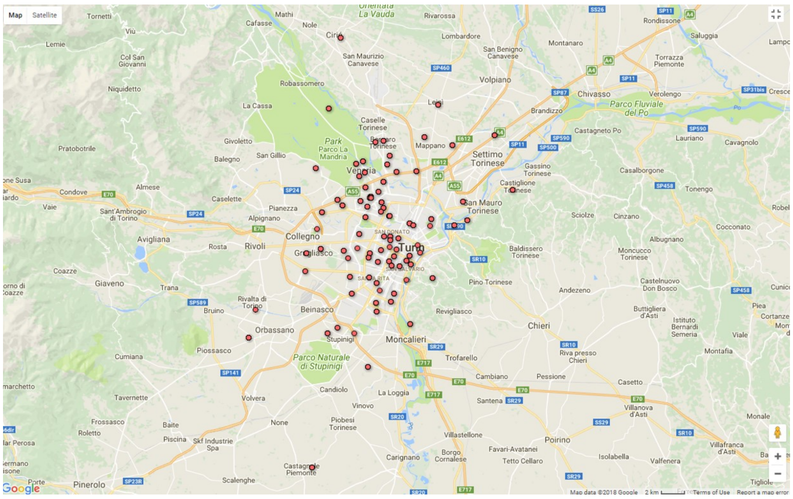


Figure 5.15: Geographical Visualization of Network Devices Involved in the First Example in Turin

Second Example in Turin

Antecedent	UBTS_unavailable(Heartbeat Failure)
	UBTS_unavailable(NTP Server Reachability Fault)
	8BTS_unavailable(PLMN Service Unavailable)
	8BTS_unavailable(Heartbeat Failure)

Consequent	UBTS_indeterminate(UtranCell_ServiceUnavailable)
	8BTS_unavailable(Heartbeat Failure)
	8BTS_unavailable(NTP Server Reachability Fault)

As we can observe in the below table information on the interestingness of a the rule is reported as an output.

Confidence	Lift	Leverage	Conviction
0.95	73.18	0.01	10.86

The antecedent holds true for 22 time bins whereas this number for consequent is 21 times and this leads to a confidence of 95%.

In this example devices are located in the different clusters. Geographical visualization of these devices is shown in figure 5.18. In this case, 101 network devices are located in the five different clusters which is assumed for Turin province and are involved in this example. Temporal correlation is observable in figure 5.16 and 5.17.

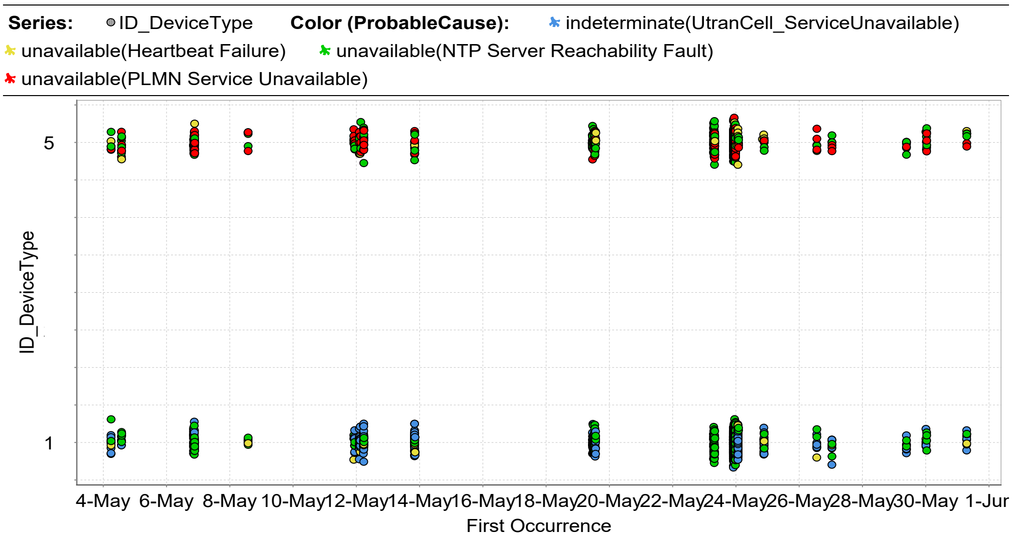


Figure 5.16: Scatter Plot: DeviceTypes vs First Occurrence in the Second Example in Turin

Third Example in Turin

Antecedent	UBTS_equipmentMalfunction
	UBTS_indeterminate(UtranCell_ServiceUnavailable)
	1BTS_unavailable(Heartbeat Failure)

Consequent	GBSC_indeterminate(Data output, Ap transmission fault)
	UBTS_unavailable(Heartbeat Failure)

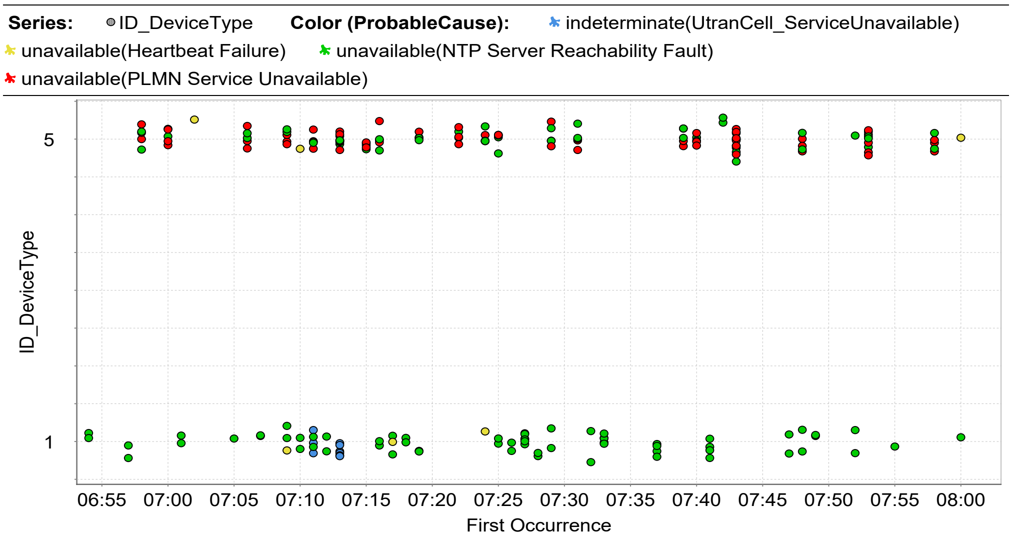


Figure 5.17: Zoomed Scatter Plot: DeviceTypes vs First Occurrence in 23$^{\text{rd}}$ of May in the Second Example in Turin

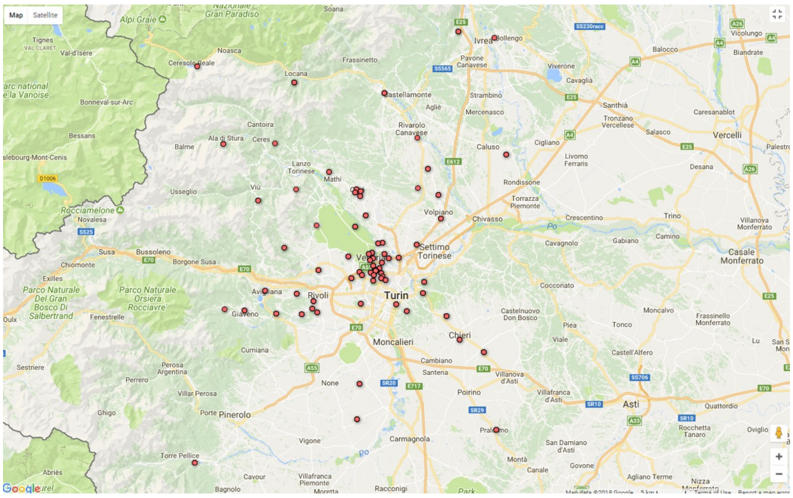


Figure 5.18: Geographical Visualization of Network Devices Involved in the Second Example in Turin

As we can observe in the below table information on the interestingness of a the rule is reported as an output.

Confidence	Lift	Leverage	Conviction
0.82	13.96	0.03	4.78

The antecedent holds true for 61 time bins whereas this number for consequent is 50 times and this leads to a confidence of 95%.

In this example all the devices are located in same clusters. Geographical visualization of these devices is shown in figure 5.21. Temporal correlation is observable in figure 5.19 and 5.20.

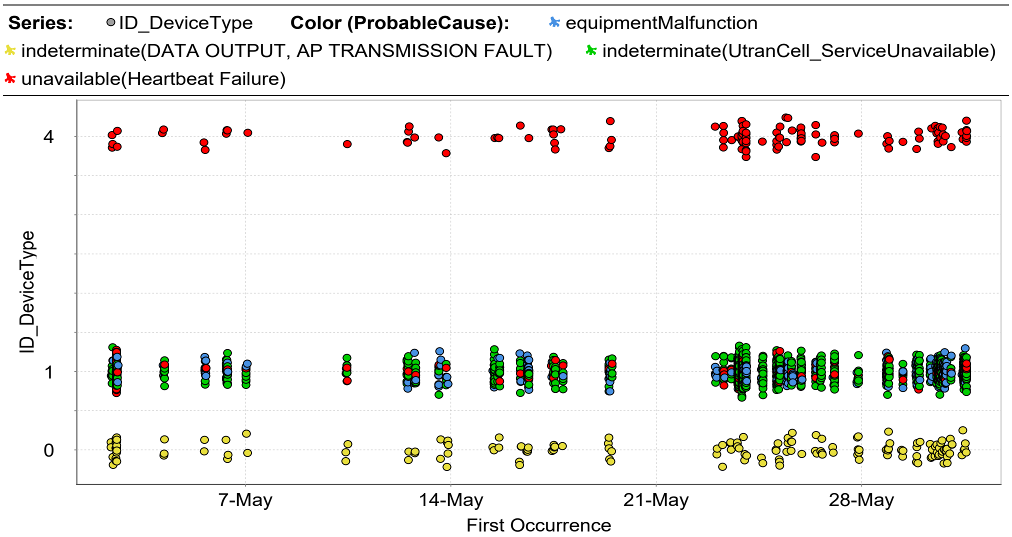


Figure 5.19: Scatter Plot: DeviceTypes vs First Occurrence in the Third Example in Turin

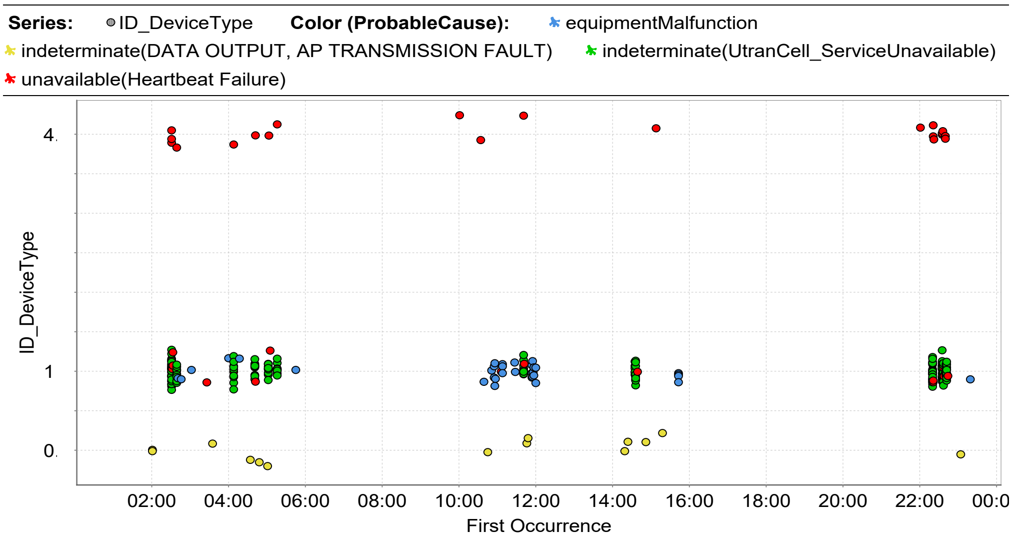


Figure 5.20: Zoomed Scatter Plot: DeviceTypes vs First Occurrence in 25th of May in the Third Example in Turin

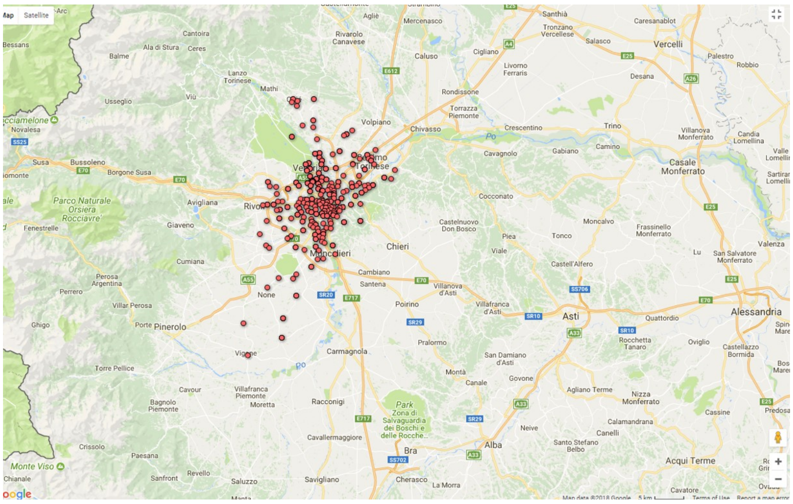


Figure 5.21: Geographical Visualization of Network Devices Involved in the Third Example in Turin

5.3.3 Rule Generation

Now that we have seen the association rules in Turin, it is useful to also search another province close to Turin. We choose Milan for this reason and the goal here is to investigate those exact same examples in Milan to see whether they are hold true in Milan. The below thresholds are set for the datasets of May and a time bin of 2 hours. Other parameters are set to the default of Rapidminer.

- Dataset: Milan, May
- Minimum confidence: 0.7
- Lower bound for minimum support: 0.01
- Number of clusters: 7

The results show that previous cases found in Turin are actually sub-patterns in Milan. However, the support for some of these rules are lower than min-support set (lower than 1%).

5.3.4 Rule visualization

The following are showing the three examples we have studied in Milan Province.

First Example in Milan

We found 11 time bins where these 7 items are held (with support=0.004 , confidence=40%) in Milan. However, we should remind the reader that it would not be identified as a "rule" in the Milan data set since the support would be lower than 0.01 which is the minimum threshold for support.

In this case, only devices located in cluster 1 and 2 (center of Milan) are involved. Geographical visualization of these devices is shown in figure 5.24. Temporal correlation is observable in figure 5.22 and 5.23.

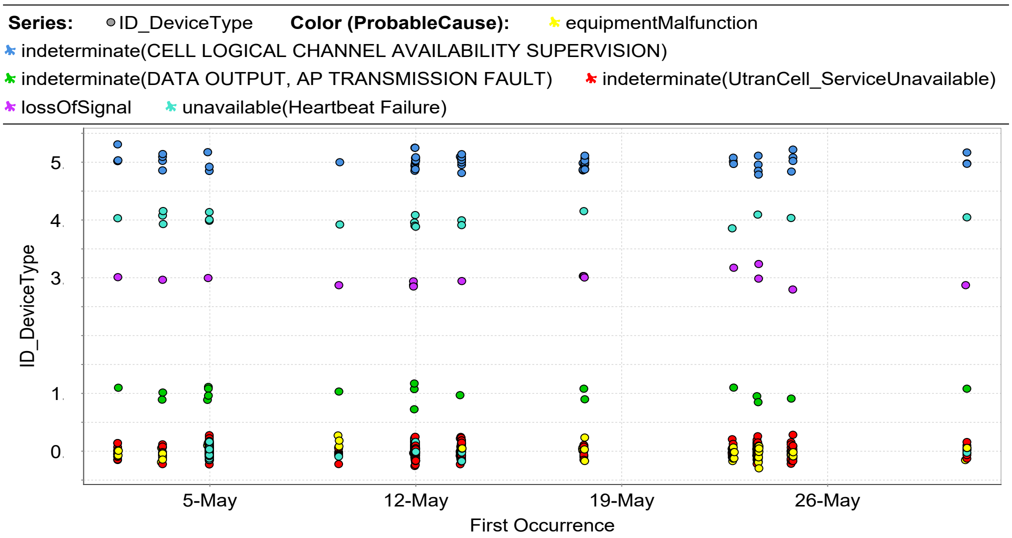


Figure 5.22: Scatter Plot: DeviceTypes vs First Occurrence in the First Example in Milan

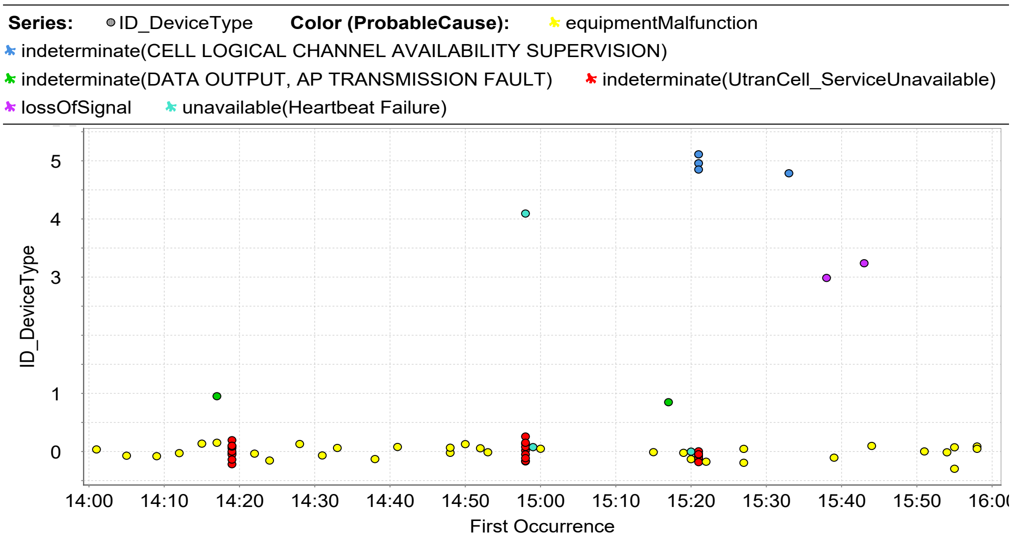


Figure 5.23: Zoomed Scatter Plot: DeviceTypes vs First Occurrence in 23rd of May in the First Example in Milan

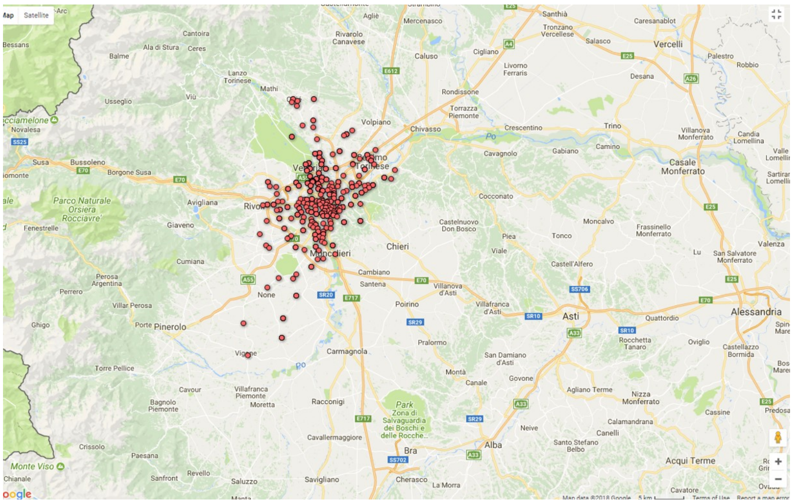


Figure 5.24: Geographical Visualization of Network Devices Involved in the First Example in Milan

Second Example in Milan

This rule suggests 6 bins in which the rule is held true (support=0.002 , confidence=0.5) for Milan. In this case, four clusters are involved. Geographical visualization of these devices is shown in figure 5.27. Temporal correlation is observable in figure 5.25 and 5.26.

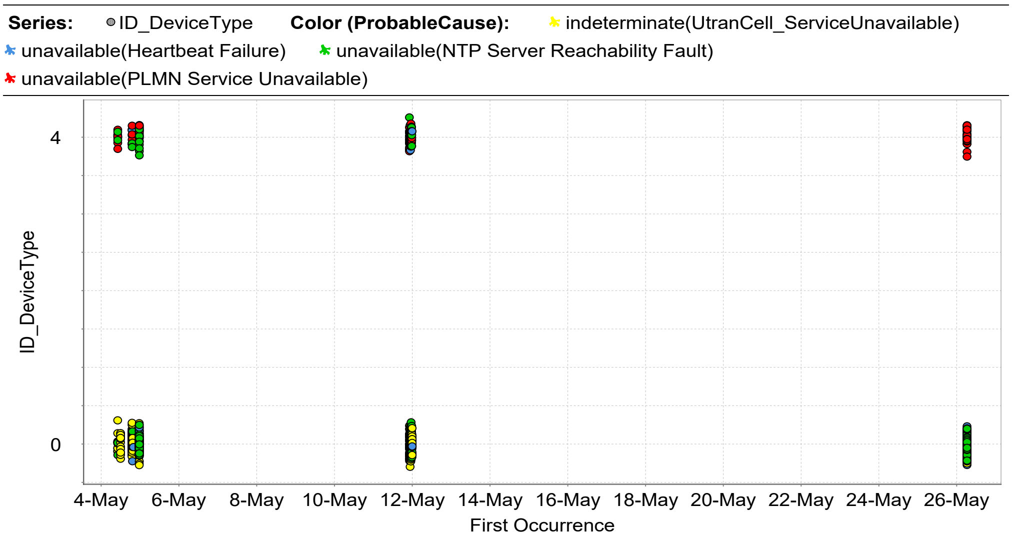


Figure 5.25: Scatter Plot: DeviceTypes vs First Occurrence in the Second Example in Milan

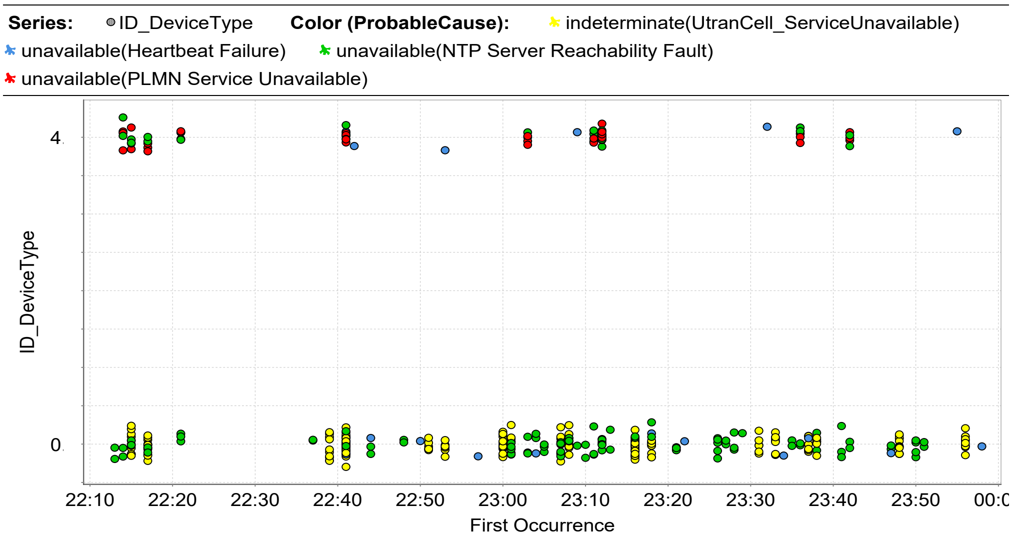


Figure 5.26: Zoomed Scatter Plot: DeviceTypes vs First Occurrence in 11th of May in the Second Example in Milan

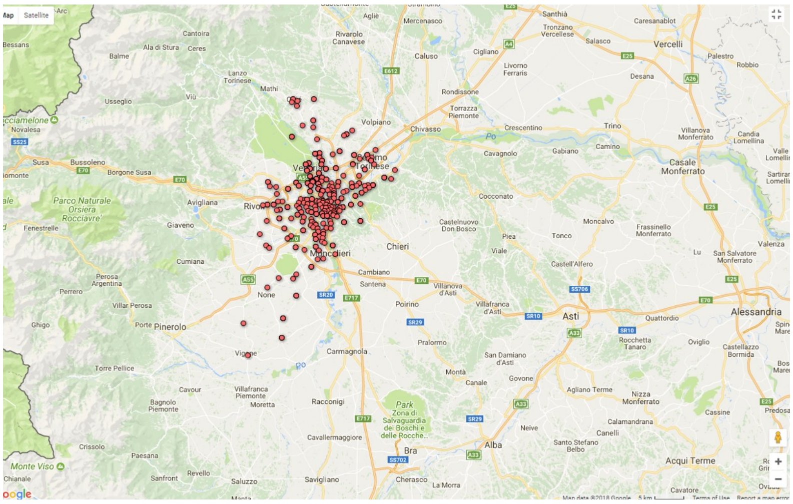


Figure 5.27: Geographical Visualization of Network Devices Involved in the Second Example in Milan

Third Example in Milan

This rule has been found 34 bins. The for this rule is support higher than 1% and moreover the confidence is equal to 0.58. If we set the value for minimum confidence lower, both of these mentioned values will be higher than the minimum support and confidence. Therefore this example would be found a rule also in Milan.

In this case, four clusters are involved. Geographical visualization of these devices is shown in figure 5.30. Temporal correlation is observable in figure 5.28 and 5.29.

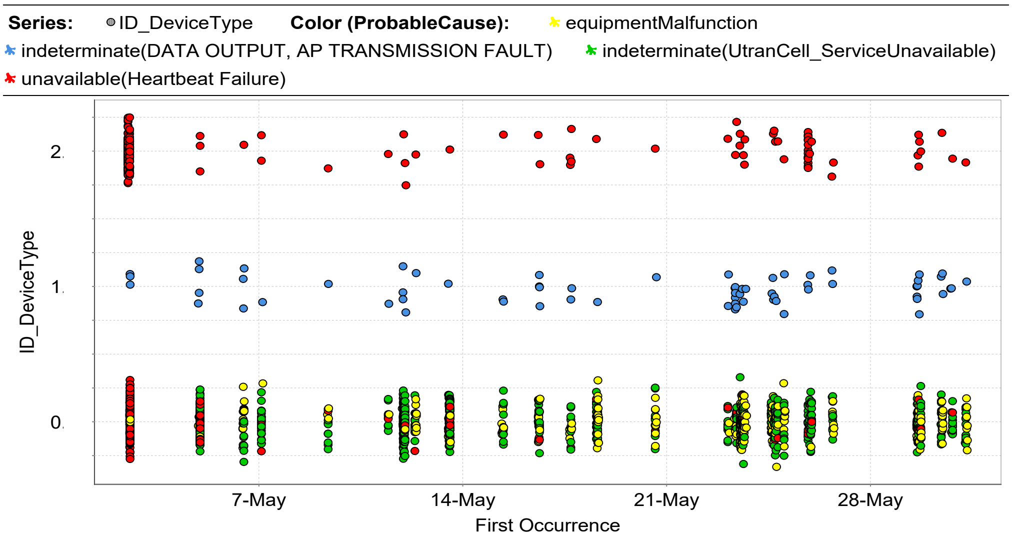


Figure 5.28: Scatter Plot: DeviceTypes vs First Occurrence in the Third Example in Milan

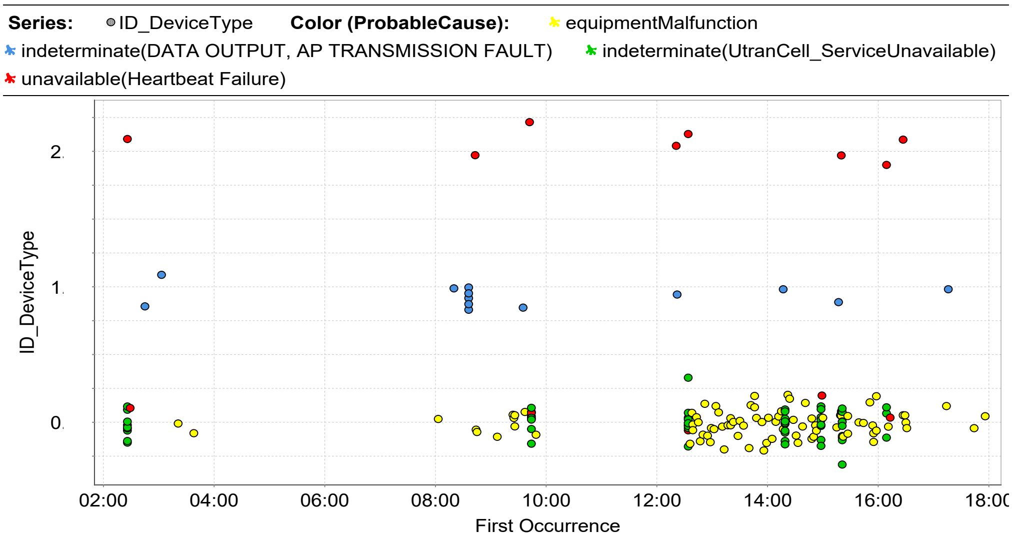


Figure 5.29: Zoomed Scatter Plot: DeviceTypes vs First Occurrence in 23rd of May in the Third Example in Milan

5.4 Mutual Rules

Previously, we showed the process of acquiring a suitable definition for the matrix of transaction and itemsets. We observed that the genesis of this idea is based on the most important features in our data set. Since in this way rules are more meaningful. Turning now to focus on those mutual rules that are held among different provinces and months, can make the results more remarkable. The concept is to start from rule set of one arbitrary

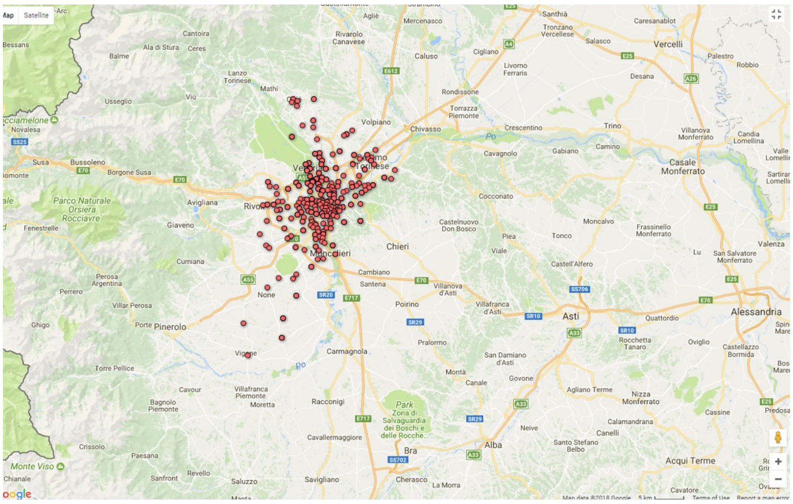


Figure 5.30: Geographical Visualization of Network Devices Involved in the Third Example in Milan

province such as Turin which is obtained in an arbitrary month such as May and search for all mutual rules within transaction sets of the other provinces in month of May and September.

As mentioned before, in order to explore mutual rules more efficiently, we should order them by measures of interestingness as lift, support, and confidence. Among 70 mutual rules of these datasets we select two of more interesting ones based on Turin to investigate and then continue to do the same for other data sets.

5.4.1 First Mutual Rule in Turin data set in May

We selected this rule based on Turin data set in May.

Antecedent	UBTS_equipmentMalfunction
	GBTS_indeterminate(Cell Logical Channel Availability Supervision)
	GBSC_indeterminate(Data Output AP Transmission Fault)
	UBTS_indeterminate(UtranCell_ServiceUnavailable)

Consequent	UBTS_unavailable(Heartbeat Failure)

As we can observe in the below table information on the interestingness of a the rule is reported as an output.

Confidence	Lift	Support
0.9	7	3%

Figure 5.31 and 5.32 details the temporal visualization of devices types for each device involved in the mentioned rule during month of May in province of Turin. It confirms a correlation between base transceiver station working on UMTS technology with those working on GSM. However, it seems this correlation is kind of peculiar because it is due to the failure of specific BSC devices that are same for month of May and September.

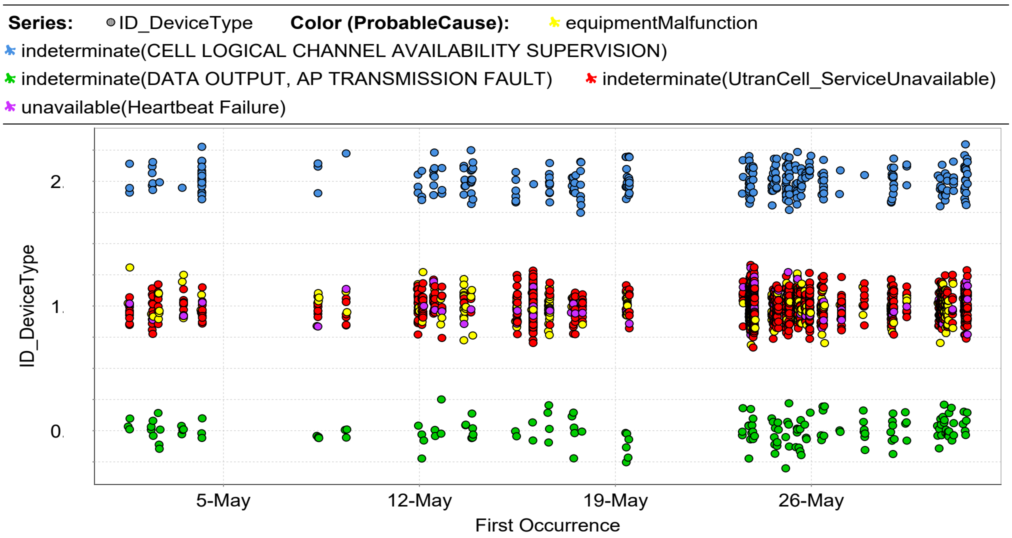


Figure 5.31: Scatter Plot: DeviceTypes vs First Occurrence in the irst Mutual Rule in Turin in May

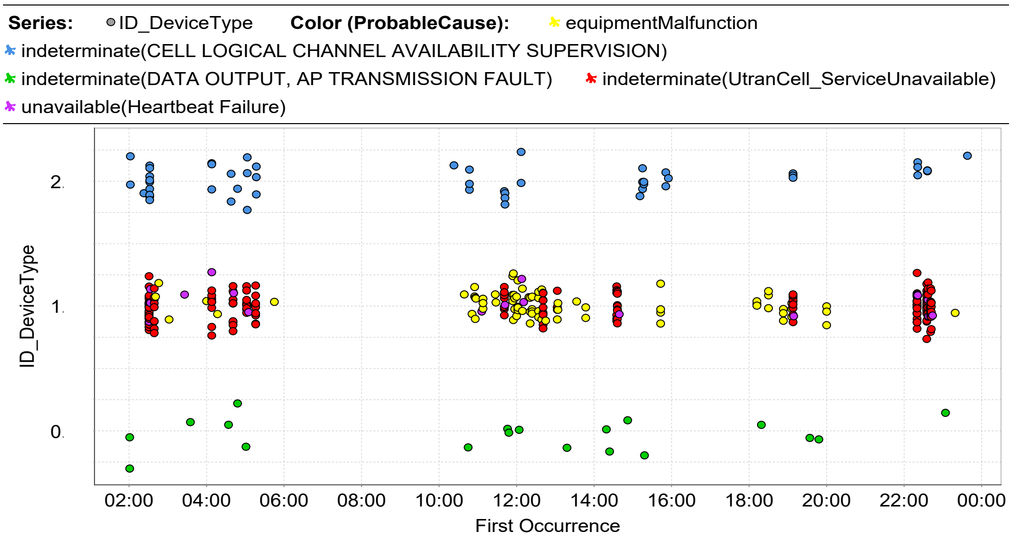


Figure 5.32: Zoomed Scatter Plot: DeviceTypes vs First Occurrence in 25th of May in the First Mutual Rule in Turin

5.4.2 First Mutual Rule in Milan data set in May

As we can observe in the below table information on the interestingness of a the rule is reported as an output.

Confidence	Lift	Support
0.9	6	3%

Figure 5.33 and 5.34 details the temporal visualization of devices types for each device involved in the mentioned rule during month of May in province of Milan.

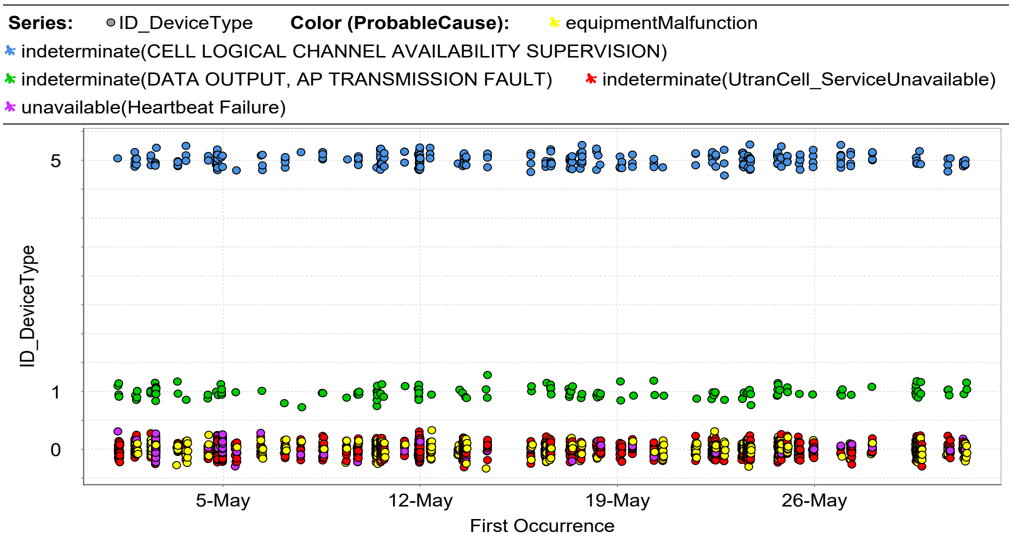


Figure 5.33: Scatter Plot: DeviceTypes vs First Occurrence in the First Mutual Rule in Milan in May

5.4.3 First Mutual Rule in Turin data set in September

As we can observe in the below table information on the interestingness of a the rule is reported as an output.

Confidence	Lift	Support
0.9	13.9	1%

Figure 5.35 and 5.36 details the temporal visualization of devices types for each device involved in the mentioned rule during month of September in province of Turin.

5.4.4 First Mutual Rule in Milan data set in September

As we can observe in the below table information on the interestingness of a the rule is reported as an output.

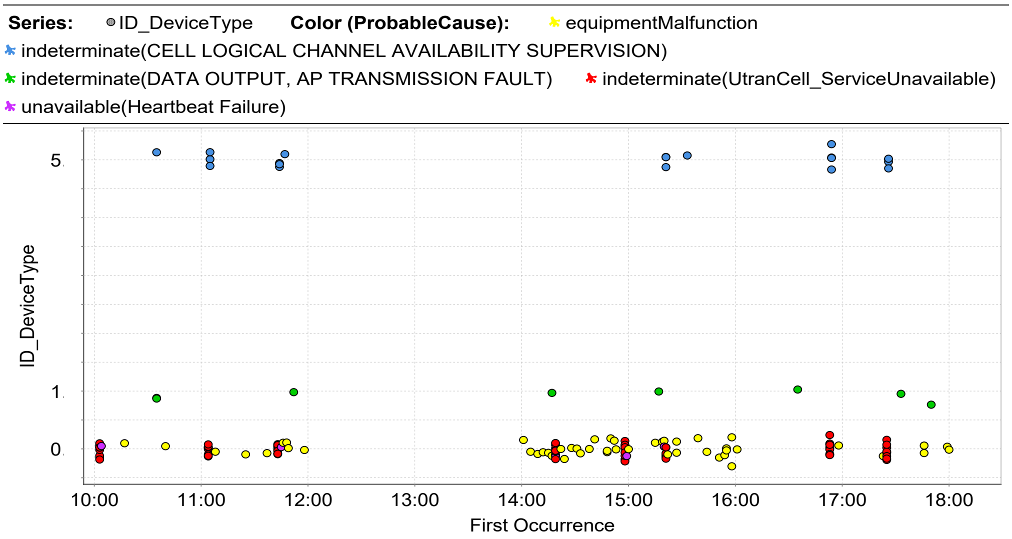


Figure 5.34: Zoomed Scatter Plot: DeviceTypes vs First Occurrence in 23$^{\mathrm{rd}}$ of May in the First Mutual Rule in Milan

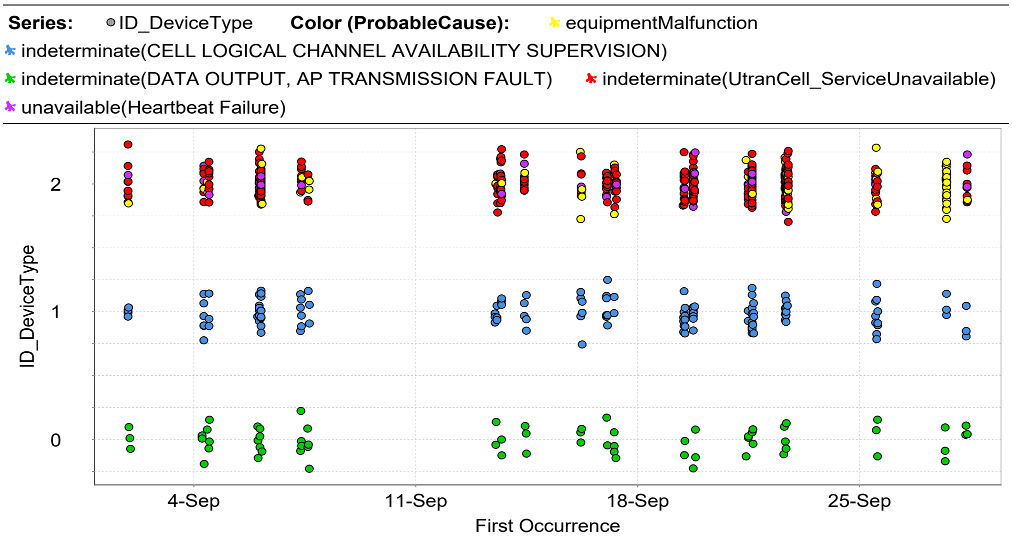


Figure 5.35: Scatter Plot: DeviceTypes vs First Occurrence in the irst Mutual Rule in Turin in September

Confidence	Lift	Support
0.9	9.9	1%

Figure 5.37 and 5.38 details the temporal visualization of devices types for each device involved in the mentioned rule during month of September in province of Milan.

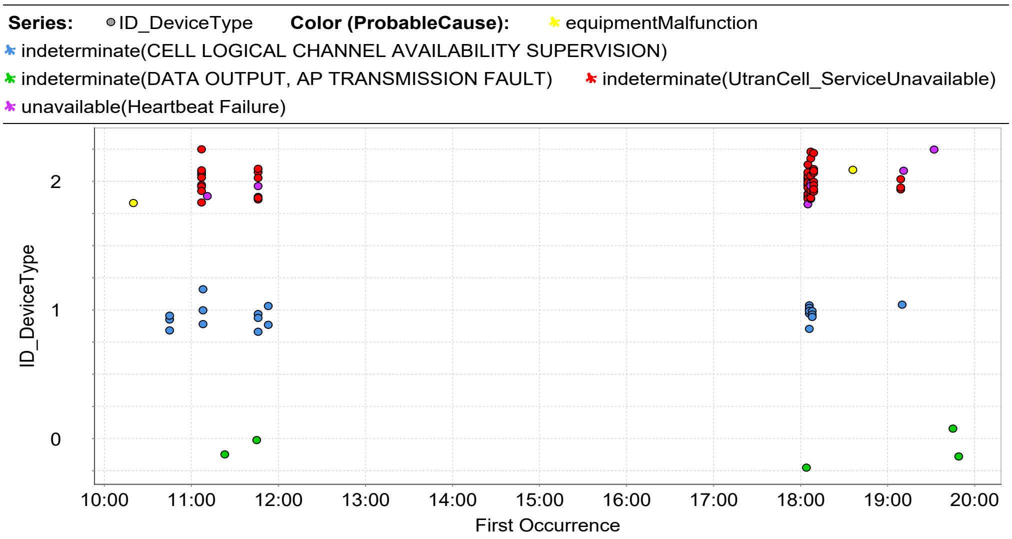


Figure 5.36: Zoomed Scatter Plot: DeviceTypes vs First Occurrence in 25th of May in the First Mutual Rule in Turin

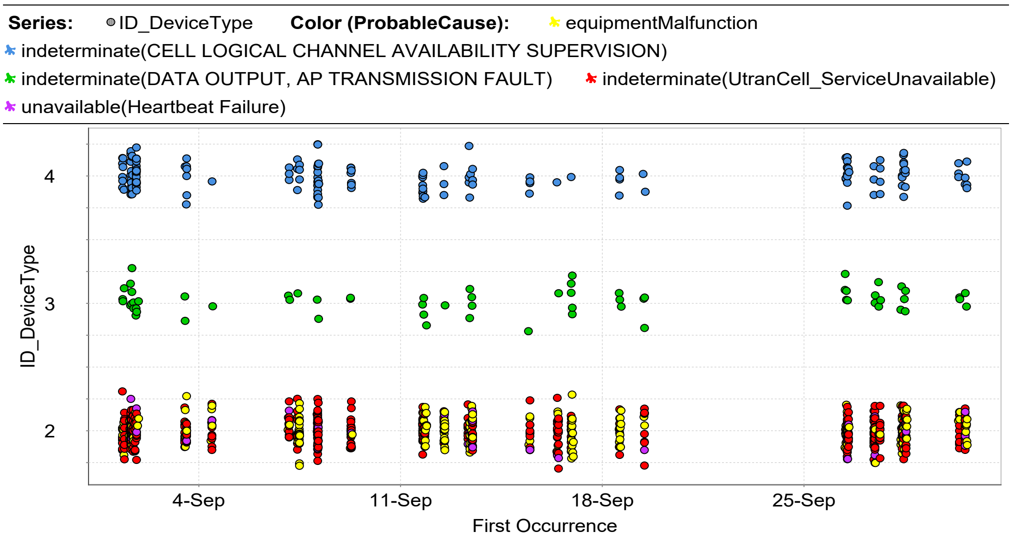


Figure 5.37: Scatter Plot: DeviceTypes vs First Occurrence in the irst Mutual Rule in Milan in September

5.4.5 Second Mutual Rule in Turin data set in May

As we can observe in the below table information on the interestingness of a the rule is reported as an output.

Figure 5.39 and 5.40 details the temporal visualization of devices types for each device involved in the mentioned rule during month of May in province of Turin.

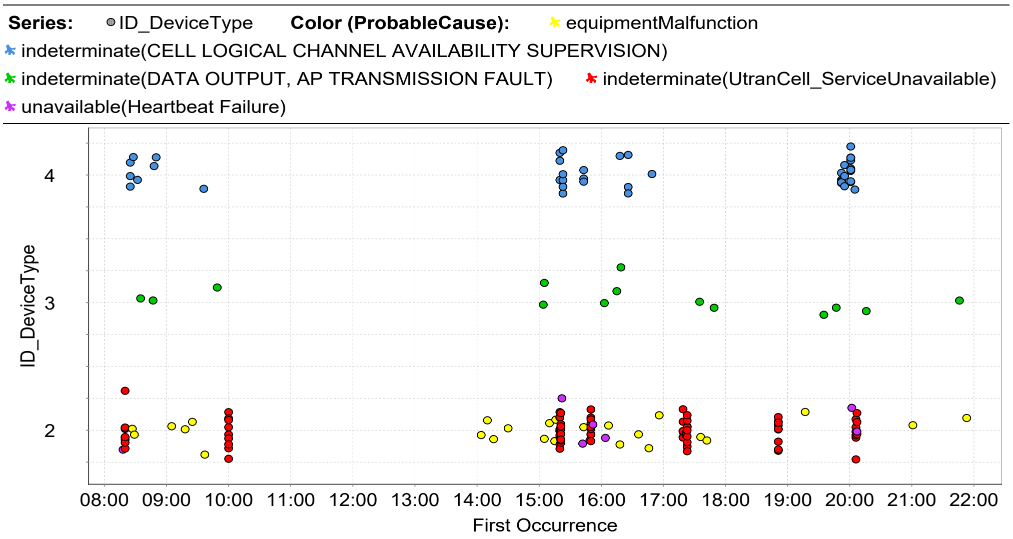


Figure 5.38: Zoomed Scatter Plot: DeviceTypes vs First Occurrence in 1st of September in the First Mutual Rule

Confidence	Lift	Support
0.8	6	3%

5.4.6 Second Mutual Rule in Milan data set in May

As we can observe in the below table information on the interestingness of a the rule is reported as an output.

Confidence	Lift	Support
0.8	5.5	2%

Figure 5.41 and 5.42 details the temporal visualization of devices types for each device involved in the mentioned rule during month of May in province of Milan.

5.4.7 Second Mutual Rule in Turin data set in September

As we can observe in the below table information on the interestingness of a the rule is reported as an output.

Confidence	Lift	Support
0.8	10	1%

Figure 5.43 and 5.44 details the temporal visualization of devices types for each device involved in the mentioned rule during month of September in province of Turin.

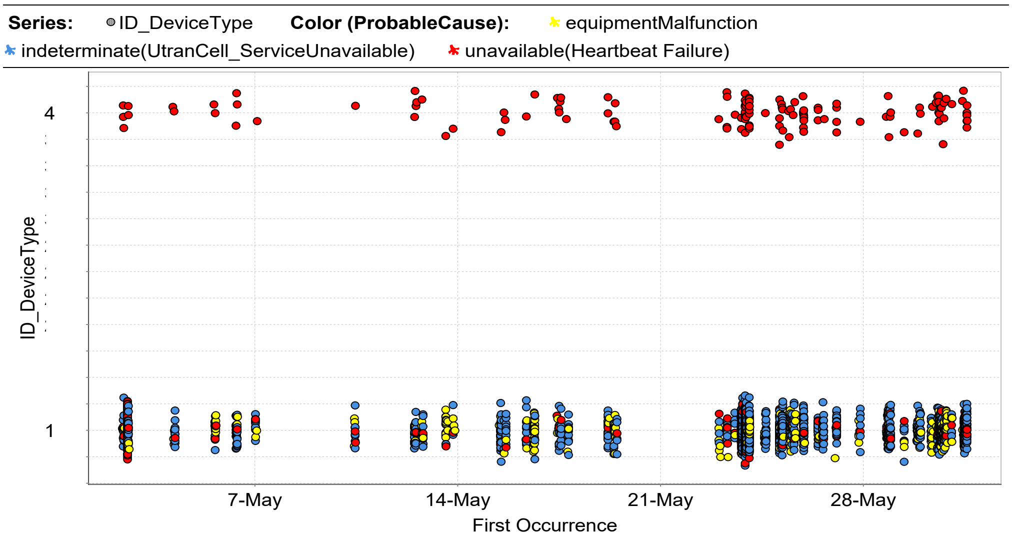


Figure 5.39: Scatter Plot: DeviceTypes vs First Occurrence in the Second Mutual Rule in Turin in May

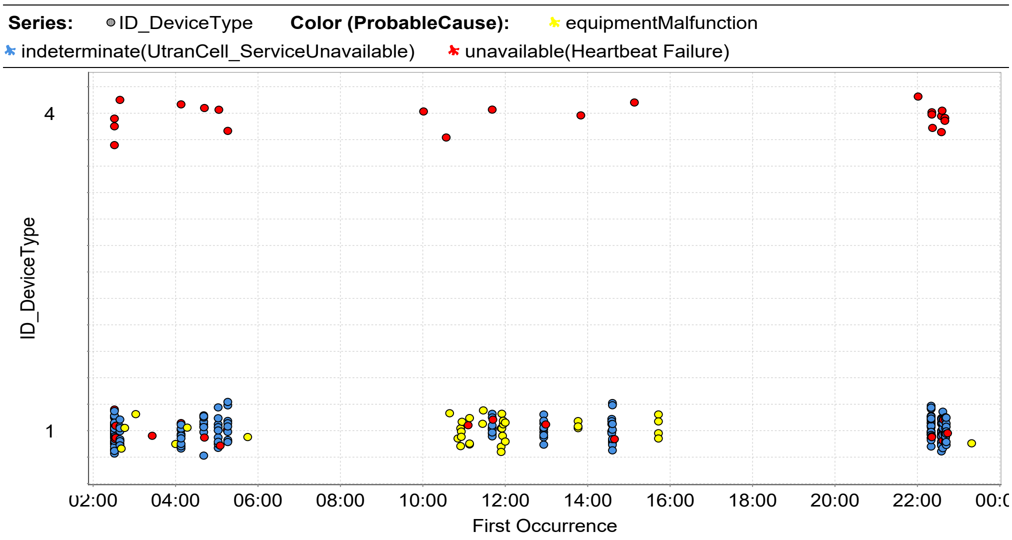


Figure 5.40: Zoomed Scatter Plot: DeviceTypes vs Second Occurrence in 25th of May in the First Mutual Rule in Turin

5.4.8 Second Mutual Rule in Milan data set in September

As we can observe in the below table information on the interestingness of a the rule is reported as an output.

Figure 5.45 and 5.46 details the temporal visualization of devices types for each device involved in the mentioned rule during month of September in province of Milan.

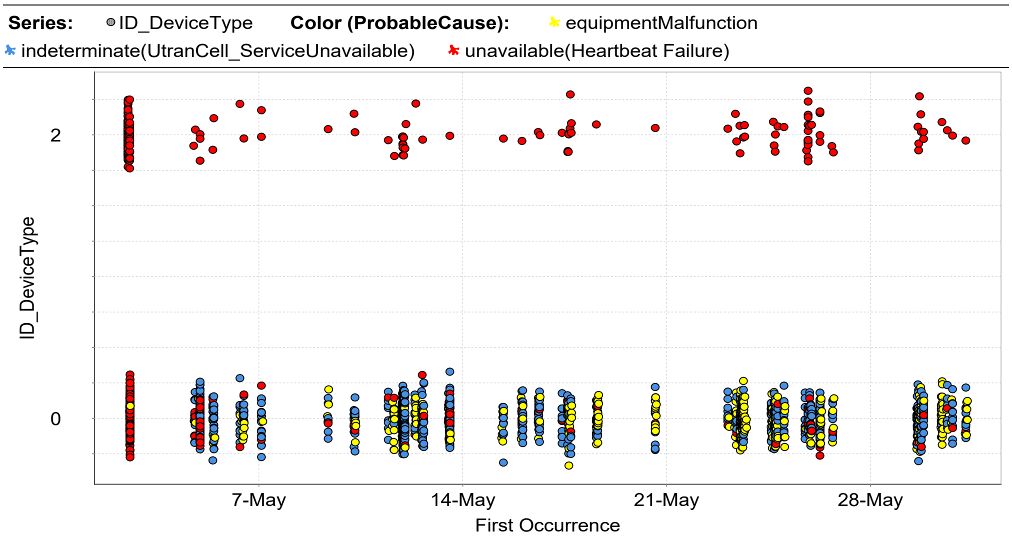


Figure 5.41: Scatter Plot: DeviceTypes vs First Occurrence in the Second Mutual Rule in Milan in May

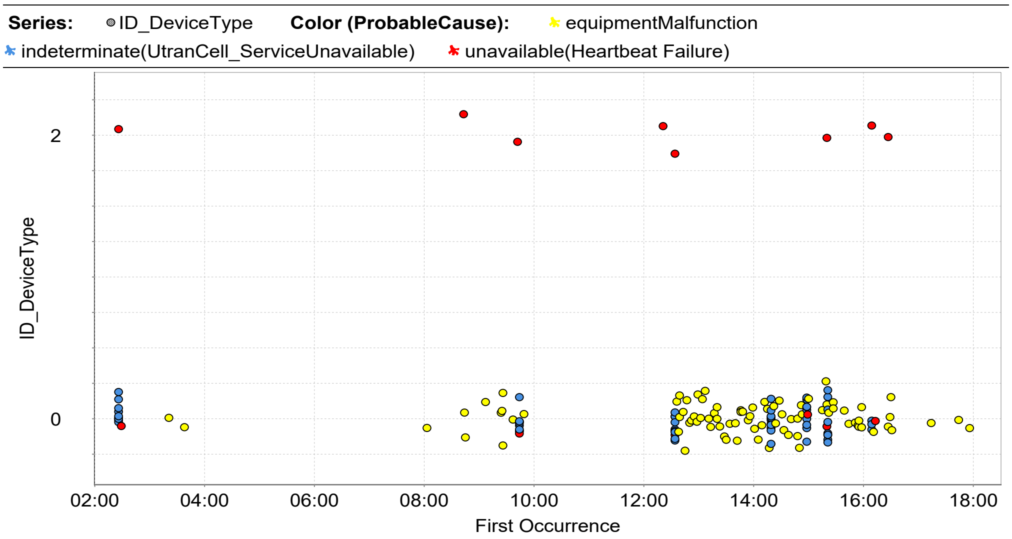


Figure 5.42: Zoomed Scatter Plot: DeviceTypes vs First Occurrence in 23rd of May in the Second Mutual Rule in Milan

Confidence	Lift	Support
0.9	9.4	2%

As we have seen in the examples, lift is higher in September because there are less devices involved in compared to May. We know that consecutively support is lower, too.

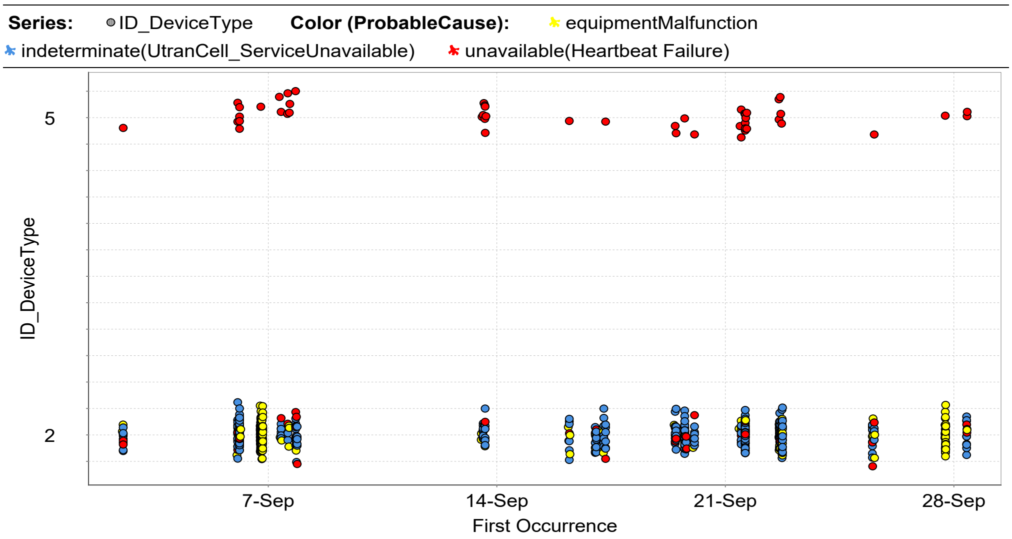


Figure 5.43: Scatter Plot: DeviceTypes vs First Occurrence in the Second Mutual Rule in Turin in September

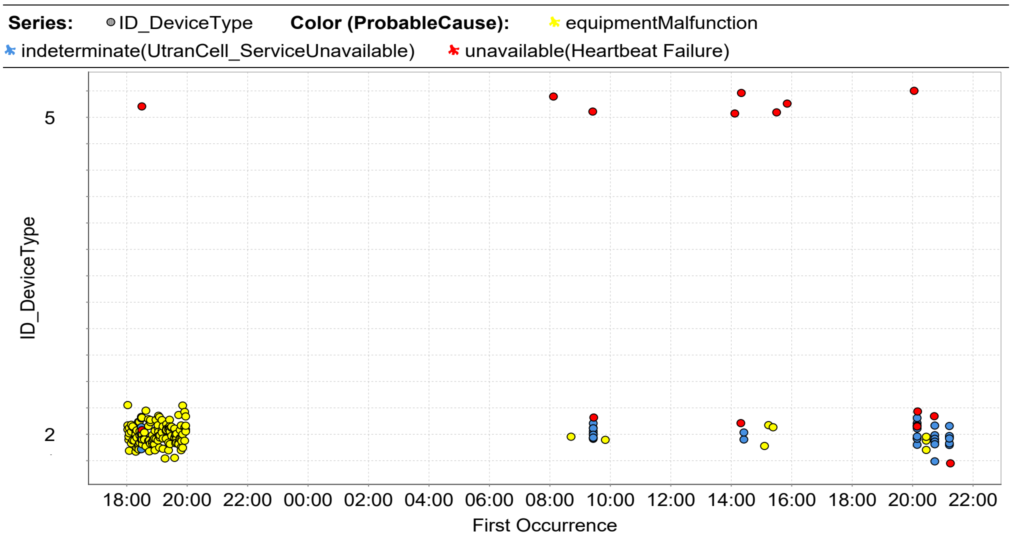


Figure 5.44: Zoomed Scatter Plot: DeviceTypes vs First Occurrence in 25th of May in the Second Mutual Rule in Turin

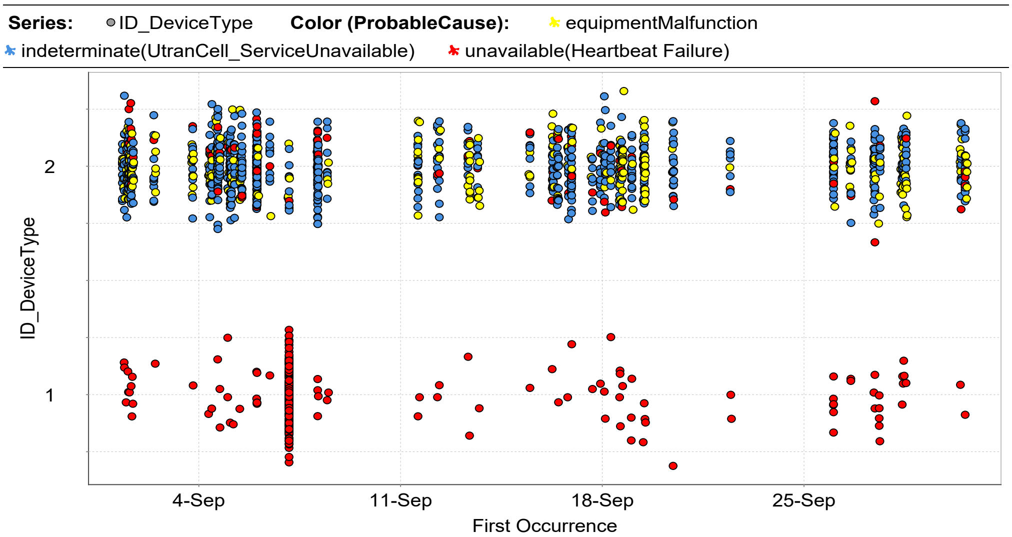


Figure 5.45: Scatter Plot: DeviceTypes vs First Occurrence in the Second Mutual Rule in Milan in September

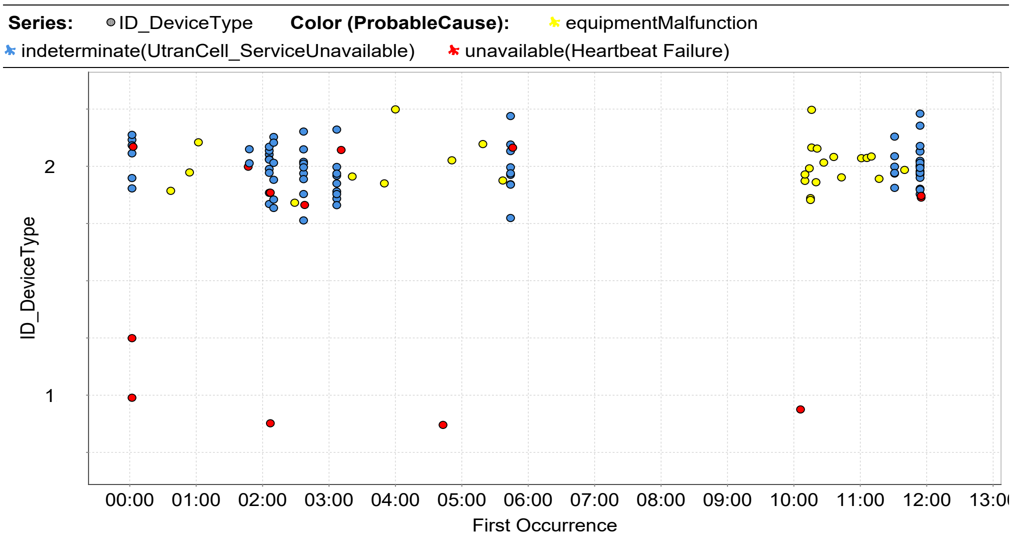


Figure 5.46: Zoomed Scatter Plot: DeviceTypes vs First Occurrence in 1st of September in the Second Mutual Rule in Milan

Chapter 6

Conclusion and Future Work

6.1 Conclusions

Throughout the thesis, we found out that an alternative way to gain more information out of data is to apply data mining and knowledge discovery methods [15]. A discussion on different methodologies representations based on market basket analysis concept has been provided. First an experiment aimed at finding temporal correlations among each network device has been performed. Next, a different experiment on types of each device has been done. The comparison between the two methodologies has been discussed through the previous chapters. To recap, the latter methodology extracts more general associations because it is focused on types not the devices themselves.

Since we are using machine learning algorithms, let us go through the main steps of machine learning used in our thesis to build a predictive model.

- Data Gathering: This step was done by TIM.
- Data Preparation: This step involves data cleaning and manipulation. De-duping, removing old or unused fields, error correction and changing them into the right format is part of these step. It also involves looking for any data imbalances that could make data heavily biased.
- Choosing an ML model: We chose frequent pattern mining and then extracted association rules.
- Evaluation and Parameter Tuning: This steps identifies how well an algorithm performs. By adding more data or changing the parameters, model gets more accurate.

With that in mind, before stating the conclusion, we remind the reader the key role of domain expert knowledge in interpreting the results of discovered patterns. Our work is to aid the experts in recalling and formulating correlation patterns in an efficient way. Given obtained rules derived from an alarm database, domain expert is able to verify whether the rules are useful or not. Some of the rules may reflect causal correlations and give new insights into the behaviour of the network elements whereas others may be irrelevant.

As we observed, parameterization is needed when searching for proper methods in order to find the required information from the data. In our approach, we apply this with different thresholds and data selections. As a result, the method reveals a set of selected informative rules. Then experts can learn quite a lot from data and find the answer to questions such as: "What are the distributions of alarms types and their causes?", "What are the most common combinations of devices that generated alarms?", "Is there any correlation among the alarms coming from different sources?", and so on. By logic, this kind of information and knowledge about the network could be even more valuable than the rules found in the data because such information can relatively easily be interpreted.

The feedback from the TIM network maintenance team in Rome confirmed that rules similar to what we showed were already presented in their system. So our automatic rules are useful for their systems. Moreover, TIM would like to use the rules we extracted as an input of machine learning algorithms to "detect patterns". These rules are stored in the systems as a list of "situations" (e.g., our rules), presented together with meta-data (location, resolution and etc).

6.2 Future Work

Since experiments with real data are often very time consuming, different tests and adaptations of our methods have been left for the future. However, there are several interesting research directions to be considered outside the scope of the thesis. First, in addition to data sources we have used, it would be interesting to also investigate network devices from other cities and provinces such as the south of Italy. Second, we would like to validate more temporal-spatial patterns and investigate how to model them. Finally, the methodologies proposed in our work to capture temporal-spatial correlations could be quite general hence we would like to investigate other critical domains such as medical and crime applications. In the arena of deeper analysis, our mechanism could aid to distinguish rules and patterns that appear in some cases with the ones that rarely do.

Bibliography

[1] http://www.thelifenetwork.org/about.html

[2] Agrawal, Imielinski, Swami. *Mining association rules between sets of items in large databases*, Proceedings of the SIGMOD international conference on Management of data, ACM 1993.

[3] Agrawal, Srikant. *Mining sequential patterns* In Proc. of the Eleventh International Conference on Data Engineering (ICDE95), pages 3-14, 1995.

[4] Agrawal, Han. *Frequent Pattern Mining* Published by Springer, 2014.

[5] Vaarandi.*A data clustering algorithm for mining patterns from event logs*, 2003 Workshop on IP Operations & Management ,IEEE (IPOM), 2003.

[6] Burns, Hellerstein, Ma, Perng, Rabenhorst, Taylor. *A systematic approach to discovering correlation rules for event management.* In Proc. of IEEE/IFIP International Sysmposium on Integrated Network Management, 2001.

[7] Qiu, et al. *What happened in my network: mining network events from router syslogs*, Proceedings of the 10th ACM SIGCOMM conference on Internet measurement, ACM 2010.

[8] Tee, Parisis, Wakeman. *Towards an approximate graph entropy measure for identifying incidents in network event data*, Network Operations and Management Symposium (NOMS), IEEE/IFIP 2016.

[9] Kobayashi, Otomo, Fukuda, Esaki. *Mining causality of network events in log data*, Transactions on Network and Service Management, IEEE 2017.

[10] Otomo, Kobayashi, Fukuda, Esaki. *An Analysis of Burstiness and Causality of System Logs*, Proceedings of the Asian Internet Engineering Conference, ACM 2017.

[11] Kaufman, Rousseeuw. *Finding groups in data an introduction to cluster analysis*, John Wiley 1990.

[12] Hahsler. *A Probabilistic Comparison of Commonly Used Interest Measures for Association Rules*, 2015

[13] Hahsler. *Introduction to arules: A computational environment for mining association rules and frequent itemsets*, Journal of Statistical Software, 2005

[14] Witten, Frank. *Data mining: practical machine learning tools and techniques, 2nd edition*, by Elsevier Inc 2005.

[15] Klemettinen, Mannila, Toivonen. *Rule Discovery from Telecommunication Network Alarm Databases*, Journal of Network and Systems Management, 7(4):395-423, 1999.

[16] Raeder, V. Chawla. *Market basket analysis with networks*, Springer-Verlag 2010.

[17] Tan, Steinbach, Karpatne, Kumar. *Introduction to Data Mining* Addison-Wesley, 2nd ed., 2013.

[18] Gunopulos, Khardon, Mannila, Saluja, Toivonen, Sharma. *Discovering all most specific sentences*, ACM Transactions on Database Systems (TODS), vol. 28, no. 2, pp. 140–174, 2003.

[19] Liang, Benson, Kanuparthy, He. *Finding Needles in the Haystack: Harnessing Syslogs for Data Center Management*, 2016.

[20] Wei, Li, Zhou, Zhang, Yang . *IWFPM: Interested Weighted Frequent Pattern Mining with Multiple Supports*, Journal of Software, 2014.

[21] Tseng. *An Efficient Method for Mining Association Rules with Item Constraints*, University of California, Berkeley.

[22] Annie M.C, Kumar D. *Market Basket Analysis for a Supermarket based on Frequent Itemset Mining*, IJCSI International Journal of Computer Science Issues, Vol. 9, Issue 5, No 3, September 2012.

[23] Hatonen, Klemettinen, Mannila, Ronkainen and Toivonen. *Knowledge discovery from telecommunication network alarm databases*, Proceedings of the 12th International Conference on Data Engineering (ICDE'96), pages 115 – 122, New Orleans, Louisiana, USA, IEEE Computer Society Press, February 1996.

[24] Hatonen, Klemettinen, Mannila, Ronkainen, and Toivonen. *Rule discovery in alarm databases. Technical*, Report C-1996-7, University of Helsinki, Department of Computer Science, P.O. Box 26, FIN-00014 University of Helsinki, Finland, March 1996.

[25] Hatonen, Klemettinen, Mannila, Ronkainen, and Toivonen. *TASA: Telecommunication alarm sequence analyzer, or "How to enjoy faults in your network".* In Proceedings of NOMS'96 - IEEE Network Operations and Management Symposium (NOMS'96), pages 520 – 529, Kyoto, Japan, April 1996. IEEE.

[26] Jakobson and Weissman. *Alarm correlation. IEEE Network*, 7(6):52 – 59, November 1993.

[27] Jakobson and Weissman. *Real-time telecommunication network management: extending event correlation with temporal constraints.* Proceedings of the fourth international symposium on Integrated network management IV, pages 290 – 301, London, United Kingdom, 1995.

[28] Latha and Ramaraj. *Algorithm for Efficient Data Mining.* International Conference on Computational Intelligence and Multimedia Applications, ICCIMA 2007.

[29] Tan, Bu, Yang. *An Efficient Frequent Pattern Mining Algorithm.* Sixth International Conference on Fuzzy Systems and Knowledge Discovery, 2009

[30] https://www.python.org/

[31] https://www.r-project.org/

[32] https://rapidminer.com/

[33] Weka Extension in RapidMiner Marketplace

[34] Lift in an association rule, IBM Knowledge Center