Master Thesis in Statistics and Data Mining

Anomaly detection and analysis on dropped phone call data

Paolo Elena

Division of Statistics
Department of Computer and Information Science
Linköping University

Supervisor

Prof. Patrik Waldmann

Examiner

Prof. Mattias Villani

"It's a magical world, Hobbes, ol' buddy... Let's go exploring!" (Calvin and Hobbes - Bill Watterson)

Contents

Abstract **1**

Acknowledgments **3**

1 Introduction **5**

1.1 Background 5

1.2 Objective 7

1.3 Useful terms 7

2 Data **9**

2.1 Data sources 9

2.2 Raw data 10

2.2.1 Data variables 11

2.3 Secondary data 14

3 Methods **15**

3.1 Anomaly Detection 15

3.1.1 Markov-Modulated Poisson Process (MMPP) 16

3.2 Anomaly Description 20

3.2.1 Exploration of individual anomalies 21

3.2.2 Dimensionality reduction and extraction of main patterns . . . 24

3.3 Technical aspects 25

4 Results **27**

4.1 Exploratory analysis 27

4.1.1 Main data 27

4.1.2 Secondary data 29

4.2 Detection of anomalous periods using the MMPP model 33

4.2.1 Preprocessing 33

4.2.2 Choice of parameters 33
4.2.3 Output 35
4.3 Exploration of anomalous periods 39
4.4 Analysis of main patterns common to anomalous periods 44

5 Discussion 49

6 Conclusions 53

Bibliography 55

Abstract

This thesis proposes methods to analyze multivariate categorical data with a time component, derived from recording information about dropped mobile phone calls, with the aim of detecting anomalies and extracting information about them. The detection relies on a time series model, based on a Markov-modulated Poisson process, to identify anomalous time periods; these are then analyzed using various techniques suitable for categorical data, including frequent pattern mining. Finally, a simpler approach based on Principal Component Analysis is proposed to complement the aforementioned two-stage analysis. The purpose of this work is to select data mining techniques that are suitable for extracting useful information from a complex, domain-specific dataset, in order to help expert engineers with identifying problems in mobile networks and explaining their causes - a process known as troubleshooting.

Acknowledgments

I would like to thank the following people:

- First and foremost my family, for supporting my decision to move to Sweden and pursue this Master's Degree, as they supported me in every step of the way, and for realizing ten years ago that studying English properly would be a great use of my free time.
- My EO friends, for having been the best part of my university experience and for filling my travels home with fun.
- My friends studying abroad, for giving me great memories of Germany, Portugal and France.
- My Linköping study mates, in particular Uriel for having been a good friend and teammate in group works, and Parfait for his company during the thesis work and his valuable feedback on this report.
- My other Linköping and Stockholm friends, for the dinners, parties and for letting me crash on their couch whenever I needed it.
- My Helsinki tutors and tutor group, for making me feel among friends from the first days of exchange.
- The Helsinki Debating Society, for introducing me to the great hobby of debating.
- The 2012 iWeek team, for hosting me in Paris and somehow kickstarting all of this.
- My former professors in Genova, in particular Marco Storace for teaching me what it means to write a proper scientific report and Davide Anguita for getting me interested in machine learning.
- My Linköping professors, in particular my thesis supervisor Patrik Waldmann for suggesting the MMPP model and Mattias Villani for helping me with

finding this thesis project, providing support and having a positive attitude towards external thesis projects.

- Ericsson AB, for giving me the opportunity to work with them and providing the data for this thesis; in particular my mentor Martin Rydar, for welcoming me to the company and always being available and supporting, Mikael Milevski, for his role in assembling and maintaining the dataset, and Leif Jonsson, for overseeing all thesis projects and providing valuable input.

1 Introduction

1.1 Background

Telecommunication companies also provide services and support to the network operators, and one of the major tasks that the support personnel has to carry out is *troubleshooting*: finding problems in the system and understanding their causes. Due to the ever increasing amounts of information that are recorded daily by the systems during their operation, support engineers are increasingly turning to data mining techniques to facilitate their work.

The particular scope of this thesis is analyzing abnormally dropped calls and connections: that is to say, communications that are ended by signal problems, limitations in the network capacity or other causes explained below. While a certain number of such events is expected and cannot be avoided, it is interesting to detect increases in the frequency of these errors, as they are often triggered by malfunctions in the network. Moreover, the percentage of connections that are dropped is one of the indicators of performance that phone operators are most interested to minimize, which provides an added incentive to focusing on this problem. Whenever increases are noticed, troubleshooters have to understand what happened and look for solutions. This is a complex process, as degradation in performance can have many possible causes: from system updates introduced by the technology provider itself (which might have unforeseen consequences) to configuration changes made by network operators, appearance of new user device models and operating systems, or simply particular user behaviors (for instance a geographical concentration that exceeds the capacity of the network, as in the case of large sporting events or demonstrations).

Explaining and solving problems helps with establishing trust in the customers (that is, the phone operators), who want to be guaranteed the best possible service and need to be confident that their supplier is in control of the situation. Therefore, making the troubleshooting process as fast and reliable as possible is very important to telecommunication companies. At Ericsson, the company at which this work was carried out, the current approach is heavily reliant on investigations by expert engineers; this thesis sets out to apply data analysis techniques that can automate and standardize the detection and description of anomalies, making their work more efficient. The following paragraphs detail some of the constraints and challenges inherent to this type of analysis.

The size of the systems and the volumes of data that are processed and transmitted daily are enormous: in 2012 there were over 6 billion mobile subscriptions, for a global monthly traffic nearing 900 Petabytes (Ericsson, 2012). To give a scale comparison, Facebook reported a total storage of 100 Petabytes while filing for its Initial Public Offering in 2012. Millions of phone calls and data connections are completed every day in each city, and with such numbers it is unthinkable to store, let alone analyze thoroughly, the whole data. Numerous trade-offs have to be made between depth of detail and breadth of the timespan examined, therefore troubleshooters have to work with different data sources depending on such constraints.

These trade-offs represented a major challenge in the development of this thesis work: full system logging for a single network controller can produce Terabytes of information per day; in order to prioritize collecting data for long periods of time, the maintainers of the dataset faced storage space restrictions. Since most of the established connections terminate normally and are considered uninteresting, only information about the errors that terminate the remaining ones (a few parts per thousand) was retained for this analysis.

While it is beneficial to have a relatively long time period represented in the data, the loss of information about normal calls rules out the traditional machine learning approaches, based on classification algorithms, that are reported in papers about the same subject, such as Zhou et al. (2013); their focus is analyzing quantitative parameters about ongoing connections to identify predictors of untimely drops, while this work is framed as an unsupervised learning problem in the family of anomaly detection. It looks at descriptive information about dropped connections, as well as their amount, to establish if and how the network was deviating from normal beavior at any given time.

Another challenge involved in the analysis is the fact that all the information about the errors is made up of categorical variables (which part of the software reported the drop, what kind of connection was ongoing, what were the devices involved and many more) and the raw data is wide (around 50 columns are consistently present for every data point and some have up to 100): due to the aforementioned constraints on collecting and integrating data, it was not practical at this stage of development to include information about the quality of the phone signal, the time it took to start the connection or its duration. This limits the applicability of the main machine learning techniques for exploration of data and dimensionality reduction, such as Principal Component Analysis and clustering, which rely on the data being numeric; even though PCA was indeed applied, this was only possible after a transformation of the original dataset that entailed losing a significant amount of detail.

1.2 Objective

This thesis sets out to apply statistical modeling and data mining techniques to the detection and description of anomalies in dropped call databases, by establishing a model of normal behavior, pinpointing periods of time in which the data departs from this model and extracting statistics and patterns that can guide domain experts in the troubleshooting process. Since this is a new way to analyze this particular data, attention is focused on defining a concept of anomaly in this context and understanding common features of anomalies. Rather than concentrating on modelling a specific dataset in its details, the aim is to develop methods that can yield a summary analysis of many different datasets of the same type, and are able to adapt to various patterns of normal behavior and data sizes.

1.3 Useful terms

The following are definitions of recurring concepts that are required to understand the context of the problem.

Troubleshooting

The systematic search for the source of a problem, finalized to solving it. In this thesis, the concept is extended to include the detection and identification of problems.

User devices

Phones, tablets and all other devices that connect to the network to transmit and receive data.

Network, phone network, central network

Collective term for the antennas that exchange signals with user devices and the computers in charge of managing connections, processing transmitted data and communicating with external systems (such as the Internet). More information will be provided in the Data chapter.

Speech connections

Instances of communication between a mobile user device and the network that correspond to an ongoing phone call. Given the continuous nature of phone calls, stricter timing constraints apply to these and more importance is given to their continuation.

Packet connections

Instances of communication between a mobile user device and the network in which packet data is exchanged (such as mobile internet traffic), not associated to a phone call. As mobile internet traffic has less real-time requirements than speech and lost data can usually be retransmitted, problems in packet connections are better tolerated.

(Abnormally) Dropped calls

Also referred to as (abnormal) call drops, this refers to connections between a mobile user device and the network that terminate unexpectedly (due to signal problems, limitations in the network capacity or other causes) rather than reaching their expected end. The word “call” is used somewhat ambiguously to refer to both speech and packet.

Radio Network Controller

An element of 3G radio networks, which manages resources (for instance transmission frequencies) and controls a group of Radio Base Stations which have antennas to communicate with user devices. RNCs are responsible for areas on the scale of cities and are the level of hierarchy at which most data collection for analysis purposes is performed; the datasets considered in this work come from individual RNCs.

2 Data

2.1 Data sources

This work was carried out in the offices of Ericsson AB, a multinational telecommunication company based in Kista, Sweden; over 1000 telecommunication networks, both mobile and fixed, in more than 180 countries use Ericsson equipment, and 40% of the world's mobile traffic relies on the company's products (Ericsson, 2012). Ericsson supplied the datasets and computer resources necessary for the thesis.

All the data analysed in this thesis comes from UMTS (Universal Mobile Telecommunications System) technology, more commonly known as 3G. It supports both standard voice calls and mobile Internet access, also simultaneously by the same user; having been introduced in 2001, 3G has undergone many developments and is still the most widely used standard for cellular radio networks.

3G networks, as well as other types of networks, have a hierarchical structure, corresponding to their partition of the geographical area that they cover. Devices connecting to the network, obviously equipped with a transceiving antenna, are termed User Equipment (UE); they can be phones, tablets, mobile broadband modems or even payment machines. The smallest division of the network is named *cell*; stationary nodes named Radio Base Stations (RBS), equipped with transceiver antennas that service each cell, are deployed on the territory. RBSs are linked at a higher level to a Radio Network Controller (RNC), which takes care of managing the resources at lower levels, keeping track of ongoing connections and relays data between user devices and a Core Network. The Core Network (CN) routes phone calls and data to external networks (for instance the Internet).

The most important level of hierarchy for this work is the Radio Network Controller: most data collection tools operate on RNCs, and the datasets considered in this work come from them. RNCs are responsible for fixed geographical areas on the scale of cities, therefore it is reasonable to expect homogeneity of daily patterns within the dataset. However, the developed techniques are going be applied in the future to data coming from parts of the world that have differing timezones, cultures and customs, which obviously influence the concept of normal behavior.

Overall, the huge size of the systems involved implies that there exists an enormous amount of sources of variation, possible parameters and internal states, most of them only understandable by very experienced engineers. This is reflected in the data analyzed, which was not assembled specifically for data mining purposes and was therefore challenging to approach.

2.2 Raw data

The datasets on which the work has been carried out come from an internal company initiative to combine information about call drop events into lines of a *csv* (Comma Separated Values) file. Several distinct logs are processed and matched by a parser developed internally; in the live networks that the methods will eventually be applied to, data sources may occasionally be turned off for maintenance or upgrades, meaning that periods with incomplete or missing data are to be expected. The period of time covered by the main dataset, upon which the methods were tested and the results are based, is two months; over this time, roughly two million drops were recorded, which corresponds to as many data points. Due to company policies, the real string values of many columns are not shown here, replaced by integer identifiers.

Each row of the dataset contains information on a different event, coming from the diagnostic features that are enabled in the Radio Network Controller; as many as 191 different columns may contain information in the raw data, though it became immediately apparent that it was wise to focus on a smaller amount. In fact, most of the columns had characteristics that would have required dedicated preprocessing and were therefore left for future consideration (some contained information only for a specific subset of drops, others contained themselves value-name pairs of internal parameters).

Ultimately, I agreed with the commissioners of the project to focus on only 26 columns, leaving the others for future work, and proceeded to clean the data using Unix command line utilities such as *cut*. Exploratory data analysis showed some of these to be redundant and I settled on retaining 15 for further processing. I will illustrate them in this section, dividing them into groups with similar meaning, in order to provide insight into this unusual problem.

2.2.1 Data variables

The first group of columns, shown in Table 2.1 contains information about the time of each call drop event.

- *uTimeReal* is the UTC timestamp of the moment the drop was recorded, with millisecond precision.
- *uTimeRopLocal*: the 15-minute interval in which the drop was recorded. The parser used to assemble the dataset groups data in such intervals, named Rops, and it was decided to maintain this level of granularity for further analysis. One of my inputs in the data preprocessing was to take into account the possibility of a different timezone, in order to have a more accurate understanding of the day-night cycles in different areas once the methods will be applied to real data. This is the reason why in the table the two columns appear not to match and are, in fact, two hours apart.
- *uWeekDayLocal*: the day of the week in which the drop was recorded; this is especially useful to spot weekly periodicities, for instance discrepancies between weekdays and weekends.

	uTimeReal	uTimeRopLocal	uWeekDayLocal
21278	2013-03-21 22:00:02.628	2013-03-22 00:00	Tuesday
21279	2013-03-21 22:00:02.792	2013-03-22 00:00	Tuesday
21280	2013-03-21 22:00:03.704	2013-03-22 00:00	Tuesday
21281	2013-03-21 22:00:09.300	2013-03-22 00:00	Tuesday
23277	2013-03-22 00:11:24.500	2013-03-23 02:00	Tuesday
66277	2013-03-23 05:12:09.412	2013-03-23 07:00	Wednesday
111277	2013-03-24 13:47:38.892	2013-03-24 15:45	Thursday

Table 2.1: Table of time-related columns

The second group, shown in Table 2.2 contains information about the software exception associated to the call drop. Exceptions are anomalous events occurring during the execution of the RNC software, altering its flow. When a call is dropped, an exception is *raised* in the RNC, and the type of exception gives valuable information on its cause.

- *uEC* is the software exception code that was reported in relation to the call drop; it has a 1:1 correspondency with an explanatory string, which was removed due to redundancy. Each code corresponds to a different failure, ranging from expiration of timers to reception of error signals from various parts of the system. Over fifty unique codes are present in the dataset.
- *uEcGroup* is a grouping of the above codes that puts related exceptions together into 15 categories; I requested this to be included so that collective increases could be detected, on the hypothesis that related exceptions might have related causes.

	uEC	uEcGroup
21278	43	INTERNAL_REJECT_AND_FAILURE_SIGNALS
21279	42	INTERNAL_TIMER_EXPIRY
21280	53	INTERNAL_TIMER_EXPIRY
21281	43	INTERNAL_REJECT_AND_FAILURE_SIGNALS
23277	27	EXTERNAL_REJECT_AND_FAILURE_SIGNALS_(RRC)
66277	43	INTERNAL_REJECT_AND_FAILURE_SIGNALS
111277	46	INTERNAL_TIMER_EXPIRY

Table 2.2: Table of software-related columns

The third group, shown in Table 2.3 contains information related to the geographical area in which the user device was located. In addition to the cell, higher level groupings were included to spot geographical patterns that might go unnoticed by looking at a variable that is naturally very sparse (Over 800 different cell identifiers appear in the data).

- *uCellId* is the identifier of the cell the device was connected to. Cells are associated with a stationary antenna, but there is no univocal correspondence to location: a device is usually within range of multiple cells and the system attempts to balance the load between them. Moreover, the serviced area can expand or contract in size to balance the load (a process known as cell breathing).
- *kIubLinkName* is an identifier for a higher level grouping, corresponding to the interface between radio base stations and the RNC. One potential problem is

that the mapping between cell and IubLink is not guaranteed to be fixed. Roughly 150 unique values appear in the dataset.

- *kRncGroup* is a yet higher level grouping; this is generally associated with IubLink and Cell, but this mapping too can potentially be redefined.

	uCellId	kIubLinkName	kRncGroup
21278	97	67	9
21279	171	32	22
21280	677	16	15
21281	513	107	10
23277	77	135	26
66277	531	33	20
111277	829	79	9

Table 2.3: Table of location-related columns

The fourth group contains information related to the type of connection that was unexpectedly terminated.

- *uUehConnType* is a string indicating what type of connection was ongoing; the identifier summarizes whether it was speech, data or mixed connection and the speed of the data transfer.
- *uUehTargetConn* is a string indicating what connection was being switched to: the system allows for connection types to be changed and it is common for drops to happen in this phase.
- *uUehConnTypeTransition* is the concatenation of the two above strings; it is known from expert knowledge that the aggregation can be more informative than the two parts.

The fifth group contains information about the state of the connection that was interrupted. The process of setting up and maintaining a connection is very complex and the states represent various phases, most of them related to waiting for a specific signal. A different thesis work carried out at the same time as this one was aimed at modelling these state transitions, following connections from start to end.

- *uCurrentState* is the state in which the connection was at the moment of the drop. Over 100 distinct values are present in the data.
- *uPreviousState* is the state from which the connection reached the current one. Over 100 distinct values are present in the data.

The sixth group, shown in Table 2.4, contains information about the service that was interrupted by the drop.

- *iuSysRabRel* is a string reporting exactly which services were being offered during the connection (speech and/or packet data, at which speeds). Roughly 50 unique levels are present.
- *iuSysRabRelGroup* is a grouping of the above, synthesizing the type of service: speech call alone, standard packet data alone, simultaneous speech and standard data, high speed data, speech and high speed data, then legacy services (e.g. video calls sent as speech calls), for a total of 7 possible values.

	iuSysRabRel	iuSysRabRelGroup
21278	18	PS_R99
21279	32	PS_HSUPA
21280	32	PS_HSUPA
21281	32	PS_HSUPA
23277	18	PS_R99
66277	32	PS_HSUPA
111277	32	PS_HSUPA

Table 2.4: Table of Service-related columns

2.3 Secondary data

One important part of information that is missing in the main dataset is the amount of connections that terminated without any problems: that is, how much normal traffic was going on while the errors happened. This information is stored in a different source and it is therefore complicated to obtain it and synchronize it with the errors dataset. Exploratory data analysis (see sec. 4.1.2) was carried out to judge whether it was reasonable to include this data in the analysis.

3 Methods

As previously mentioned, this thesis work can be conceptually divided into two parts: the first, which I'll refer to as *detection*, is concerned with finding time periods with anomalously high numbers of dropped call events and works with time series data; the second, which I'll refer to as *description*, is concerned with characterizing each anomalous period, comparing it with other periods that the detection has marked as normal and finding patterns that can be linked to specific failures.

3.1 Anomaly Detection

Detection is handled by analyzing the time series of dropped call counts, employing a model that can learn the usual amount of dropped calls and identify points in the time series that deviate from learned behavior. An important consideration is that the normal behavior of errors is closely related to the normal behavior of users on the network: there is a positive correlation between the amount of connections established and the amount of connections terminated unexpectedly. Since the activity of users of the network is expected to mirror daily routines and weekly cycles, any model applicable to this data has to have time as an input and account for periodicity: what is normal during daytime might be extremely anomalous during nighttime.

Works such as Weinberg et al. (2007) and Aktekin (2014) analyze similar data (call-center workload), however their focus is on studying normal behavior rather than on considering anomalies. As we are working in an unsupervised context, where unusual events are not labeled, learning a model of normal behavior is not straightforward. Ideally we would like to remove abnormal events from the data before training a model, but the meaning of abnormal itself depends on having a reliable idea of the baseline of normal behavior. This constitutes a circular problem that was likened to the ancient “chicken-and-egg” dilemma in the research paper that provided inspiration for this work. The chosen model deals with this problem by not requiring the training set to be free of anomalies, building instead a concept of anomaly in relation to other parts of the same time series. The technique, presented in Ihler et al. (2007), utilises Poisson processes with variable mean to model counts while accounting for periodicity, integrating a Markov state that represents whether or not each time point is considered anomalous or normal. A posterior probability estimation of each time point being anomalous can be obtained by averaging the Markov states over numerous iterations. Thanks to the Bayesian inference framework employed, this model can also answer additional questions about the data, for instance indicating whether or not all days of the week share the same normal behavior.

Due to the completely categorical nature of the dataset and the practical obstacles associated with integrating potentially useful covariates (for instance the conditions of the network, the signal strength or other sorts of continuous measurements related to connections), in this work detection operates on the time series alone.

3.1.1 Markov-Modulated Poisson Process (MMPP)

3.1.1.1 Basic Concepts

This technique models time-series data where time is discrete and the value $N(t)$ represents the number of noteworthy events occurred in the time interval $[t-1,t]$; it was developed having in mind counts of people entering buildings and vehicles entering a freeway and is therefore suitable for phenomena exhibiting the natural rhythms of human activity. The model assumes that the observed counts $N(t)$ are given by the sum of a count generated by normal behavior, $N_0(t)$, and another generated by anomalous events, $N_E(t)$, as seen in 3.1. These two variables are assumed to be independent and modeled in separate ways, as described in the following two paragraphs.

$$N(t) = N_0(t) + N_E(t) \tag{3.1}$$

Periodic Counts Model (N_0) The regular counts $N_0(t)$ are assumed to follow a nonhomogeneous Poisson distribution, the most common probabilistic model for count data. The probability mass function (the density's equivalent for discrete random variables) in the standard case is

$$P\left(N_0;\lambda\right)=\frac{e^{\lambda}\lambda^{N_0}}{N_0!}\qquad N_0=0,1,\ldots,\tag{3.2}$$

with λ being the average number of events in a time interval (*rate*), which is also the variance. In the nonhomogeneous case, the constant λ is replaced by a function of time, $\lambda(t)$; more specifically in this case the function is periodic. The reasonableness of the Poisson assumption must be tested on the data; in the nonhomogeneous case, this means testing separately each rate and the portion of the dataset associated with it (preferably after filtering out outliers). A common problem with Poisson models is that of *overdispersion*, when the sample variance is significantly higher than the sample mean; the authors of Ihler et al. (2007) provide a qualitative way of testing for this and state that this assumption does not need to be strictly enforced for the model's purpose of event detection.

In order to model periodic patterns, $\lambda(t)$ is decomposed multiplicatively as

$$\lambda(t)=\lambda_0\delta_{d(t)}\eta_{d(t),h(t)}\tag{3.3}$$

with $d(t)$ indicating the day of the week and $h(t)$ the part of the day in which t falls. λ_0 is the average rate, δ_j is the relative day effect for day j (for instance, Mondays could have higher rates than Saturdays) and $\eta_{j,i}$ is the relative effect for time period i if day equals j. The constraints $\sum_{j=1}^{7}\delta_j=7$ and $\sum_{i=1}^{D}\eta_{j,i}=D$ (with D being the number of subdivisions of a day) are imposed to make the parameters more easily interpretable. Suitable conjugate prior distributions are chosen, specifically $\lambda_0\sim\Gamma(\lambda;a^L;b^L)$ (a Gamma distribution), $\frac{1}{7}\left[\delta_1,\ldots,\delta_7\right]\sim Dir(\alpha_1^d,\ldots,\alpha_7^d)$ and $\frac{1}{D}\left[\eta_{j,1},\ldots,\eta_{j,D}\right]\sim Dir(\alpha_1^h,\ldots,\alpha_7^h)$ (two Dirichlet distributions).

Aperiodic, Event-Related Counts Model (N_E) As seen in Equation 3.1 anomalous measurements are seen to be due to a second process, present only for a small part of the total time, which adds itself to the normal behavior. In order to indicate the presence of an event, a binary process $z(t)$, with Markov probability distribution, is used. The process is the state of a Markov chain with transition probability matrix $M_z=\begin{pmatrix} z_{00} & z_{01}\\ z_{10} & z_{11}\end{pmatrix}$, with each row having a Dirichlet conjugate prior; $z=0$

indicates no event and $z = 1$ indicates that an event is ongoing. z_{01} is therefore the probability of an event occurring during normal behavior and z_{10} is the probability of an ongoing event to terminate; the usual constraints on Markov transition matrices apply, therefore $z_{00} = 1 - z_{01}$ (probability of staying in a state of normal behavior) and $z_{11} = 1 - z_{10}$ (probability of an ongoing event to continue). The count produced by anomalous events can be modelled itself as Poisson with rate $\gamma(t)$, written as $N_E(t) \sim z(t) \cdot P(N; \gamma(t))$ using the above definition of $z(t)$. The rate $\gamma(t)$ is drawn itself from a Gamma distribution with parameters $a^E$and b^E; this parameter can however be integrated over, yielding $N_E(t) \sim NegativeBinomial(N; a^E, \frac{b^E}{(1+b^E)})$.

The inclusion of a Markov chain, with its concept of state, allows the model to account for different types of anomalies; in addition to single counts that decisively deviate from the norm, the memory given by the state makes it possible to detect *group anomalies*, i.e. several values in a row that might not signify anything when taken individually but could, as a whole, be indicative of an event.

3.1.1.2 Fitting the Model and Detecting Events

Assuming to know how $N(t)$ is decomposed into $[N_0(t), N_E(t), z(t)]$, it is easy to draw samples a posteriori of the parameters $[\lambda_0; \delta; \eta; M_z]$; this allows to obtain posterior distributions for them by inference using Markov chain Monte Carlo methods. The algorithm alternates between drawing samples of the hidden data variables $[N_0(t), N_E(t), z(t)]$ given the parameters and samples of the parameters $[\lambda_0; \delta; \eta; M_z]$ given the hidden variables. Each iteration has $\mathcal{O}(T)$ complexity, where T is the length of the time series. As is customary with MCMC methods, initial iterations are used for *burn-in*, while subsequent ones are retained in the results.

Sampling of the hidden variables proceeds in the following way: given $\lambda(t)$ and M_z a sample of $z(t)$ is drawn using the forward-backward algorithm (Baum et al., 1970). Then, given this hidden state, $N_0(t)$ is set as equal to the data value $N(t)$ when $z(t) = 0$ and drawn from a distribution proportional to

$$\sum_i P(N(t) - i; \lambda(t)) NBin(i; a^E, \frac{b^E}{1 + b^E}) \tag{3.4}$$

when $z(t) = 1$ (notice how this is weighed by the probability that the count $N(t) - i$ came from the supposed Poisson distribution). This makes the model resistant to incorporating the anomalous events into its model of normal behavior: rather than simply averaging over all the training data (obviously accounting for periodicities), which might lead to skewed normal counts if a massive event was registered, the model adjusts downwards the contribution of counts that are deemed to be unusually big.

In case of missing data, $N_0(t)$ is drawn from the appropriate Poisson and $N_E(t)$ is drawn independently. By writing out the complete data likelihood, it is possible to derive the distributions from which to draw the parameters given the data; because of the choice of conjugate prior distributions, the posteriors have the same form with updated parameters obtained from the data; the detailed description appears on Ihler et al. (2007).

The presence or absence of events is coded in the model by the binary process $z(t)$, thus the posterior probability $p(z(t) = 1|\{N(t)\})$ is used as an indicator of when events occurred. This is obtained by averaging all samples of $z(t)$ given by the MCMC iterations. A threshold Θ is then set to classify a time period t as anomalous if $z(t) > \Theta$.

The prior parameters with the most influence over the performance of the model are claimed by the authors to be the transition parameters of $z(t)$, which regulate how much the event process is allowed to compensate for high variance in the data and thus possibly overfit; adjusting them obviously also affects the sensitivity and therefore the number of events detected.

3.1.1.3 Testing differences between days

The model allows for each day to have its separate time profile, but that implies a high number of degrees of freedom and as a consequence a high requirement on the amount of data needed for training; however, it is possible to force some or all the days to share their δ and η_i values: a separate profile can be learned for each day of the week, two profiles can be learned for weekends and weekdays, or a single profile can be fit to all days. If the days that are grouped have similar behavior, this leads to an easier to train, more robust model. It is possible to test whether the full model or any constrained model is a better fit for the data by computing marginal likelihoods (likelihood of the data under the model, integrating out the parameter values) as shown in 3.5. This is a particularly desirable characteristic of the model, as traffic data can exhibit different patterns depending on the geographical area from where the data is sourced and it is important to choose the best parameters depending on the individual dataset.

$$p(N|constraint_k) = \int p(N|\lambda_0, \delta, \eta)\partial\lambda_0\partial\delta\partial\eta \tag{3.5}$$

This integral can be computed using the samples from the MCMC iterations; comparing the likelihoods obtained under different constraints, both on δ and on η, it is possible to judge how much freedom to allow the model to have. Since the expression contains the uncertainty over the parameters, there is no need to penalize explicitly a higher number of parameters.

3.2 Anomaly Description

Description deals with the actual multivariate categorical data; various techniques are employed to correlate increases in the time series to changes in the underlying dataset. Two main approaches are adopted, one extracting details about individual anomalies and another extracting common features among anomalies.

The first approach operates directly on the data; it uses the output of the time series detection to label specific time periods as anomalous and examines changes in the distribution of each categorical variable. Data from each anomalous period is compared to data from related periods (previous and past days at the same hours) that are deemed to be normal, highlighting the specific attribute values showing a significant increase; then techniques for anomalous pattern detection in categorical datasets, inspired by Das et al. (2008), are applied to test for correlations and interactions between variables and select the patterns to output as a description of the anomaly.

The second approach, undertaken on the basis of the results of the previous analysis, consists in selecting a subset of interesting variables and transforming the corresponding data into a multivariate time series, with the aim of applying dimensionality reduction techniques like PCA to gain a picture of the main trends. A simple anomaly detector, based solely on PCA, is then compared to the original model in order to judge which methods would be suitable for integration in real-time monitoring of the network.

3.2.1 Exploration of individual anomalies

Most of the anomaly detection literature concerning itself with multivariate data (for reference, see the review paper Chandola et al. (2009)) defines anomalies as single points that are outliers in respect to the rest of the data, while collective, or *group* anomalies, are addressed much more rarely. Some of the techniques used here were presented in works aimed at *detecting* anomalous patterns; however, it was decided to approach the detection otherwise because of the clear time-related patterns present in the data and the domain-specific definition of anomaly as a period with higher counts regardless of the composition.

The exploration is carried out in two complementary steps: first, each categorical variable is examined separately, comparing the distribution in the anomaly with that of the reference period; second, frequent pattern mining is used to identify associations between specific categorical values across different variables (sets of values appearing together), as in Brauckhoff et al. (2012).

For each anomalous period (a continuous stretch of 15-minute intervals flagged as anomalous) the corresponding portion of the call drop dataset is extracted; in order to provide a reference dataset to compare against, each of these portions is associated to a reference dataset, made up of data from the same time period in different days. For instance, if an anomaly is detected on Day 10 between 15:00 and 16:00, the reference will be built with the same hour between 15:00 and 16:00 from Day 9, 8, 7 and so on, skipping days that contain anomalies themselves. The number of reference days affects the data size of the reference period and therefore the time needed for computations; a balance needs to be struck between having an accurate picture of the normal activity and getting results quickly. A maximum amount of reference days is set depending on the amount of data available and on assumptions about its homogeneity (if the dataset is small or there is concern that normal patterns might be changing, a lower amount is preferable).

3.2.1.1 Comparing distributions

The basic idea, borrowed from Kind et al. (2009), involves selecting a few variables (each representative of one of the groups outlined in sec. 2.2.1) and representing the distribution of drop events by constructing a histogram for each of them, using the possible attribute values as bins; then histograms from anomalous periods (showing the count of records having the respective value in the period) are compared to histograms from the reference data (showing the average number of appearances of the same attribute in the reference periods). Probabilistic measures of difference between distributions, such as the Kullback-Leibler divergence found in Brauckhoff et al. (2012), were discarded because they emphasize relative, rather than absolute, differences, and are not defined when the support of the two distributions is not overlapping (in cases where some attributes only appear during the anomalous period, they are assigned probability 0 in the normal data: this yields an infinite divergence regardless of the number of occurrences, which is undesirable).

The comparison is based on a simpler measure, the Manhattan distance. Representing a portion of the original dataset labeled as anomalous with p, rows of the original dataset as r_j and choosing a categorical variable a, having n possible values $\alpha_1, \alpha_2, \ldots, \alpha_n$,

$\mathbf{x} = (x_1, x_2, \ldots, x_n)$ is the histogram for p, with $x_i = \#(r_j \subset p : a_j = \alpha_i)$.

Similarly, a histogram $\mathbf{y}$ can be defined for the same variable in the reference data. Representing the m chunks of data taken for reference as $q_k \quad (k \in [1, m])$, we can write

$\mathbf{y} = (y_1, y_2, \ldots, y_n)$, with $y_i = \frac{1}{m} \sum_{k=1}^{m} \#(r_j \subset q_k : a_j = \alpha_i)$

A second vector $\mathbf{s} = (s_1, s_2, \ldots, s_n)$ is then computed for the reference period, with s_i the standard deviation on each of the averages y_i across the m subsets.

Then $d_i = |x_i - y_i|$ is the contribution from attribute i to the Manhattan distance between the two histograms $d = \sum_{i=1}^{n} |x_i - y_i|$ and $t_i = \frac{d_i}{s_i}$ is a score of how unusual the deviation is. Attribute values whose absolute and relative deviation scores d_i and s_i are higher than specified thresholds are reported (thresholds for s_i can be determined with the same criteria as a statistical test, while thresholds for d_i depend on the level of detail desired by the user).

Variables with special treatment A particular treatment was reserved to the variables corresponding to the geographical location of the records. The interesting aspect is to determine whether or not the problems are confined to a specific subdivision of the network: if the problem is happening at the center of the network, many areas will see increases which are not ultimately interesting. Therefore it is sensible to introduce a measure of the concentration of the distribution across these variables: the kurtosis, computed on $\mathbf{x}$ as $\frac{\mu_x^4}{\sigma_x^4} - 3$, was chosen empirically as it correlated well with expert judgment.

3.2.1.2 Frequent pattern mining and co-occurrences

The previous step returns a list of abnormal attribute values for each individual variable, but the transformation of the data in histograms hides the underlying structure of each row. The most suitable techniques to uncover subsets of nominal values appearing frequently together in the same row belong to the family of association rules/frequent patterns mining, originally defined in Agrawal et al. (1993). Specifically, the example of Brauckhoff et al. (2012)was followed, using the Apriori algorithm to obtain maximally frequent itemsets (that is, itemsets whose supersets are not frequent); the amount of patterns output can further be reduced by only including those that contain values selected as anomalous in the previous step.

Seen that many frequent patterns are not specific to the anomalous period but also appear in the reference data, another filtering step of this output may be useful: as seen in Das et al. (2008), the support of each candidate pattern is computed also in the reference dataset, obtaining a 2x2 contingency table. Then, Fisher's exact test (Fisher, 1922) is applied to this table, to judge if the pattern is significantly more common in the anomalous data and therefore characterizes it.

In order to compute the support of the patterns in the reference data, where they may not be frequent, the standard algorithms were unsuitable; the cited work employs a dedicated, optimised data structure for computing contingency tables. Given the manageable data size, it was decided to adopt a simpler approach, described in Holsheimer et al. (1995), better suited to quick coding in the R environment. The data is transformed from the usual row-based format, where the value of each variable in the row is shown, to the dual representation of TID-lists: TID stands for transaction identifier and each TID corresponds to a row. Each possible attribute value is assigned a list of the rows it appears into; in order to obtain the rows in which a group of attributes appear together, it is sufficient to perform the set intersection of the respective TID lists, and simple counting yields the support.

Given the nature of the data, the amount of frequent patterns returned as output grows rapidly as more variables are added, therefore a small number of main columns was chosen for this step; another requirement was to then be able to inspect the association of the main variables with others that were otherwise disregarded. This was implemented similarly to frequent pattern mining, taking as input the specific sets of values to examine and using TID lists to explore their co-occurrences. In addition to the aforementioned statistical testing on the support, a measure of the confidence of each co-occurrence was computed: with A being the value of the main variable and B that of the other, $conf = Support(A \Rightarrow B)/Support(A)$, in other words how many of the records containing A also contain B (in association rule mining, the support of a rule is the number of records in the dataset that contain the rule). This will be useful for instance to determine if any single user is causing a significant proportion of the dropped calls in a geographical area, as this may indicate that the problem lies in the user device rather than in the network.

3.2.2 Dimensionality reduction and extraction of main patterns

The above techniques extract automatically interesting details about anomalous time periods, which are potentially useful for in-depth analysis of root causes and ongoing problems for the network or for subsequent correlation with other data sources in the network; however, it is also desirable to obtain quick, summary information of the main features of anomalous data. Dimensionality reduction techniques often prove very useful in unsupervised problems of this kind; however the most common ones, such as Principal Component Analysis, are not applicable to categorical datasets such as this one. In order to overcome this constraint, it was necessary to transform the data, in the process losing information about the structure of individual rows. Each categorical column was transformed into a set of indicator variables, one for each possible value, and the indicators were summed for every 15-minute period, obtaining a multivariate time series: structure is lost as it is infeasible to add terms for all the possible combinations of attributes appearing in the same row. Regression techniques having the total count for each period as output would be meaningless, as the response is simply given by adding all the counts for one variable; on the other hand, correlation analysis and PCA can show which attributes, both within the same variable and across different variables, grow together. Some principal components can then be associated to the normal behavior, while others encode spikes, local increases and other deviations; the attractiveness of PCA lies in the fact that it is possible to interpret the loadings in order to understand which variables influence each principal component score, and correlated variables contribute to the same component. Thus, a big component score corresponds to an extreme value of its contributing variables and it is possible to label anomalies in the total count according to the principal components that have high values.

In order to make PCA loadings easier to interpret, it is possible to apply the varimax rotation (Kaiser, 1958), a transformation that, while maintaining orthogonality, yields loadings with very high or very low coefficients, associating most variables to a single loading. This trades some precision in the representation for higher interpretability of the scores, with the latter being more desirable in this specific use case.

It is important to note, however, that the PCA approach does not retain the time component of the data: the time series is seen simply as a multidimensional dataset, with no importance given to the order of the rows, thus limiting the ability to recognize abnormal behavior that is only anomalous in relation to the time during which it happens.

3.3 Technical aspects

This thesis work has been carried out using the R programming language and software environment; Unix command line utilities were employed for data cleaning, pre-processing and subsetting in order to keep the data to a manageable size - the original, raw comma-separated values file weighing 2 GB. For the MMPP algorithm, I rewrote into R the MATLAB code released by the authors of Ihler et al. (2007), testing its results on the enclosed example datasets. The *ggplot2* plotting library was used for graphics, employing a “color-blind-friendly” palette, and frequent pattern mining relied on the *apriori* R package.

4 Results

4.1 Exploratory analysis

4.1.1 Main data

As previously mentioned, the standard way to study this data in the time domain is to look at the count of drops in every 15 minute period; this was maintained through most of the analysis. The time series is shown in Figure 4.1. The daily periodicity is immediately noticeable in Figure 4.2, with lower counts in the early hours of the day and peaks in the evenings; several spikes of various magnitudes are present, with a concentration during a week on either side of the fifth Sunday, during which every day presents an abnormal peak as compared to the rest of the time series. Figure 4.3 shows in higher detail a week-long portion of the data; each of the first four days present spikes in which the drop count reaches up to 5 times the usual values. The peaks appear at irregular times, even during the nighttime. Other, much smaller spikes can be seen in other days, while the overall pattern appears quite regular, with small differences between the overall shapes of days.

One important consideration to make is that the categorical data can be seen as compositional data: by transforming each categorical variable into a set of indicator variables, one for each possible value, and compressing the information into counts every 15 minutes, we will obtain a time series of counts, with the constraint that the total count shown in Figure 4.1 is always equal to the sum of the counts of the indicators for any one of the variables. This is shown in Figure 4.4, using the service group variable, showing how most of the dropped traffic is either normal or high-speed packet data (mobile internet).

By breaking down the total count into its parts, however, an increase in the total time series will correspond to increases in some categories for every variable; discerning which variables actually contain valuable information about the nature of the event

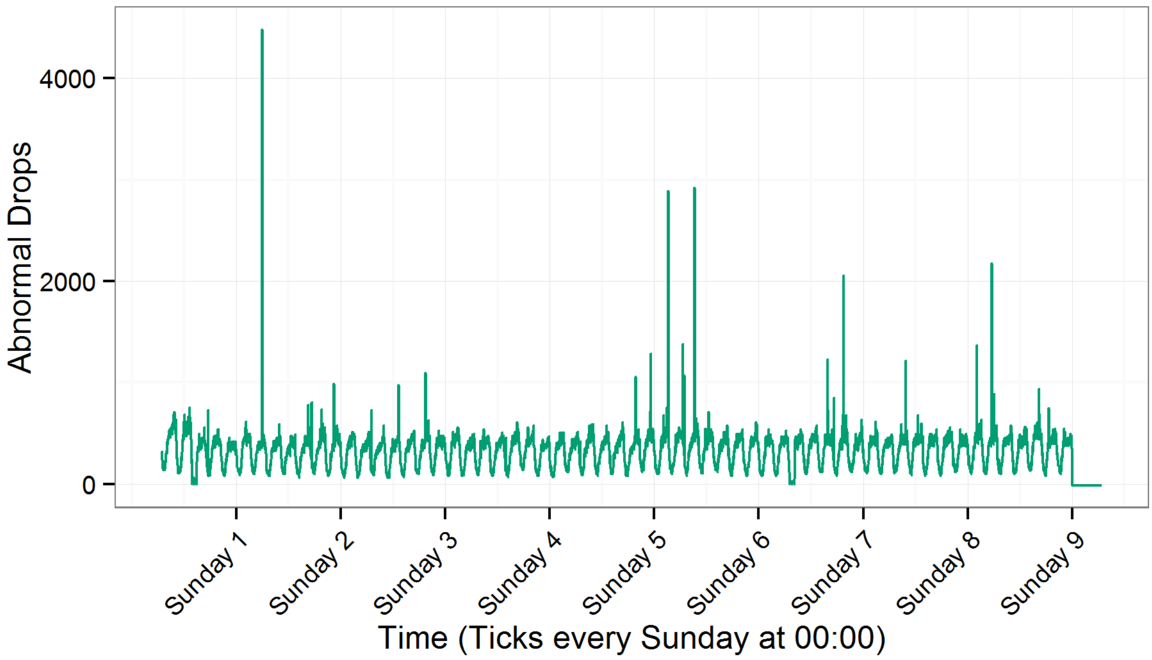

Figure 4.1: Time series of dropped call counts every 15 minutes.

that caused the increase would be desirable but is not always possible, even for domain experts.

Figure 4.5 shows how similar spikes can have very different underlying data; pictured in green are the counts of drops reporting one of the most common exception codes, 40, while in yellow is a rare code, 119. The spike on the fifth day corresponds to a raise in frequency of exception 40, while spikes on the seventh and first days are associated to a burst of the rarer 119 exception; yet another spike, on the third day, does not correspond to any of the two. This is a good example of a variable that gives valuable information: looking at service types the spikes are indistinguishable, while the exception code graph shows that two of them are associated with an uncommonly high number of occurrences of a specific exception. It would then be interesting to find out if these exceptions happen only with a specific type of connection or in a specific part of the network; this has been one of the focal points of this work.

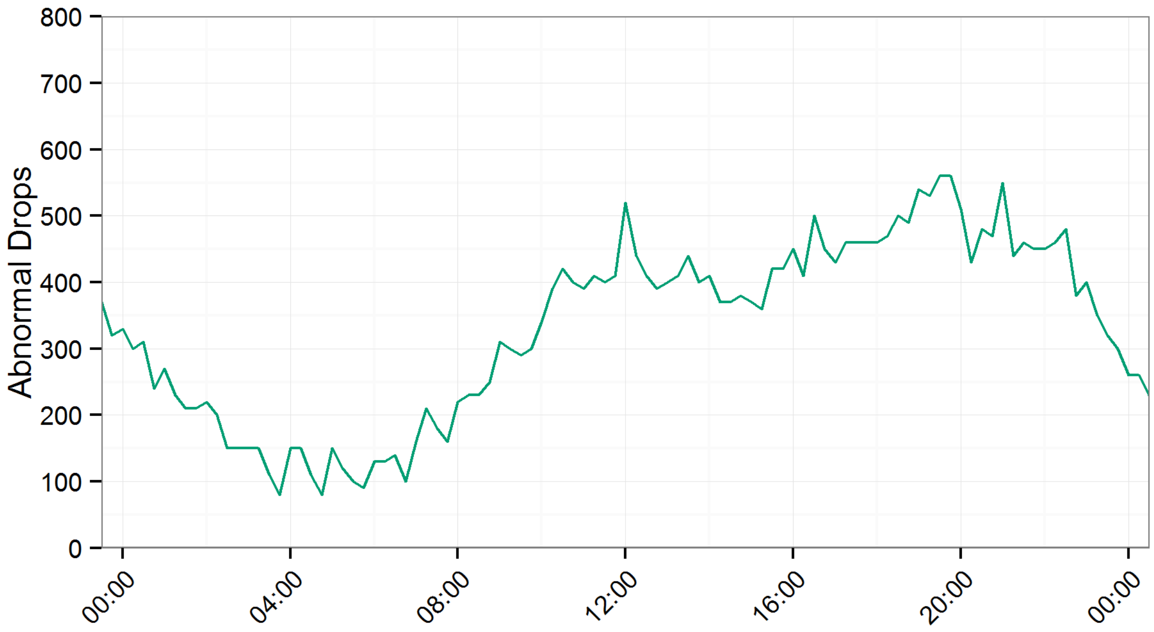

Figure 4.2: Detail of a regular day, showing the daily pattern

4.1.2 Secondary data

Another goal of the exploratory analysis was to understand whether or not it was necessary to include information about the amount of normal connections for every given time period, which is stored in a different data source, in order to understand the normal behavior and judge how necessary the integration of this source is. From Figure 4.6 it can be seen how the normal connection counts, in blue (rescaled in order to have the same magnitude as the abnormal ones) appear much more regular than abnormal drops counts, with only a few spikes of much smaller relative magnitude. Also, it can be seen how the amplitude of the oscillations is different, suggesting that there is a dependency between normal connections and abnormal drops: in other words, the ratio of connections that drop is not normally constant but depends on the amount of traffic on the network. This was tested with a simple linear regression, having the number of normal connections as dependent variable and number of abnormal drops as response variable, yielding a significant slope (the resulting plot appears in Figure 4.7). The presence of many outliers means that the standard assumptions for linear regression are violated, but this analysis is enough

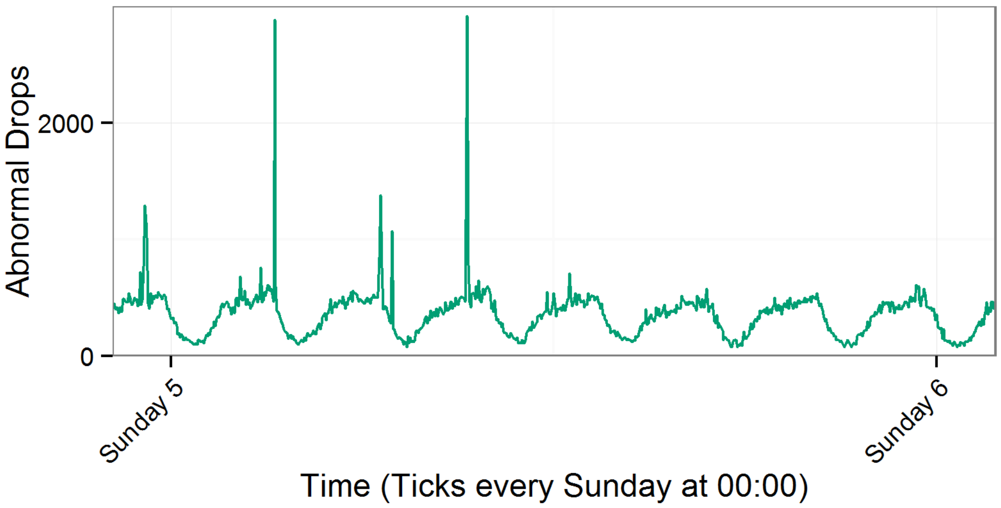

Figure 4.3: Detail of the sixth week.

to affirm that there is a relation. The correlation between the two variables is 72%, and this figure jumps to 87% when the most evident outliers are replaced by the median, indicating that most information in the two sets of counts is overlapping. For all these reason, and especially since the number of spikes and outliers in the abnormal drops data is much higher than in the normal connections data, I deemed it sufficient to concentrate on analyzing the abnormal drops data alone, since most departures from normal behavior can be seen and explained only in it. Moreover, the integration process required days-long parsing procedures to obtain the correct data, plus manual intervention to ensure alignment, and the information was limited to counts with no possibility to explore further, unlike with abnormal drops.

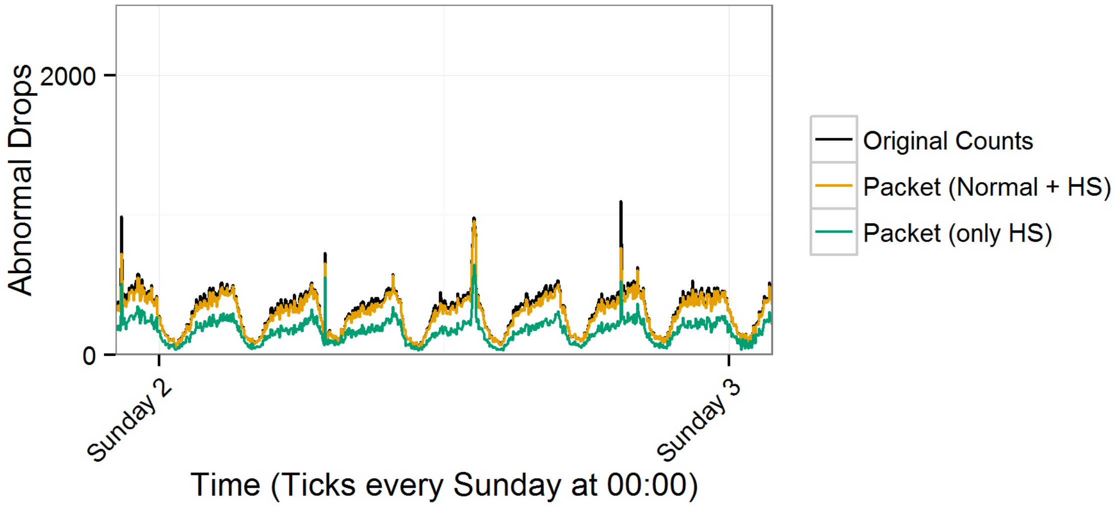

Figure 4.4: Cumulative breakdown of drop counts by service

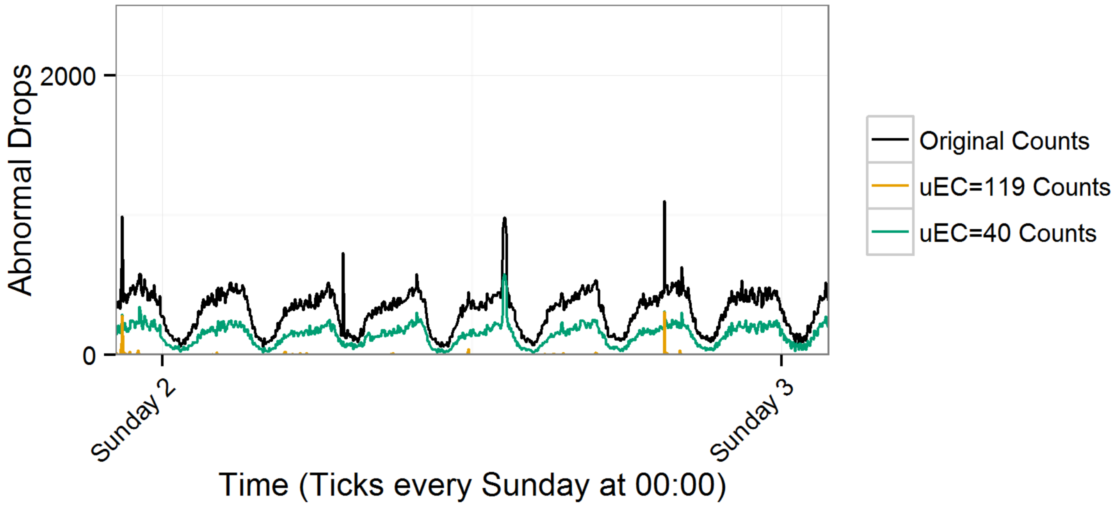

Figure 4.5: Breakdown of drop counts by exception code

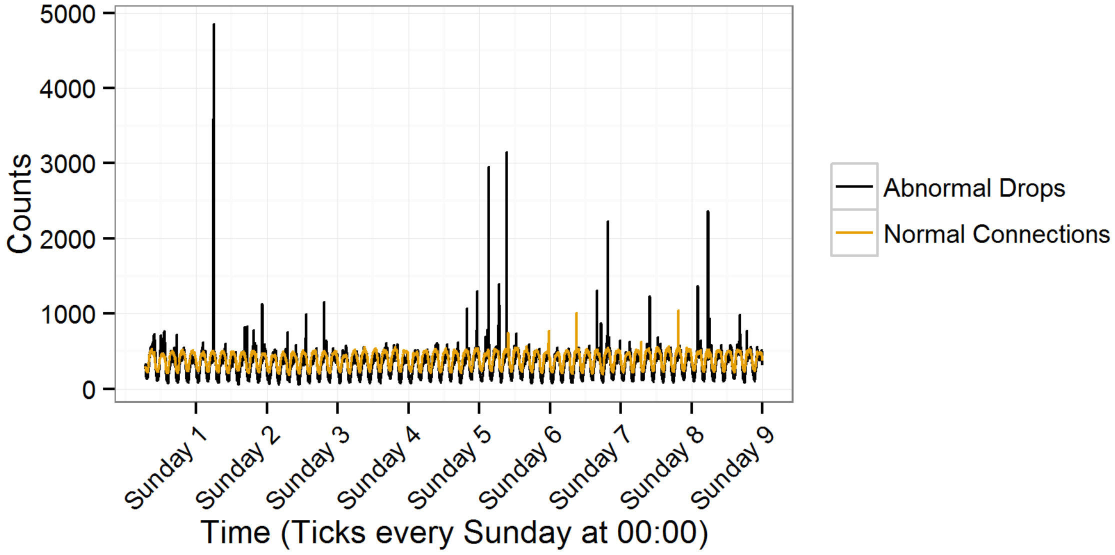

Figure 4.6: Overlapping abnormal drops and normal connections

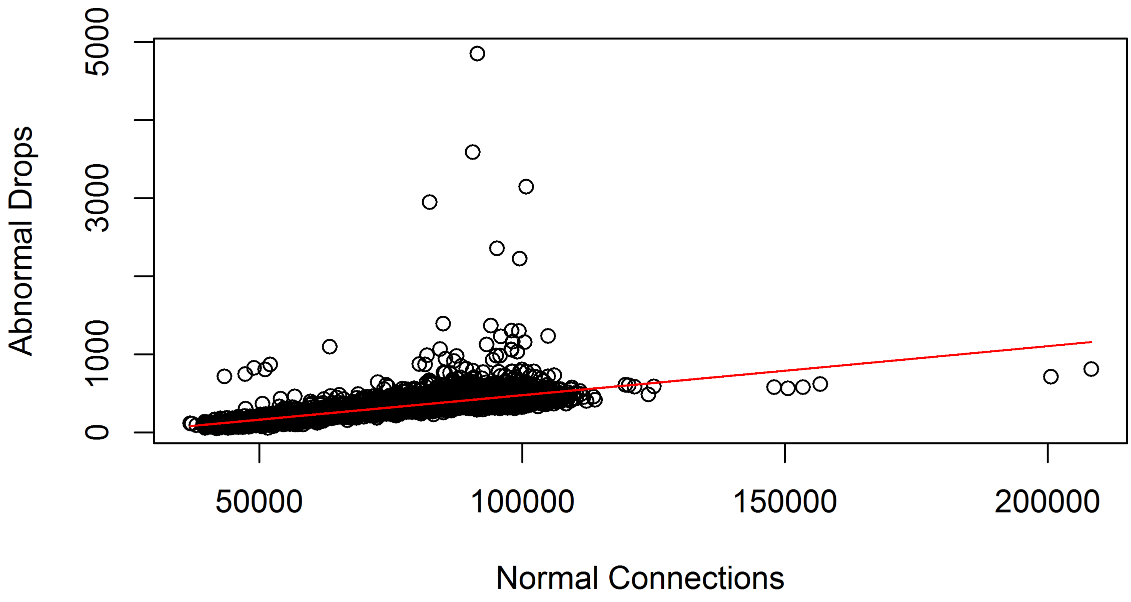

Figure 4.7: Scatterplot of abnormal drops versus normal connections, with linear regression line

4.2 Detection of anomalous periods using the MMPP model

4.2.1 Preprocessing

Fitting the MMPP model on data in the original scale presented problems with the convergence of the algorithm: due to the equivalence between mean and variance, the distribution becomes more and more flat as the mean increases. This means that the likelihood obtained with the "correct" mean that the algorithm estimates is similar to those obtained with slightly different mean values, causing the algorithm not to converge stably on its λ values and therefore impacting all other inference. Adjusting prior parameters did not relieve this problem, but a simple rescaling of the data eliminated the problem: all counts were divided by a constant K and rounded to the nearest integer, and model fitting was performed on the resulting time series. Eventually $K = 10$ was chosen as it is intuitive to grasp the original scale with powers of 10, the detection results were satisfying and a rounding of this magnitude did not lose any interesting details, as noisy variations in the data are themselves bigger. Figure 4.8 shows detection results on the original time series, with immediate evidence of non-convergence. An added benefit of rescaling is that it lowers the mean-to-variance ratio, which reduces problems related to overdispersion of the data (a common problem with Poisson models as they do not allow to specify mean and variance separately); once anomalous points are removed, the mean/variance ratios were computed separately or each of the 96 15 minute periods in a day, which are assumed come from the same distribution, yielding values between 1.2 and 3.8, ruling out overdispersion.

4.2.2 Choice of parameters

Once transformed the data to a suitable scale, the first step of fitting the MMPP model involved choosing how many degrees of freedom to allow for the normal behavior: the assumption from the data providers was that weekends would be significantly different from workdays, while the above plots only seem to suggest daily, rather than weekly patterns. The MMPP model was therefore fit to the dataset with various constraints on the day and time of day effects: each day with the same profile, different profiles for weekends and weekdays and different profiles

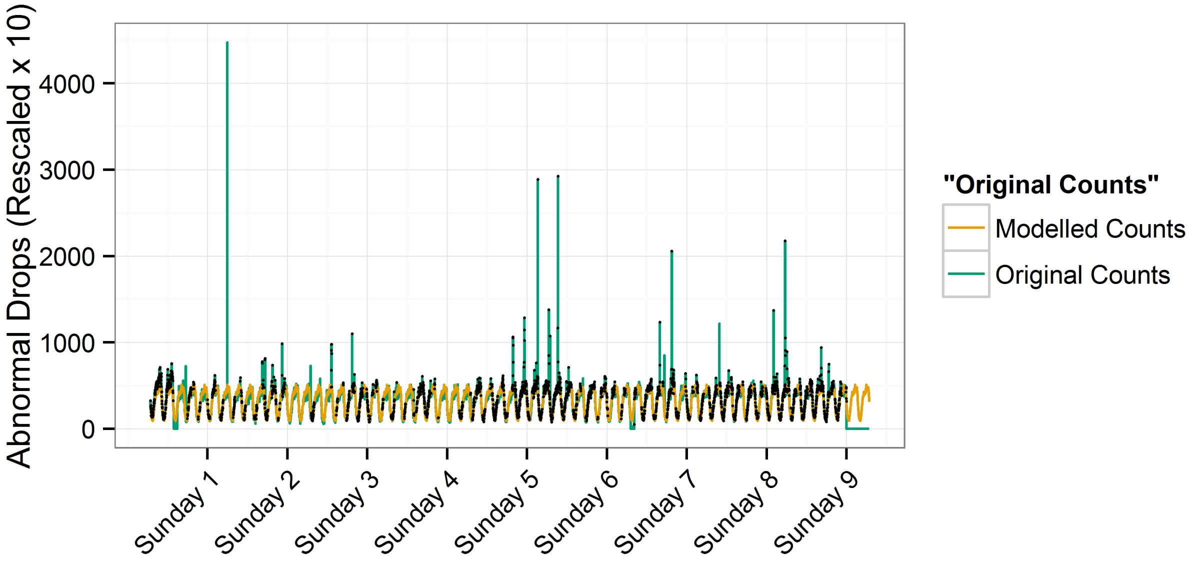

Figure 4.8: Output of the MMPP model run on the original (non-rescaled) dataset; notice how almost half of the points are labelled as anomalous, while some clear outliers are not.

for each day. The computation of the estimated marginal likelihood, shown in Table 4.1, confirmed the exploratory analysis: differences between days are not big enough to justify separate modelling, therefore the advantage of having a bigger ratio between data size and parameters makes the simpler model preferable.

	AllDaysSameShape	WeekendShape	AllDaysOwnShape
AllDaysSameEffect	-25904.00	-25903.00	-25918.00
WeekendEffect	-26121.00	-26172.00	-26136.00
AllDaysOwnEffect	-27902.00	-27902.00	-27913.00

Table 4.1: Table of log-likelihoods for different parameter choices

The prior transition probabilities for the Markov chain modelling the presence of events were $z_{0\to1} = 0.99$, indicating that we want jumps to the anomalous state to be rare, and $z_{1\to0} = 0.25$, indicating that we prefer the process to stay in the anomalous state once an anomaly is detected, but this jump should be much easier than the reverse.

4.2.3 Output

The estimated probability for each of the 5856 time points, obtained with 10 burn-in iterations, 100 sampling iterations and considering all days to have the same profile, is shown in Figure 4.9. For subsequent analysis, it was necessary to translate this probability in a binary decision rule; the choice was to consider points with $P(anomaly) > 0.5$ as anomalous. The 0.5 threshold was preferred to the more conservative 0.95 as the latter would exclude some group anomalies that, due to the stochastic nature of the method, were not detected reliably at every iteration. Figure 4.10, Figure 4.11 and are closeups on week-long sections of the data, demonstrating how both short-term, spike anomalies and smaller, continuous deviations from the normal behavior are successfully detected. 289 of the 5856 data points are labelled as anomalous, grouped in 37 sequences. 30 are less than two hour long (8 or less points), 5 are between 2 and 5 hours and the remaining 2 are 13 and 19 hours long.

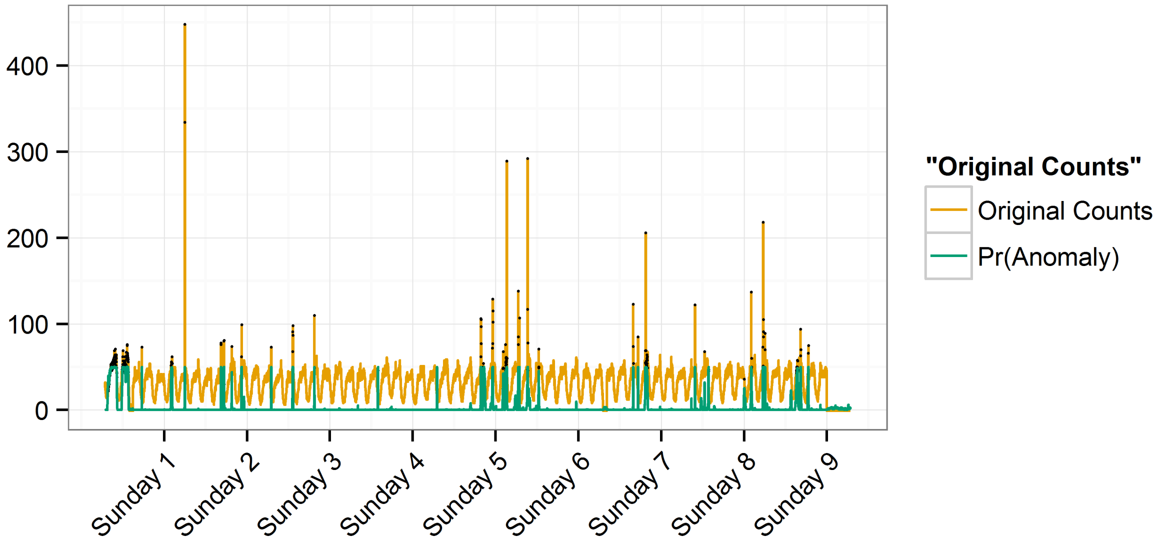

Figure 4.9: Plot showing the estimated probability of anomaly for each time point

In addition to the main dataset, a time series extracted from another dataset was also available. The output is shown in Figure 4.13; this data had a different nature, with the first days exhibiting very different behavior with respect to the rest of the

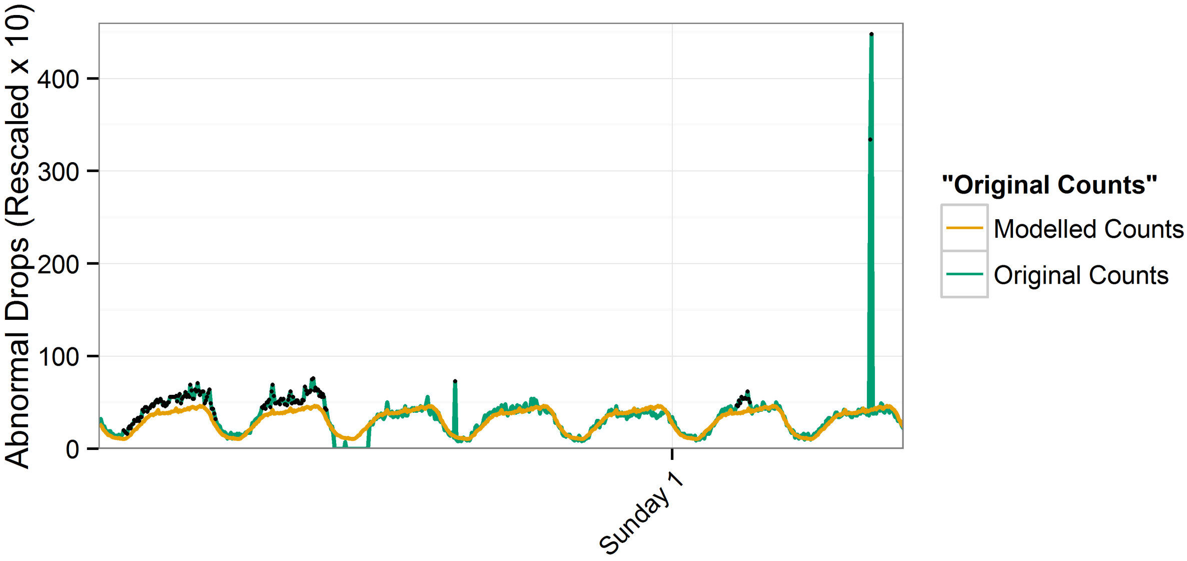

Figure 4.10: Close-up on the first week of the time series

time series. Nevertheless, the MMPP model was still able to derive its estimate of normal behavior from the right side of the data, labelling the whole left part as anomalous.

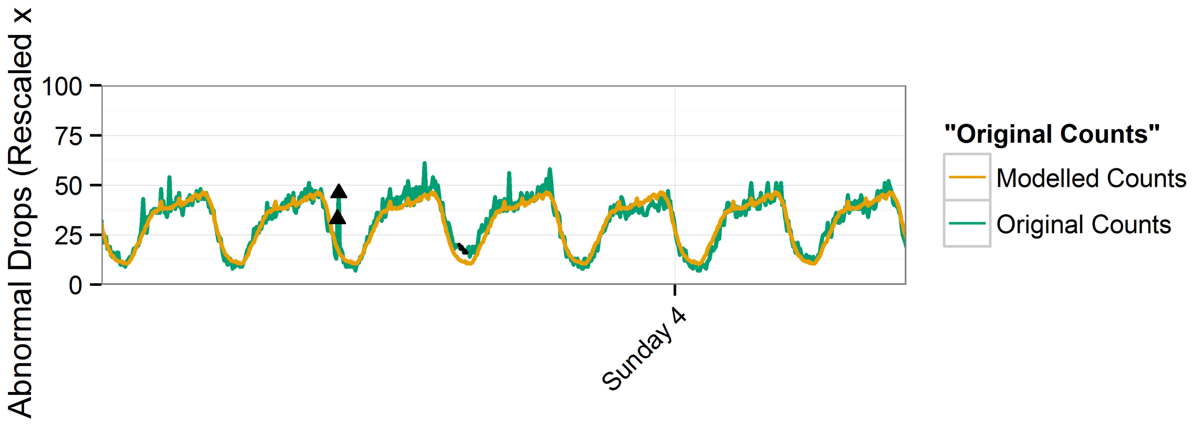

Figure 4.11: Close-up on the fourth week of the time series

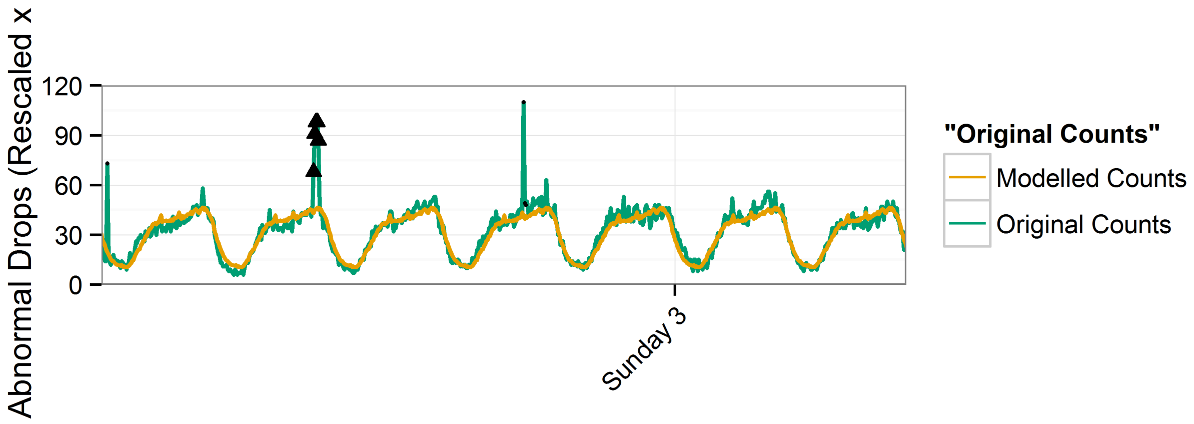

Figure 4.12: Closeup on the third week of the time series

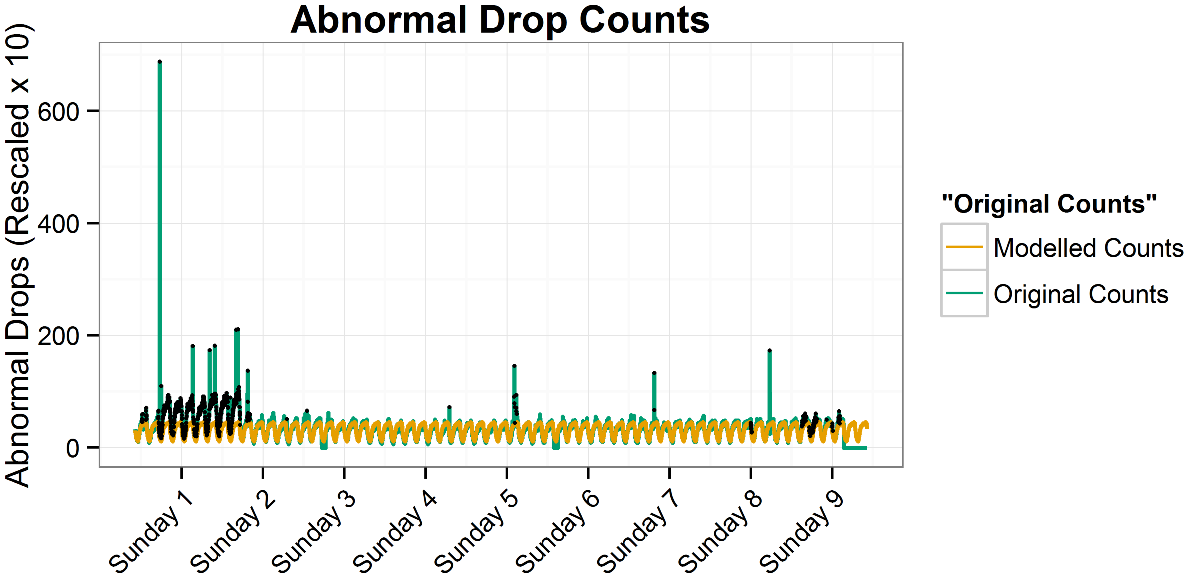

Figure 4.13: Output of the MMPP detection on a second dataset

4.3 Exploration of anomalous periods

In the following section, output regarding two anomalous time periods is presented; the techniques used are intended to summarize information for an expert user, who can then rule out or confirm own hypotheses about the cause of the problem. Only a subset of the analysed variables (already a subset of the available ones) is presented here. The first is the spike marked with triangles on the left-hand side in Figure 4.11, occurring during nighttime with a duration of less than 30 minutes (notably, the count wouldn't have been considered anomalous if occurring during the evening), reported in Table 4.2. Three exceptions and one service type are reported as meaningfully deviating, while nothing is reported about geographical distribution, since the flat shape suggests a problem happening centrally in the network controller. Barplots for the three variables are reported in Figure 4.14.

Frequent pattern mining, run on 4 variables with a 2.5% threshold on support, returned 98 patterns. Inspection of the output, which is not included for brevity, allows to see that code number 119, making up 15% of the total drops, is associated with high-speed data (PS_HSUPA) and speech calls (SPEECH), but not with standard data (PS_R99); that specific exception code never appeared in the reference data, therefore its expected frequency is 0 and all frequent patterns that include it are significant for the Fisher check on the ratio. As for code number 40, despite an increase in the overall number, the ratio in the anomalous period is 36% against the normal 44%, as the emergence of a large amount of records with different codes inflates the denominator; this means that most patterns involving code 40 do not pass the Fisher test. Without careful interpretation, this may give a distorted picture of the situation, but the benefit is that the most interesting patterns, those that increased in frequency in line with the overall increase of dropped calls, are highlighted. In this case, the only reported pattern involves a specific connection transition (C_TO_D), which can be singled out for further analysis as it appears to be one of the symptoms of the event that has caused an increase in the amount of dropped calls. Looking at the confidence analysis, the confidence of encountering C_TO_D given all the drops with exception 40 went from 20% to 40%, confirming that the increase was meaningful; this is given as an example of the possible relations that can be quickly checked by inspecting the description output. As will be discussed in chapter 5, only integration with other datasets and contribution of expert knowledge can lead to a better automation of the interpretation of these results; it appears clear, however, that this fault has its origin in the central system, showing errors that are never seen during normal operation.

	Variable	Value	Reference	StDev	Anomaly	Score
1	uEC	40	75.20	21.30	145.50	3.30
2	uEC	119	0.00	0.00	60.00	Inf
3	uEC	188	5.20	2.30	50.50	19.80
7	iuSysRabRelGroup	PS_HSUPA	101.60	25.60	301.50	7.80

Table 4.2: Table of deviating values for the first period

The second is the spike marked with triangles during the second day of Figure 4.12, occurring in the evening with a duration of 5 time points (between 60 and 75 minutes); its summary is reported in Table 4.3. Here 2 exception codes and 2 service types are reported, but the most interesting aspect is that a single grouping in the network controller, corresponding to a few cells of the network, reported as many drops as the rest of the network combined, and was the sole responsible of the anomaly: the total average number of drops went from 458 to 883 (a difference of 425) while the specified geographic area went from 14 to 415 (a difference of 401, almost as big as the total increase). Barplots for the three selected variables are shown in Figure 4.15, the last being especially interesting as it demonstrates how concentrated the anomaly was.

Frequent pattern mining, run with the same parameters, returned 80 patterns; patterns including the anomalous geographic area show how all main service types are affected, as well as one exception code, 902, that is the third most frequent in that area and only the fifth overall. More interesting is the related analysis of co-occurrences, which allows to see how the main geographic subdivision variable, *kRncGroup*, correlates with others in the same group. Two cells with consecutive identifiers, indicating close geographical proximity, are responsible for 95% of the drops reported in the specific RncGroup: in a real setting this could indicate that a concentration of many people inside the coverage area of the antennas has led to an overload of the network, with a subsequent impact on its ability to retain phone calls. In that case the blame for the decrease in performance could be attributed to external factors; if multiple such events were reported it would be advisable to investigate the causes and possibly install more antennas in the area. However, there were no other anomalies regarding those cells in the dataset.

	Variable	Value	Reference	StDev	Anomaly	Score
1	uEC	40	223.10	13.20	492.20	20.40
2	uEC	163	81.80	9.70	137.80	5.80
3	kRncGroup	15	13.70	4.80	415.80	84.20
8	iuSysRabRelGroup	PS_HSUPA	259.90	19.70	519.20	13.20
9	iuSysRabRelGroup	PS_R99	178.10	11.10	337.80	14.30

Table 4.3: Table of deviating values for the second period

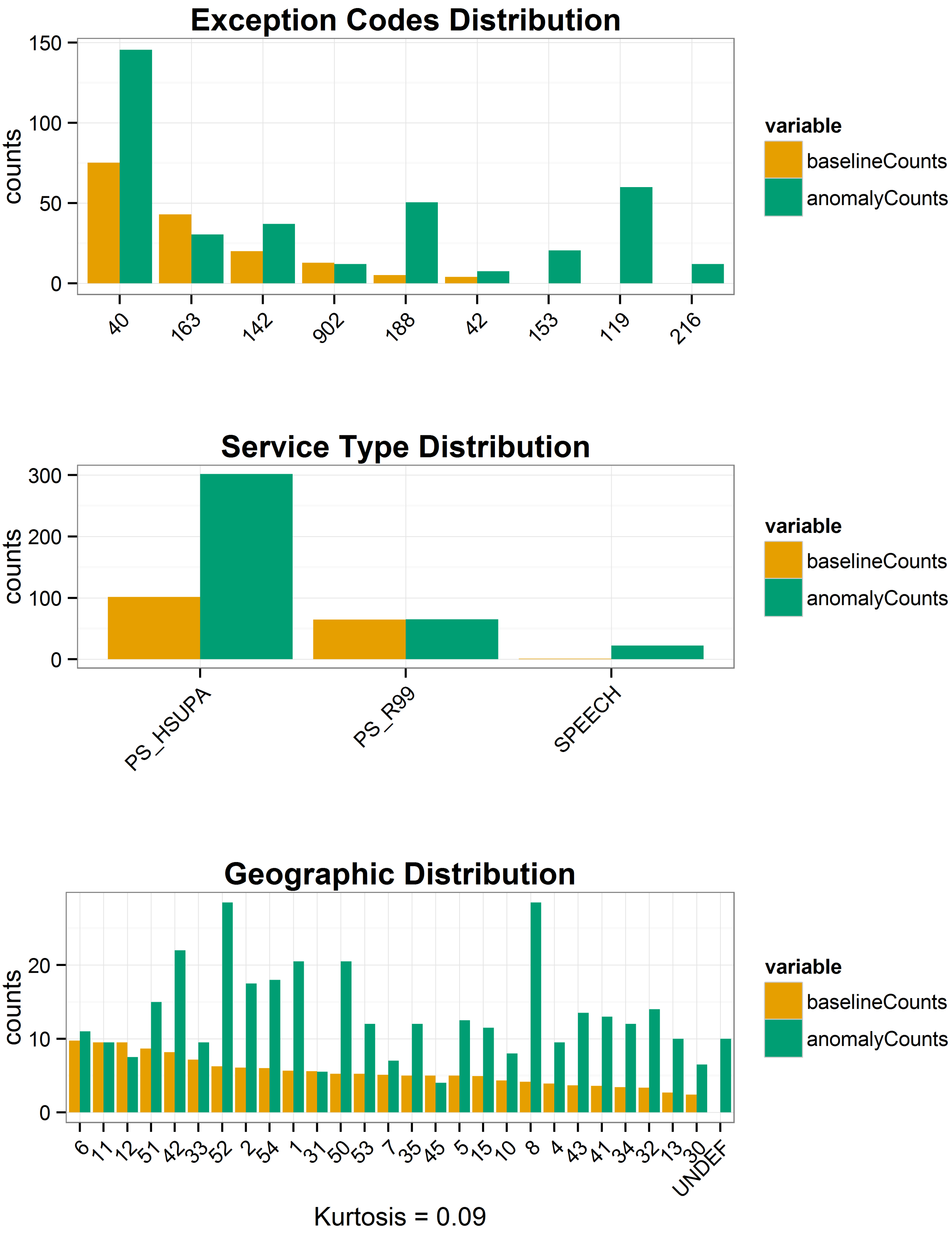

Figure 4.14: Barplot of distribution across three variables of interest for the first anomaly

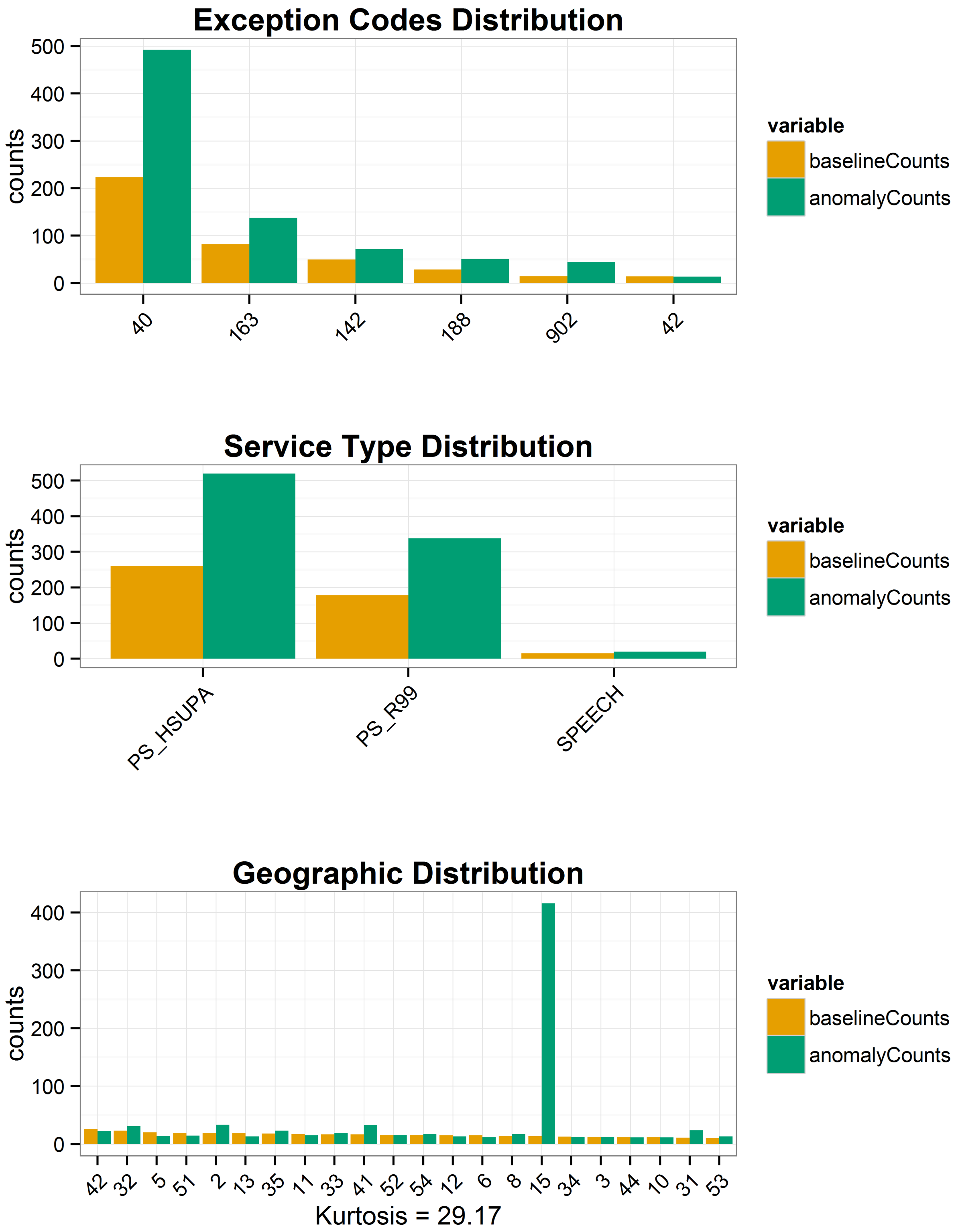

Figure 4.15: Barplot of distribution across three variables of interest for the second anomaly

4.4 Analysis of main patterns common to anomalous periods

The findings of the previous steps allowed to select some interesting variables, as well as features obtained from others, that can summarize the main interesting patterns of behavior in the data; these were transformed in a multivariate time series, as described in sec. 3.2.2. Exceptions and service types related to phone calls were retained, eliminating attributes appearing less than 50 times in the 2 million rows; then two new variables were added: *NoConnChng,* the amount of drops not associated with a transition between connection types, and *MaxRncMod*, the number of drops in the busiest area at every time step. Plotting the correlation matrix of the resulting 39-dimensional dataset, with rows ordered by hierarchical clustering (a common visualization technique) confirms some previous intuition: the bulk of exception codes (from 221 to 188) can be thought of as having similar origin, as they show quite big correlations between them. All the main, most frequent exceptions, which are found in all periods, belong to this group. Then in the bottom right corner there is a group of exception codes that are in many cases (including one of the two presented in sec. 4.3) unique to anomalous periods, and characterize a subset of all anomalies. These show very strong correlation, indicating that they are likely to represent the same underlying phenomenon, and are strongly associated with losing phone calls (these are prioritised during normal operation and it is intuitive that they are only dropped in large numbers when a severe problem happens). Other smaller groups exist, for instance two codes that correlate with high activity in a geographical area and therefore may indicate congestion, and there are also a few codes that seem to be completely unrelated to others, for example because they appear only once in a big spike (87).

PCA can be expected to summarize the structure of the data; in order to retain information on the size of the variables and prioritize more frequent attributes (which have higher counts) it was decided to run the algorithm on unscaled data using the covariance matrix; this is legitimate as all variables have the same scale. Results of the PCA are shown in Table 4.4; as the proportion of variance explained decreases substantially around the 10th component, following components can be discarded without loss of interesting information. By inspecting the PCA scores, it is clear that the first principal component accounts for the main daily pattern in the time series, while subsequent components capture deviations. Since the main focus of

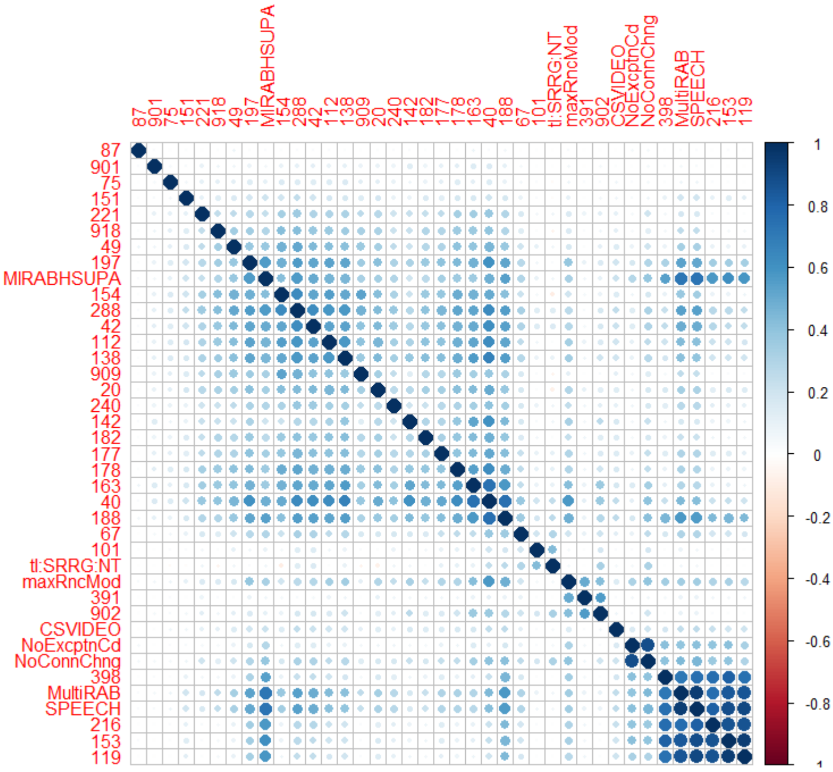

Figure 4.16: Correlation matrix between time series of appearance counts of chosen attributes

the work is to obtain easily interpretable results, the scores obtained after applying the varimax rotation to the first 9 components are reported (see Figure 4.17); the first component corresponds to the most common exception, 40, while the second captures 119, 153 and SPEECH drops, the most important members of the cluster in the correlation plot. The third component corresponds to the cluster of variables related to congested cells; 4th, 5th and 7th capture other variables related to common exception codes, the 8th some interaction between common and bursty exceptions, while 6th and 9th correspond to rarer bursty exceptions (missing data and exception code 87 respectively).

A simple anomaly detector is constructed using these 9 scores, by labeling a point as anomalous for a component if its score is more than 3 standard deviations off the mean (which is always zero); the results in Figure 4.18 show how most of the anomalies shown by the more complex time series model are also captured by this simpler technique, with the exception of small, time-of-day dependent anomalies (increases during night-time that would be normal during the day). The PCA detector also labels points that are normal for the time series analysis: some of these appear normal to human inspection and can be seen as false positives, while others correspond to single anomalous points that are not big enough to satisfy the prior definition of anomaly given for the MMPP model.

	StandardDeviation	ProportionOfVariance	CumulativeProportion
1	83.74	66.01	66.01
2	46.39	20.26	86.28
3	20.99	4.15	90.42
4	17.16	2.77	93.20
5	13.49	1.71	94.91
6	12.52	1.48	96.39
7	9.17	0.79	97.18
8	8.41	0.67	97.84
9	8.04	0.61	98.45
10	7.23	0.49	98.94
11	4.50	0.19	99.14
12	3.74	0.13	99.27

Table 4.4: Summary of PCA

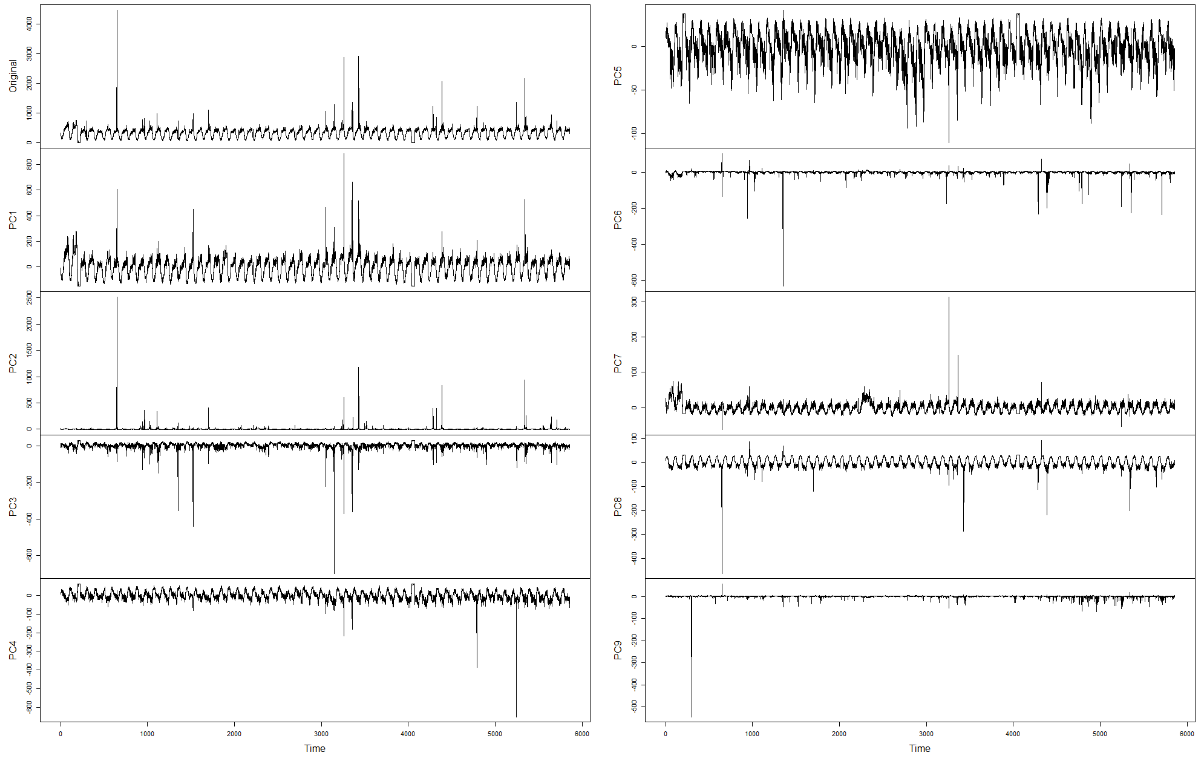

Figure 4.17: Plot showing the original time series and the scores on 9 transformed principal components

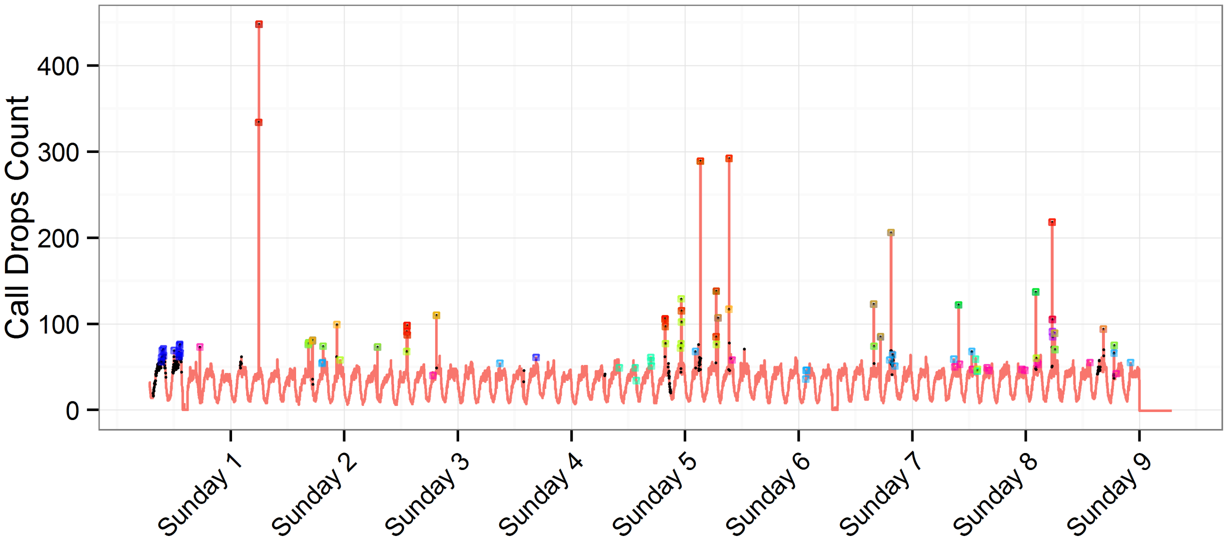

Figure 4.18: PCA anomaly detector (colored squares) compared to MMPP detector (black dots)

5 Discussion

To our knowledge, this thesis work has been the first data mining approach to anomaly detection in the specific type of dataset considered; as such, it can be seen as a pilot test of suitable methods, rather than a thorough, definitive solution to the problem. Prior to this, there was little systematic knowledge of what could be interesting hypotheses to test or informative patterns to find; for this reason, it was preferred to explore various techniques, some proving more effective than others, rather than develop extensively a single method. Another element that complicated the choice of the methods was the format of the dataset: having originally been developed for inspection with spreadsheet software, this data is not readily comparable to what is commonly analyzed in standard literature. Works on similar datasets exist and were used as inspiration, but differences were sufficiently big that it was not possible to simply replicate their methodology.

A large amount of time was spent on understanding the data, determining which techniques showed the most promise and which features were worthy of being retained; goals constantly shifted every time the methods revealed more about the structure of the dataset. For instance, a requirement that appeared to be very important at the start was to be able to analyze many variables, which led to the inclusion of pattern mining and co-occurrences analysis as a big part of the work.

However, after examining the output it became clear that anomalies could be more easily characterised with simpler methods, while frequent patterns provide additional information that is only useful when analyzing the details. These techniques were retained as a helpful analysis tool, to improve the way in-depth analysis is performed by troubleshooters, but ultimately contribute a smaller part of the interesting results than originally expected. Histogram analysis itself was originally considered as a tool to detect anomalies, before turning into one of the more easily interpretable exploratory techniques; using the Manhattan distance, despite it being a relatively crude approach compared to a statistical divergence, made it possible to focus on the single attribute values that define anomalies and are ultimately more important to the end goal of the work.

Despite not being exhaustive or systematic, histogram analysis and pattern mining were nevertheless very useful for understanding the dataset and selecting interesting features for further analysis. Among the 37 anomalous periods found in the development dataset, the analysis exposed many different patterns, from geographically concentrated episodes to central failures. Even though various periods showed unique characteristics, common features to many anomalies were identified, in terms of exception codes present and areas affected.

Having mentioned exploratory techniques, it is time to comment on the other methods employed in this work. Specifically, the MMPP method was able to give a solid, understandable foundation with which other, more empirical methods could be combined and compared; nevertheless, some considerations have to be made about its effective applicability in automated analysis. One possible problem regards its suitability for continuous monitoring of the data: the algorithm in its current form does not work *on-line,* therefore it has to be retrained every time new data is added. Also, the concepts it is based on are quite sophisticated and a substantial amount of data is required for training; as the data was found to be quite regular, it might be sensible to consider substituting the model with a simpler, threshold-based model built using the insight gained from MMPP. The resulting model would be less powerful or attractive from a purely statistical viewpoint, but it could prove faster and easier to maintain. One indication of the aforementioned regularity of the data was the fact that the estimated λ computed by the MMPP model for each 15-minute part of the day and the median of the corresponding population had a correlation of 0.9992, proving both that the inference was working well and that a much simpler technique would have given a good estimate of the expected normal frequency of call drops. However, as new datasets are collected and tested, the need for flexibility is highlighted: in many cases, significant differences between workdays and weekdays can be observed, confirming the validity of choosing a more complex model such as MMPP.

Regarding the PCA techniques, they were the ones that showed the biggest potential of solving with a single method the two problems of detecting anomalies and understanding them; unfortunately they were introduced late in the development of the work, as for a long time it appeared more important to retain the full structure of the data and operate on the categorical database itself, and therefore they could not be fully developed. More attention would have to be devoted to choosing the right parameters and settings, for instance whether or not to scale the data (scaled data gave better prediction but complicated interpretation) and how to construct the anomaly detector from the component scores. These decisions would require a more thorough literature review, as well as extensive testing on more datasets; the recent emergence of datasets with more complex patterns of normal behavior, while strengthening the case for choosing the MMPP model, further complicates the adoption of the PCA approach as a possible single solution. Canonical Correlation Analysis was also briefly considered as an alternative to PCA, and ultimately discarded due to longer processing times that made tuning parameters and assessing results infeasible during the late stage of development of this work; this might represent an interesting future development.

As for implementation, which is ultimately an important practical concern, the performance achieved is undoubtedly satisfying: execution of the full stack of developed R scripts, on a single Intel Core i5-3427U 1.8 GHz CPU, takes less than 15 minutes on a dataset of about 2 million rows and 388 megabytes of compressed size. This allows for deployment of the methods in automated analysis without having to migrate the code to faster programming languages and indicates how the R coding in this thesis has been efficient.

A final consideration to be made is that having separated anomaly detection from subsequent description makes it possible to use the database mining techniques for other, related tasks. For instance, it is often desirable in troubleshooting to compare two pre-defined periods of time, searching for the emergence of new patterns brought on by system changes or new user devices. Collection of data for this purpose is still underway, therefore this use case cannot be reported here, but the techniques have been developed to be readily adaptable.

6 Conclusions

With this work being, to our knowledge, the first step towards applying data mining to troubleshooting of dropped calls, it is important to focus the conclusion on the wider picture and the work ahead. Evaluating results on their own is complicated, mostly due to a lack of ground truths to be used for comparison - a situation common to many unsupervised learning problems.

In this case, those ground truths could be found either by human evaluation of many sample datasets or by correlation with other data sources. The former is currently a very time-consuming operation, but this work can accelerate it by bringing interesting patterns to the attention of the domain expert. The latter is a very interesting and promising possibility for future development, moving towards a wider adoption of data mining techniques in troubleshooting.

However, the associated challenges cannot be underestimated: in the context of a company managing huge systems, obtaining and integrating data is very complex. This thesis has developed methods to extract interesting features from one data source; once they have gained acceptance, it will be possible to set up joint monitoring of the same network with different tools and compare anomalies in dropped calls with changes in performance reporting. Another interesting development would be to follow the calls which we see are dropping from the beginning, as was done in Zhou et al. (2013), to understand the phenomenon in a way that cannot be achieved with this dataset alone.

To sum up, while the methodologies proposed are by no means exhaustive, this work has succeeded in leading to a better, more systematic understanding of the properties of dropped calls data, paving the way for further analysis which will be able to benefit from integration with other datasets.

Bibliography

Agrawal, R., Imieliński, T., and Swami, A. (1993). Mining association rules between sets of items in large databases. *SIGMOD Rec.*, 22(2):207–216.

Aktekin, T. (2014). Call center service process analysis: Bayesian parametric and semi-parametric mixture modeling. *European Journal of Operational Research*, 234(3):709 – 719.

Baum, L. E., Petrie, T., Soules, G., and Weiss, N. (1970). A maximization technique occurring in the statistical analysis of probabilistic functions of markov chains. *The Annals of Mathematical Statistics*, 41(1):164–171.

Brauckhoff, D., Dimitropoulos, X., Wagner, A., and Salamatian, K. (2012). Anomaly extraction in backbone networks using association rules. *Networking, IEEE/ACM Transactions on*, 20(6):1788–1799.

Chandola, V., Banerjee, A., and Kumar, V. (2009). Anomaly detection: A survey. *ACM Comput. Surv.*, 41(3):15:1–15:58.

Das, K., Schneider, J., and Neill, D. B. (2008). Anomaly pattern detection in categorical datasets. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '08, pages 169–176, New York, NY, USA. ACM.

Ericsson (2012). Traffic and market report.

Fisher, R. A. (1922). On the interpretation of chi-squared from contingency tables, and the calculation of p. *Journal of the Royal Statistical Society*, 85(1):pp. 87–94.

Holsheimer, M., Kersten, M., Mannila, H., and Toivonen, H. (1995). A perspective on databases and data mining. In *In 1st Intl. Conf. Knowledge Discovery and Data Mining*, pages 150–155.

Ihler, A., Hutchins, J., and Smyth, P. (2007). Learning to detect events with markov-modulated poisson processes. *ACM Trans. Knowl. Discov. Data*, 1(3).

Kaiser, H. F. (1958). The varimax criterion for analytic rotation in factor analysis. *Psychometrika*, 23(3):187–200.

Kind, A., Stoecklin, M., and Dimitropoulos, X. (2009). Histogram-based traffic anomaly detection. *Network and Service Management, IEEE Transactions on*, 6(2):110–121.

Weinberg, J., Brown, L. D., and Stroud, J. R. (2007). Bayesian forecasting of an inhomogeneous poisson process with applications to call center data. to appear. *Journal of the American Statistical Association*.

Zhou, S., Yang, J., Xu, D., Li, G., Jin, Y., Ge, Z., Kosseifi, M., Doverspike, R., Chen, Y., and Ying, L. (2013). Proactive call drop avoidance in umts networks. In *INFOCOM, 2013 Proceedings IEEE*, pages 425–429.

LIU-IDA/STAT-A--14/005—SE