# Time series forecasting used for real-time anomaly detection on websites

author: **Georgios Galvas**

A thesis presented for the degree of
MSc Business Analytics

VU supervisor: Evert Haasdijk
VU second reader: Sandjai Bhulai

Faculty of Sciences
Vrije Universiteit
Amsterdam, Netherlands
October 2016

# Abstract

The development of new methods of manipulating big data has made it possible for users to constantly monitor the traffic behaviour of networks and websites, as well as enabling them to manage and identify potential failures, or even intrusions. Due to the increasing complexity of computer networks and the great number of traffic-behaviour time-series data generated in real time, manual inspection of the aforementioned by a human factor is rendered impossible. To deal with this problem, anomaly or failure detection models have been developed to identify potential deviations from pre-existing behavioural patterns. In the current paper we are presenting an anomaly detection model for identifying potential errors or failures in websites. Since the data of our metrics is time series data, we first introduce forecasting methods for time series. The focus will only be on the exponential smoothing family techniques, especially the Holt-Winters model for time series with trend and seasonal variation, and Taylor's models which are designed to account for multiple seasonality patterns. A comparison is made between these models in terms of forecasting accuracy and the results are presented. By using these forecasting models to indicate the expected future behaviour of our metrics, we have built an anomaly detection model based on the observation that the residuals between the forecasts and actual values follow a Gaussian distribution centered around zero.

**Tags:** website performance, time series, exponential smoothing, Holt-Winters, Taylor, anomaly detection, Gaussian distribution.

# Declaration

I, Georgios Galvas, declare that this thesis titled, 'Real-time short-term time series forecasting used for anomaly detection on websites' and the work presented in it are my own. I confirm that:

- □ This work was done wholly during the master project *Business Analytics*.
- □ Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- □ Where I have consulted the published work of others, this is always clearly attributed.
- □ Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- □ I have acknowledged all main sources of help.
- □ Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

# Aknowledgements

This internship report was composed as a part of the *Business Analytics* Master program at the Vrije Universiteit of Amsterdam. The goal of the internship is to provide in depth experience on the latest methods of today's business environment, tackled with tools from the field of Mathematics and Computer Science, the intersection of which together with Business concepts composes the program of *Business Analytics*. The current thesis report contains the results of my research performed at *MeasureWorks*, a company offering services and products having as its main goal the improvement of the performance of its clients' websites.

I would like to thank Jeroen Tjepkema, CEO and founder of *MeasureWorks*, for giving me the opportunity to undertake my internship in the company, for his generosity, benevolence and trust in me. Furthermore, I am much obliged to Fernando Flores, data scientist and my supervisor in the company, for introducing me to the company's work, for providing me with the background knowledge of the existing models and the dataset I worked on, together with his useful aid on technical matters of the software I used. Special thanks also to all the employees of *Measureworks*, for their willingness to help whenever needed. Finally, I would like to thank Evert Haasdijk, assistant professor and my supervisor from the Vrije Universiteit, for his guidance and his useful recommendations, essential to the fulfillment of my project.

Georgios Galvas
October 2016,
Amsterdam

# Contents

# CHAPTER 1

# Introduction

In a rapidly developing and highly technological world, "we are all now connected by the Internet, like neurons in a giant brain" as Stephen Hawking stated. In this giant brain, the websites give sense and meaning to its existence through the variety of educational, communicational and entrepreneurial purposes they fulfill. Nowadays, websites are central in the distribution of information, exchange of goods and services as well as the monitoring process of everyday socioeconomical activities. Thus, they should have a consistent and high-speed performance in order to satisfy the users' needs and reduce discontent. With the purpose of supporting websites' smooth functionality, downfalls and errors should be immediately identified and overhauled by *anomaly detection* tools. The demand for efficient anomaly detection tools becomes imminent since the new kinds of data (i.e. sensor data) emerge to mound on top of the growing volumes of the existing ones.

*MeasureWorks* is a Web-performance solutions provider. For a wide range of Dutch companies, they provide performance tooling and consultancy to make websites faster, more reliable and ultimately deliver more conversion. The company, was founded in 2008, and for the past 2 years has been developing its own technology and products. Based on a real time tracking mechanism they have built algorithms that can predict website outrages, and track online sentiment to provide insight on these potential downfalls. This technology is currently available to a small set of clients. The current research is conducted on this downtime prediction platform, working with large sets of real time tracking data from a specific client.

## 1.1 Time series anomaly detection

When searching at the Oxford university dictionary about what an anomaly is, we find the following definition:

| | |
|---|---|
| **anomaly** | *something that deviates from what is standard, normal, or expected* |

*Anomaly detection*, as Dunning and Friedman [2014] highlight, is the science of *"spot-*

*ting the unusual, catching the fraud, of discovering the strange activity"*. It is carried out by a machine-learning program and is the method of finding any abnormalities in the behaviour of a system or identifying outliers from a set of observations. Could it be possible to identify abnormal observations based on the known and used methods? To effectively answer this, supervised machine learning techniques such as classification and their ability of detecting anomalies must be called in question. The primary idea would be to gather instances of the normal and abnormal kinds of data. This process would facilitate the classifier to be trained efficiently in order to correctly classify future observations and subsequently identify the potential abnormalities.

However, by tackling the problem like this, a weakness is revealed: the inability to identify new forms of anomalies previously unobserved. The process of retraining the classifier, using some examples of this new irregularity, would have to be repeated, a time-consuming and costly procedure during which the classifier would be incapable of detecting these new kind of anomalies. Thus, the method of classification cannot be applied due to the fact that the characteristics of an anomaly are generally not known. In a nutshell, according to the writings of Dunning and Friedman [2014]:

*"Anomaly detection is all about finding what you don't know to look for"*

In contrast to other methods, anomaly detection tools are capable of identifying new forms of anomalies, previously unobserved. By knowing what *normal* is, they can define anything different or "far" from that normal to be labeled as *anomalous*. We can say then that an anomaly is defined by contrast to what normal is. With this reasoning, to use the exact words of [Dunning and Friedman, 2014], the important aspect of anomaly detection can be summarized in the following words:

*"Anomaly detection is based on the fundamental concept of modeling what is normal in order to discover what is not"*

Consequently in our context, the challenge of identifying potential anomalies in the performance of a website can be viewed as a problem regarding the creation of a model describing a website's ideal behaviour and its comparison to the observed one.

The data we have been working on, as is more extensively explained in the next chapter, is time series data. According to Brutlag [2000], detecting anomalous behaviour in time series data consists of three stages, each one building on its predecessor:

1. An algorithm for predicting the values of a time series one time step ahead.

2. A measure of deviation between the predicted values and the observed values.

3. A mechanism to decide if and when an observed value or sequence of observed values is 'too deviant' from the predicted value(s).

In the present paper we are only focusing on the *class of exponential smoothing models* and not on the *ARIMA (AutoRegressive Integrated Moving Average)* when forecasting the future values of a time series. The reason for this is due to the simplicity and transparency of the first, in combination with a greatly satisfying performance. Furthermore the investigation focuses on single time series forecasting only (i.e. we do not consider an explanatory variable for the time series $y_t$), and the reason for that is that our main goal is to perform real time forecasting and anomaly detection. Thus ARIMA methods are not appropriate due to their computational complexity (see paper Au et al. [2011]). On the other hand, exponential smoothing methods, and especially the Holt-Winters method and its variations are known for their adaptable, robust character and their straightforward implementation.

## 1.2 Research questions

The model that ***MeasureWorks*** currently uses for time series forecasting is the *Simple Exponential Smoothing* or, as most commonly known, the *Exponential Weighted Moving Average* (*EWMA*). Even though a robust forecasting method, EWMA produces accurate forecasts for constant level time series only, with no signs of seasonal repetitions. One of our objectives in this paper is, after a thorough investigation and understanding of the available dataset, to select and implement the most appropriate to the data model, the one that outperforms *EWMA* in terms of forecasting accuracy. Our first and main research question then is the following:

1 *According to the behavioural pattern of the measure variables, what is an appropriate forecast method for each metric and to what extent does it improve the forecast accuracy of the already existing predictive model?*

The main goal of Measureworks is to guarantee maximum website performance to its clients. Therefore, after the selection of the most appropriate time series forecasting tool for all the metrics that describe the good behaviour of a website, a method which evaluates and distinguishes normal from abnormal behaviour is needed. This naturally leads to the second research question we aim to answer:

2 *How can we identify an appropriate method, based on the new forecasting model, to automatically detect anomalous behaviour on websites, and subsequently raise an alarm?*

## 1.3 Organization

The paper is organized as follows: In chapter 2 we introduce the dataset, which is time series data for 81 days from a website of a Dutch company, and we investigate its features. Special interest is given in the seasonal patterns of each one of the different measurement variables/metrics. Chapter 3 introduces the forecasting techniques, the one currently used which we want to replace and the new models to be tested and compared, all belonging to the exponential-smoothing family of forecasting methods. In chapter 4 we evaluate and compare the performance of all the new forecasting models. Chapter 5 describes the creation and implementation of the anomaly detection model, which is based on the assumption of normally distributed forecasting errors. Finally, chapter 6 is a conclusion together with some ideas for future further improvement.

# CHAPTER 2

## Data insights and visualisation

The first and maybe most important step before building a model intended to perform a specific task is to have a good understanding of the dataset of interest. Each dataset has different characteristics and the choice of the most appropriate model is closely related to the identification of these unique features. For that reason we start our investigation with an explanatory data analysis before we delve into the details of selection and implementation of the most appropriate for our case models.

## 2.1 Dataset general info

The dataset that was made available for the current thesis comes from the website of a Dutch consumer and business finance company. It consists of time series data for almost 81 days for the period between March 19 and June 7, for the year 2016. For this time period we have a total of 108787 instances. The metrics of interest are analyzed in one minute intervals, a practice also followed by Pena et al. [2013] when forecasting the traffic volume of an IP network, due to the need of instant reaction when unusual behaviour is monitored. Since we are concerned about finding evidence of any abnormal behaviour on websites, we choose the following two metrics which are indicative of the performance of a website:

1. *Number of pageviews*
   The total number of users that viewed the website within each minute of the day.

2. *Median pageload time*
   The time that a website needs to load is defferent for every user, and mainly depends on the Internet connection. Among all the visitors of the website within a minute, we take the median of all these numbers - as a more robust location estimator - to indicate the median pageload time of the website for each minute of the day.

A complete day has $60 \times 24 = 1440$ minutes and consequently, since the measuring is performed every one minute, a daily cycle will have 1440 measurements or instances, and therefore a whole week should idealy consist of $1440 \times 7 = 10080$ measurements. For easiness, we choose to view the dataset in terms of weeks. That way, the dataset can also be described as being composed of 11 complete weeks and 4 seperate days. We make the convention that a complete week starts from the midnight hour, i.e. 00:00, of a Monday and ends to one minute before midnight , i.e. 23:59, of the Sunday of the same week.

By seperating the data set in terms of weeks, knowing the starting and finishing date of each week we can find which weeks have missing data by counting the instances. On table of figure 2.1 below there is in detail the number of missing values of the entire dataset. Apart from these numbers, the table also includes the percentage of missing values with respect to the total expected number of instances for the specfic period, giving us a general view of how complete each week and consequently the whole dataset is.

| Time period | Date | Number of missing values | % of missing values with regard to time period |
|---|---|---|---|
| First 2 days | 19/3 - 20/3 | 61 | 2.1% |
| Week 1 | 21/3 - 27/3 | 73 | 0.72% |
| Week 2 | 28/3 - 3/4 | 5 | 0.05% |
| Week 3 | 4/4 - 10/4 | 227 | 2.2% |
| Week 4 | 11/4 - 17/4 | 39 | 0.39% |
| Week 5 | 18/4 - 24/4 | 3124 | 31% |
| Week 6 | 25/4 - 1/5 | 135 | 1.3% |
| Week 7 | 2/5 - 8/5 | 259 | 2.6% |
| Week 8 | 9/5 - 15/5 | 81 | 0.8% |
| Week 9 | 16/5 - 22/5 | 2700 | 26.8% |
| Week 10 | 23/5 - 29/5 | 12 | 0.12% |
| Week 11 | 30/5 - 5/6 | 211 | 2.1% |
| Last 2 days | 6/6 - 7/6 | 926 | 62% |

Figure 2.1: Table of missing values for the whole dataset

From the table of figure 2.1, the following important conclusions can be drawn: by summing the total number of missing values and dividing it with the number of instances ideally expected for the specific time period, we find that the percentage of missing values for the entire dataset is less than 7%. Unfortunately, most of these lay on the 5th and 9th week, where 31% and 26.8% of the values of these weeks respectively are missing. This translates in more than two complete days missing for each of these two weeks, which makes week 5 and 9 less informative than the rest of the weeks. In addition to that, the last two days have 62% of the values missing, a high percentage to consider these days valuable for providing enough information.

## 2.2 Seasonality patterns

It is quite often the case that human activities show some form of repetition patterns. A brief look in the literature proves the cyclical nature of users' behavior: the electricity demand of a country examined by Souza et al. [2007], the number of mobile phone calls at a particular cell tower site investigated by Au et al. [2011], the number of call center arrivals from Taylor [2010a] and the amount of traffic in TCP/IP based networks analyzed by Cortez et al. [2012] are just some of the many examples in which the cyclical character of the human behaviour causes these measurements to exhibit several seasonality patterns. In this section we aim to reveal any seasonal patterns hidden in our dataset, by producing plots for the two different metrics of our interest.

### *Number of pageviews*

We start by presenting plots for the total *number of pageviews*. The plots of the complete dataset for each one of the weeks can be found in the appendix at the end of this documentation. Here we present only some of them, those that we consider to be more indicative. For every graph that is produced within a period of time - whether this is for a day, a week, several consecutive days or the entire dataset - the x-axis starts from midnight hours 00:00 of the first day and ends at 23:59 of the last day. To protect the proprietary information of our clients, each time series in the graphs is scaled by a random factor.

On figure 2.2 a plot of the whole dataset for the metric *number of pageviews* is produced. A first general observation is that the data appear to have a changing level, but not an apparent increasing or decreasing trend. Furthermore, regarding the seasonal repetitions, it is not easy to make clear conclusions about presence of any seasonal patterns for the metric. It seems though that there must be a daily pattern, due to the multiple spikes, each one indicating the peek during one day.

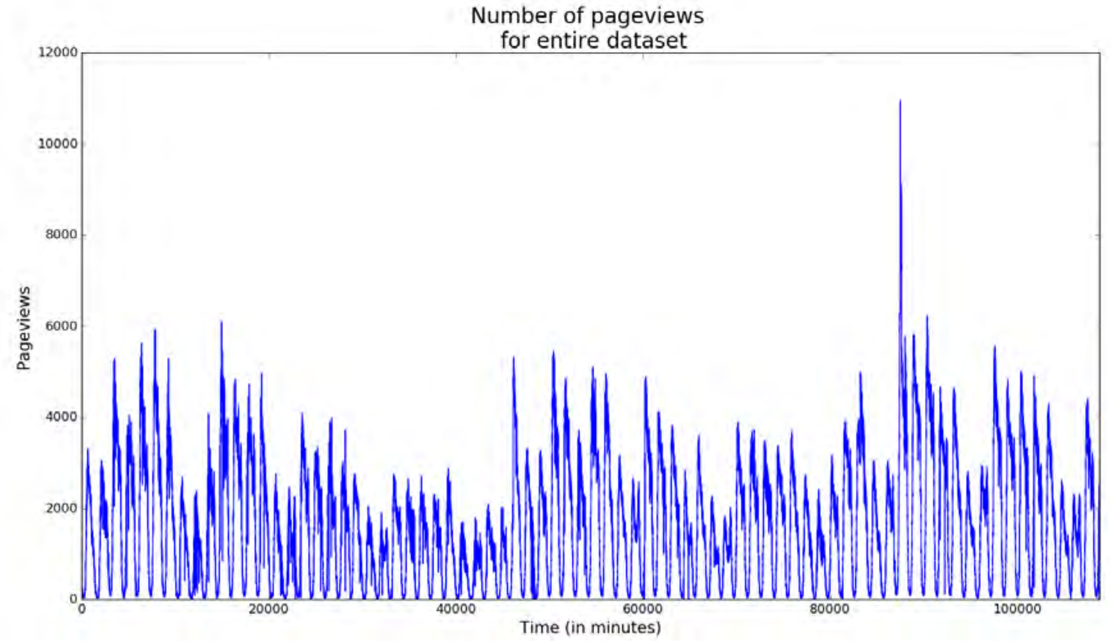


Figure 2.2: The number of pageviews for the entire dataset.

A plot for the number of pageviews during a time period of 20 consecutive days, starting from a Monday, follows on figure 2.3:

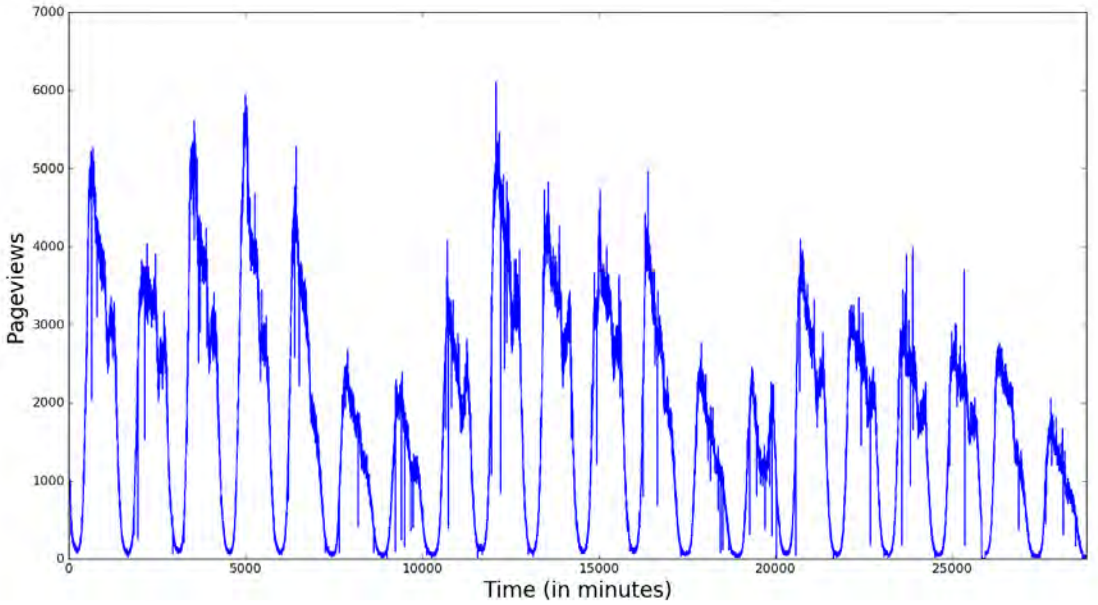


Figure 2.3: Number of pageviews for a total of 20 consecutive days. The daily pattern is obvious.

As we can see from figure 2.3, there are 20 peeks, each one for a seperate day which indicates that indeed our data follow at least a daily pattern. The behaviour of the metric for each day seems to be the same, the only apparent difference is the level it reaches - for some days the number of pageviews is higher and for others it is lower - but the daily cycle is obvious. To acquire some insight on the behavior of the users, we continue by a plot of the intra-day cycle of a normal weekday.

On figure 2.4 some general observations regarding the common patterns of the daily cycle for the weekdays can be made. The pageviews start increasing in the morning hours between 06:00am and 07:00am. The increase is vast up to around 10:00am, where it reaches the maximum value of the day between 11:00 and 12:00. After this peek the curve starts diclining - probably due to lunch time - until the time between 4:00pm and 5:00pm. From that point a rapid decrease follows which can perhaps be attributed to the fact that it is the time that people leave their works, reaching a local minimum at around 6:00pm. From this point there is an increase between 6:00pm and 7:30pm, followed by a constant behavior of 2-3 hours, until 10:00pm when a rapid decline starts again.

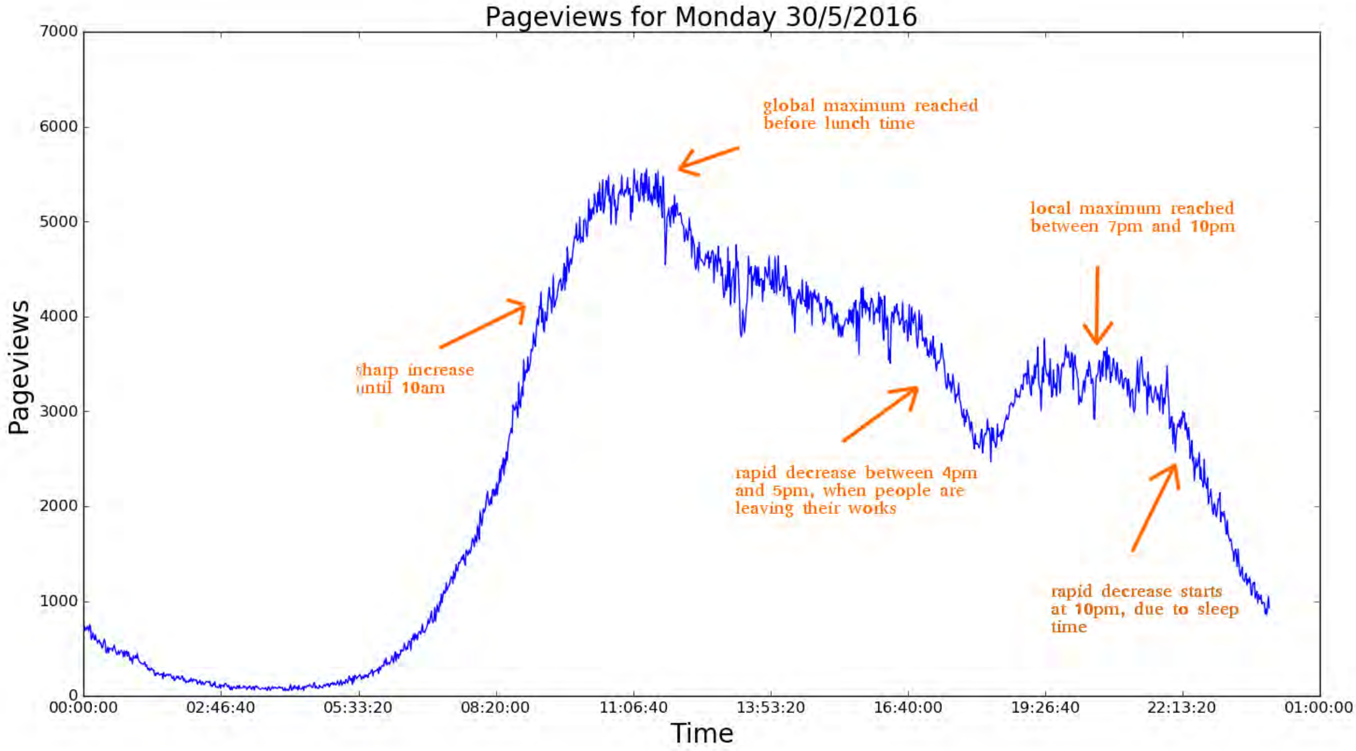


Figure 2.4: Daily behavior of the metric ”number of pageviews” for a normal weekday. Two peeks are obvious, one in the morning and one in the evening: a global maximum reached at around 11:00 to 12:00 and a local one reached beween 19:30 and 22:00.

To see in a clearer view we produce a one-week plot about the total number of pageviews, starting from Monday up to Sunday. The weekly pattern is easily seen from this plot in figure 2.5:

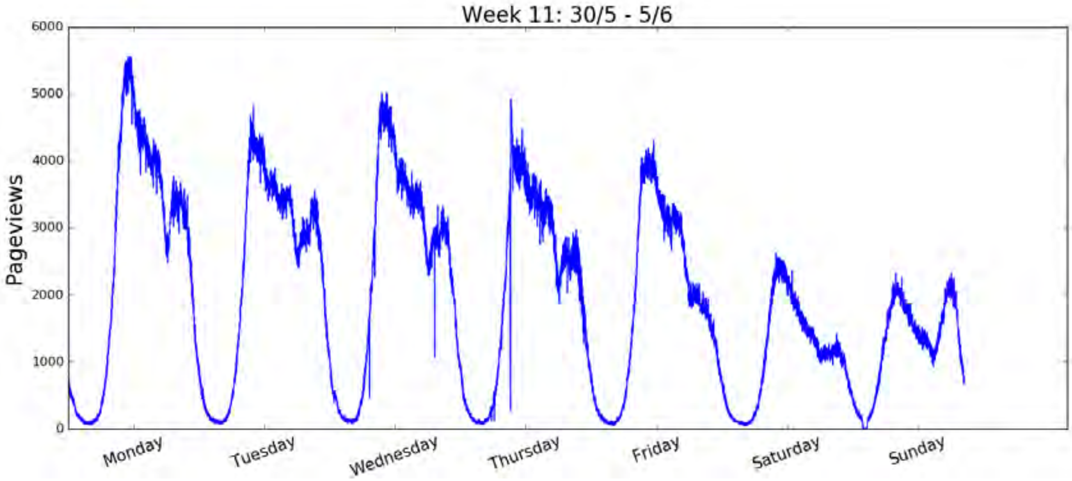


Figure 2.5: A whole week for the number of pageviews. This metric during weekdays reaches a higher level compared to Saturday and Sunday.

From the weekly plot of figure 2.5, as well as figure 2.6 following, some important realizations can be made regarding the behaviour of this metric:

1. Week days from Monday to Thursday seem to have similar behaviour with each other, same like the one explained before when describing the daily pattern of figure 2.4: pageviews during these days appear to have two peeks, one global maximum taking place in the morning hours followed by a local maximum occuring in the evening hours.

2. The last weekday before weekend, Friday, appears to show a relatively different behaviour than the rest of the weekdays: even though the morning peek occurs, just like the rest of the weekdays, the evening increase is of lower magnitude, almost not existing. This could be attributed to the fact that users' behaviour on Friday evening is affected by the upcoming weekend.

3. Regarding the weekend, it is apparent that Saturdays and Sundays behave differently than the weekdays, since they reach a lower level. Furthermore, they also seem to have differences between each other. Saturday appears to have a morning peek followed by a notably smaller evening peek (quite similar to Friday's evening peek, maybe a slightly more obvious). On the other hand, the Sunday morning and afternoon peeks seem to be of the same magnitude with each other.

From all the above, two important conclusions can be drawn regarding the seasonality patterns of the metric *number of pageviews*. This metric not only exhibits a daily seasonal pattern, but also a strong weekly pattern. In particular, its weekly behaviour could be devided in four distinct categories: the weekdays between Monday and Thursday all behaving in a very similar way, and Fridays, Saturdays and Sundays, each one forming a seperate category due to the different behaviour of their daily cycle.

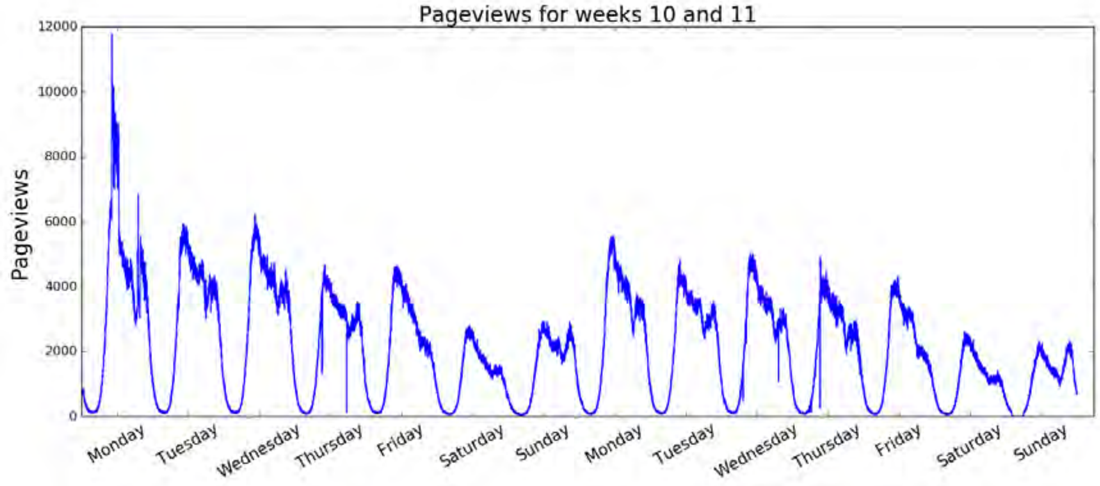


Figure 2.6: The number of pageviews for the last 2 weeks of the dataset. Apparently an anomaly occured on the first Monday, which is the reason that the first daily cycle on the current graph is different than expected.

## *Median pageload time*

We continue with the second metric, the *median pageload time* and we plot the time series for the whole data set and for 20 consecutive days, starting from a Monday, just like we did with the previous metric. From figures 2.7 and 2.8 one general but obvious observation can be made: there are observations that are really higher in magnitude than the rest. This is clearly visible from the plot of the entire dataset.

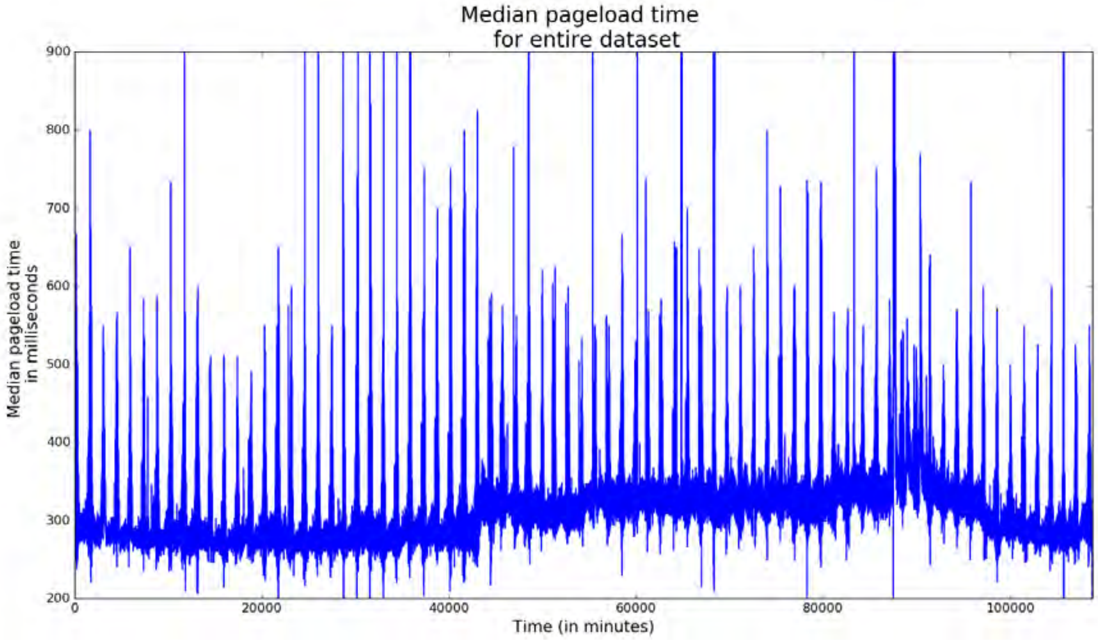


Figure 2.7: The median pageload time for the entire dataset.

On the other hand, a daily pattern can be observed with easiness from figure 2.8, where the intra-day cycles are seen as different spikes.

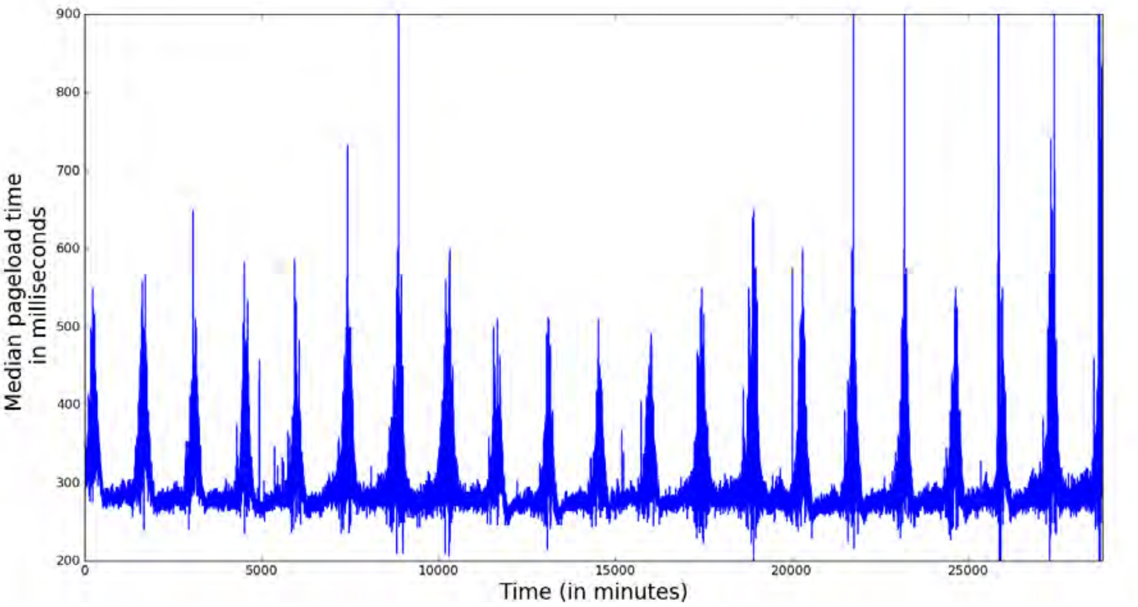


Figure 2.8: The median pageload time for a total of 20 consecutive days.

To reveal if there is any weekly pattern, we plot the *median pageload time* for one whole week, starting from the early hours of Monday to late midnight of Sunday.

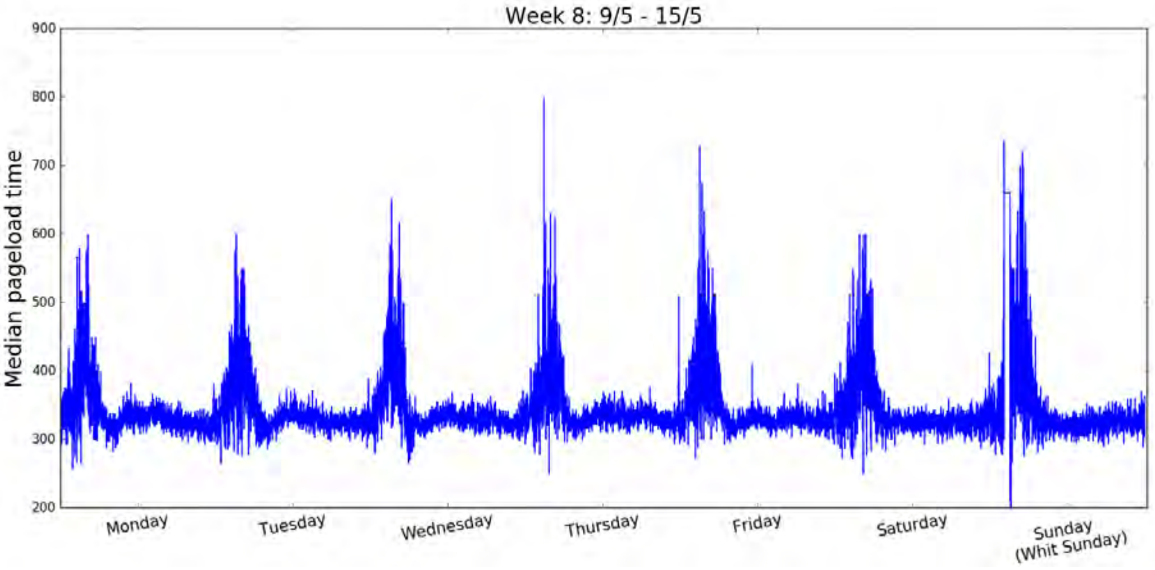


Figure 2.9: The median pageload time for an entire week.

From figure 2.9 there is no apparent evidence that this metric exhibits a weekly seasonal pattern. Unlike *number of pageviews*, the behaviour of *median pageload time* seems very similar - if not the same - for all the days of the week. To ensure that this is indeed the case, on figure 2.10 we plot the weekly behaviour of *median pageload time* focusing only on the 200-400 range. It seems that there are slight differences between weekdays and weekends: the deviation in the weekends is higher, and the underlying level of the metric takes lower values in weekends compared to weekdays.

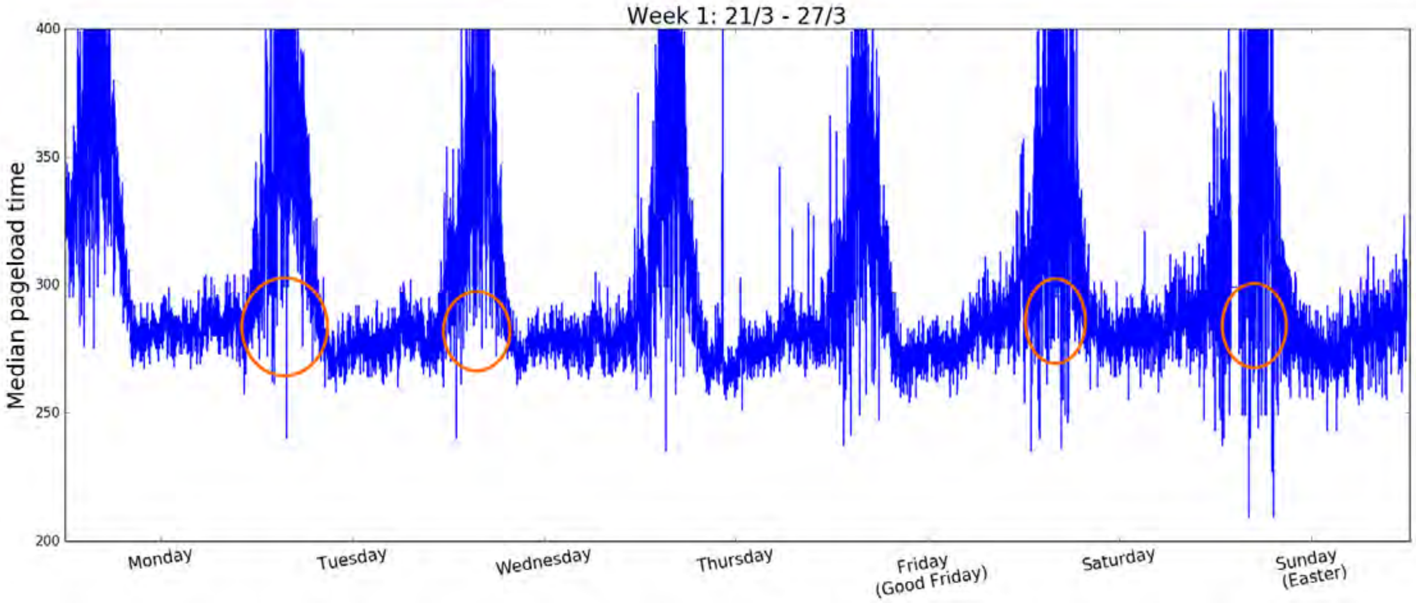


Figure 2.10: Weekly seasonal patterns - even though weak - shown for a random week for the *median pageload time*. Weekends seem to deviate more, compared to weekdays. The differences between some weekdays and weekends are circled in red.

Regarding the intra-day cycle, we plot the behaviour of the *median pageload time* for a random day, a Wednesday, on figure 2.11. It is visible that between midnight hours and the time around 8:00am the curve resigns a triangular movement: an increase starts a few minutes before the midnight hours, where it reaches a global maximum between the early morning hours of 3:00am to 4:00am. From that point and forward the curve declines reaching a minimum value at around 8:00am, where it keeps a forthcoming steady behavior until almost the end of the evening hours.

If we compare this behaviour with the one of the previous metric this seems contradicting to some extent, since these high values of the *median pageload time* are reached exactly during the hours that the *number of pageviews*, are taking the lowest values possible during the day, as we have seen from figure 2.4. A naive but reasonable enough assumption would expect the pageload time of a website to be higher at the time where most of the users are visiting the website.

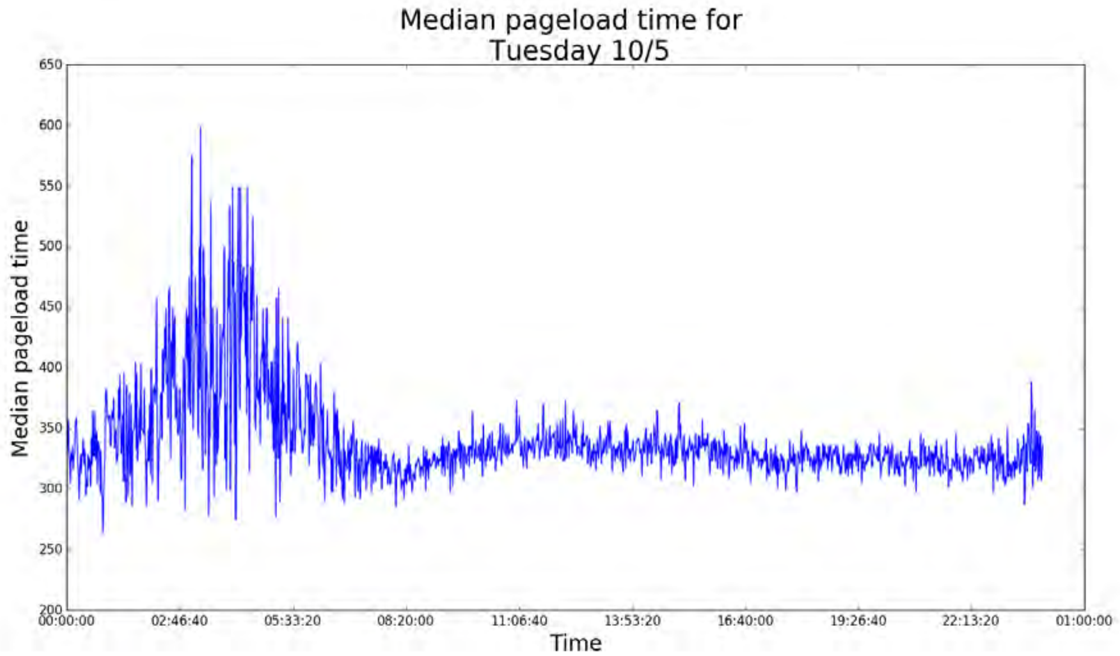


Figure 2.11: The intra-day cycle of the metric *median pageload time*, for a day of the week of figure 2.9.

One important thing to point out from the graph of figure 2.11 is the high deviation of the values during the hours where the *median pageload time* is increrasing and decreasing, i.e. where the triangular curve is revealed - between 00:00am and 08:00am. On the contrary, the steady behaviour taking place between 8:00am and 00:00am is characterized by small deviations.

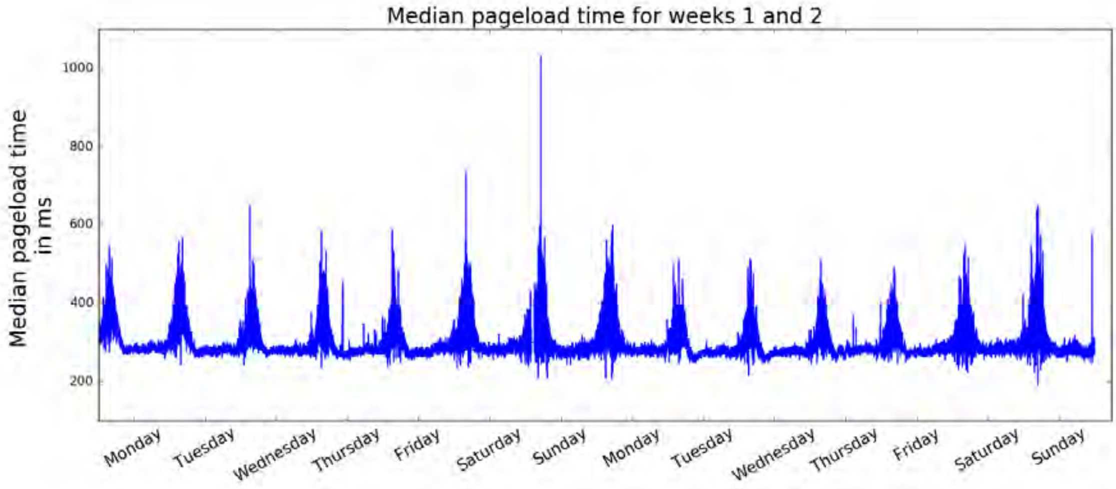


Figure 2.12: Plotting the metric *median pageload time* for two consecutive weeks.

To conclude, as regards the behaviour of *median pageload time*, there is strong evidence that this metric too has an intra-day cycle where its behaviour repeats after the completion of one day. Furthermore, apart from this daily seasonal cycle, it does seem to exhibit a weak weekly seasonality pattern. One other observation coming from the graphs is the high values that the outliers have, and the frequency this happens. Apparently this metric seems more prone to errors or falses.

## 2.3 Public holidays

One thing that we should take into account, which is clearly visible from the graphs, is the effect that public holidays have on the metrics we study. The public holidays of the Netherlands for the year 2016 for the period of our dataset, are the following:

| **Holiday** | **Date of holiday** |
|---|---|
| Good Friday | Friday, March 25 |
| Easter Sunday | Sunday, March 27 |
| Easter Monday | Monday, March 28 |
| King's Day | Wednesday, April 27 |
| Liberation Day | Thursday, May 5 |
| Whit Sunday | Sunday, May 15 |
| Whit Monday | Monday, May 16 |

For the whole data of the 9 complete weeks (after excluding the less informative data, as explained before), we plotted a graph comparing the behaviour of each of the seven days of the week seperately. From these graphs, which can be found in the appendix in the end of the documentation, the effect of the public holidays is clear. Below is an indication example, for the case of Sundays for the metric *number of pageviews*:

As we explained before, the metric *number of pageviews* on Sundays has a different behaviour than the rest of the days, with two peeks of almost the same magnitude appearing during midday and evening. As we can see from figure 2.13 though, this does not hold for holidays. On Easter Sunday, as well as on Whit Sunday we clearly see a change on the pattern, namely the midday peek being almost double in size than the evening peek.

Similar realizations about the effect of holidays on the metrics can be made from the weekly plots. From figures 2.14, 2.15 and 2.16 we can also see the influence of holidays on the metric *number of pageviews.* Even though, as we observed previously, weekdays from Monday to Thursday have a very similar pattern, the Easter Monday of week 2, the Wednesday of week 6 which is King's day and the Thursday Liberation day of week 7 take smaller values than the rest of the weekdays of the week they belong.

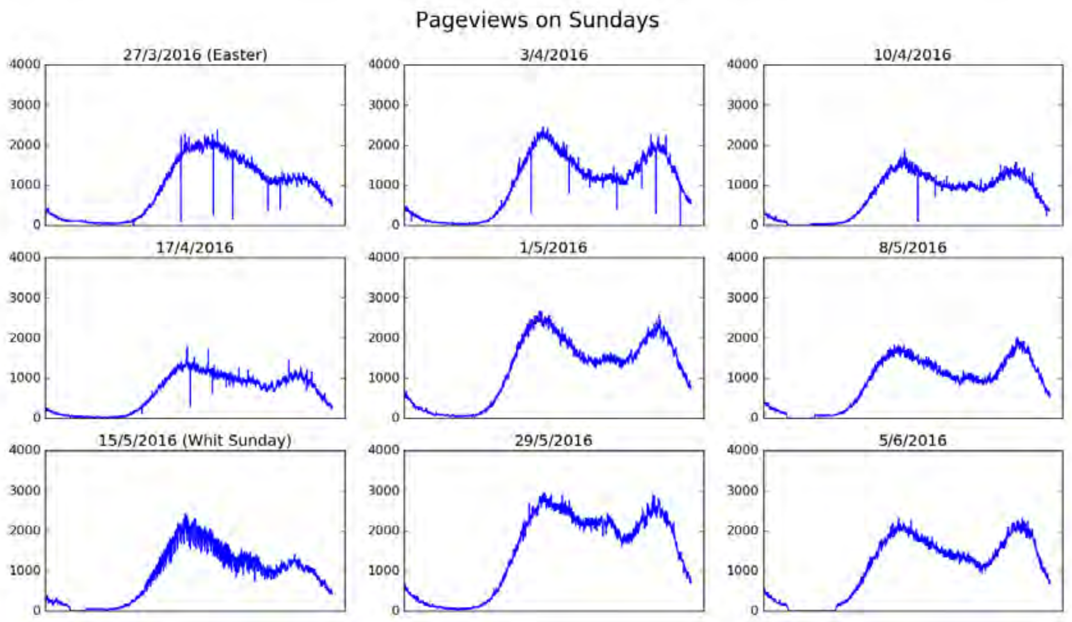


Figure 2.13: Daily cycle of pageviews for all Sundays

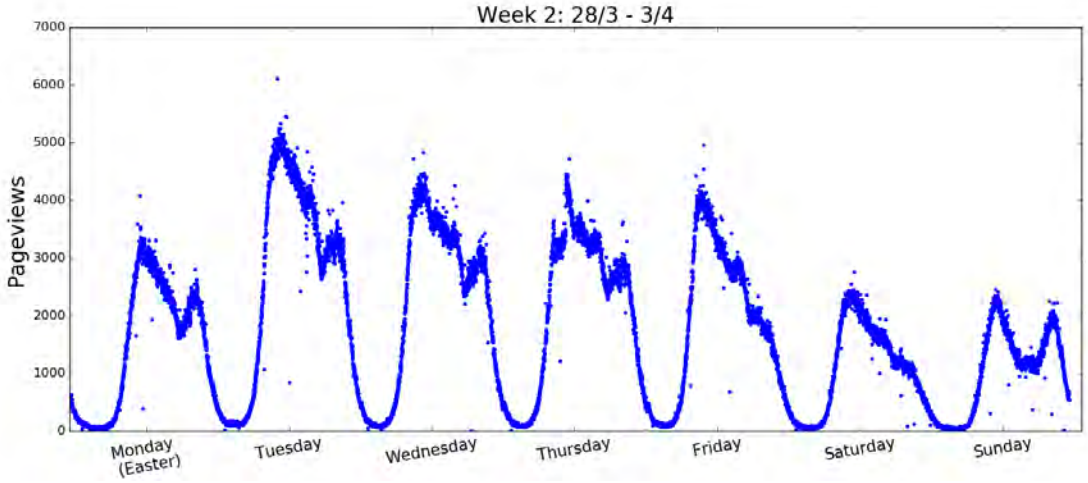


Figure 2.14: Users of the website are fewer during Easter Monday than the rest of the weekdays.

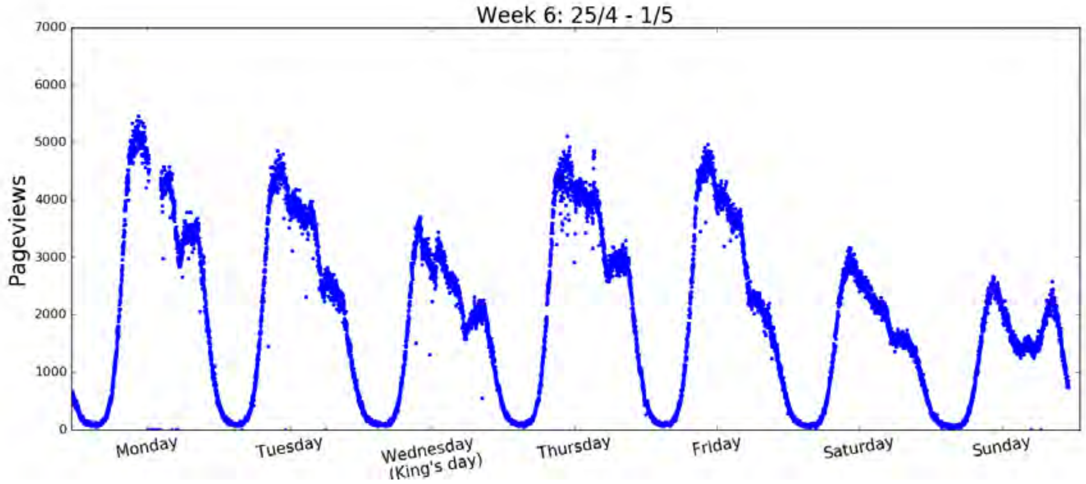


Figure 2.15: The impact of King's day is obvious: pageviews on Wednesday reach a lower maximum compared to the rest of the weekdays.

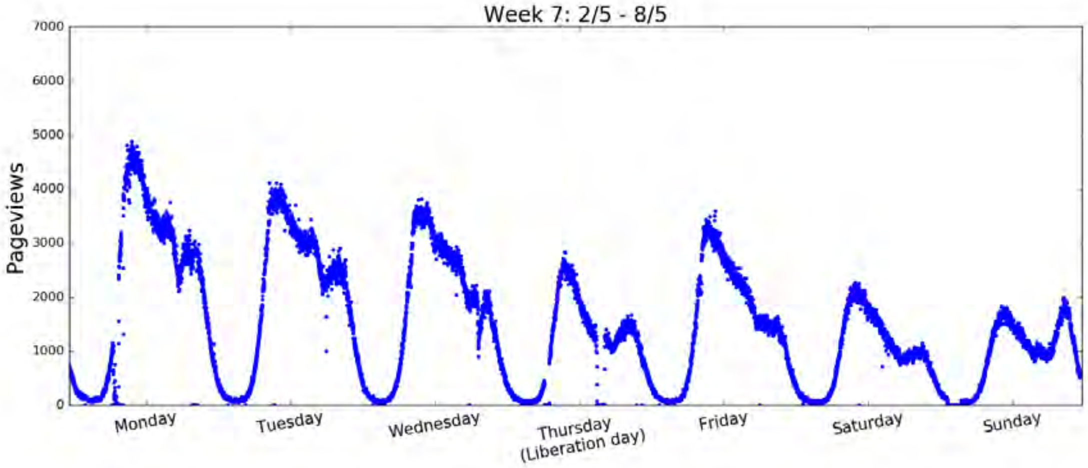


Figure 2.16: The fact that Thursday is Liberation day results in the number of pageviews to reach a lower maximum compared to the rest of the weekdays.

On the contrary with the *number of pageviews*, the metric *median pageload time* does not show any significant relevance or connection to the Netherlands' national holidays. From the graphs of the appendix we see that, for this metric, holidays behave similarly as the rest of the days.

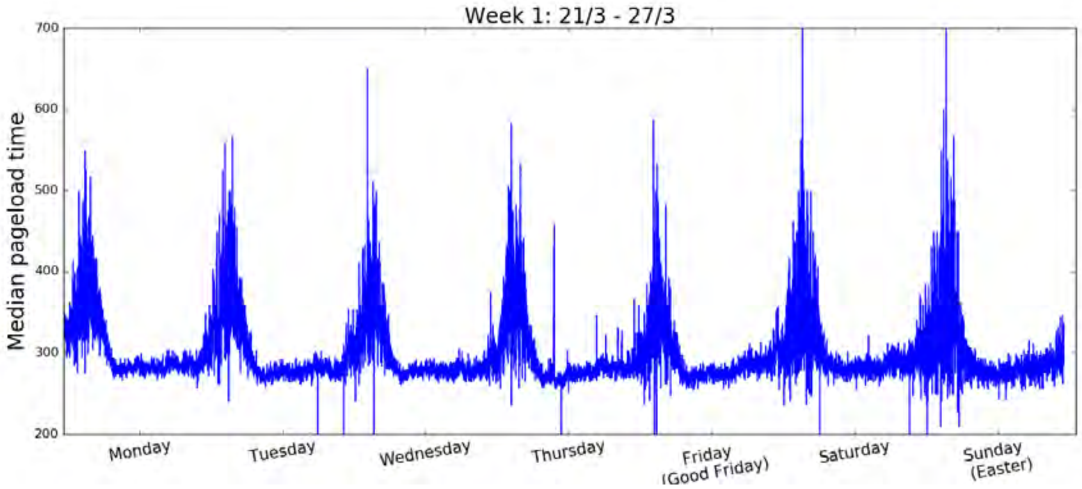


Figure 2.17: There is no sign of significant change on the behaviour of the median pageload time that can be attributed to public holiday.

## 2.4 Missing values and outliers

As it happens with most datasets, two are the main problems which prevent them from being declared as a good quiality dataset:

1. Missing values
2. Presence of Outliers

The fact that there are missing values in our data was proved and presented in the beggining of this chapter. Namely, the problem lies mostly in weeks 5 and 9, which due to the high amount of values that were not available, we concluded that no essential information can be gained from these two weeks (see also fig 2.19).

Our historical data come from a website which - like most of the websites - has experienced several malfunctions in the past, and these malfunctions are depicted in the two metrics we study. For example, an unexpectedly low value of the *number of pageviews* may be the result of an error in the website, whereas a very high value could indicate a potential upcoming failure due to the high volumes of users. Furthermore, a high value of the metric *median pageload time* may be prognosticating a future anomalous behaviour. The presence of such unexpected values in our dataset, mostly know as *outliers*, is then not surprising. We will talk about the importance of treating outliers in our context of anomaly detection, together with methods of dealing with them, on the following chapters.

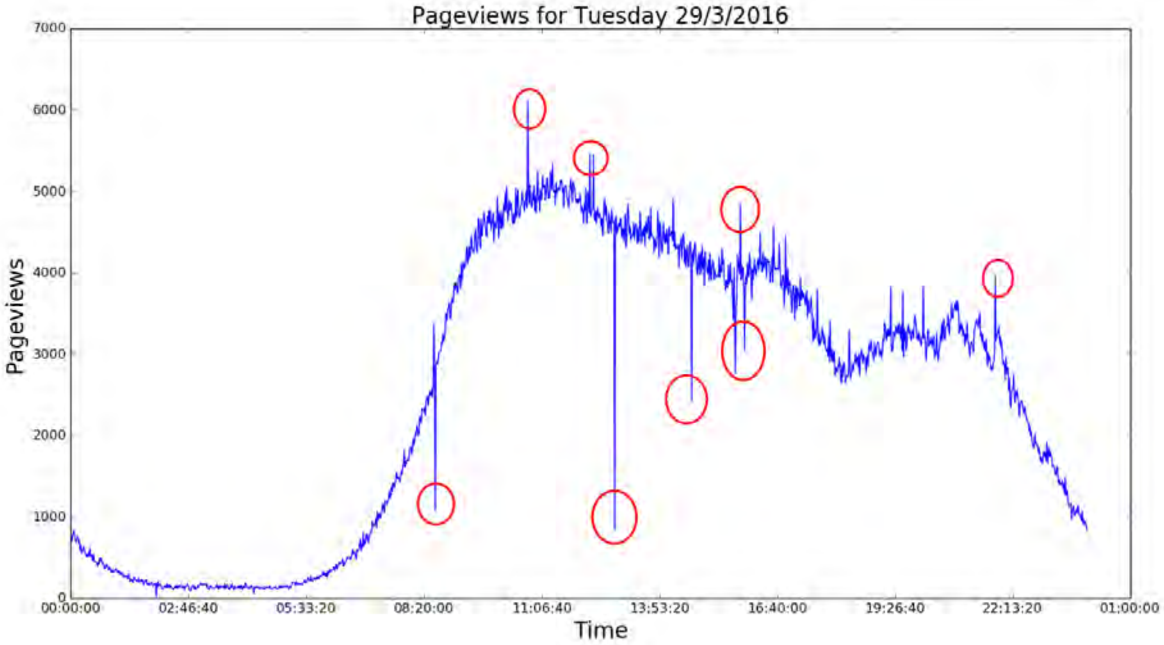


Figure 2.18: Pageviews for a daily cycle containing abnormal values, highlighted with red circles. These vertical lines which indicate sudden rapid increase could possibly result to an imminent website breakdown, while those unexpected falls could be attributed to an instant collapse of the web page.

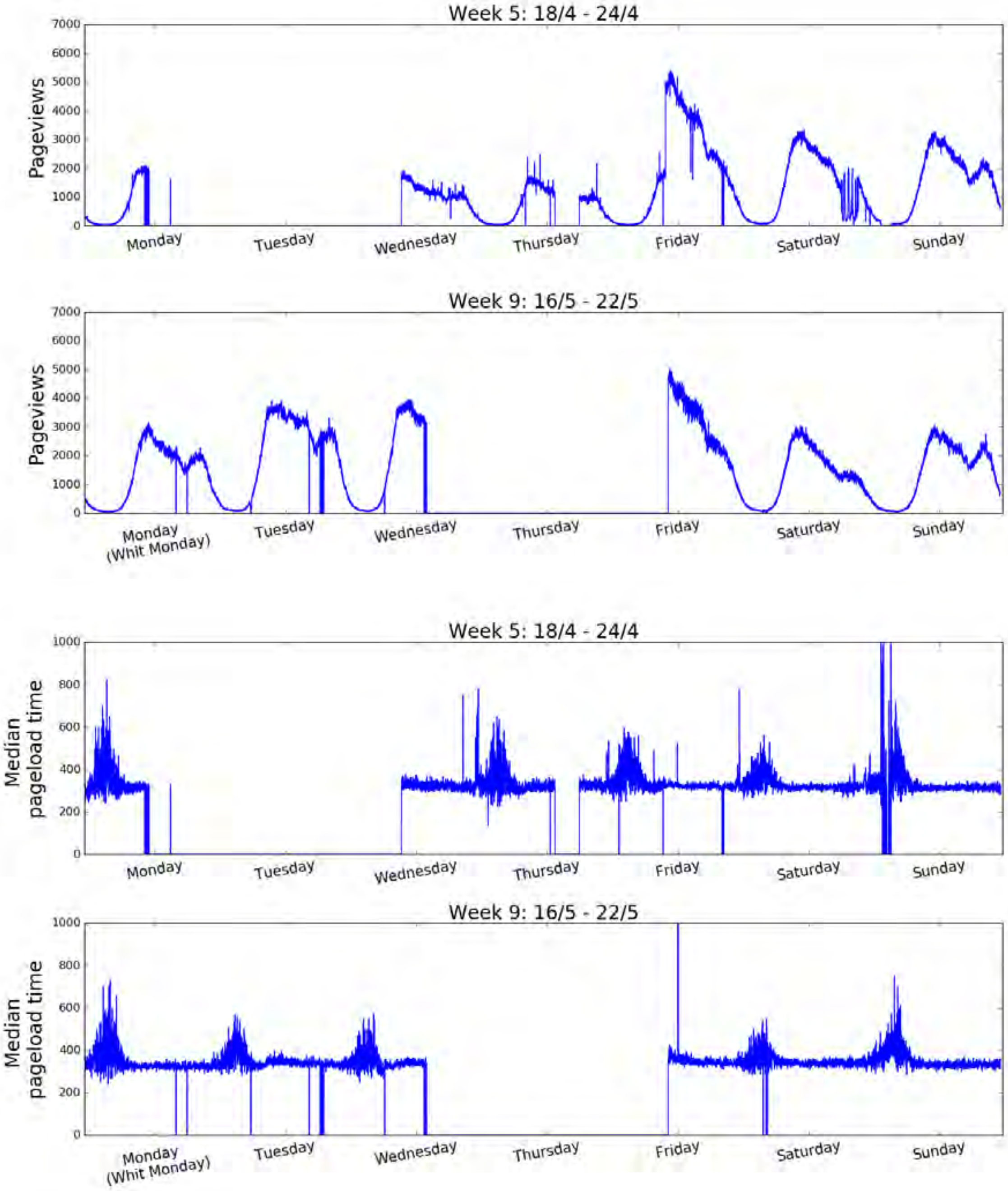


Figure 2.19: The weeks with the highest number of missing values for both metrics. As it is seen, these weeks provide no valuable information and therefore are excluded from the dataset.

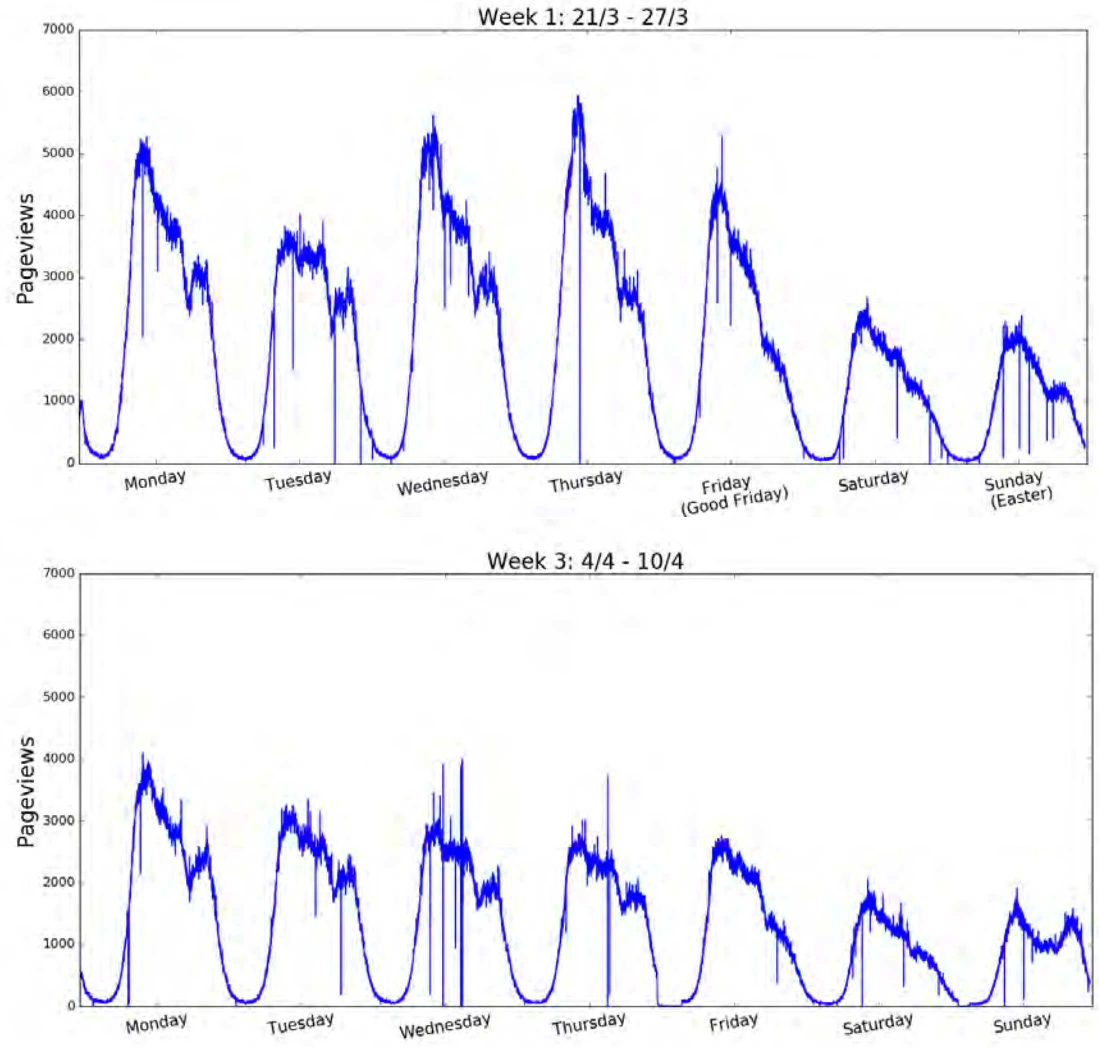


Figure 2.20: Indicative examples of weeks containing abnormal behaviour, shown as vertical lines.

# CHAPTER 3

# Exponential smoothing forecasting methods

Consider a time series $\{y_t\}$, i.e. data measured in equally spaced points in time, and suppose we want to make estimations about future values. For a start, lets assume that this time series does not show any form of trend or seasonality, i.e. no constant increase or decrease nor any pattern of repetition, just random inherent variation which depends on factors that we cannot see or understand. In mathematical terms, this can be expressed as

$$y_t = c + \epsilon$$

where $y_t$ is the value of the time series at time $t$, $c$ is the constant that determines the level of the time series and $\epsilon$ is the noise - this random inherent variation - with a mean of zero and variance of $\sigma^2$.

Given this time series, our first attempt to forecast the future would be the *simple average* of these values. Since the data show no trend or seasonality, they tend to gather around the mean and therefore the simple average seems the most plausible forecast to begin with. This simple forecasting technique, though naive, takes into account all the available historical data and thus has the advantage of exploiting all information about past behaviours.

There may be cases where the context of the problem suggests that only recent values really play an important role in predicting the future. As an example consider the price of a stock. Knowing the price of the stock one year ago provides little or no information about its future behaviour. In these cases we could agree that taking the mean of the last $k$ values, known as the *k-period moving average* - where $k$ is to be determined - could give more accurate predictions than the simple average.

Furthemore, another possible technique would be to use the *weighted average*, i.e. to consider all the previous values but give different weight to each one of them: bigger weight to data points that are closer to today in terms of time, and less weight to values that are further in the past. By doing so, we have the advantages of both preceding methods: we consider all the information we have from the historical data and at the same time we give more importance to recent than to past information.

Another option would be to consider not the whole dataset but part of it, lets say the last $k$ values, and assign different weights to each data point on the same reasoning as we explained before in the weighted average. This method is known as the *k-period weighted moving average* and, even though by excluding the very old values we lose some past information, it can produce accurate results in the cases where only recent past is believed to be informative.

The preceding forecasting methods, simple as they look, they raise some important questions: Can we determine $k$, the number of values that is informative enough to consider when predicting future values? In each context and dataset this number will be different and it seems difficult to choose a specific value other than an arbitraty one. At the same time, if we take all information into consideration and give more weight to recent than far-distant in time values, how can we determine the rule under which to assign the weights to each of the data points?

## 3.1 Simple and double exponential smoothing

The *simple exponential smoothing* method, also known as *Exponential Weighted Moving Average* (or shortly *EWMA*) is a measure of central tendency that can be used to make very accurate predictions for constant-level time series with no seasonality. The EWMA is a simple method that uses all the available historical data and assigns to them weights depending on how recent they are. These weights decrease exponentially as we consider data points of the further past.

From now on and until the rest of the paper we denote with $\hat{y}_t$ the forecast of the time series at time $t$ and with $y_t$ the actual true value at time $t$. Given this convention, the EWMA can be described in mathematical terms as follows:

$$\hat{y}_{t+1} = \alpha \cdot y_t + (1-\alpha) \cdot \hat{y}_t \tag{3.1}$$

where $\alpha$ is the *smoothing parameter* with the restriction that $0 \leq \alpha \leq 1$.

As we can see the only value that identifies EWMA is the parameter $\alpha$, which can intuitively seen as a measure of the impact of the last measurement - the greater the $\alpha$, the greater the impact of the last measurement, the lower the value of $\alpha$ the better the model "remembers" the values of the distant past.

By placing the values of the forecasts in formula 3.1 and working recursively we get the following alternative formula:

$$\hat{y}_{t+1} = \alpha \cdot y_t + \alpha(1-\alpha) \cdot y_{t-1} + \alpha(1-\alpha)^2 \cdot y_{t-2} + ... + \alpha(1-\alpha)^{t-1} \cdot y_1 + (1-\alpha)^t \cdot \hat{y}_1 \tag{3.2}$$

which can also be written

$$\hat{y}_{t+1} = \left( \sum_{n=0}^{t-1} \alpha(1-\alpha)^n \cdot y_{t-n} \right) + (1-\alpha)^t \cdot \hat{y}_1 \tag{3.3}$$

From formula (3.2) it is more clear that EWMA uses all the past available data in order to make predictions for the future.

We can easily verify that the weights in equation 3.2 add up to one:

$$\begin{aligned}\left(\sum_{n=0}^{t-1}\alpha\cdot(1-\alpha)^n\right)+(1-\alpha)^t &= \alpha\cdot\left(\sum_{n=0}^{t-1}(1-\alpha)^n\right)+(1-\alpha)^{t+1}\\ &= \alpha\cdot\frac{(1-\alpha)^t-1}{(1-\alpha)-1}+(1-\alpha)^t\\ &= 1-(1-\alpha)^{t+1}+(1-\alpha)^{t+1}\\ &= 1\end{aligned}$$

## Component form

We now present an other alternative representation of the Simple Exponential Smoothing, which will also be particularly useful in the coming sections when we discuss about time series with trend and seasonality. According to Hyndman and Athanasopoulos [2013], in this simple case of constant level time series with no seasonal patterns, we can "break" the predictions in two equations, the forecast equation and the level smoothing equation:

$$\begin{aligned}\text{level smoothing equation:}\quad & \ell_t = a\cdot y_t+(1-\alpha)\cdot\ell_{t-1}\\ \text{forecast equation:}\quad & \hat{y}_{t+1}=\ell_t\end{aligned}\tag{3.4}$$

The forecasted value at the next time step is equal to the estimated level on the previous time step, where the level is a weighted average of the actual value and the estimated level computed up to and including the previous time instant.

## Initialization

As equation (3.1) reveals, the forecast at time $t+1$ is equal to a weighted average between the most recent observation $y_t$ and the most recent forecast $\hat{y}_t$. Moreover, looking at equation (3.2) and specifically on the last term, we can see that the initial value $\hat{y}_1$ at $t=1$ needs to be determined before we are able to make the first forecast.

Some initialization techniques are, either to set

$$\ell_0 = y_1$$

or to set the intial forecast value equal to the mean of the sample, i.e.:

$$\ell_0 = \frac{1}{T}\cdot\sum_{t=1}^{T} y_t$$

considering that our sample consists of values measured up to time $T$, starting from $t=1$.

## Double exponential smoothing

It may be the case that a time series shows some increasing or decreasing behaviour regarding the average value it takes as it evolves in time. We then say that the time series shows some form of trend and the technique we use is known as *double exponential smoothing* or - by the name of the mathematician that developed it - *Holt's linear trend model.*

The component form of Holt's model will be the same as the one of the simple exponential smoothing with only difference that a second smoothing equation for the trend is added. Considering then that we have a sample of $T$ in number time series values $y_1, y_2, ..., y_T$, the smoothing and forecast equations of Holt's model will be:

$$\begin{aligned} \text{level smoothing equation:} \quad & \ell_t = \alpha \cdot y_t + (1-\alpha) \cdot (\ell_{t-1} + b_{t-1}) \\ \text{trend smoothing equation:} \quad & b_t = \beta \cdot (\ell_t - \ell_{t-1}) + (1-\beta) \cdot b_{t-1} \\ \\ \text{forecast equation:} \quad & \hat{y}_t = \ell_{t-1} + b_{t-1}, \qquad \text{for } t = 1, 2, ..., T. \end{aligned} \tag{3.5}$$

When forecasting future values, i.e. values that we don't have data measured at the time instances, then the prediction is given by the future-forecast equation below:

$$\hat{y}_{T+h|T} = \ell_T + h \cdot b_T$$

where $h = 1, 2, 3, ...$ is the step which we want to forecast further in the future starting measured from time $T$ and forth, which from now on and for the rest of the paper we will refer to it as *forecast horizon* or *lead time* and denote it by $h$. Using the notation of Hyndman and Athanasopoulos [2013], $\hat{y}_{T+h|T}$ is the forecast for time $T+h$, where $T$ is the time of the last observed value of the time series, or from a statistical perspective the size of the data sample.

From equations (3.5) we can see that both the level and trend components need to be initialized. We will talk about how we determine these initial values $\ell_0$ and $b_0$ in the next section when describing the Holt-Winters model.

## 3.2 Holt-Winters method

The *Holt-Winters method* (HW), named after its inventors, is a continuation of Holt's model for a time series with a linear trend that also takes into account seasonal patterns, and was proposed by Holt's student, Peter Winters in 1960 [source: Wikipedia]. In this method, at every time step estimates for the level, trend and seasonality components are revised, which also justifies the name of *triple exponential smoothing.*

There are two different versions of the Holt-Winters method, for the two different forms of seasonality patterns that define the model: the *additive* and the *multiplicative* form. In its additive form, which we present first, the change of the seasonal fluctuation between two successive seasonal factors differ by a constant number, while in its multiplicative form this change is a percentage and thus for greater values the change of the successive cycle will be greater.

### Additive model

$$\begin{aligned} \text{forecast equation:} \quad & \hat{y}_t = \ell_{t-1} + b_{t-1} + s_{t-m} \\ \\ \text{level smoothing equation:} \quad & \ell_t = \alpha \cdot (y_t - s_{t-m}) + (1-\alpha) \cdot (\ell_{t-1} + b_{t-1}) \\ \text{trend smoothing equation:} \quad & b_t = \beta \cdot (\ell_t - \ell_{t-1}) + (1-\beta) \cdot b_{t-1} \\ \text{seasonality smoothing equation:} \quad & s_t = \gamma \cdot (y_t - \ell_t) + (1-\gamma) \cdot s_{t-m} \end{aligned} \tag{3.6}$$

for $t = 1, 2, ..., T$.

The future forecast equation for $h-$time steps forward in the future since the time of the last observed value will be

$$\hat{y}_{T+h|T} = \ell_T + h \cdot b_T + s_{T+h-m}, \qquad h = 1, 2, 3, ...$$

where $m$ is the periodicity of one whole seasonal cycle, i.e. the number of time steps of one full season. In our case where the measuring is done every one minute, if we were to consider Holt-Winters model assuming daily seasonality, then $m = 1440$ while for the case of weekly seasonal data this would be equal to $m = 10080$. We refer to $m$ from now on as *length of the seasonal cycle* and by the term "seasonal cycle" we denote any pattern that repeats (with variation) periodically.

## Multiplicative model

$$
\begin{aligned}
\text{forecast equation:} \qquad & \hat{y}_t = (\ell_{t-1} + b_{t-1}) \cdot s_{t-m} \\
\\
\text{level smoothing equation:} \qquad & \ell_t = \alpha \cdot \left( \frac{y_t}{s_{t-m}} \right) + (1-\alpha) \cdot (\ell_{t-1} + b_{t-1}) \\
\text{trend smoothing equation:} \qquad & b_t = \beta \cdot (\ell_t - \ell_{t-1}) + (1-\beta) \cdot b_{t-1} \qquad (3.7) \\
\text{seasonality smoothing equation:} \qquad & s_t = \gamma \cdot \left( \frac{y_t}{\ell_t} \right) + (1-\gamma) \cdot s_{t-m}
\end{aligned}
$$

where $t = 1, 2, ..., T$.

The forecast equation for $h-$time steps ahead in the future since the time of the last observed value will be

$$\hat{y}_{T+h|T} = (\ell_T + h \cdot b_T) \cdot s_{T+h-m}, \qquad \text{for } \ h = 1, 2, 3, ...$$

The three smoothing parameters of the Holt-Winters model are:

$$
\begin{aligned}
\alpha &= \text{the level parameter} \\
\beta &= \text{the trend/slope parameter} \\
\gamma &= \text{the parameter for the seasonal cycle}
\end{aligned}
$$

which follow the restriction of

$$0 \leq \alpha, \beta, \gamma \leq 1$$

## Initializing *Holt-Winters* method

For the first method on how to initialize the Holt-Winters model we will need at least two full seasonal cycles and, according to Hyndman and Athanasopoulos [2013] and Rob J Hyndman and Wheelwright [1997], is as follows:

The *initial level* component will be the average of the values of the first seasonal cycle. The *initial slope/trend* component will be the average of all the $m$-slopes computed in the first two seasonal cycles. Finally, the *initial seasonality* components at each time period for the additive case will be the observation minus the initial level and for the multiplicative case it will be the observation devided by the initial level. In mathematical notation, this initialization method can be summarized as follows:

$$
\begin{aligned}
&\text{initial level component:} && \ell_0 = \frac{y_1 + y_2 + ... + y_m}{m} \\
&\text{initial slope component:} && b_0 = \frac{\sum_{t=m+1}^{2m} y_t - \sum_{t=1}^{m} y_t}{m^2} \\
&\text{initial seasonal component:} && s_i = y_i - \ell_0, \qquad \text{additive case} \\
& && s_i = \frac{y_i}{\ell_0}, \qquad \text{multiplicative case}
\end{aligned}
\tag{3.8}
$$

for i=1,2,...,m.

## 3.3 Taylor's model for double seasonality

So far we have described methods for forecasting time series, all belonging to the *exponential smoothing family of forecasting methods.* For constant level time series with no seasonality we have showed that EWMA is adequate for forecasting, Holt's linear trend method is capable of predicting time series that show some form of linear trend, and finally the Holt-Winters method is used for times series with trend and seasonality. Furthermore we have noted that Holt-Winters, even though a very powerful and robust method for forecasting, that stays strong even now more than 50 years after its develepement, it takes into account one single seasonal pattern with seasonal cycle of length equal to $m$ periods. But in real life problems, particularly in the business field, it is quite common that time series exhibit multiple seasonal patterns.

In his paper, Taylor [2003] further improved the work of Holt and Winters, allowing the inclusion of one seasonal cycle nested within another bigger one. In his proceeding paper, Taylor [2010b] extended his earlier work by presenting a model for forecasting a time series with intraday, intraweek and intrayear seasonal cycles. The philosphy underlying his theory is mainly adding one smoothing equation for each different type of seasonality.

Our data, as examined on the previous chapter, exhibit at least two seasonal patterns: daily and weekly. Consequently, in the current paper we seek to capture only these two seasonal patterns, and thus here we only introduce Taylor's model for double seasonality which we refer to as *Taylor's double seasonal exponential smoothing method* or shortly *Taylor's Double Seasonal model.*

### Additive method

Similar to the case of Holt-Winters, there are two versions of Taylor's DS model depending on the form of seasonality in the data. In its additive form which we present first, the seasonal variation of the time series is not affected by the level of $y_t$. In the state space equations below, this results in the fact that the seasonal and trend components enter the forecasting equation in an additive manner:

$$\begin{aligned}
\text{forecast equation:} \quad & \hat{y}_t = \ell_{t-1} + b_{t-1} + D_{t-m_1} + W_{t-m_2} \\
\\
\text{level equation:} \quad & \ell_t = \alpha \cdot (y_t - D_{t-m_1} - W_{t-m_2}) + (1-\alpha) \cdot (\ell_{t-1} + b_{t-1}) \\
\text{trend equation:} \quad & b_t = \beta \cdot (\ell_t - \ell_{t-1}) + (1-\beta) \cdot b_{t-1} \\
\text{daily seasonality :} \quad & D_t = \gamma \cdot (y_t - \ell_t - W_{t-m_2}) + (1-\gamma) \cdot D_{t-m_1} \\
\text{weekly seasonality :} \quad & W_t = \delta \cdot (y_t - \ell_t - D_{t-m_1}) + (1-\delta) \cdot W_{t-m_2}
\end{aligned} \tag{3.9}$$

where $t = 1, 2, ..., T$. The forecast equation for predicting $h-$time steps ahead is the following:

$$\hat{y}_{T+h|T} = \ell_T + h \cdot b_T + D_{T+h-m_1} + W_{T+h-m_2}$$

## Multiplicative method

On the other hand, when the level of the time series affects the variation caused by seasonal factors, causing larger seasonal variation at higher values of $y_t$, the multiplicative version is appropriate:

$$\text{forecast equation:} \qquad \hat{y}_t = (\ell_{t-1} + b_{t-1}) \cdot D_{t-m_1} \cdot W_{t-m_2}$$

$$\begin{aligned}
\text{level equation:} \qquad & \ell_t = \alpha \cdot \left( \frac{y_t}{D_{t-m_1} \cdot W_{t-m_2}} \right) + (1-\alpha) \cdot (\ell_{t-1} + b_{t-1}) \\
\text{trend equation:} \qquad & b_t = \beta \cdot (\ell_t - \ell_{t-1}) + (1-\beta) \cdot b_{t-1} \\
\text{daily seasonality :} \qquad & D_t = \gamma \cdot \left( \frac{y_t}{\ell_t \cdot W_{t-m_2}} \right) + (1-\gamma) \cdot D_{t-m_1} \\
\text{weekly seasonality :} \qquad & W_t = \delta \cdot \left( \frac{y_t}{\ell_t \cdot D_{t-m_1}} \right) + (1-\delta) \cdot W_{t-m_2}
\end{aligned} \tag{3.10}$$

where $t = 1, 2, ..., T$.

$$\text{future-forecast equation:} \qquad \hat{y}_{T+h|T} = (\ell_T + h \cdot b_T) \cdot D_{T+h-m_1} \cdot W_{T+h-m_2}$$

where $h = 1, 2, 3, ...$ defines the $h-$time steps forward prediction.

## Initializing *Taylor's Double Seasonal* model

In his paper about forecasting and anomaly detection of network traffic, Szmit and Szmit [2012] use an arbitrary method to initialize the level, trend, and the daily and weekly seasonal components specifically for the *additive* case:

$$\begin{aligned}
\ell_0 &= y_1 \\
b_0 &= 0 \\
D_{0,1} = D_{0,2} = ... = D_{0,m_1} &= 0 \\
W_{0,1} = W_{0,2} = ... = W_{0,m_2} &= 0
\end{aligned} \tag{3.11}$$

The corresponding initialization method for the *multiplicative* case would then be

$$\begin{aligned}
\ell_0 &= y_1 \\
b_0 &= 1 \\
D_{0,1} = D_{0,2} = ... = D_{0,m_1} &= 1 \\
W_{0,1} = W_{0,2} = ... = W_{0,m_2} &= 1
\end{aligned}$$

Another initialization method is proposed by Taylor [2003] when forecasting electricity demand with *multiplicative* daily and weekly seasonality, which is also implemented on the paper of Jalil et al. [2013], and uses simple averages of the first few data observations. Adapted to our sesonality characteristics where the length of the daily cycle is $m_1 = 1440$ and the length of the weekly cycle is $m_2 = 10080$, these initial components were chosen as follows:

- The *initial trend*, $b_0$, was chosen as the average of (1) $\frac{1}{10080}$ of the difference between the mean of the first 10080 and second 10080 observations, and (2) the average of the first differences for the first 10080 observations.

- The *initial level*, $\ell_0$, was chosen as the mean of the first 20160 (= 2 × 10080 ) observations minus 10080.5 times the initial trend, i.e.

$$\ell_0 = \frac{1}{2m_2} \cdot \sum_{t=1}^{2m_2} y_t - (m_2 + 0.5) \cdot b_0$$

- The initial values for the *within-day seasonal index*, $D_t$, were set as the average of the ratios of actual observation to 1440-point centered moving average, taken from the corresponding minute in each of the first seven days of the time series.

  More specifically, the $n^{th}$ component of the initial daily index $D_0$, which corresponds to the $n^{th}$ minute of the day, $n \in [1, 1440]$, will be

$$\frac{1}{7} \cdot \sum_{k=1}^{7} \frac{y_{n+(k-1)\cdot 1440}}{\text{mean of the } k^{th}\text{day of the first week}}$$

- The initial values for the *within-week seasonal index*, $W_t$, were set as the average of the ratios of actual observation to 10080-point centered moving average, taken from the corresponding minute on the same day of the week in each of the first two weeks of the time series, devided by the corresponding initial value of the smoothened within-day seasonal index, $D_t$.

  In mathematical terms, the $n^{th}$ component of the initial weekly index $W_0$, which corresponds to the $n^{th}$ minute of the week, $n \in [1, 10080]$, will be

$$\frac{1}{2} \cdot \frac{1}{D_{(n \bmod 1440)}} \cdot \sum_{k=1}^{2} \frac{y_{n+(k-1)\cdot 10080}}{\text{mean of the } k^{th} \text{ week}}$$

  where ($n$ mod 1440) is the modulo operation, i.e the remainder of the Euclidean division of $n$ by 1440.

The corresponding initialization method for the *additive* case would produce the same initial level and trend component, while the initial seasonal components would be

$$\frac{1}{7} \cdot \sum_{k=1}^{7} \left( y_{n+(k-1)\cdot 1440} - [\text{mean of the } k^{th}\text{day of the first week}] \right)$$

for the within-day seasonal index, and

$$\frac{1}{2} \cdot \sum_{k=1}^{2} \left( y_{n+(k-1)\cdot 10080} - [\text{mean of the } k^{th} \text{ week}] - D_{(n \bmod 1440)} \right)$$

for the within-week seasonal index.
We refer to this initialization method for the two different cases as (3.12). The initialization method influences the values of the smoothing parameters. How accurate the predictions on the first days or weeks, which is mainly influenced by the initialization method, determines whether the smoothing parameters will be close to zero or close to one.

## The smoothing parameters

The values that characterize the performance of this model are the smoothing parameters, which are the following:

$$\begin{aligned}
\alpha &= \text{the level parameter}\\
\beta &= \text{the trend/slope parameter}\\
\gamma &= \text{the parameter for the small seasonal cycle (daily in our case)}\\
\delta &= \text{the parameter for the bigger seasonal cycle (weekly in our case)}
\end{aligned}$$

which follow the restriction of

$$0 \leq \alpha, \beta, \gamma, \delta \leq 1.$$

These four smoothing parameters define how much weight the components will give to the recent observations and how much to those from the distant past, before they get updated. As an example consider the daily seasonal component $D_t$. The value of $\gamma$ will define whether for the update of this component more weight will be given in the seasonal variation between the minute $t$ of the current day and the time instant $t$ of the previous day (when $\delta$ is close to 1) or whether for this computation the variations between consecutive days belonging in the distant past are considered more informative, and thus $\gamma$ will take a lower value, close to 0.

Furthermore, the weekly seasonal component $W_t$, where $1 \leq t \leq 10080$, describes the seasonal variation between a specific day of a week and the same weekday of the precedeing week, for this particular minute $t$. A high value of $\delta$ would mean that for the computation of this $W_t$ more weight is given in the variation of this specific time instant between the current and the precedent week than in the variation of the precedent weeks. Similarly for the trend parameter, a very high value of $\beta$ - close to one - translates into a system where for computating the trend in the dataset more weight is given in the change of the underlying level between the present and one time step ago, that the trend as was computed from the past observations.

CHAPTER 4

# Experimental setup

After a thorough investigation of the properties of the available dataset in chapter 2 and the explanation of the theory behind the exponential smoothing forecasting techniques in chapter 3, we move on to the implementation of these techniques in the dataset, which is the topic of the current chapter. Apart from the implementation details and a brief description of the obstacles found, together with solutions proposed, we also present ways of evaluating the accuracy of the proposed models, as well as a comparison between them. As with most datasets, there are mainly two types of problems: the missing values and the outliers.

## 4.1 Data engineering

Before we proceed to the evaluation and comparison of the models, we first prepare the data set. This includes a method on treating the missing values, whether ouliers occur, how informative they are and how to handle them, and finally defining what part of the dataset will be used for training and what part for testing the models.

As we have seen in Chapter 2, the sample data we work on show daily and weekly seasonality patterns for both *median pageload time* and *number of pageviews*. Because of the presence of abnormal points in the dataset, which we are interested in detecting with our anomaly detection model described in the following chapter, as well as the presence of missing values, the dataset needs to be ”cleaned”.

The dataset is large enough, and so choosing to exclude weeks 5 and 9, together with the first and last two days when training the model should not be a problem (see table of figure 4.4. If doing so, the percentage of the missing values for this new dataset yielded from the original one, which consists of 9 complete weeks in total, would be around 1%, making this dataset even more appropriate to work with. However, by excluding the two complete weeks leaves us with discontinuous time series, since the behaviour of the 6th week for example will follow that of the 4th week, and not the 5th as it should have been. This discontinuity of the time series would cause problems when training the model, especially when there is a considerable trend in the dataset.

In graph 4.1 we plot the metric *number of pageviews* for weeks 3, 4 and 6, in the same graph. We see that for weeks 3 and 4 the level of the series, when looking at each of the two weeks independently, is quite similar. However, this does not hold when looking at weeks 4 and 6. The underlying level of the time series seems to rapidly increase during the transition from week 4 to week 6, making us assume that, by not considering week 5 when training the model we fail to capture the trend in the data - if any - leading to the trend component and subsequently the trend parameter $\beta$ to be distorted when searching for the optimal smoothing parameters.

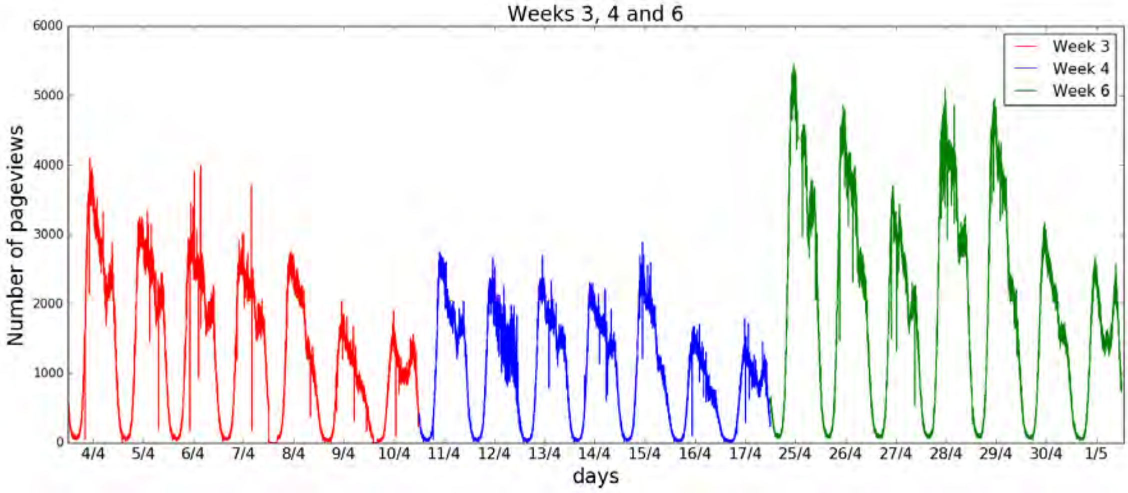


Figure 4.1: The big change in the level of the time series between weeks 4 and 6, may indicate that the discontinuity of the time series caused by excluding week 5 could lead to problems while training the models, making our performance results less reliable.

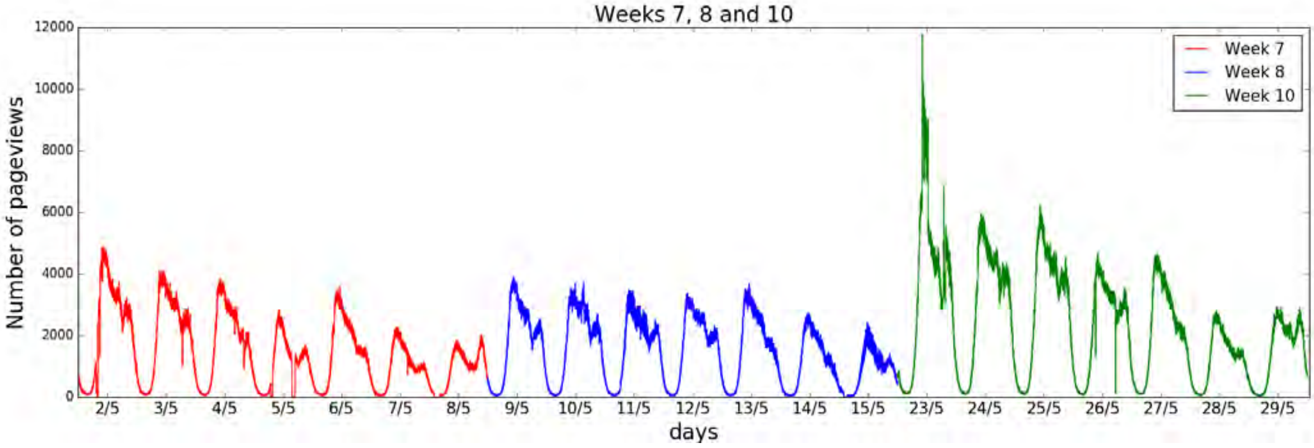


Figure 4.2: The difference between the underlying level of week 8 and week 10 may indicate that week may contain valuable information

From graph 4.2 it is also clearly seen that weeks 7 and 8 seem to resemble each other in terms of their underlying level while week 10 is considerably different, which may denote potential missing information of the dataset that the exclusion of week 9 may cause. On the other hand from figure 4.3, it is visible that the change of the level between consecutive weeks is not always increasing or decreasing smoothly. This rapid change between the underlying level of some successive weeks can be attributed to external factors which are not known and will not be investigated in the current documentation, or probably to the existence of other extra seasonal patterns. The data seem to have a changing level rather than a trend, and so we conclude that the exclusion of weeks 5 and 9 from our dataset will not have a big effect on the computation of the trend components, nor on parameter estimation of the forecasting models.

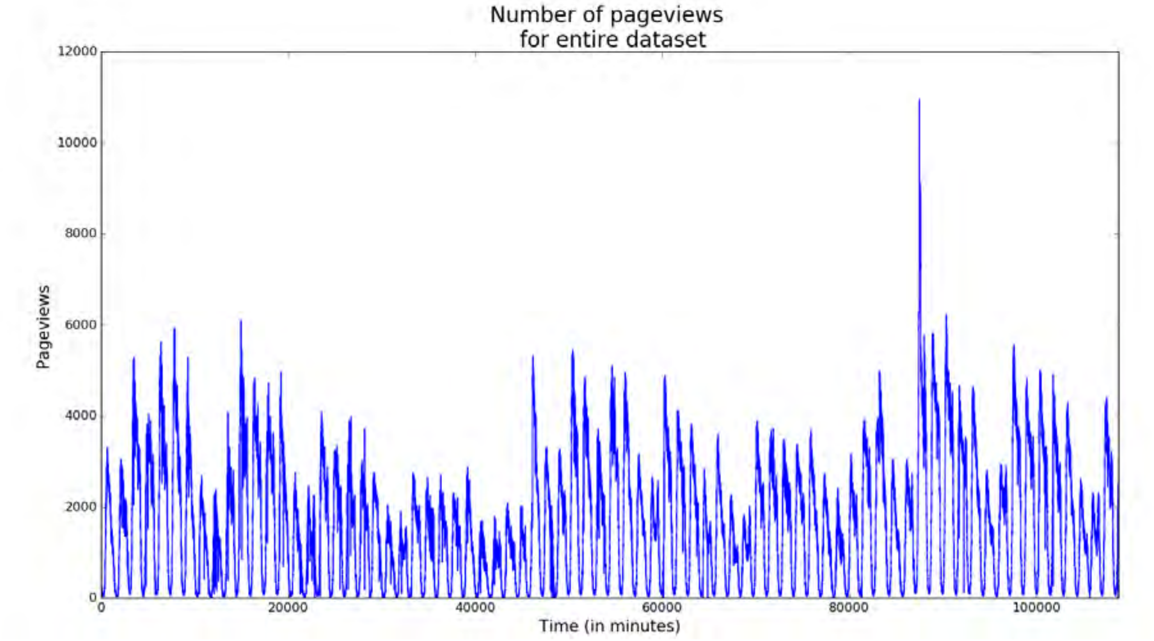


Figure 4.3: The number of pageviews for the entire dataset.

| Time period | Date | Number of missing values | % of missing values with regard to time period |
|---|---|---|---|
| First 2 days | 19/3 - 20/3 | 61 | 2.1% |
| Week 1 | 21/3 - 27/3 | 73 | 0.72% |
| Week 2 | 28/3 - 3/4 | 5 | 0.05% |
| Week 3 | 4/4 - 10/4 | 227 | 2.2% |
| Week 4 | 11/4 - 17/4 | 39 | 0.39% |
| Week 5 | 18/4 - 24/4 | 3124 | 31% |
| Week 6 | 25/4 - 1/5 | 135 | 1.3% |
| Week 7 | 2/5 - 8/5 | 259 | 2.6% |
| Week 8 | 9/5 - 15/5 | 81 | 0.8% |
| Week 9 | 16/5 - 22/5 | 2700 | 26.8% |
| Week 10 | 23/5 - 29/5 | 12 | 0.12% |
| Week 11 | 30/5 - 5/6 | 211 | 2.1% |
| Last 2 days | 6/6 - 7/6 | 926 | 62% |

Figure 4.4: Table of missing values for the whole dataset. Due to the high amount of missing values, we choose to exclude weeks 5 and 9, as well as the first and last two days, since we chose to work with complete weeks.

Since we are dealing with time series and each value is used for the forecast of the next time step, it is invalid to not consider the values of all the minutes during each day. None of the models can tolerate gaps in the series - they need to use observations for each minute of the day, for all the weeks. That is why we need to fill in the all the missing instances with a number. In the present moment we fill in the missing values with the average of the preceding and proceeding available observation. Moreover, concearning the outliers, we do not replace them with other, more "normal", values. In this initial model comparison we only deal with the missing values, and later we will present a method for identifying and replacing the anomalous values, together with the impact of this on the models' performance.

## Train and test set

To start with, we note the following: in order to produce an unbiased estimate of the perfomance of the models on unseen data, we will not consider how well a model fits the historical data. The accuracy of forecasts can only be determined by considering how well a model performs on new data that were not used when estimating the model. For this reason, it is a common practice among the statisticians and researchers of the forecasting community to devide the available initial dataset into two subsets: the *train set* and the *test set*.

The *train set* is used for building or training the models, which mainly means defining the optimal parameters for this model, i.e. the parameters that better describe the behaviour of the available historical data. In our case, this is equivalent with defining $\alpha, \beta$ and $\gamma$ in the case of the Holt-Winters or defining $\alpha, \beta, \gamma$ and $\delta$ in the case of Taylor's double seasonal model.

Furthermore, after training the model, or - as it sometimes said - after tuning the model parameters, we are to test how well the model performs with these optimal parameters. To do that, we use an appropriate accuracy measure - depending on the circumstances and dataset - which in principle is measuring the forecast error, i.e. the deviation of the forecasts from the actual values. It is important that this evaluation is done in observations that were not used to define the parameters of the model, and so, we use the *test set* for this purpose.

As a rule of thumb, adopted also by Hyndman and Athanasopoulos [2013], the 80%-20% rule is used, which requires the 80% of the dataset to be used as train set and the remaining 20% as test set.

| Set | Week | % of entire dataset |
|---|---|---|
| train set | 1, 2, 3, 4, 6, 7, 8 | 78% |
| test set | 10, 11 | 22% |

Table 4.1: The partition of our dataset into a train and a test set follows the 80%-20% rule of thumb.

## 4.2 Parameter estimation

We explained before that the smoothing parameters depict how much "memory" the models should have, i.e. how far in the past they should look when considering historical observations for making predictions about the future. A big value of a parameter - a value closer to 1 - means that the model is taking more recent observations in higher account than past observations. On the contrary, a parameter value close to zero makes the model consider more the observations of the past than the more recent ones.

One common practice for selecting the optimal smoothing parameters from the available historical data is to use a data-driven procedure optimizing a certain criterion. For every combination of the parameters $\alpha, \beta$ and $\gamma$ for Holt-Winters method or $\alpha, \beta, \gamma$ and $\delta$ for Taylor's double seasonal model the one-step-ahead forecast errors are computed. We then choose the set of parameters that produced the smallest one-step-ahead forecast errors.

Throughout the literature, mainly the following measures are considered for the estimation:

1. *Mean Squarred Error* (*MSE*)
   Not good when there are many outliers. Used when you want to give bigger importance to the influence of outliers (used by Souza et al. [2007], Gould et al.).

$$MSE = \frac{1}{T} \cdot \sum_{t=1}^{T} (y_t - \hat{y}_t)^2$$

2. *Mean Absolute Error* (*MAE*)
   which is mostly preffered when there are many outliers present in the dataset (used by Szmit and Szmit [2012] ).

$$MAE = \frac{1}{T} \cdot \sum_{t=1}^{T} |y_t - \hat{y}_t|$$

In our anomaly detection context, as we have also visually proved on previous chapters, there is a high amount of outliers. Consequently, the MSE criterion is not the most appropriate when dealing with contaminated time series, as also stated by Gelper et al. [2010], since some very big forecast errors can cause an explosion to the MSE, which will lead to smoothing parameters being biased towards zero. For this reason, we choose the MAE instead, which is less influenced by the presence of outliers. The parameters are then found by minimizing the mean of the one-step-ahead forecast errors.

For all models, optimal parameters were found by an itterative search which minimizes the errors for a total of $30^3 = 27000$ combinations of $\alpha$, $\beta$ and $\gamma$ for the Holt-Winters case, and $20^4 = 160000$ combinations of $\alpha$, $\beta$ ,$\gamma$ and $\delta$ for the Taylor's double seasonality model. We chose to repeat this process for two different measures, the Mean Squared Error (MSE) and the Mean Absolute Error (MAE), yielding 12 different sets of optimal parameters. Those parameters found by minimizing the MSE are not included in the paper though, since they yielded less accurate forecasts than thos found by minimizing the MAE.

## 4.3 Evaluation and comparison

First we present the results when all models to be tested are applied on the dataset where ouliers were not treated differently than the other data points. The table below features all the forecasting models that will be tested and compared with the old model and with each other:

| Reference name | Model | Seasonal pattern | Seasonality type |
|---|---|---|---|
| EWMA (old model) | EWMA | - | - |
| HW-add(1440) | Holt-Winters | daily | additive |
| HW-mult(1440) | Holt-Winters | daily | multiplicative |
| HW-add(10080) | Holt-Winters | weekly | additive |
| HW-mult(10080) | Holt-Winters | weekly | multiplicative |
| Taylor DS-add | Taylor's DS | daily+weekly | additive |
| Taylor DS-mult | Taylor's DS | daily+weekly | multiplicative |

Table 4.2: A summary of all the forecasting models to be tested for each of the metrics seperately.

On tables 4.3 and 4.4 following we can see the performance of all the forecasting models. Concerning the metric *"Number of pageviews"*, it is obvious from table 4.3 that all models outperform the already existing predictive model, *EWMA*. More specifically, the EWMA yields a mean absolute error of 1012.13, where for all other models this value lays between 76.48 and 107.72, an improvement between 89% and 92%. This was to be expected since the behaviour of this metric changes fast due to seasonal factors taking place during the day and week. A model unable to grap and consider these seasonal changes, such as EWMA, is weak in terms of performance and accuracy.

Moreover, another obvious realization about this metric, coming also from table 4.3, is that the additive models perform better than the multiplicative ones. This confirms our visual realization made on chapter 2, where we observed that the magnitude of the seasonal pattern does not increase as the data values $y_t$ increases. Furthermore, among the three different additive models, Taylor's model with double seasonality (daily and weekly) performs better than both Holt-Winters models, yielding a MAE value of 76.48.

One surprising result is that, even though in the data analysis chapter the graphs we produced showed a clear weekly pattern for this metric, the model of Holt and Winters assuming daily seasonality performs better than when assuming weekly seasonality, where the MAE is reduced from 82.10 to 80.10. On the other hand, when both the change between two consecutive days and the change between two consecutive weeks is considered - which is what Taylor's model does - the accuracy is considerably improved.

| Model id | Model | MAE |
|---|---|---|
| 1 | EWMA | 1012.13 |
| 2 | HW-add(1440) | 80.10 |
| 3 | HW-mult(1440) | 94.16 |
| 4 | HW-add(10080) | 82.10 |
| 5 | HW-mult(10080) | 107.72 |
| 6 | Taylor DS-add | 76.48 |
| 7 | Taylor DS-mult | 78.06 |

Table 4.3: Comparison of the performance of all models for the metric *"Number of pageviews"* on the test set. These performance results were computed without considering the filled-in missing values with an average, in the test set.

For the metric *"Median pageload time"* there is a major imporvement in the performance of the models compared to EWMA, even though it is of lower magnitude compared to the improvement of the previous metric. Taylors' model considering double additive seasonality outperforms the rest of the models.

| Model id | Model | MAE |
|---|---|---|
| 1 | EWMA | 78.62 |
| 2 | HW-add(1440) | 22.06 |
| 3 | HW-mult(1440) | 20.84 |
| 4 | HW-add(10080) | 23.51 |
| 5 | HW-mult(10080) | 22.09 |
| 6 | Taylor DS-add | 20.14 |
| 7 | Taylor DS-mult | 20.41 |

Table 4.4: Comparison of the performance of all models for the metric *"Median pageload time"* on the test set.

For Taylor's model, two values of the MAE are included in the tables. The first one refers to the performance when the initialization method (3.11) was used, while the second refers to (3.12), from which the following realization is made: the initialization method is closely related to the performance of the model.

## Analysis of Variance for the errors

We use the one-way *ANOVA* to test whether the differences on the results of the performance of the models are statistically significant.

For each of the two metrics, we will test the null hypothesis

$$H_0 : \mu_1 = \mu_2 = ... = \mu_7$$

where $\mu_i$ is the mean aboslute error of model $i$ as presented on tables 4.3 and 4.4, in contrast to the alternative hypothesis

$$H_1 : \text{At least one mean is different than the others}$$

For *number of pageviews*, and for a significance level of 95%, we get a p-value equal to zero when comparing all the seven models, and consequently we reject the null hypothesis that all the means are equal. Clearly the reason for that is the greater value of the mean of absolute errors of model 1, i.e. EWMA. By excluding it, the p-value becomes 0.03, which is still lower than 0.05, occuring again to the rejection of $H_0$. When excluding model 5, the multiplicative Holt-Winters with weekly seasonality, we get the p-value of 0.13, a value greater than 0.05 (see table 4.6). We conclude then for the metric *number of pageviews* that, even though Taylor's model with additive seasonality performs better than the rest of the models, the differences of the MAE with the one of models 2,3,4 and 7 are not statistically significant. On the other hand, these differences could be that were greater when more data were available for testing the model, than just 2 weeks. For that reason we consider Taylor's model with additive daily and weekly seasonality to be the one that best describes the behaviour of the metric *number of pageviews*.

| models | p-value |
|---|---|
| 1, 2, 3, 4, 5, 6, 7 | 0 |
| 2, 3, 4, 5, 6, 7 | 0.03 |
| 2, 3, 4, 6, 7 | 0.13 |

Table 4.5: The p-values occuring from the one-way ANOVA, for the metric *number of pageviews*.

For *median pageload time*, and after going through the same process, the one-way ANOVA yields that, appart from EWMA, all remaining models do not have a statistically significant difference on mean absolute errors. For the same reasons explained about the other metric though, we choose Taylor's model with additive daily and weekly seasonality to describe the behavior of this metric, too.

| models | p-value |
|---|---|
| 1, 2, 3, 4, 5, 6, 7 | 0 |
| 2, 3, 4, 5, 6, 7 | 0.64 |

Table 4.6: The p-values occuring from the one-way ANOVA, for the metric *median pageload time*.

## Forecast horizons: Why do we forecast only 1 time step ahead?

In most of the available literature about predictive models and their accuaracy, it is a common practice when evaluating the performance of a predictive model to consider the accuracy of the model for several different forecast horizons. Most of these evaluations have shown that in general a short term forecast can be more accurate than forecasting a time series several time steps in the future. In simple words, the further in the future a model is forecasting, the most possible it is to produce inaccurate predictions, as noted also by Jalil et al. [2013] and , among others. On their paper about forecasting time series with multiple seasonal patterns, Gould et al. produce a graph of the mean squarred forecast error (MSFE) in terms of $h$, where they conclude that the MSFE is minimized when considering the minimum lead time, i.e. $h = 1$, or a lead time that is multiple of the seasonal period $m$, meaning that forecasts are in general more accurate when they are made for the same time of day as the last observation.

On the other hand, the reader by now must have realised that in our evaluation and comparison of the different models in this chapter, we only considered one value for the forecast horizon, the value of $h = 1$. The logic behind this decision is simple: We know that, after correct training, a predictive model is capable of knowing the expected behaviour of a random variable, with some - hopefully small - deviations. When the observed behaviour is deviating by a big amount from the expected behaviour then in general this is considered an anomaly. To detect an anomaly means to make a forecast for a specific point in time and then compare this predicted value with the actual value: if the difference is big it may be possible that we have identified an anomaly. In the case of real-time anomaly detection though, the true/actual value is needed for this evaluation to be carried out. For that reason, even if we made a prediction for $h = 60$ for an hour ahead or $h = 1440$ for a day ahead, we would need to wait until that time when we have the actual value, before we make the evaluation and thus conclude on whether we identified an anomaly or not.

In addition to that, when forecasting for one time step ahead, all the historical data gathered up to now will be used for predicting the value in the next time points. This makes it more possible for an accurate prediction to occur. On the other hand, if we would use a larger forecast horizon, it would mean that the current value would have been predicted $h$ time steps ago, taking into account less historical data. Considering all these, it seems reasonable to only consider a forecast horizon equal to $h = 1$.

This is also the reason that we estimated the smoothing parameters of the models by minimizing the sum of squarred one-step ahead errors instead of $h$-step ahead errors: because our only case for forecasting will be for the next minute.

## Summary

As we have proved in this chapter, Taylor's double seasonal model assuming additive daily and weekly seasonality is the one that performs best when compared to the other models of the exponential smoothing family of forecasting methods, for both metrics. After conducting an ANOVA test the results showed that the differences on the MAE of these models are not statistically significant, but evaluation on a bigger test set could produce more concrete differences between the models. Regarding the second best performing model, Holt-Winters accounting additive daily seasonaliy, it could also be beneficial in cases where there are not enough time series data for the two metrics. It would require only one day of time series data to be initialized, and thus a couple of complete days of data would be enough for parameter estimation and evaluation. On the other hand, Taylor's model, which also accounts for weekly seasonality, would require seven times more data than Holt-Winters, making it less preferable when there is inadequate data. In the following section where we build the anomaly detection model, only Taylor's model is considered for forecasting.

# CHAPTER 5

# Anomaly detection model

## 5.1 The Gaussian model

We have seen so far methods on how to forecast a time series. Both metrics we investigated had seasonal patterns, and the importance on using a model that captures these seasonalities was underlined by the superior performance in contrast to those that fail to capture the seasonal variations.

By now, the importance of learning the behaviour of the two metrics, and how to achieve that for time series data should be clear: by training the forecasting models of the exponential smoothing family, we were able to make accurate predictions about how the time series is expected to behave in the near future. It is now of greater importance to define a measure of how much the observed behaviour of the time series deviates from the expected patterns.

We use the forecast error of each time instant to define this deviation betwen the actual and predicted value, for a specific point $t$ in time:

$$e_i = y_i - \hat{y}_i$$

A thorough analysis of the errors follows in figures 5.1 and 5.2 and the important realization that

$$e_i \sim N(0, \sigma^2)$$

is made. By using the fact that the errors follow a Normal distribution centered around zero, with a standard deviation $\sigma$, we try in this chapter to define what constitutes an anomaly for our dataset for both metrics, followed by an evaluation of the performance of this *Gaussian anomaly detector*.

### *Number of pageviews*

The dataset of course contains several outliers, the anomalous datapoints as explained before. It is reasonable to assume that the expected values of these observations will be lying far away from the observed, and consequently their forecast errors will be extremely high or low. For both metrics we will consider the forecast errors that are computed by the predictions produced from Taylor's double seasonal model with additive seasonality, the one performing better than the rest. A summary of some statistics regarding the errors of the entire dataset for the metic *number of pageviews* follows below:

$$\begin{aligned} Min &= -3771.8459176 \\ Max &= 5166.85623111 \\ \mu &= -0.0467871619061 \\ \sigma &= 116.152475686 \end{aligned}$$

From this summary we see that there are outliers that are almost 50 standard deviations higher than the mean, and outliers that are more than 35 standard deviations lower than the mean. It is obvious that the presence of outliers will make the normaliy of the errors problematic. A question that naturally arises then is: What is the reason that big errors appear? It may be that either non-accurate training of the forecast model lead to it, overstimating or underestimating a forecast, or either the forecast was accurate enough but the actual value was way higher or lower than the prediction, due to it being an anomalous observation.

Regardless what the reason is, the normality is influenced when considering the outliers. We consequently choose to exclude them, and plot the histogram of the errors for several different definitions of what an outlier is considered to be, in terms of how many standard deviations from the mean it lies. On figure 5.1 we present the histograms of the errors, for the cases where outliers are excluded and are defined to be observations lying further than 3, 5, 8 and 10 standard deviations from the mean.

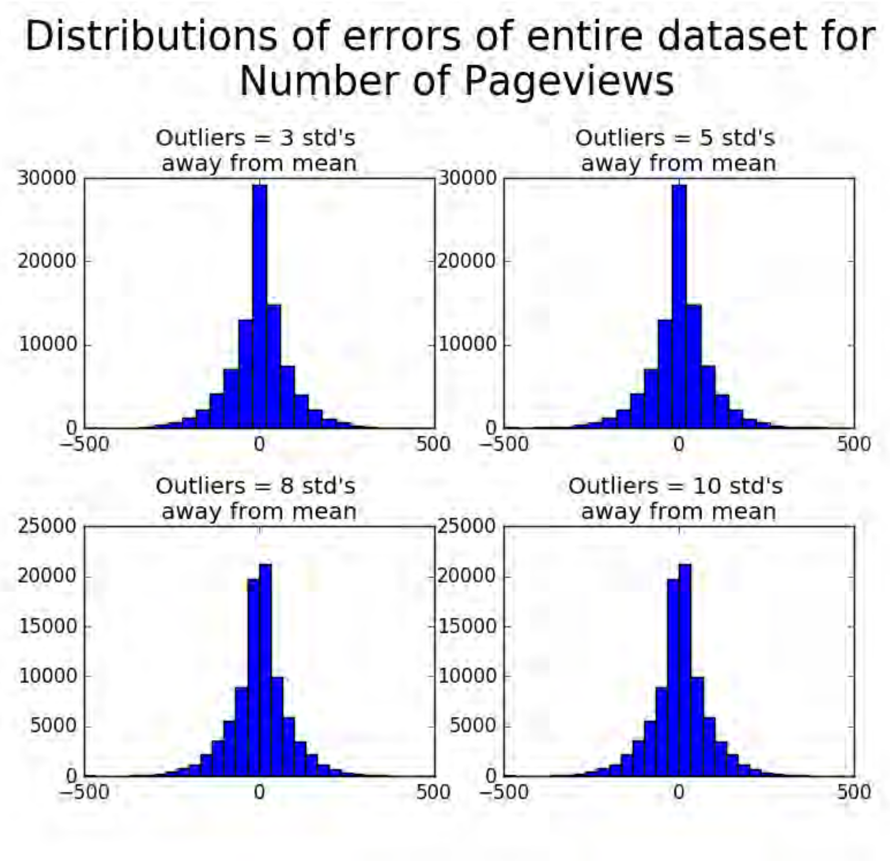


Figure 5.1: The distribution of the forecast errors for the metric *”Number of pageviews”*, excluding outliers. It is clear that, after excluding the outliers for all cases the forecast errors are normally distributed around zero. As we see, the normality of the data is not influenced whether we define an outlier to be 3, 5, 8 or 10 standard deviations away from the mean. These outliers comprise less than 3% of the entire dataset.

### *Median pageload time*

Same conclusions can be drawn also for this metric: the forecast errors of the *median pageload time*, after excluding outlying observations which are less than 1% of the entire dataset, are normally distributed around zero (see figure 5.2 below).

$$\begin{aligned} Min &= -3706.46053739 \\ Max &= 4402.27936453 \\ \mu &= 0.108581646373 \\ \sigma &= 72.8083377369 \end{aligned}$$

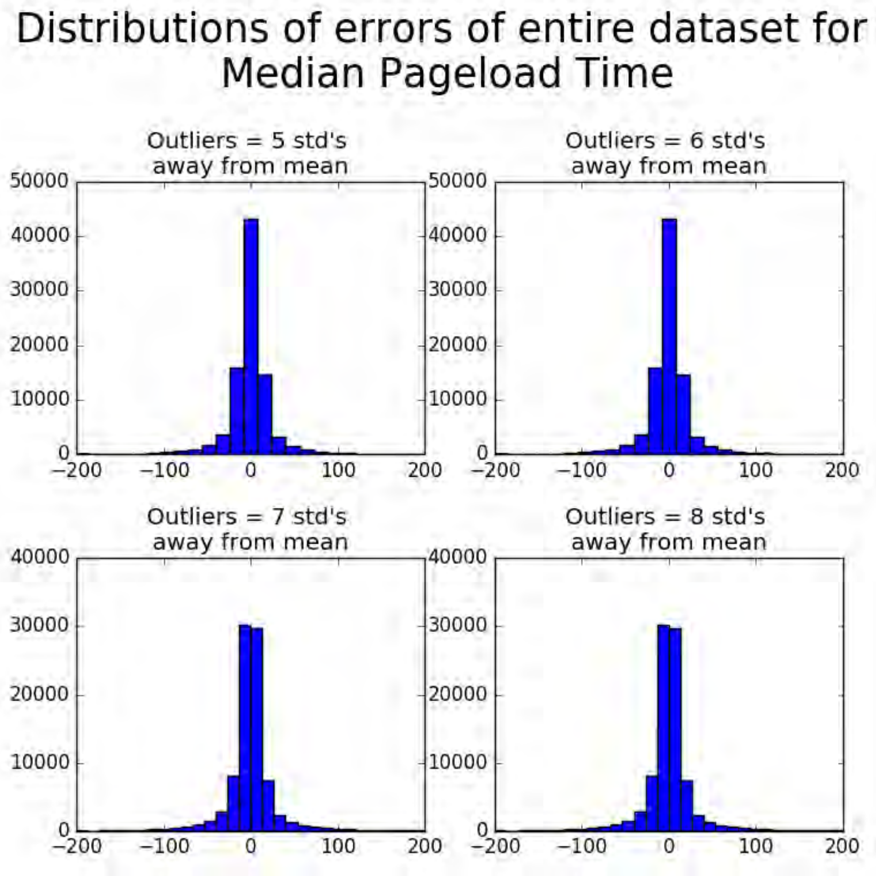


Figure 5.2: For this metric too, the distribution of the forecast errors is normal around zero.

One important note here, concerning which errors should be included when presenting the above histograms, is the following: on chapter 3 when explaining the initialization methods of Holt-Winters, a method of averaging the observations of the first seasonal cycle - which is the first day or the first week of the dataset, depending on which seasonality pattern we chose - was explained. For Taylor's model, initialization (3.12) was also based on the same reasoning of averaging. It is important here to stress that, the forecasts produced for the first day or week of Holt-Winters and Taylor's forecasts for the first week when using the initialization (3.12) will produce extremely accurate forecasts, the errors of which will be extremely low, almost negligible. This will result in a biased-towards-zero mean and in a standard deviation that is in fact higher than this computed. The errors of these period are thus not representative and should be excluded when testing their normality.

The next and final step, after defining a measure of deviation between the expected and observed behaviour, is a rule which decides whether each observation, according to this deviation is marked as normal or abnormal. This choice is made according to the distance of the forecast errors from the mean in terms of standard deviations. We seek to define this threshold - the number of $\sigma$'s - that makes the most accurate distinction between a normal observation and an anomaly (similarly with the model of Ward et al. [1998]).

## 5.2 Generating an alarm and evaluating performance

We should not assume that a failure in a website is caused by only one anomalous data point of a metric, but instead we should expect that an anomalous behaviour occurs when for multiple time steps the behaviour observed is not compatible with the expected behaviour. Consequently we will use a moving window of consecutive anomalies to be a reason for identifying an abnormal behaviour. A question that naturaly follows and needs to be answered before all is:

*How many consecutive anomalies/outliers should cause the raise of an alarm?*

This question can also be translated in the business context as follows:

*How many time steps (or minutes in our case) is it appropriate for MeasureWorks to inform the website owner company about an anomalous behaviour?*

Of course this number depends on the preference of the company, but also depends on the dataset. There might be cases where there is information, coming from background knowledge, that an error or failure on a website results after $k$ consecutive abnormal observations, where $k$ is to be determined. In our case we make the following conventions:

- Individual abnormal observations will be refered to as *outliers* or *anomalous data points.*
- An alarm is raised, indicating that *anomalous behaviour* was detected and needs attention, when $k = 3$ consecutive outliers are detected, or when $k = 3$ outliers are observed within 5 minutes. We refer from now on to any succession of 3 or more outliers as an *anomalous area.*

# Evaluating True/False positives

As we explained in the beginning of chapter 4, the missing values of the dataset were filled in with the average of the preceeding and proceeding observations, a necessary process for training the model. We did this both on the train and the test set, even though the forecast errors for the replaced missing values of the test set were not considered when computing the MAE of each model. For the evaluation of the anomaly detection model though, we will use the test set where the missing values were not replaced by the average, but were set equal to zero, for the following reason: since we do not know the reason that these values were missing, we could assume that they could be potential anomalies. Treating them as so, will allow us to have more anomalous data points in the test-set for evaluating the performance of our model.

The evaluation of the Gaussian anomaly detector will be done in terms of "True Positives" and "False Positives". The terms Positive/Negative refers to whether we identified and labelled observations as anomalous or not. Since we are counting anomalies, "Positive" refers to an anomalous and "Negative" to a normal observation. The terms True/False refers to whether our labelling was correct or not (source: Flach [2012]). There are then four different possibilities, which in our anomaly detection context will be as follows:

1. *True Positives:*
   Anomalies that correctly were labelled as so. In our context we may refer to them also as *correct detections.*

2. *False Positives:*
   Anomalies that were wrongly labelled as normal observations. We will refer to them also as *missed detections* or *missed anomalies.*

3. *True Negatives:*
   Normal datapoints that were correctly labelled as so.

4. *False Negatives:*
   Normal observations that were wrongly labelled as anomalies, which we will frequently refer to as *false detectons* or *false alarms.*

A good anomaly detection model is the one which maximizes the number of correct detections while at the same time keeps the number of false detections in as low levels as possible. Unfortunately in our case, there is no information about which ones among the outliers are true anomalies or not, a valuable information when selecting which threshold is the one yielding the most *true positives* and the less *false negatives.* In order to evaluate the accuracy of our anomaly detection model we will find the anomalies by visual inspection from the graphs, and consider them to be actual anomalies, which we will seek to identify with the Gaussian anomaly detection model. This evaluation will be done, of course, on the test set.

### *Number of pageviews*

On figure 5.6 the actual behaviour of the metric *number of pageviews* on the test set is plotted. In total 22 anomalous points or anomalous areas were found, as circled with green color on figure 5.6. Our objective is to find out how many of these were identified by the model. Some of these anomalous areas are abnormal data points for more than 3 consecutive minutes. We make the convention that a raise of a single alarm on an anomalous area containing more than 3 consecutive anomalies is considered succesfull. As an example, on the first day of the test set, the 23/5, there is an abnormal behaviour marked for a couple of hours (first gren circle). The detection will be considered succesful if at least three consecutive outliers are detected at these area.

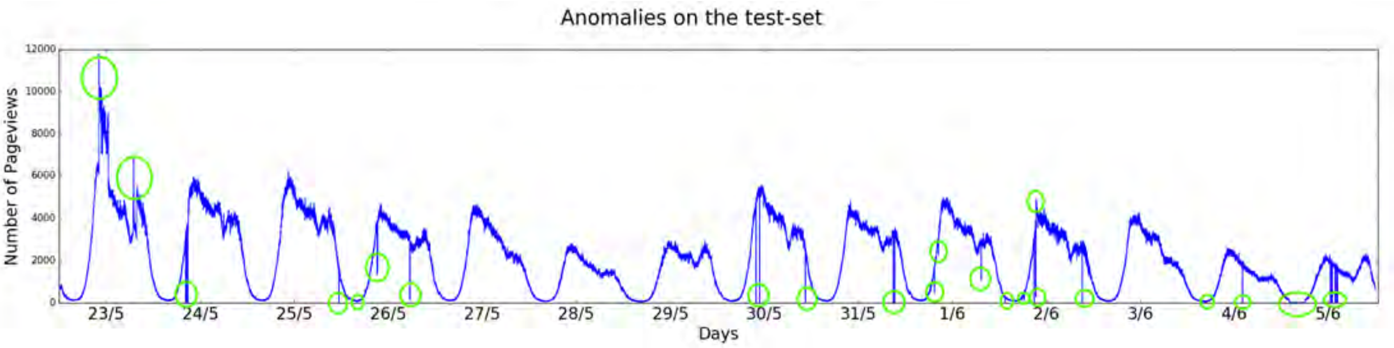


Figure 5.3: The green circles indicate the point of appearance of an outlier or the area of an anomalous behaviour. We use these as ground truth when evaluating the performance of the anomaly detector. See also figure 15 on the appendix.

We will test four different thresholds: an observation will be defined as anomalous/outlier when its forecast error has a distance greater than 3, 5, 8 or 10 standard deviations from the mean of the errors. On table 5.2, as well as on figure 5.7 on the next page we see the results of all the four cases. It seems that the threshold of $3\sigma$ detects the most anomalous areas or outliers compared to the rest. The cost for that though is the high number of False Negatives, a value way greater than the remaining thresholds. For the threshold of $10\sigma$ we observe the smallest number of false alarms, though its performance is notably inferior than the threshold of $3\sigma$.

| Threshold | Correct detections | False detections |
|---|---|---|
| $error > 3\sigma$ | 20/22 | 587 |
| $error > 5\sigma$ | 17/22 | 60 |
| $error > 8\sigma$ | 16/22 | 29 |
| $error > 10\sigma$ | 16/22 | 5 |

Table 5.1: Performance of the anomaly detector for 4 different thresholds, for the metric *number of pageviews*. An objective belief is that the threshold of $5\sigma$ is balanced between the number of correct detections and false alarms.

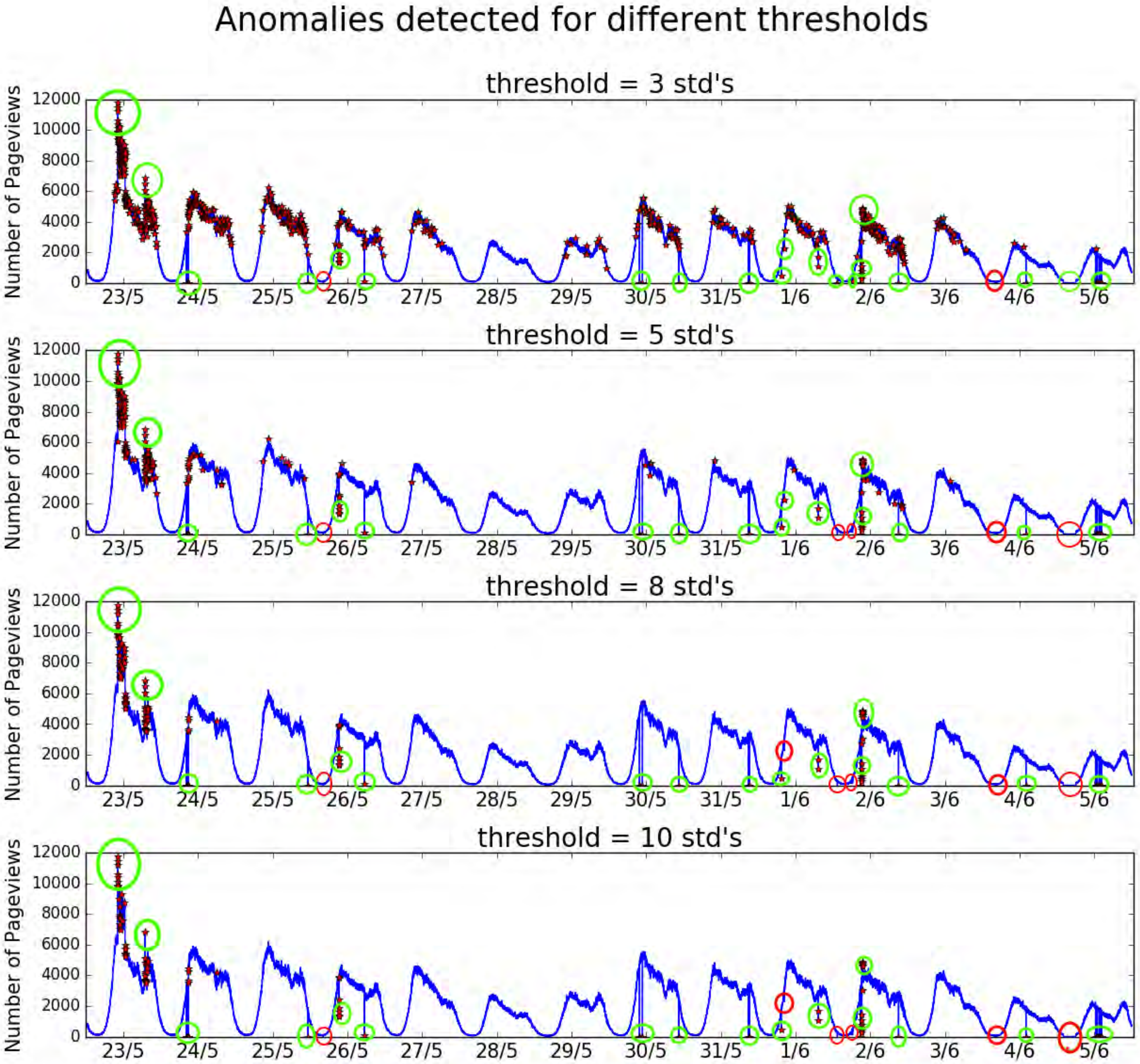


Figure 5.4: For the four different cases of thresholds chosen, the anomalous points detected are marked as red stars. The *True Positives* are marked as green circles, while the *False Positives* are circled in red color.

A closer look on figure (5.5) reveals the reason why the Gaussian anomaly detector with threshold $3\sigma$ succesfully identifies the presence of more anomalous areas compared to the other thresholds. These extra detections, the two red circles in the beggining of 2/6 and the one in the beggining of 5/6, are taking place in the early morning hours between 1:00am to 6:00am. In these early morning hours where the anomalies occured, the level of the time series data is very low and thus the drop on the number of pageviews is small enough and is captured only by the - more sensitive in small changes threshold of - value $3\sigma$.

One way to deal with this problem is to draw not one, but 1440 distributions for the residuals instead, one for each minute of the day. Then the distributions corresponding to the minutes for these morning hours would have different standard deviations - smaller in particular - than the rest, making them more sensitive to changes, resulting in the detection of future errors. We can see that the errors behave differently during the day - smaller errors during the less busy hours and greater errors during the busy hours - when plotting the forecast errors for the test set, shown in figure 15.

On the other hand, one of the drawbacks of this method is that, we would need at least 30 days of data before we could compute the mean and standard deviation of each of these 1440 normal distributions. As stated by David M. Diez [2012], the population standard deviation can reasonably be substituted by the sample standard deviation when the sample size is at least $n = 30$. An alternative to that could be to draw 2 different distributions, one for the busy hours where the errors are higher and one for the less busy hours, in which the forecast errors fluctuate in lower levels, or even use a classification method to define this number of different normal distributions.

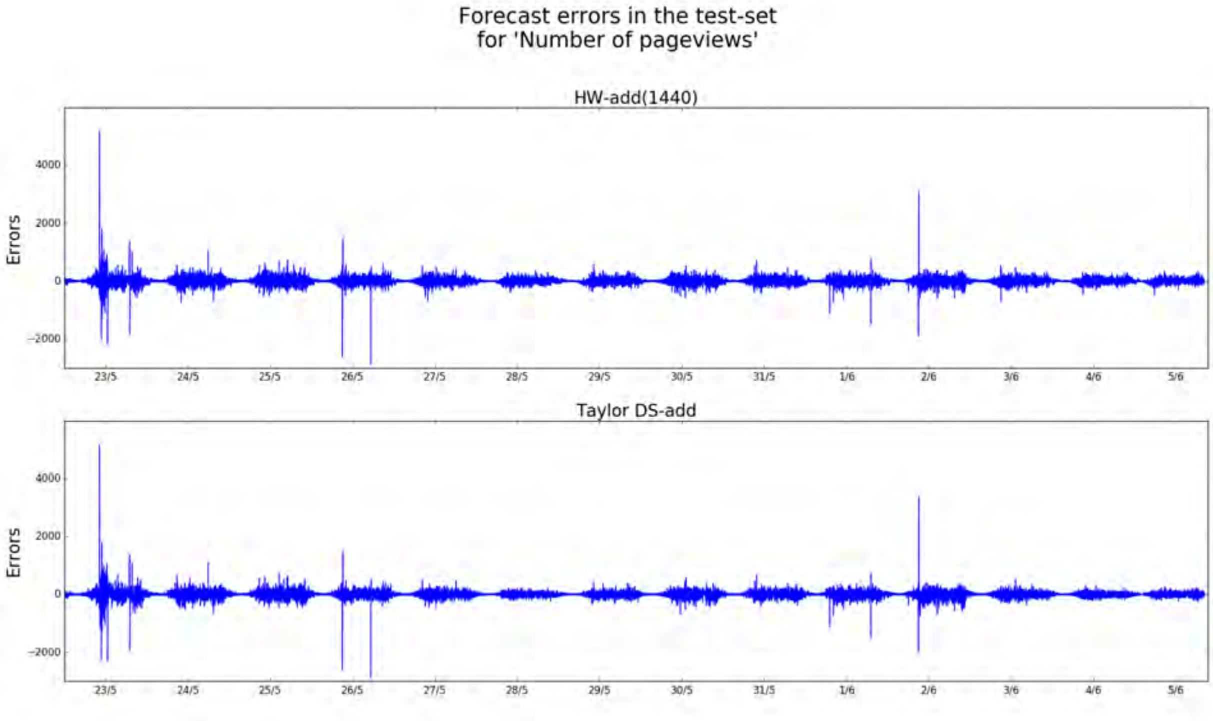


Figure 5.5: The forecast errors for the test set for the metric "Number of pageviews", for the two best performing models: Taylor's DS additive and Holt-Winter's with additive daily seasonality.

## *Median pageload time*

Since the behaviour of the metric *Median pageload time* is different than the one of *number of pageviews*, different thresholds will be used. For this metric, due to higher variability, we choose lower thresholds, mainly 5, 6, 7 and 8 standard deviations away from the mean of the forecast errors.

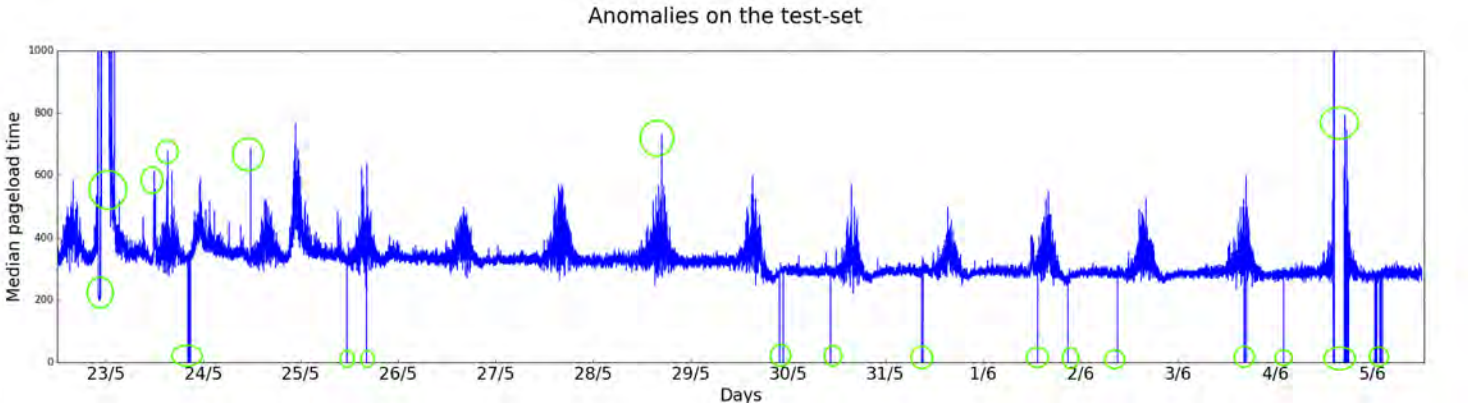


Figure 5.6: The green circles indicate the point of appearance of an outlier or the area of an anomalous behaviour. We use these as grand truth when evaluating the performance of the anomaly detector. See also figure 15 on the appendix.

| Threshold | Correct detections | False detections |
|---|---|---|
| $error > 5\sigma$ | 20/20 | 82 |
| $error > 6\sigma$ | 19/20 | 28 |
| $error > 7\sigma$ | 19/20 | 15 |
| $error > 8\sigma$ | 18/20 | 4 |

Table 5.2: Performance of the anomaly detector for 4 different thresholds, for the metric *median pageload time*. For both metrics the number of correct outliers detected is considered instead of the number of correct alarms raised.

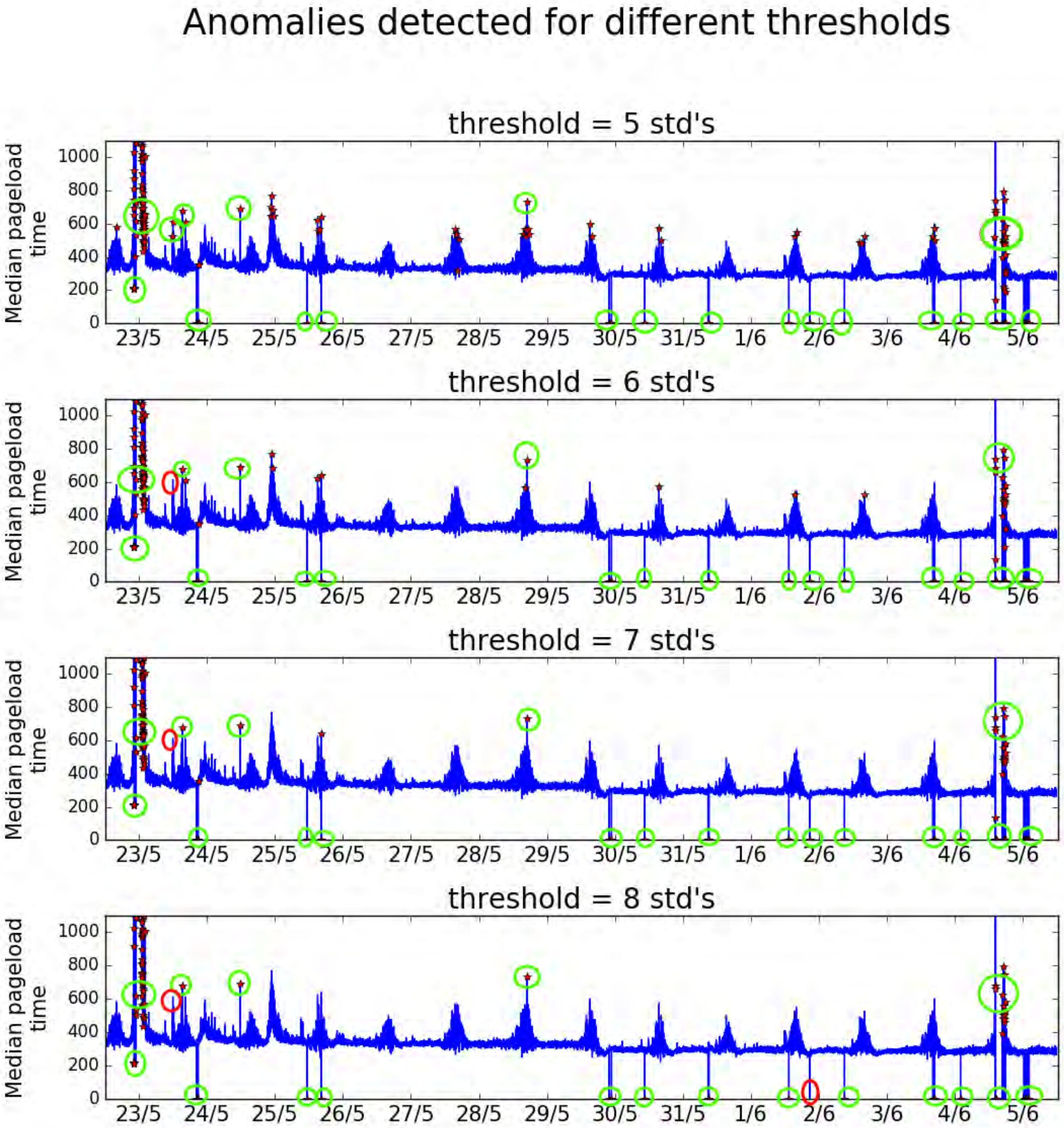


Figure 5.7: For the four different cases of thresholds chosen, the anomalous points detected are marked as red stars. Again the *True Positives* are marked as green circles, while the *False Positives* are circled in red color.

## 5.3 Building robust forecasting models using the Gaussian anomaly detector

In the previous chapter we presented the results when all forecasting models to be tested were applied on the initial dataset where outliers were not treated differently than the other data points, only the missing vaues were filled in with an average, due to the inability of the models to treat missing values. In this section we will talk about the influence of outliers in the forecast accuracy.

In his paper, [Goodwin, 2010] notes three important weaknesses of Holt-Winter's method for forecasting. One of these weaknesses, which is of greatest importance in our context, is the impact of outliers on the forecasts. Similarly, Gelper et al. [2010] remark that, even though the superiority and competitiveness of Holt-Winters' method compared to other more complicated forecast methods, its performance is considerably deteriorating in the presence of outliers. As a variation and extension of Holt-Winters, the same problem applies also to Taylor's double seasonal model.

Remember from chapter 3 that every forecast can be broken into three components, namely the level, trend and seasonality components, which are updated in every time step and their values are determined by the observations previously seen. [Goodwin, 2010] claims that unusually high or low values present in the dataset can cause the models to overestimate or understimate the rates in which the time series is increasing or decreasing, and consequently the value of the forecast of the next period will be too high or too low.

Furthermore, another issue that could arise because of the presence of outliers concerns the optimal smoothing parameters. We have seen that the smoothing parameters determine how sensitive the model is to the latest observations and a common method in finding the optimal ones is by identifying the values of the parameters that produce the most accurate forecasts of the past. As [Goodwin, 2010] notes, in the presence of unusually high or low observations there is the danger that forecasts react too slowly or too rapidly, with a consequence of distorting the smoothing parameters.

This topic on how to treat outliers and anomalous observations for improving the forecast accuracy of the models previously explained is intentionally left for this chapter, after the deployment of the anomaly detection model. The reason for that is the fact that, after an anomalous observation is detected by the Gaussian model, a different value, smoother and more 'normal' is used for determining the values of the level, trend and seasonal components instead. This technique is inspired by the work of Gelper et al. [2010], with the difference that they use robust scale estimators (such as the *median absolute deviation* for example) instead of the standard deviation, which we chose to use for now due to time limitations. This technique of using the Gaussian anomaly detector not only to detect an anomaly, but also to replace it with a 'cleaned' value for improving the future forecasts will be implemented to all the forecasting methods examined before, and their performance of the classic version will be compared with the performance of the robust one.

## Comparison between classical and robust models

The evaluation of the forecasting models will be done, of course, on the test set, with one main difference: the time points of the anomalous observations detected will not be included in the computation of the MAE. For the instances where an abnormal behavior was detected, a good forecast will be the one that actually deviates a lot from the observed abnormal behaviour. A high value of the MAE in these points translates into a prediction curve that stays robust, without being influenced by the extremelly high or low values of the curve of the observed behaviour, while a low value of the MAE for these points will mean that the prediction curve closely follows the actuall anomalous behaviour of the metric.

| Model id | Model | Classic Version | Robust Version |
|---|---|---|---|
| 1 | EWMA | 985.78 | - |
| 2 | HW-add(1440) | 78.42 | 72.43 |
| 3 | HW-mult(1440) | 75.26 | 71.76 |
| 4 | HW-add(10080) | 78.31 | 74.27 |
| 5 | HW-mult(10080) | 94.16 | 91.60 |
| 6 | Taylor DS-add | 71.58 | 68.50 |
| 7 | Taylor DS-mult | 72.65 | 69.33 |

Table 5.3: The mean absolute error (MAE) computed for the classic and robust versions of all forecasting models, for the metric *"Number of pageviews"*. The threshold $5\sigma$ was used for detecting an anomaly.

| Model id | Model | Classic Version | Robust Version |
|---|---|---|---|
| 1 | EWMA | 63.65 | - |
| 2 | HW-add(1440) | 17.75 | 15.61 |
| 3 | HW-mult(1440) | 16.60 | 15.61 |
| 4 | HW-add(10080) | 19.04 | 16.96 |
| 5 | HW-mult(10080) | 17.75 | 16.82 |
| 6 | Taylor DS-add | 15.52 | 15.04 |
| 7 | Taylor DS-mult | 16.13 | 15.68 |

Table 5.4: Comparison of the MAE for the metric *"Median pageload time"* on the test set. For both metrics these performance results were computed without considering the filled-in missing values with an average, nor the instances that anomalous behaviour was detected, in the test set. The dataset was cleanded first 2 distorted last 2, $\sigma = 5$

From tables 5.3 and 5.4 it is obvious that the robust forms of all the models perform better than the classic versions. Even though the difference is not significant, when evaluating in a bigger test set the improvement can be more obvious. In this point we would like to underline and stress the importance of using these robust forms of the predictive models, which result in the imporvement of the forecasts, something that is closely related also to the ability of the anomaly detector to correctly identify anomalies, and avoiding False Positives.

The importance of using the robust versions of the predictive models is more clearly seen when using Taylor's model with a different initialization method. Two different initialization methods were described in chapter 3 about Taylor's model: the one, method (3.11), was an arbitrary method setting the starting values of the components equal to zero or one, while method (3.12) used simple averages for producing these starting values. From tables 5.5 and 5.6 we see that, even though the improvement in the forecast accuracy for initialization method (3.11) is not significant, the same does not hold for initialization method (3.12). For the metric *number of pageviews*, the robust version improves the forecast accuracy by 12% and 15% for Taylor's additive and multiplicative model correspondingly, compared to the classic version.

| **Initialization method** | **Classic Version** | **Robust Version** |
|---|---|---|
| arbitrary: (3.11) | 71.58 | 68.50 |
| averaging: (3.12) | 81.81 | 71.39 |

Table 5.5: The MAE computed on the test set, for *number of pageviews* for the two different initialization methods of *Taylor's DS additive model.*

| **Initialization method** | **Classic Version** | **Robust Version** |
|---|---|---|
| arbitrary: (3.11) | 72.65 | 69.33 |
| averaging: (3.12) | 87.33 | 73.93 |

Table 5.6: The MAE computed on the test set, for *number of pageviews* for the two different initialization methods of *Taylor's DS multiplicative model.*

According to Taylor [2010b], the success of various optimization algorithms depends primarily on the choice of the initial values for the smoothing parameters. From their side, the optimal smoothing parameters depend to a high extend on the quality of the data and on the initialization of the forecast models. For the two different initialization methods, 2 different quadruples of parameters were found, which are presented on the appendix in the end of the documentation. Furthermore, since initialization is influenced by the quality of the data, we manually replaced the outliers of the first two weeks with a more "normal" value, namely the average of the 4 neighbouring observations.

The reason that initialization method (3.11) yields more accurate results than (3.12) is worth to investigate. It is reasonable to expect

We summarize this important conclusion, that the following are interelated and influenced by each other, regarding the class of the Exponential Smoothing forecasting techniques:

1. the initialization method,
2. the quality of the data,
3. the optimal smoothing parameters,
4. the accuracy of the forecasts.

## Computational speed in real time anomaly detection

We need to know the *mean* and *variance* of the forecast errors at each minute:

The mean of the errors will be

$$\mu = \frac{\sum_{i=1}^{n} e_i}{n}$$

The variance will be

$$\begin{aligned}
\sigma^2 &= \frac{\sum_{i=1}^{n}(e_i - \mu)^2}{n} \\
&= \frac{\sum_{i=1}^{n}(e_i^2 - 2 \cdot \mu \cdot e_i + \mu^2)}{n} \\
&= \frac{\sum_{i=1}^{n}(e_i^2 - 2 \cdot \mu \cdot e_i + \mu^2)}{n} \\
&= \frac{\sum_{i=1}^{n} e_i^2 - 2\mu \cdot \sum_{i=1}^{n} e_i + \sum_{i=1}^{n} \mu^2}{n} \\
&= \frac{\sum_{i=1}^{n} e_i^2}{n} - 2\mu \cdot \frac{\sum_{i=1}^{n} e_i}{n} + \frac{n \cdot \mu^2}{n} \\
&= \frac{\sum_{i=1}^{n} e_i^2}{n} - 2\mu^2 + \mu^2
\end{aligned}$$

which gives the following alternative expression for the variance:

$$\sigma^2 = \frac{\sum_{i=1}^{n} e_i^2}{n} - \mu^2$$

Every one minute, when a new observation becomes available, the values of the mean and variance of the forecast errors need to be recomputed. This process can be time consuming if every time all previous observations are considered for these computations, especially when the number of observations grows large. We choose instead to use the alternative formula for the variance, presented above, and store only three values, those that are needed for the computation of the mean and variance, namely the *sum of the errors*, the *sum of the squared errors* and the *number of total observations*, we achieve to speed up the process, as describd in the code below.

The whole process of detecting an anomaly in real time can then be summarized in the following six steps:

1. Forecast for the next minute is made, denote it by $\hat{y}_{t+1}$.

2. After one minute has passed, the actual value, $y_{t+1}$, is made available. The forecast error is thus computed:

$$e_{t+1} = y_{t+1} - \hat{y}_{t+1}$$

3. Evaluation of whether the forecast error $e_{t+1}$ - and thus the actual value $y_{t+1}$ - is considered an anomaly is made. This is done by computing how many standard deviations away from the mean the forecast error $e_{t+1}$ lies. If it exceeds the threshold, then it is labelled as an anomalous observation.

4. The replacement of the ouliers should be done real-time, when the forecasts are produced: Instead of using this anomalous value for computing the level, trend and seasonal components, a smoother value is chosen, mainly the one lying exactly on the threshold value.

5. The new mean and variance are computed using the above formulas. The code for this step will look like this:

```
# Python code:
sum_of_errors = sum_of_errors + e
sum_of_squared_errors = sum_of_squared_errors + e^2
number_of_observations = number_of_observations + 1
# Using the result of (1):
mean = sum_of_errors / number_of_observations
# Using the result of (2):
variance = (sum_of_squared_errors / number_of_observations) - mean^2
```

6. Repeat the process from step 1 to step 6 for the next minute.

# CHAPTER 6

## Conclusion

The purpose of this thesis is to build an anomaly detection tool that will identify potential downfalls and malfunctions on websites in real time. The problem can be broken down into three main parts. The first one is the construction of an appropriate model capable of being trained on the normal behaviour of our metrics. Due to the multiple seasonal patterns of our metrics, the class of exponential smoothing forecasting models, being capable of dealing with seasonal time series, was chosen, also due to its simplicity, transparency and lower computational demand. From this class, Taylor's model accounting for both daily and weekly seasonality patterns was found to outperform Holt-Winters accounting for either daily or weekly seasonal effects, in terms of forecasting accuracy.

Secondly, by knowing this normal behaviour, a measure of deviation of each instance from this standard behavioural pattern is defined. Due to the numerical character of time series, the choice of the forecast error is made. Finally, by observing that these errors are normally distributed around 0 with an unknown standard deviation of $\sigma$, the anomaly detection model is built to flag as anomalies those observations for which their forecast errors were found to be unexpectedly high or low.

Regarding the true nature of the outliers of our dataset, and if they can be defined as actual anomalies or not, there is not enough information about it. Therefore, the evaluation was conducted by manually setting some outliers as exemplar anomalies and then by choosing which sigma achieved the most correct detections. According to our observations and subjective views on anomalies, for each of the two metrics, the $\sigma$'s that delivered the most balanced results while managing to limit wrong detections, were highlighted.

By lowering the limitations of the threshold and thus detecting all possible anomalies, the tool will detect a range of false negatives. In this way, the amount of supposedly detected anomalies will create doubts about the efficiency of the tool. Conversely, choosing to raise the threshold could result in decreasing the quantity of detected anomalies but also preventing the tool from identifying true positives.

Judging from the aforementioned choices regarding the detection tool, the clients and their needs are the ones that will decide if they prefer capturing all the anomalies or just the most important ones. Furthermore, it is them who are to determine how many consecutive anomalies should lead to the raise an alarm. To conclude, despite the building process of an anomaly detector, there will always be other questions to be viewed and answered from the perspective of the clients. A modeler-client relationship is reciprocal and thus exchanging views and insights can lead to optimization of the offered services.

## 6.1 Further work

The further research can be centered around the maintenance and functionality of the anomaly detector in real time in the future. However, these possible improvements do not concern only anomaly detection related issues but also research ideas that will improve MeasureWork's scope as a company.

### 1. Assuming the existence of common sub-cycles of weekdays

Regarding *Holt-Winters method* accounting for seasonality of length $m$, the seasonal components $s_1, s_2, ..., s_m$ are updated only once during one seasonal cycle, i.e. once every $m$ time periods. Similarly, in *Taylor's model* considering daily and weekly seasonality patterns with seasonality cycles of length $m_1$ and $m_2$, the daily seasonal components $D_1, D_2, ..., D_{m_1}$ are updated only once every day while the weekly seasonal components $W_1, W_2, ..., W_{m_2}$ are updated only once per week. The models consequently fail to consider the seasonal changes between one day and the next, since the seasonal variation of a time instant for a specific day of the week is computed according to the same time instant of the same week day of the previous week. Any relevant information between two consecutive days of the week is not taken into account.

Furthermore, Taylor's double seasonal model would require 1 initial value for the level, one for the trend, $m_1$ for the daily seasonal and $m_2$ for the weekly seasonal components, meaning that the model would need estimates of $m_1 + m_2 + 2$ values before it starts operating. However, recall from the exploratory data analysis of Chapter 2, our finding regarding the metric *Number of pageviews* that its behaviour follows the same intra-day cycle from Mondays to Thursdays, and Fridays, Saturdays, and Sundays all form a distinct category in terms of their intra-day behaviour.

In their paper, Gould et al. describe a method of taking advantage of the common subcycles and thus reducing the number of seed values the models require. In our case, instead of assuming 7 distinct subcycles, each one for a different day of the week, we could instead use 4 subcycles, reducing the number of seed values for the seasonal terms needed from $m_1 + m_2 = 1440 + 7 \cdot 1440 = 11520$ to $m_1 + m_2 = 1440 + 4 \cdot 1440 = 7200$, by reducing the number of daily cycles from $k = 7$ to $k = 4$.

This model for multiple seasonal processes (MS) as it is named by Gould et al., also allows for each day to have its own hourly pattern or to have some days with the same pattern. In addition to that, this model allows days with different patterns to affect one another, overcoming the weakness of Taylor's double seasonal model, which assumes the same intra-day cycle for all days of the week. By considering different daily cycles for each seperate day of the week - or categorizing them in groups according to behavioural similarity - and by treating the public holidays or special days as Sundays Gould et al. succeed on reducing the mean squarred forecast error by 15% compared to the Holt-Winters with either daily or weekly seasonality or to Taylor's double seasonal model. Same practice is also followed by Taylor [2010a] with his *Intra day exponential smoothing method* or *IC exp smoothing* as he names it.

## 2. Recomputation of the smoothing parameters

According to Taylor [2010b], the success of various optimization algorithms depends primarily on the choice of the initial values for the smoothing parameters. From their side, the optimal smoothing parameters depend to a high extend on the quality of the data and on the initialization of the forecast models. A further improvement in the maintenance of the forecasting models would be to use an adaptive Holt-Winters or Taylor's model following the suggestion of Kalekar [2004]. This mainly means that the parameters $\alpha, \beta$ and $\gamma$ would be recomputed after some period of time. This way the model adapts itself to the changes. The newly computed parameters may be computed either using the $k$ most recent data, either all the available data from the start till the current point of time. In the latter case all past data need to have been stored.

## 3. Holiday effects

Holiday effects are responsible for changes in the behaviour of the data. From the plots of Chapter 2 we have seen that the national Dutch holidays have a different daily pattern, without this declination from the expected pattern to mean that there is an abnormal behavior. For that reason, in order to avoid false alarms, accounting for special days and holidays is a necessity. In the paper of Souza et al. [2007] a rule to correct the forecasting of electricity demand on holidays or special days was proposed. This method to account for holiday effects by applying exogenous corrections produced a considerable improvement in the forecasting accuracy.

## 4. Different forecast horizons

In the current research we chose to only consider and investigate performance and comparison of the forecasting models for the case where we forecast $h = 1$ minute ahead in the future. It would be interesting to investigate the effect of longer forecast horizons in model selection and model performance, and define the $h$ yielding the most accurate forecasts. Even though, as we explained in previous chapter, for the procedure of anomaly detection there is no reason of forecasting longer than 1 minute ahead, it would be valuable for *MeasureWorks* to present predictions further than one minute ahead in its dashboard.

# Appendices

# Additional graphs

## Number of pageviews

Below are the graphs for each seperate day of the week for the ”number of pageviews” metric:

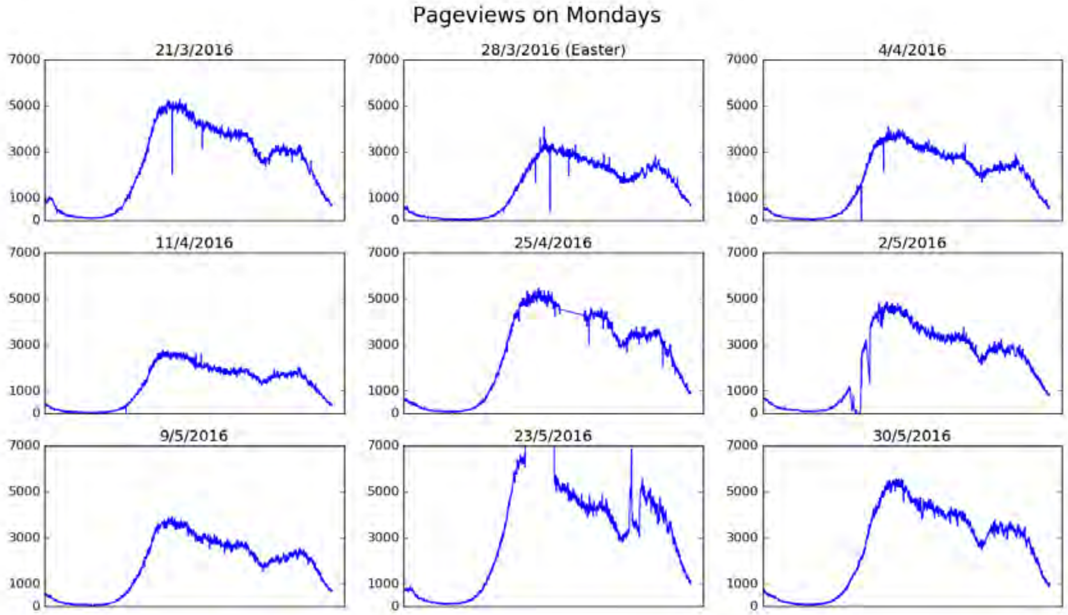


Figure 1: Daily cycle of pageviews for all Mondays

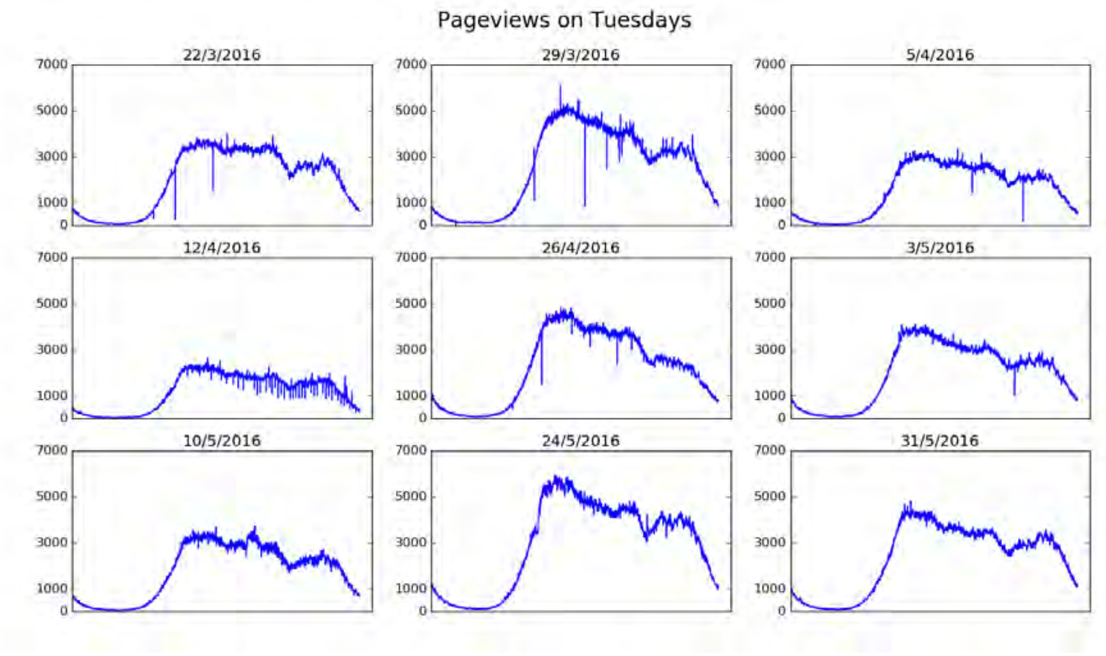


Figure 2: Daily cycle of pageviews for all Tuesdays

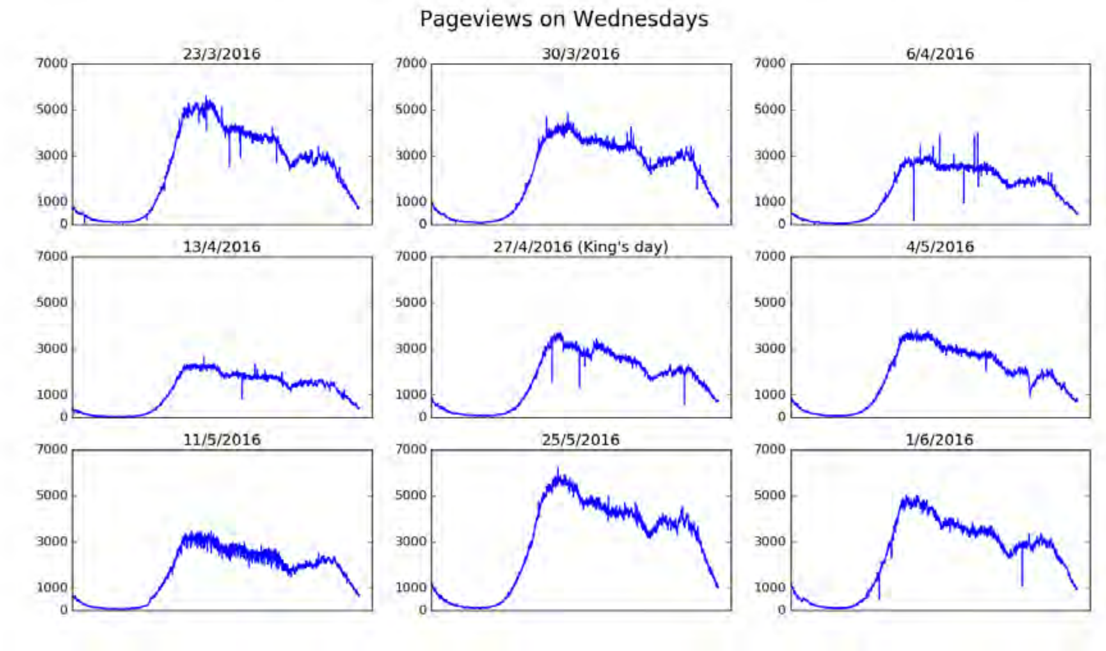


Figure 3: Daily cycle of pageviews for all Wednesdays

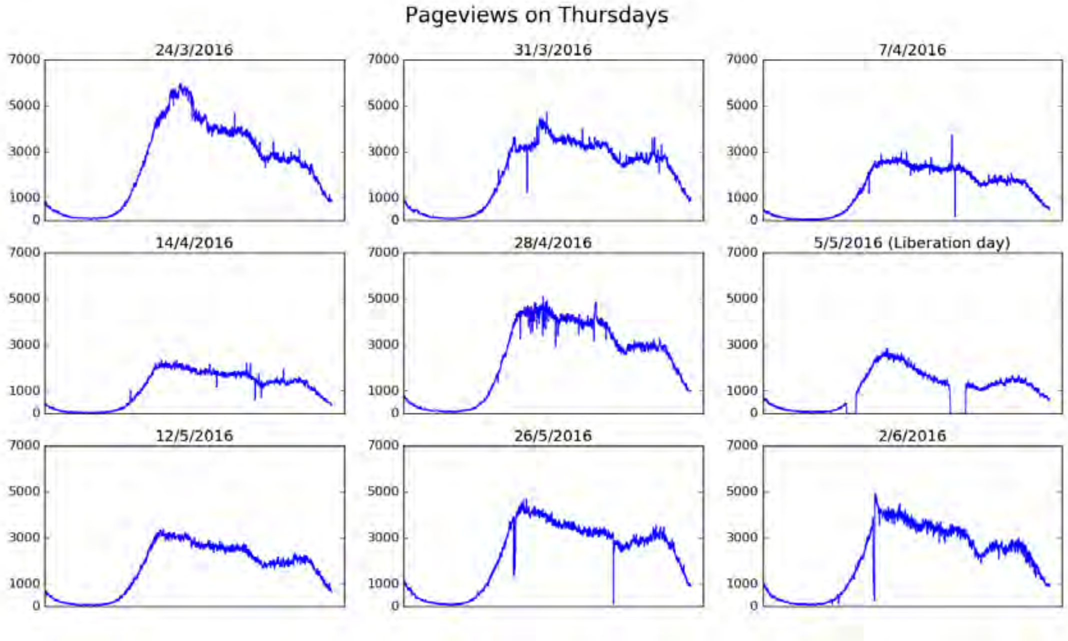


Figure 4: Daily cycle of pageviews for all Thursdays

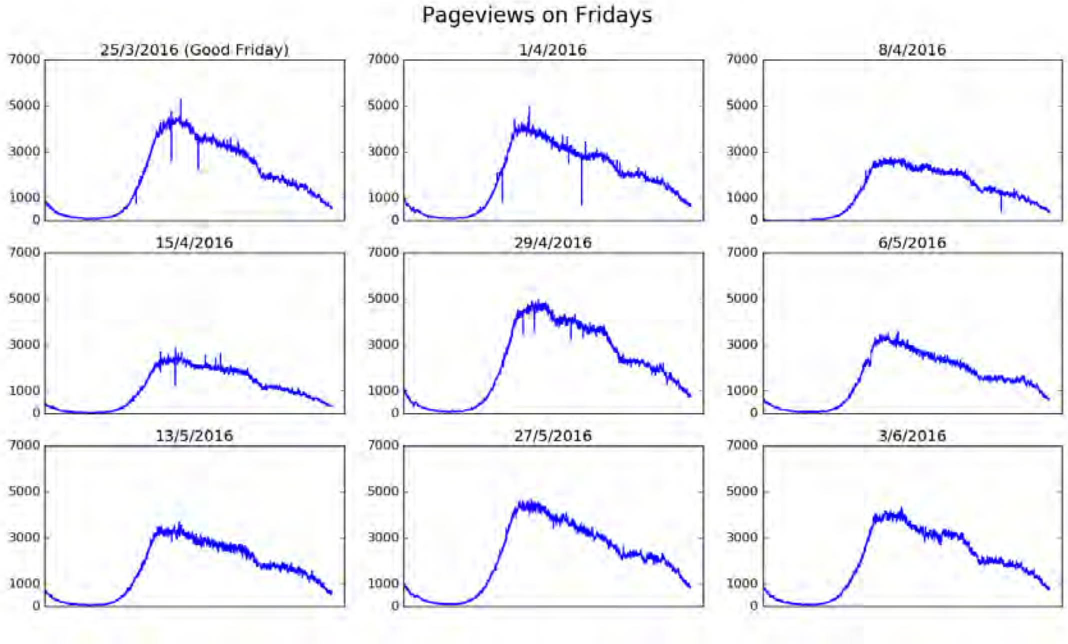


Figure 5: Daily cycle of pageviews for all Fridays

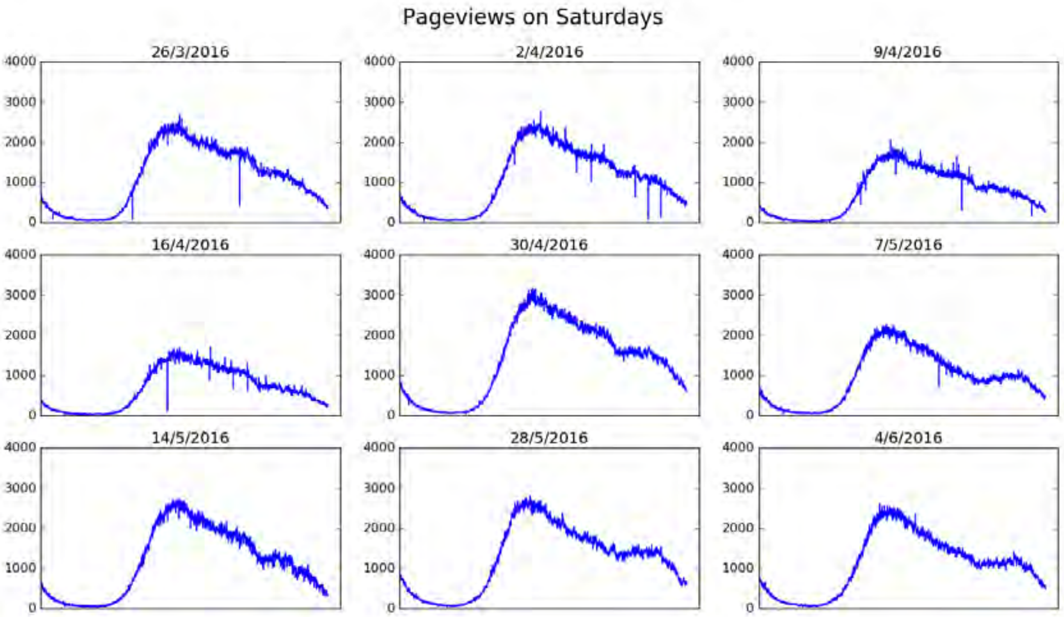


Figure 6: Daily cycle of pageviews for all Sarurdays

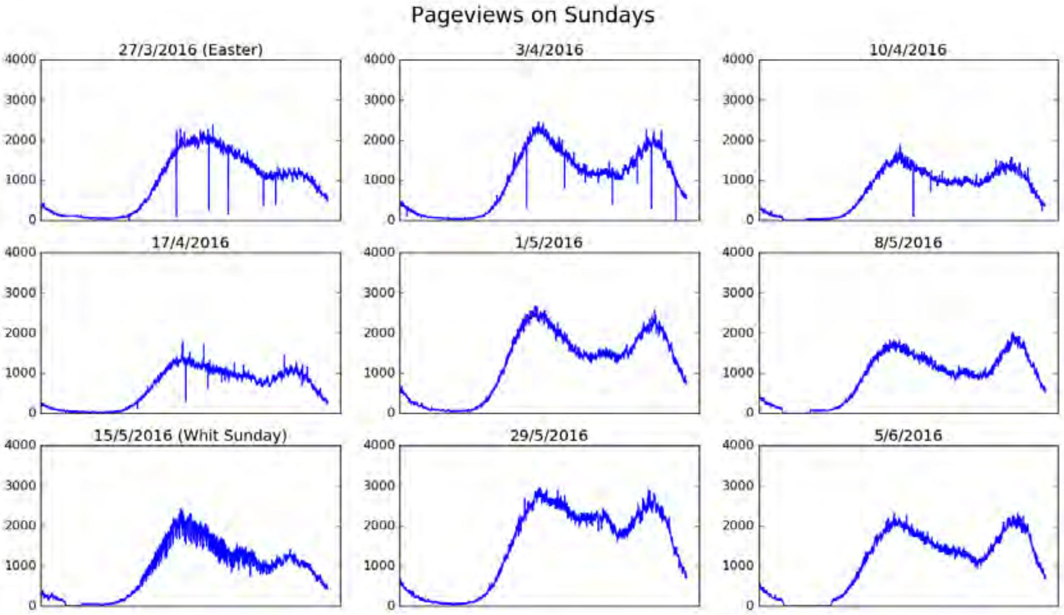
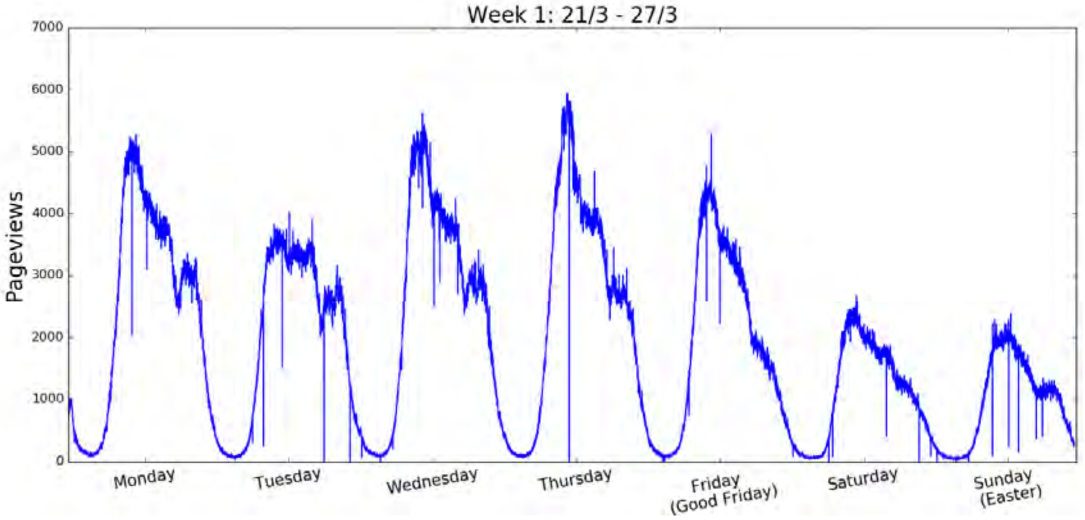
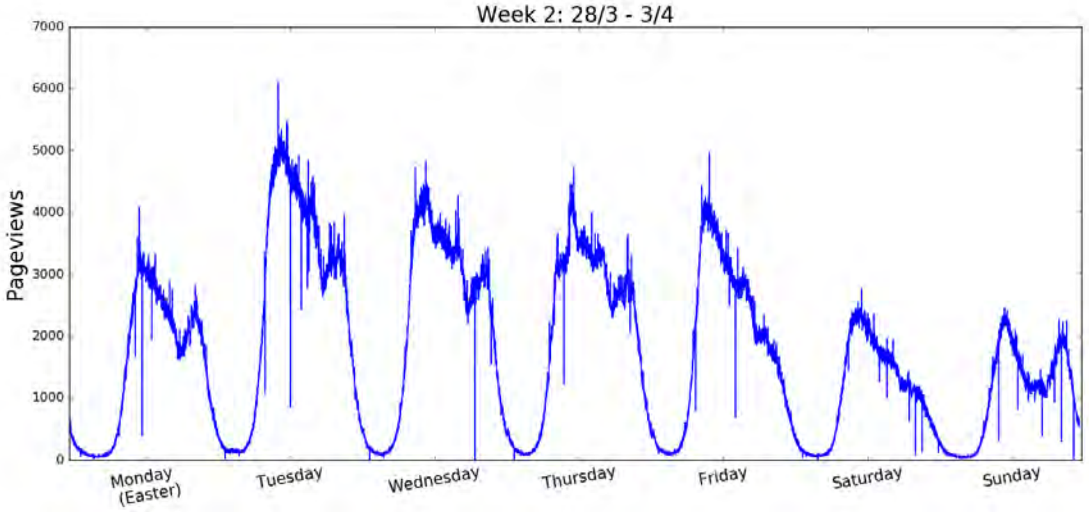
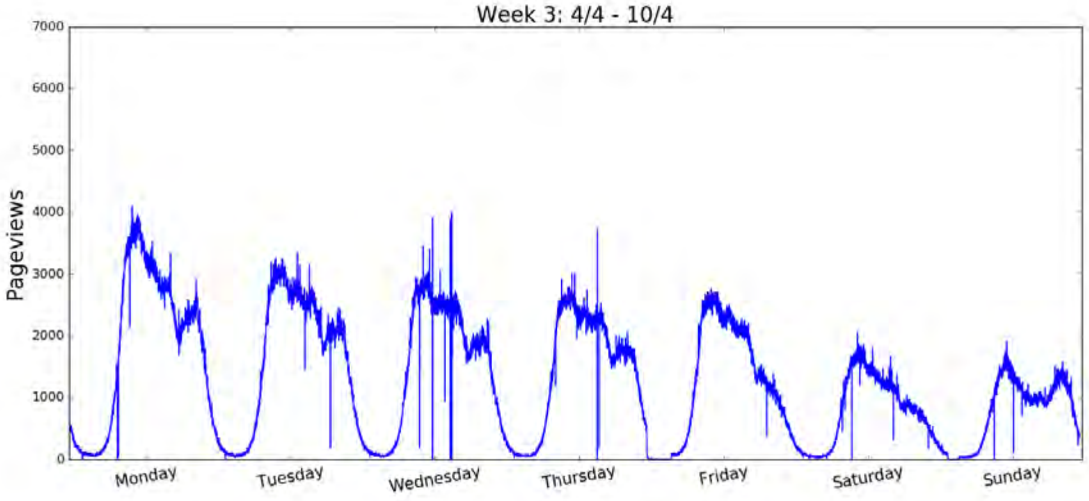
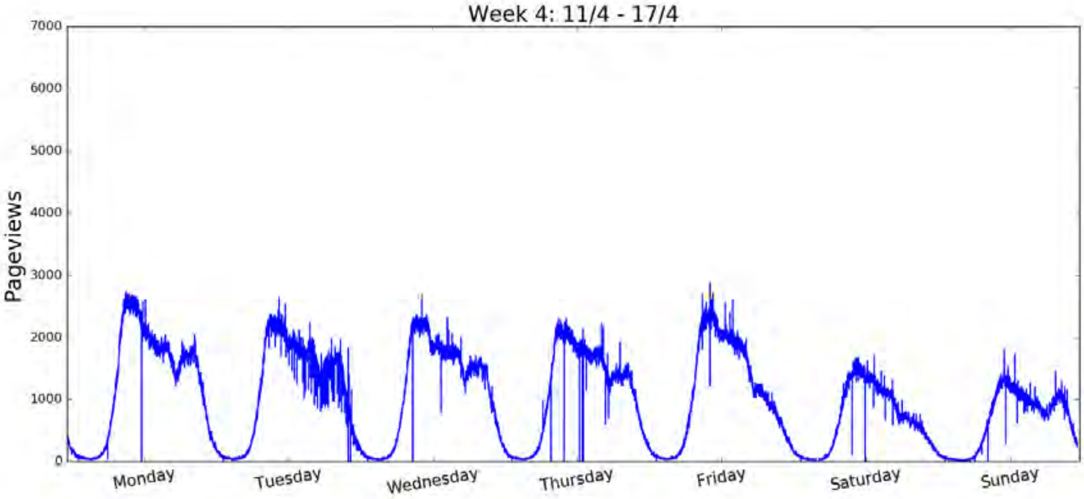


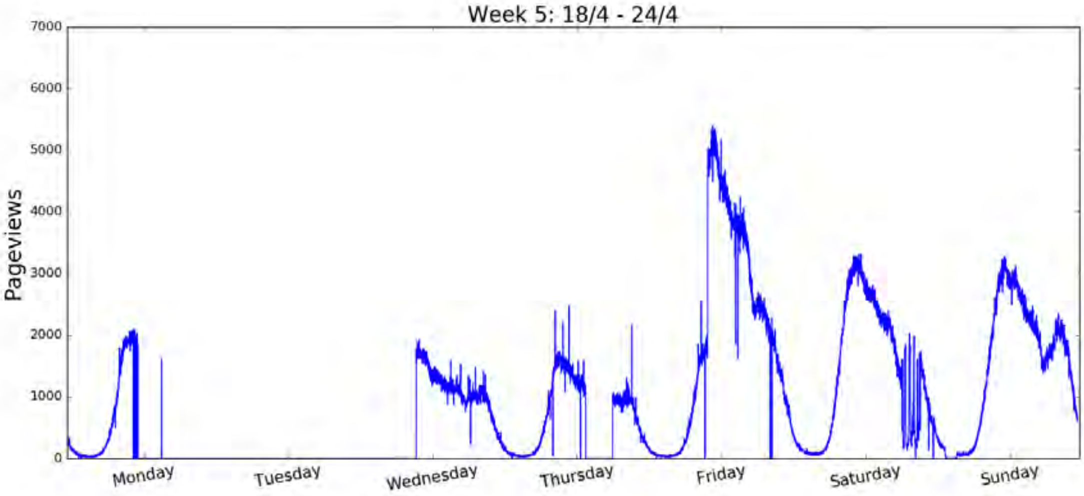
Figure 7: Daily cycle of pageviews for all Sundays

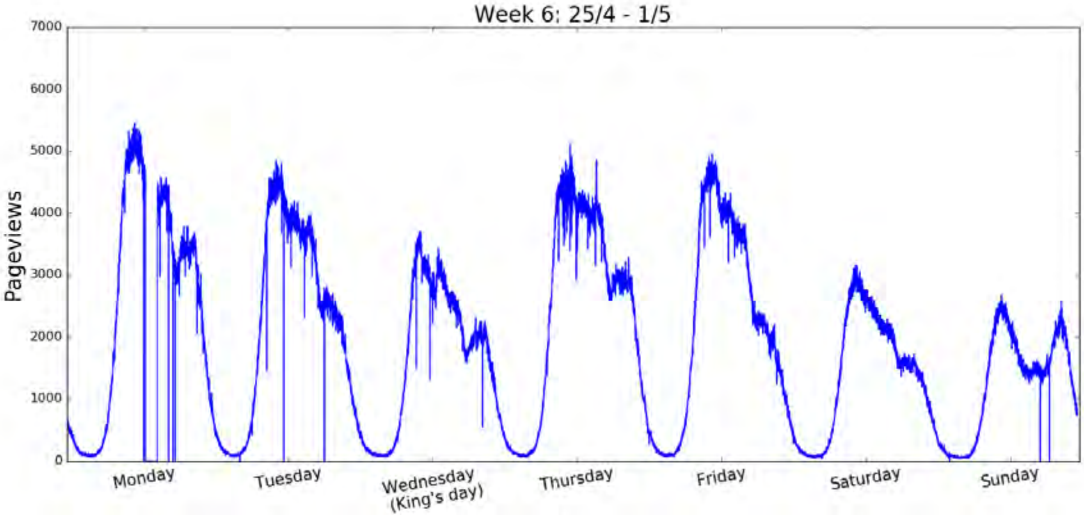

Week 1: 21/3 - 27/3
Pageviews
0
1000
2000
3000
4000
5000
6000
7000
Monday
Tuesday
Wednesday
Thursday
Friday
(Good Friday)
Saturday
Sunday
(Easter)

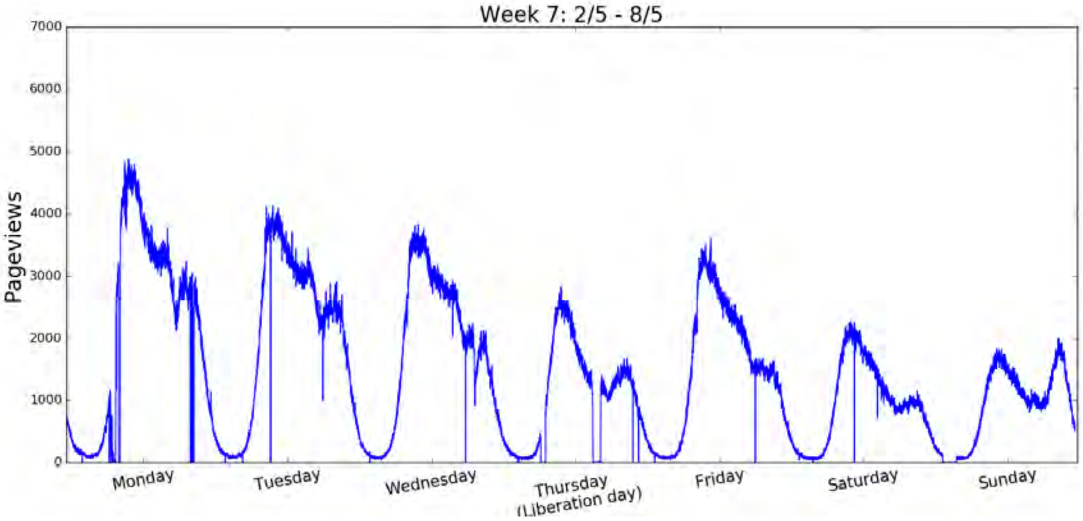

Week 2: 28/3 - 3/4
Pageviews
0
1000
2000
3000
4000
5000
6000
7000
Monday
(Easter)
Tuesday
Wednesday
Thursday
Friday
Saturday
Sunday

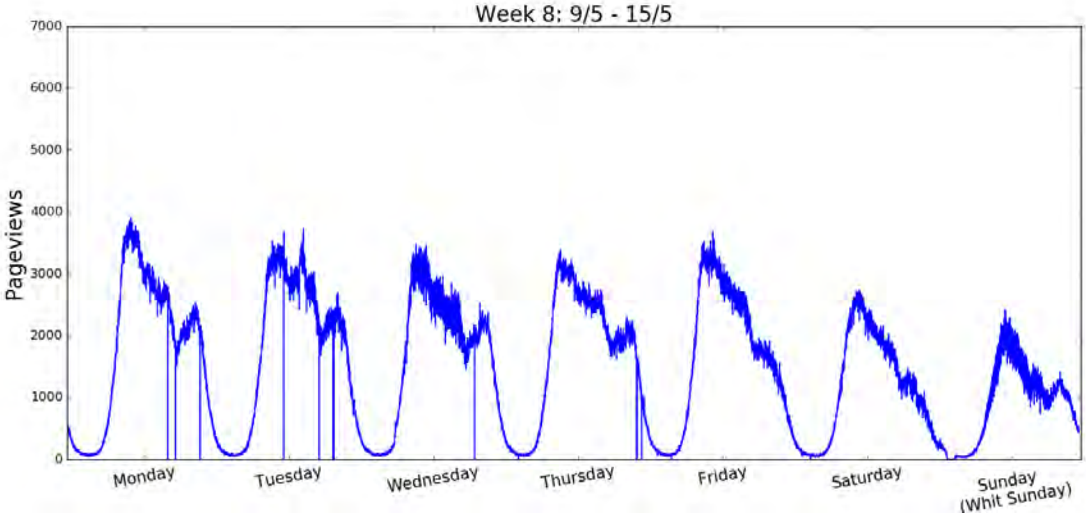

Week 3: 4/4 - 10/4
Pageviews
0
1000
2000
3000
4000
5000
6000
7000
Monday
Tuesday
Wednesday
Thursday
Friday
Saturday
Sunday

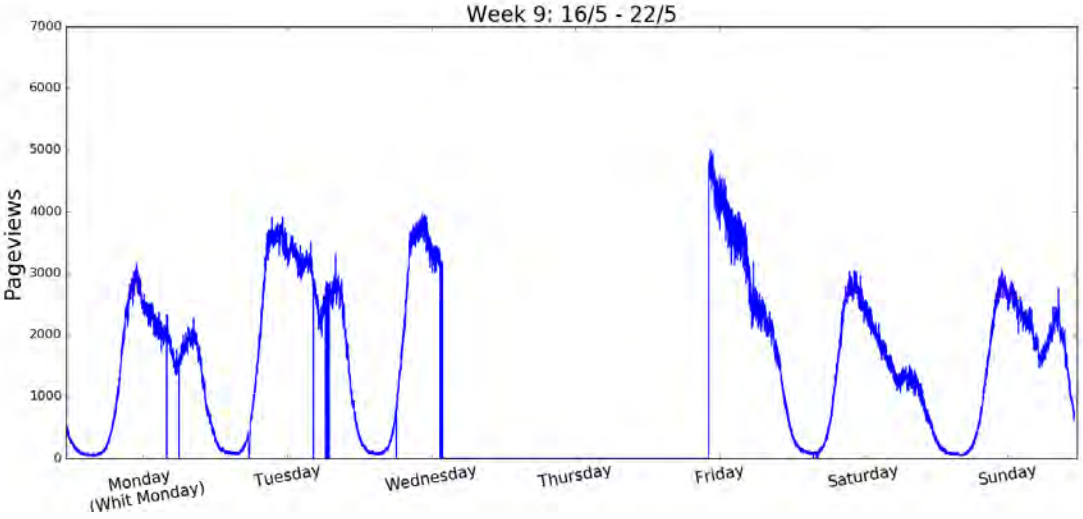

Week 4: 11/4 - 17/4
Pageviews
7000
6000
5000
4000
3000
2000
1000
0
Monday
Tuesday
Wednesday
Thursday
Friday
Saturday
Sunday

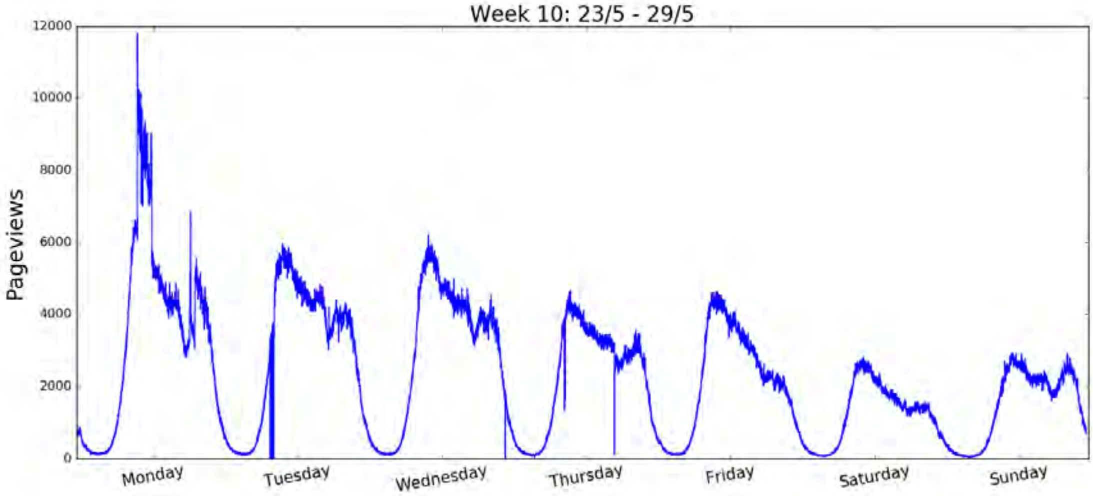

Week 5: 18/4 - 24/4
Pageviews
7000
6000
5000
4000
3000
2000
1000
0
Monday
Tuesday
Wednesday
Thursday
Friday
Saturday
Sunday

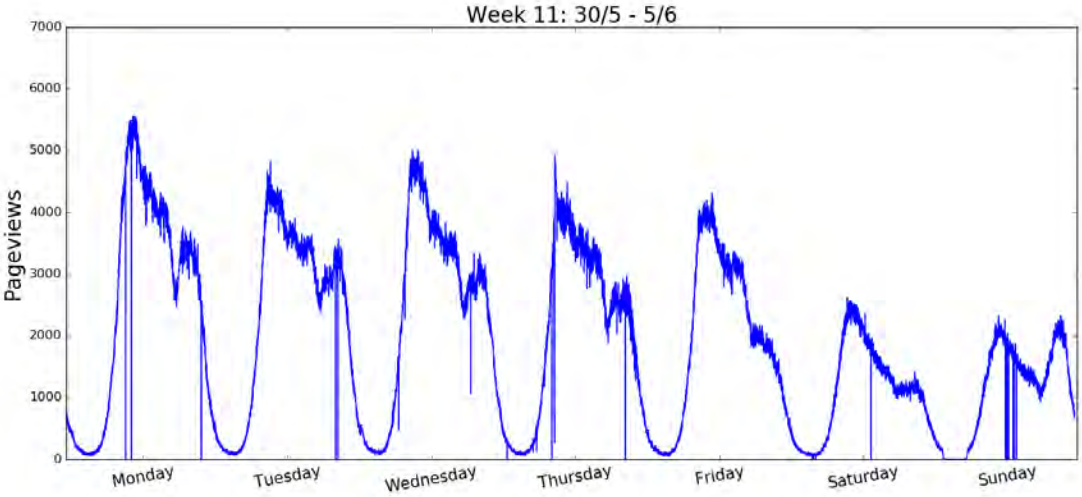

Week 6: 25/4 - 1/5
Pageviews
7000
6000
5000
4000
3000
2000
1000
0
Monday
Tuesday
Wednesday
(King's day)
Thursday
Friday
Saturday
Sunday


Week 7: 2/5 - 8/5
Pageviews
7000
6000
5000
4000
3000
2000
1000
0
Monday
Tuesday
Wednesday
Thursday
(Liberation day)
Friday
Saturday
Sunday


Week 8: 9/5 - 15/5
Pageviews
7000
6000
5000
4000
3000
2000
1000
0
Monday
Tuesday
Wednesday
Thursday
Friday
Saturday
Sunday
(Whit Sunday)


Week 9: 16/5 - 22/5
Pageviews
7000
6000
5000
4000
3000
2000
1000
0
Monday
(Whit Monday)
Tuesday
Wednesday
Thursday
Friday
Saturday
Sunday


Week 10: 23/5 - 29/5
Pageviews
12000
10000
8000
6000
4000
2000
0
Monday
Tuesday
Wednesday
Thursday
Friday
Saturday
Sunday


Week 11: 30/5 - 5/6
Pageviews
7000
6000
5000
4000
3000
2000
1000
0
Monday
Tuesday
Wednesday
Thursday
Friday
Saturday
Sunday

# Median pageload time

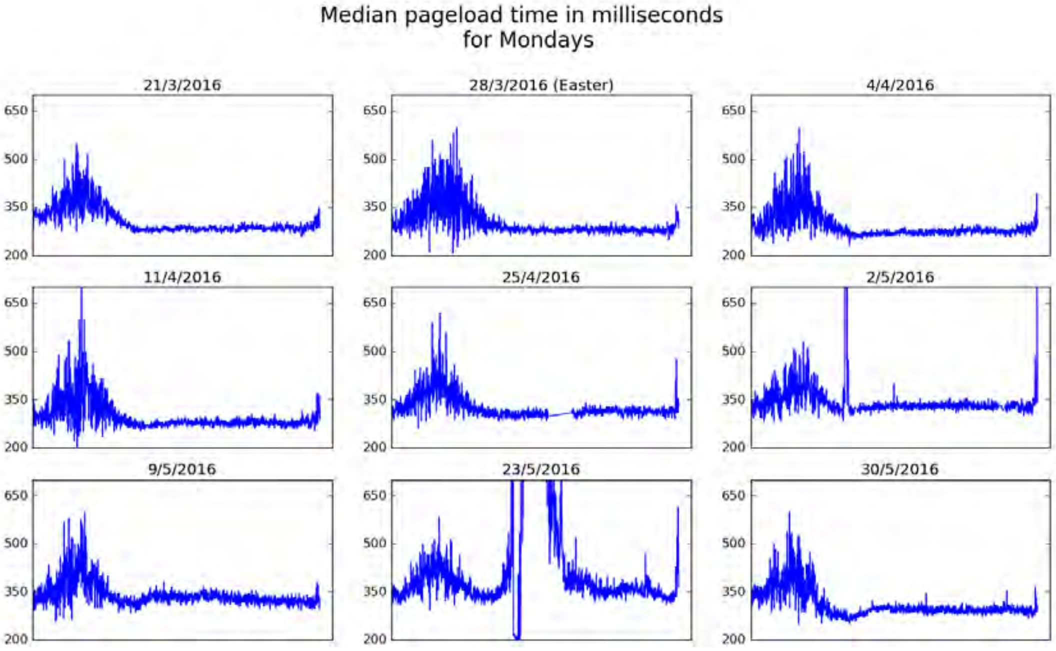


Figure 8: Daily cycle of median pageload time for all Mondays

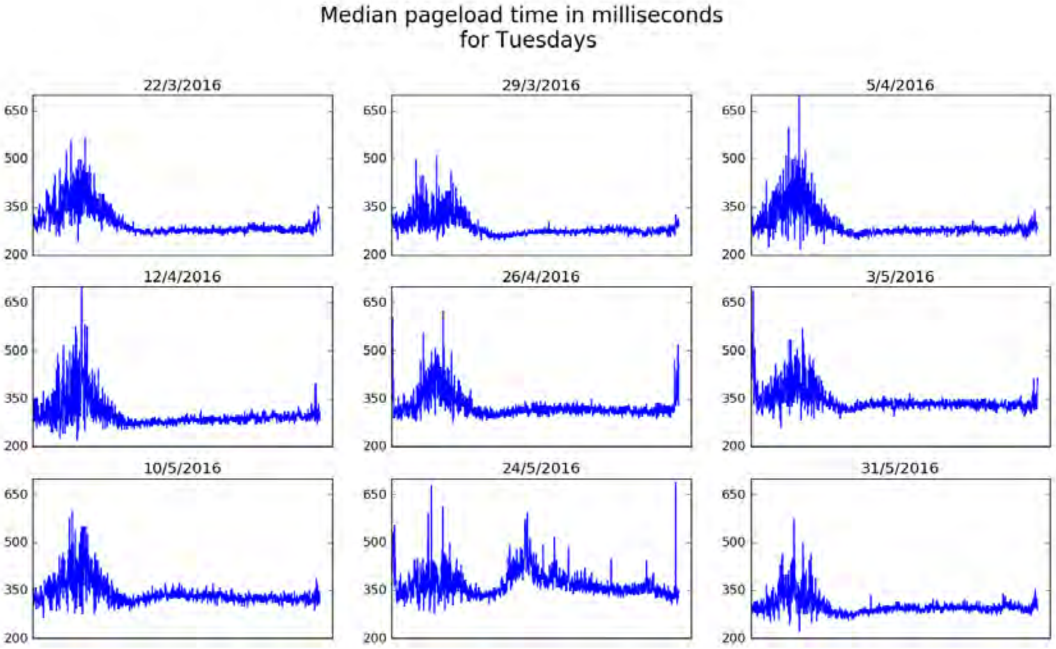


Figure 9: Daily cycle of median pageload time for all Tuesdays

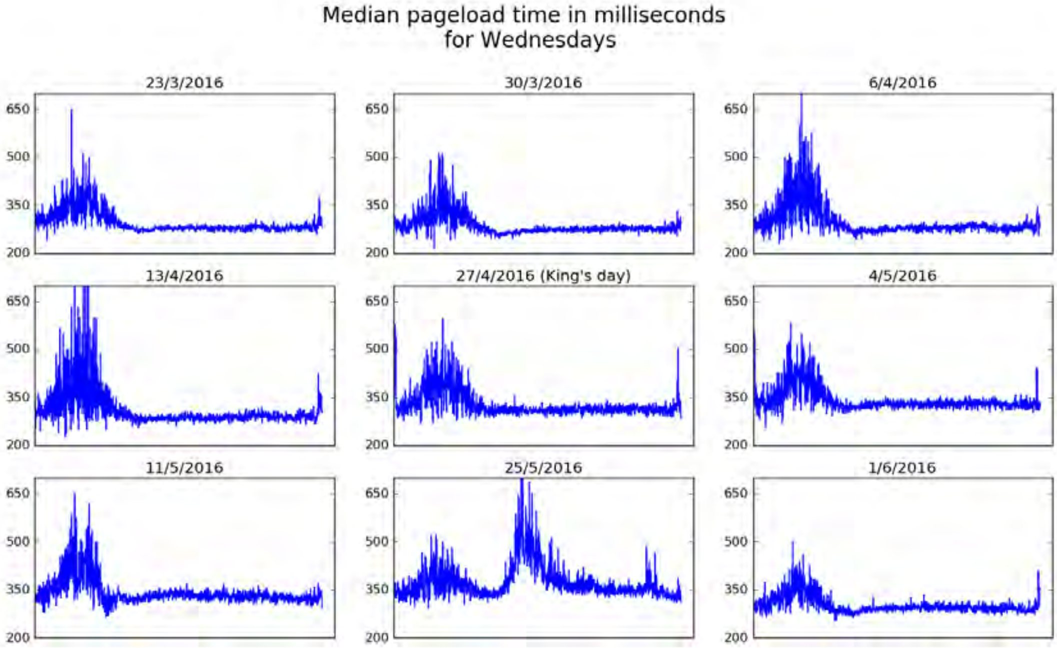


Figure 10: Daily cycle of median pageload time for all Wednesdays

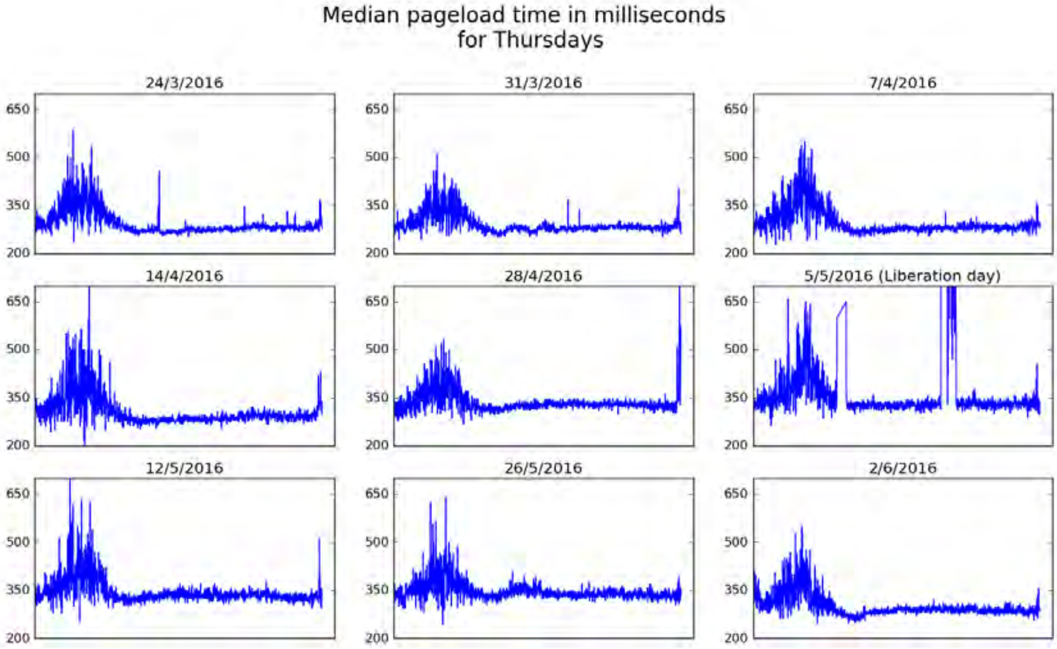


Figure 11: Daily cycle of median pageload time for all Thursdays

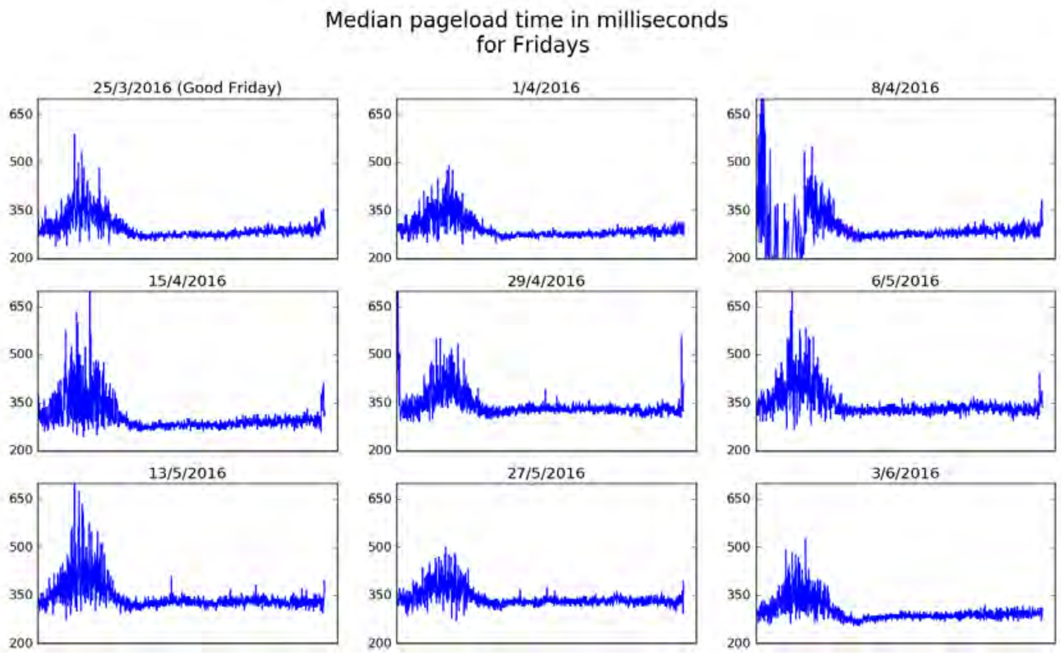


Figure 12: Daily cycle of median pageload time for all Fridays

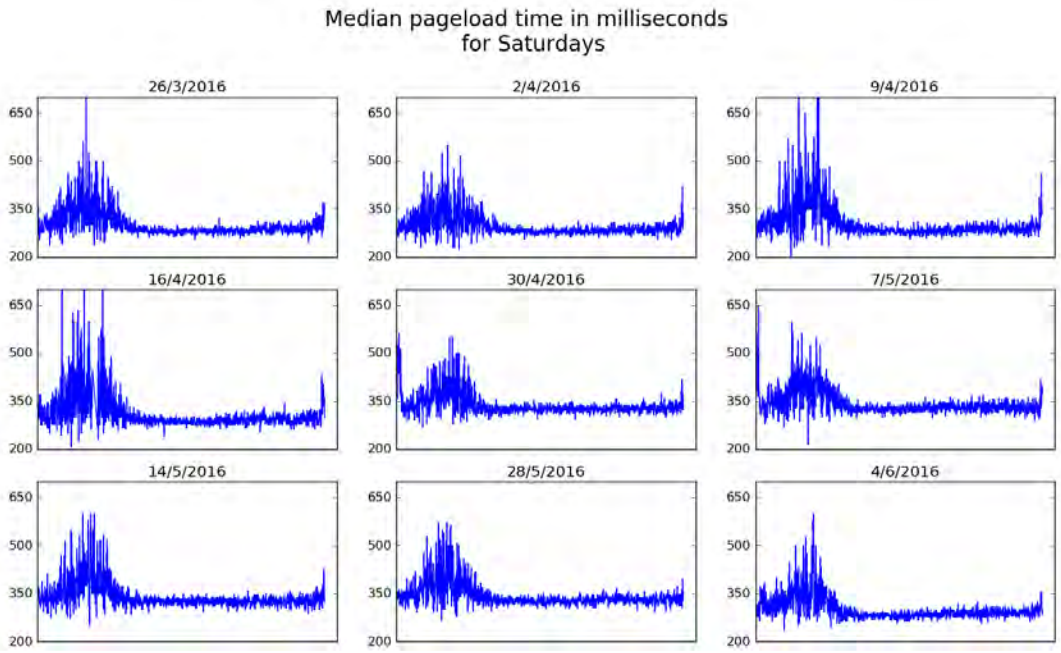


Figure 13: Daily cycle of median pageload time for all Saturdays

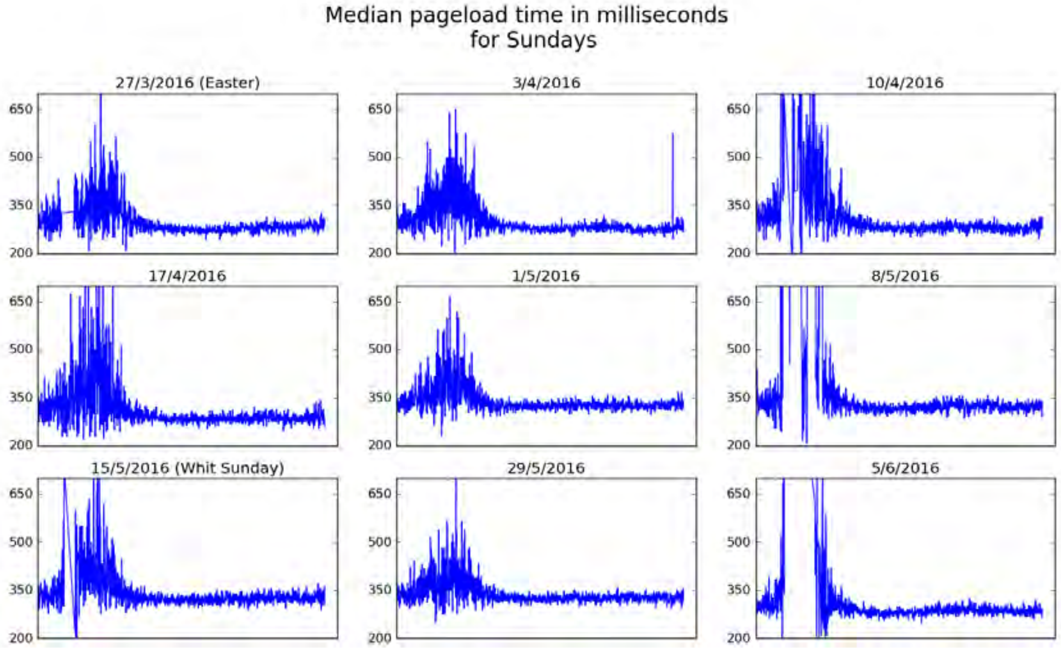


Figure 14: Daily cycle of median pageload time for all Sundays

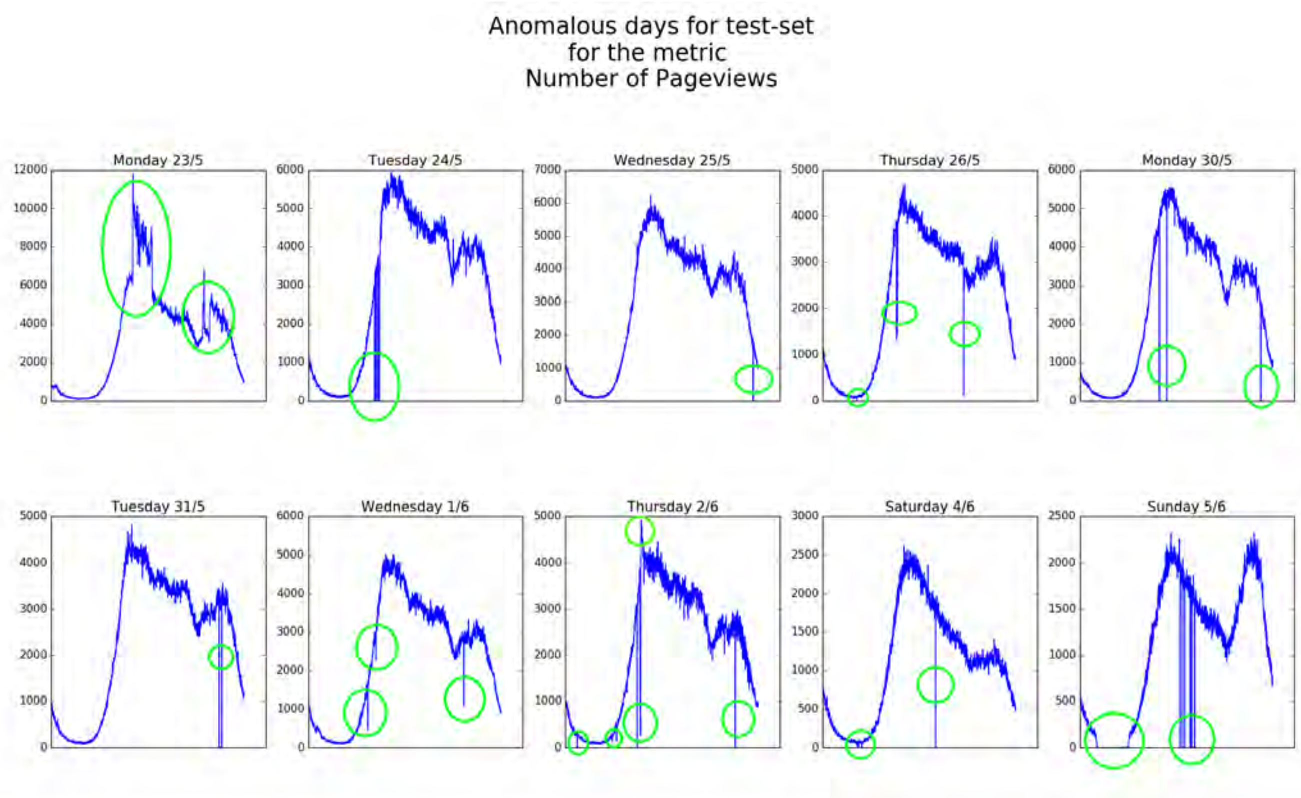
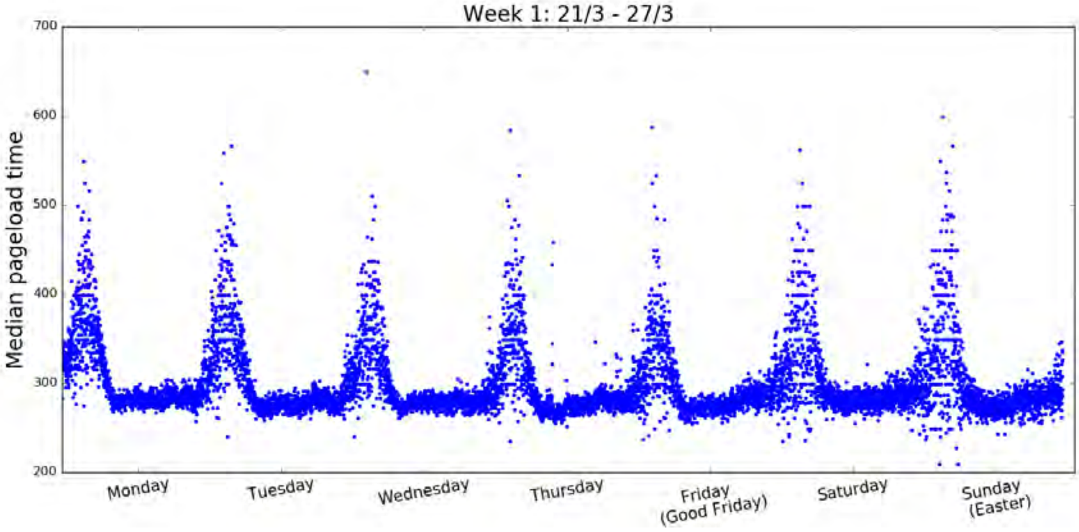
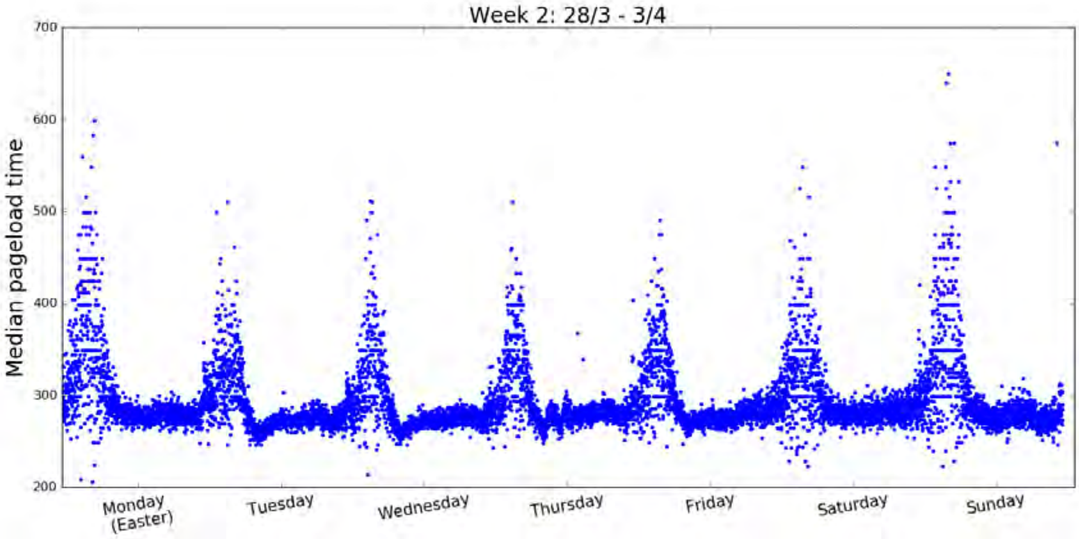
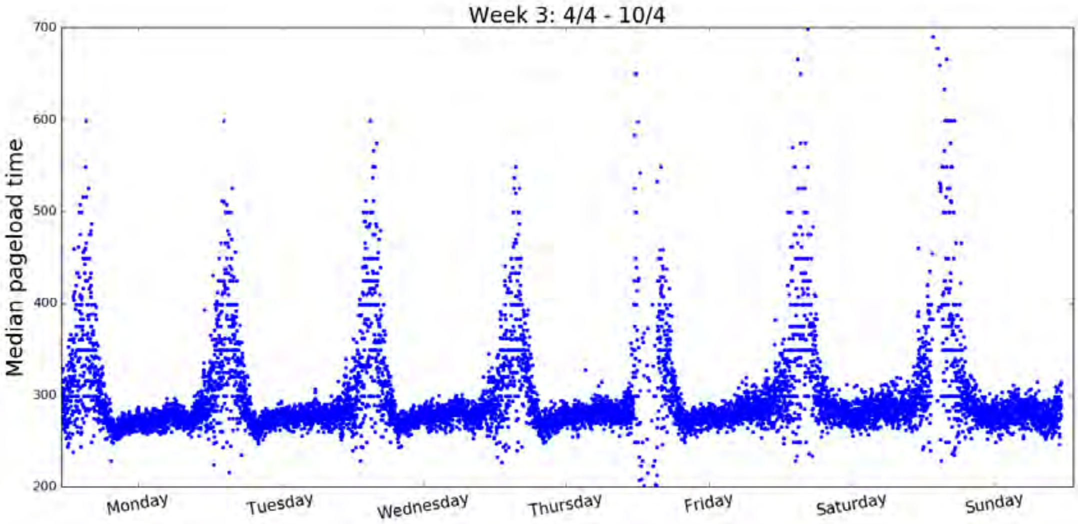
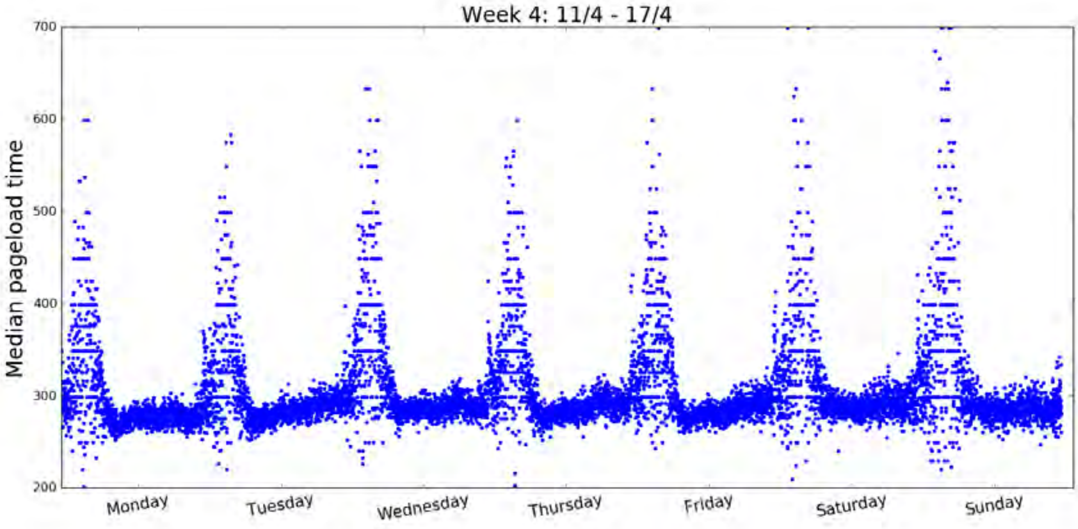


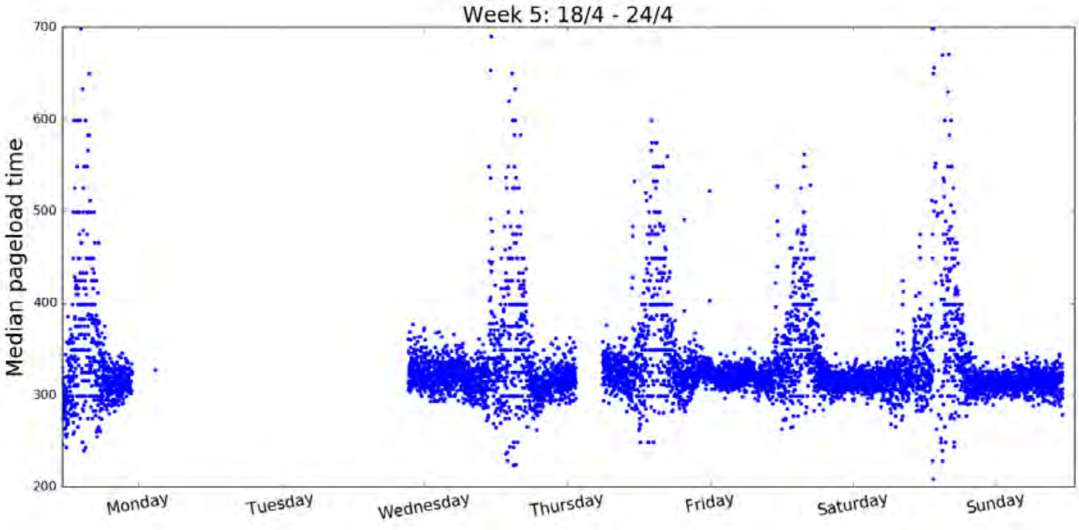
Figure 15: The anomalies we mark on the test set, 22 anomalous ”areas” in total.

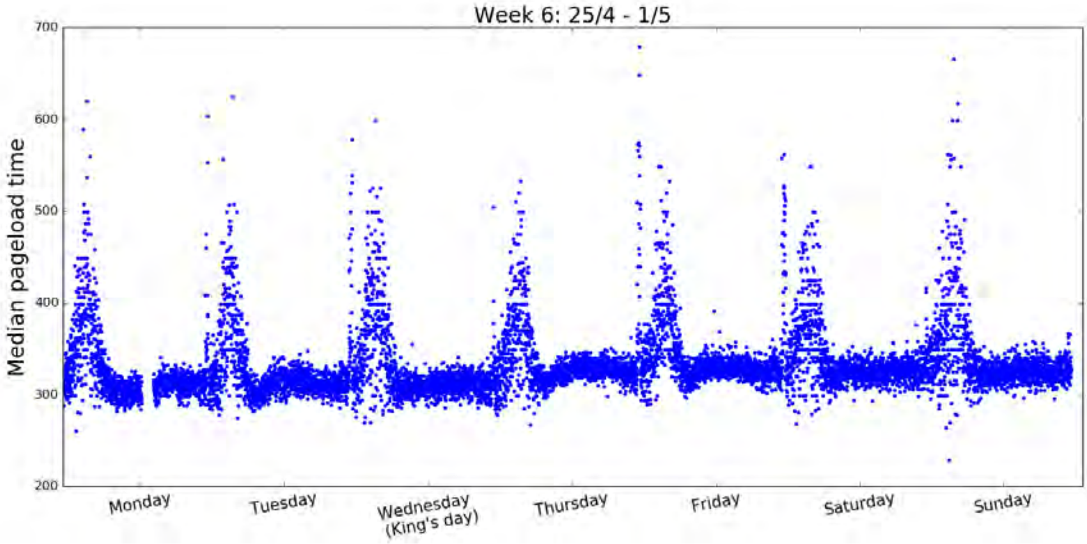

Week 1: 21/3 - 27/3
Median pageload time
700
600
500
400
300
200
Monday
Tuesday
Wednesday
Thursday
Friday
(Good Friday)
Saturday
Sunday
(Easter)

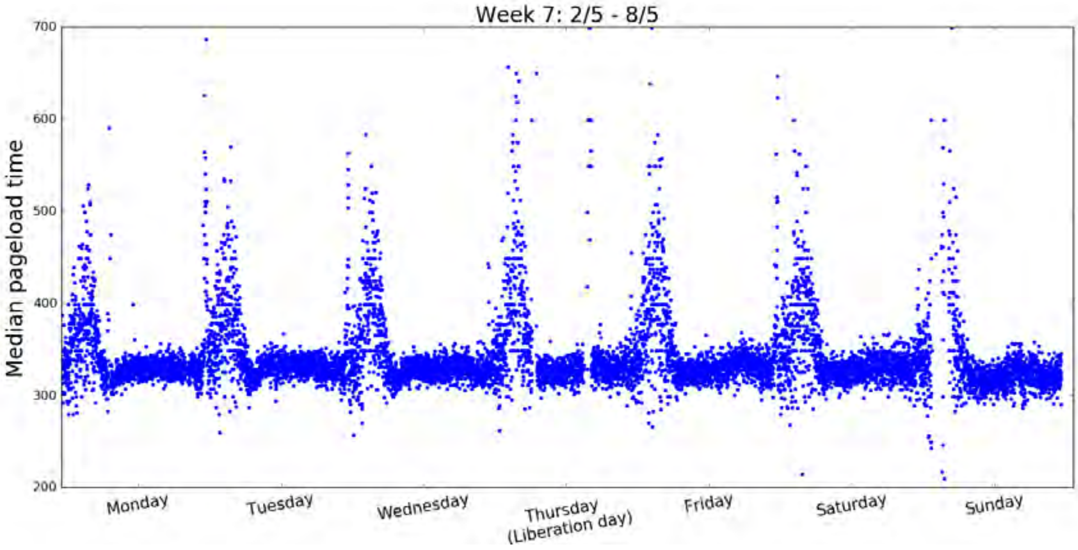

Week 2: 28/3 - 3/4
Median pageload time
700
600
500
400
300
200
Monday
(Easter)
Tuesday
Wednesday
Thursday
Friday
Saturday
Sunday

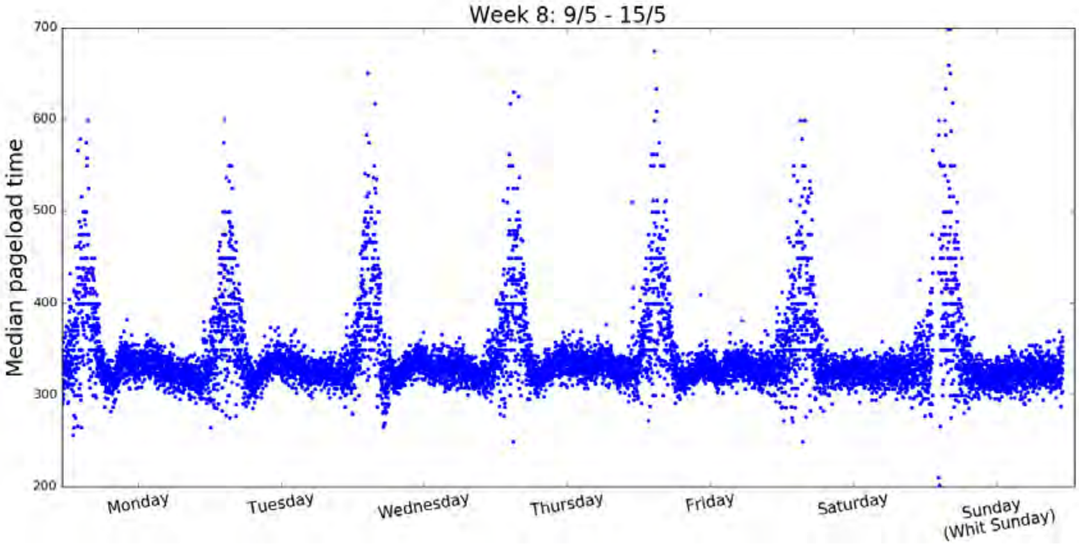

Week 3: 4/4 - 10/4
Median pageload time
700
600
500
400
300
200
Monday
Tuesday
Wednesday
Thursday
Friday
Saturday
Sunday

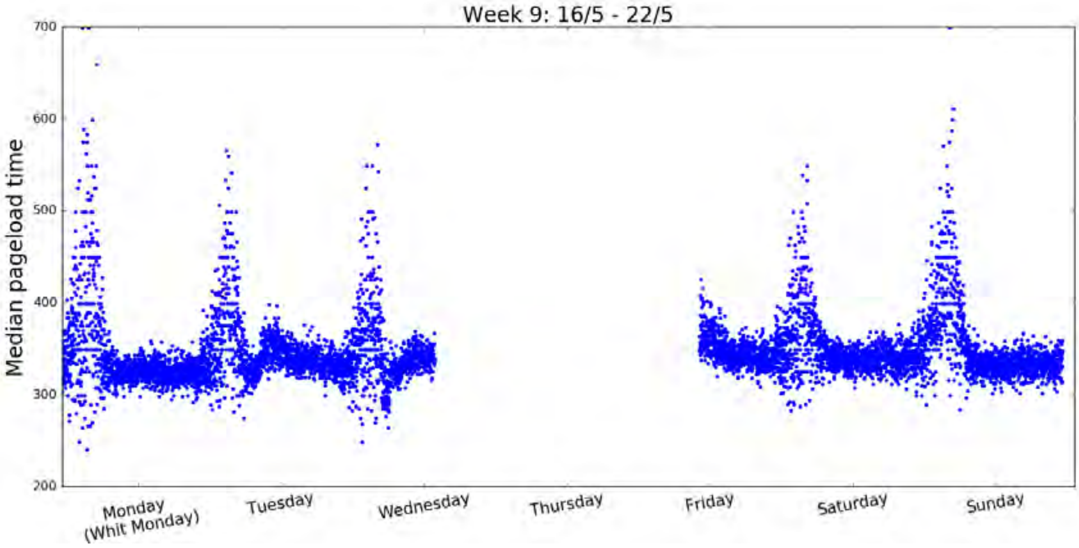

Week 4: 11/4 - 17/4
Median pageload time
700
600
500
400
300
200
Monday
Tuesday
Wednesday
Thursday
Friday
Saturday
Sunday

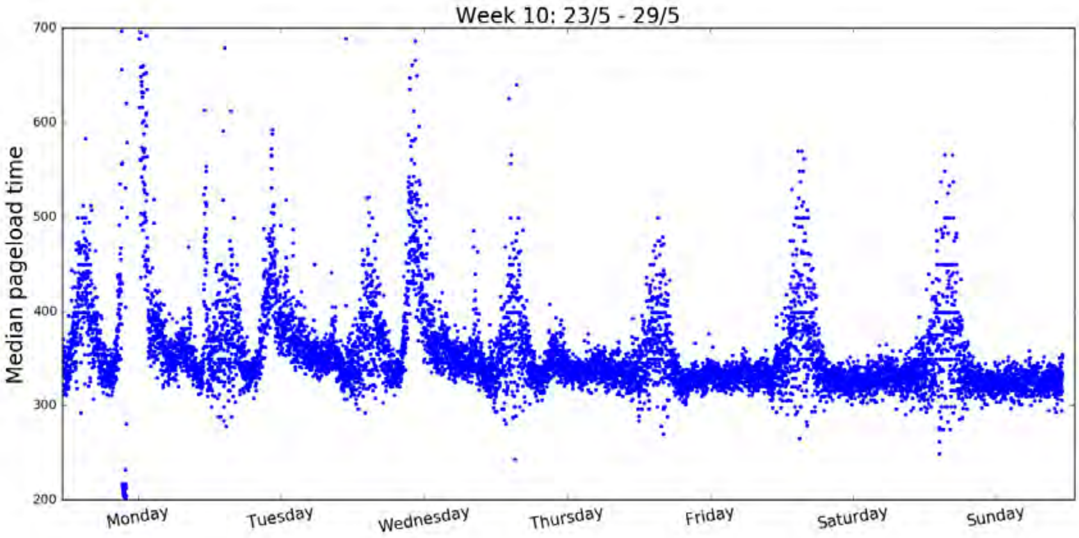

Week 5: 18/4 - 24/4
Median pageload time
700
600
500
400
300
200
Monday
Tuesday
Wednesday
Thursday
Friday
Saturday
Sunday

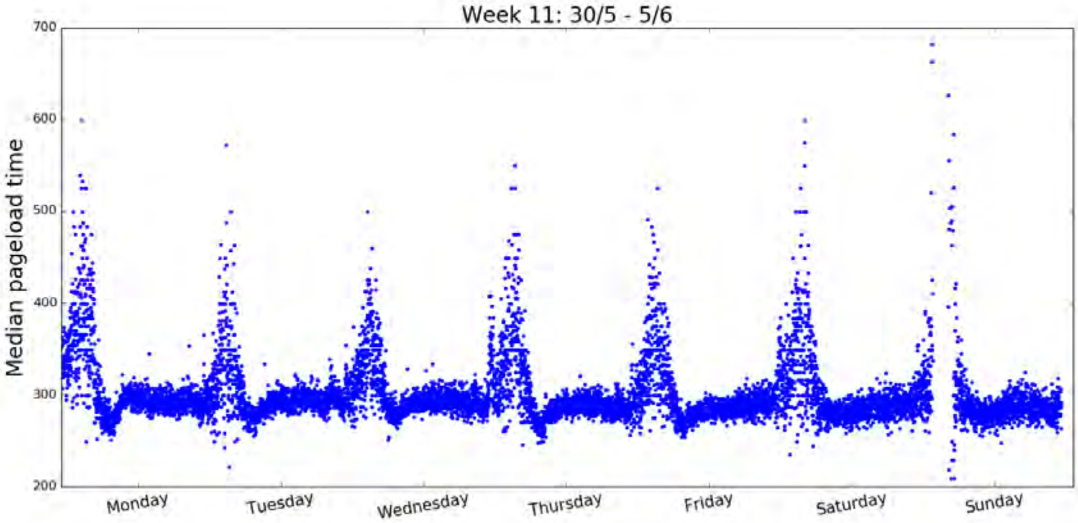

Week 6: 25/4 - 1/5
Median pageload time
700
600
500
400
300
200
Monday
Tuesday
Wednesday
(King's day)
Thursday
Friday
Saturday
Sunday


Week 7: 2/5 - 8/5
Median pageload time
700
600
500
400
300
200
Monday
Tuesday
Wednesday
Thursday
(Liberation day)
Friday
Saturday
Sunday


Week 8: 9/5 - 15/5
Median pageload time
700
600
500
400
300
200
Monday
Tuesday
Wednesday
Thursday
Friday
Saturday
Sunday
(Whit Sunday)


Week 9: 16/5 - 22/5
Median pageload time
700
600
500
400
300
200
Monday
(Whit Monday)
Tuesday
Wednesday
Thursday
Friday
Saturday
Sunday


Week 10: 23/5 - 29/5
Median pageload time
700
600
500
400
300
200
Monday
Tuesday
Wednesday
Thursday
Friday
Saturday
Sunday


Week 11: 30/5 - 5/6
Median pageload time
700
600
500
400
300
200
Monday
Tuesday
Wednesday
Thursday
Friday
Saturday
Sunday

# Tables for the smoothing parameters

Here we present analytically all the optimal smoothing parameters found for all the models and different cases:

| **Holt-Winters:** *”Number of pageviews”* | | | |
|---|---|---|---|
| Daily | Additive | MAE train = 67.208 | $\alpha = 0.310344827586$<br>$\beta = 0$<br>$\gamma = 0.137931034483$ |
| | Multiplicative | MAE train = 71.26 | $\alpha = 0.206896551724$<br>$\beta = 0$<br>$\gamma = 0.0689655172414$ |
| Weekly | Additive | MAE train = 64.74 | $\alpha = 0.241379310345$<br>$\beta = 0$<br>$\gamma = 0.310344827586$ |
| | Multiplicative | MAE train = 72.018 | $\alpha = 0.0698965517241$<br>$\beta = 0$<br>$\gamma = 0.0354482758621$ |

Table 1: Optimal parameters found for the metric *”Number of Pageviews”* for the Classical Holt-Winters model. The MAE is computed on the train set and the number of parameter combinations explored is $30^3$. (checked, correct)

| **Holt-Winters:** *”Median pageload time”* | | | |
|---|---|---|---|
| Daily | Additive | MAE train = 20.61 | $\alpha = 0.275862068966$<br>$\beta = 0$<br>$\gamma = 0.0344827586207$ |
| | Multiplicative | MAE train = 20.21 | $\alpha = 0.275862068966$<br>$\beta = 0$<br>$\gamma = 0.0689655172414$ |
| Weekly | Additive | MAE train = 20.34 | $\alpha = 0.275862068966$<br>$\beta = 0$<br>$\gamma = 0.172413793103$ |
| | Multiplicative | MAE train = 20.32 | $\alpha = 0.275862068966$<br>$\beta = 0$<br>$\gamma = 0.206896551724$ |

Table 2: Optimal parameters found for the metric *”Median pageload time”* for the Classical Holt-Winters model. The MAE is computed on the train set and the number of combinations of the parameters explored is $30^3$. (checked, correct)

| **Taylor's DS**, (3.11) initialization method | | | |
|---|---|---|---|
| Number of pageviews | Additive | MAE train = 61.15 | $\alpha = 0.25$<br>$\beta = 0.0416$<br>$\gamma = 0$<br>$\delta = 0$ |
| | Multiplicative | MAE train = 61.707 | $\alpha = 0.214857142857$<br>$\beta = 0.0722857142857$<br>$\gamma = 0$<br>$\delta = 0$ |
| Median pageload time | Additive | MAE train = 19.16 | $\alpha = 0.263157894737$<br>$\beta = 0$<br>$\gamma = 0$<br>$\delta = 0$ |
| | Multiplicative | MAE train = 19.556 | $\alpha = 0.368421052632$<br>$\beta = 0$<br>$\gamma = 0$<br>$\delta = 0$ |

Table 3: Optimal parameters found for the two metrics for Taylor's DS model using the initialization method (3.11). The number of parameter combinations explored was $25^4$ for the *Number of pageviews* and $20^4$ for the *Median pageload time*.

| **Taylor's DS**, (3.12) initialization method | | | |
|---|---|---|---|
| Number of pageviews | Additive | MAE train = 65.22 | $\alpha = 0.210526315789$<br>$\beta = 0$<br>$\gamma = 0$<br>$\delta = 0.263157894737$ |
| | Multiplicative | MAE train = 62.679 | $\alpha = 0.157894736842$<br>$\beta = 0$<br>$\gamma = 0$<br>$\delta = 0.157894736842$ |
| Median pageload time | Additive | MAE train = 19.989 | $\alpha = 0.263157894737$<br>$\beta = 0$<br>$\gamma = 0$<br>$\delta = 0.105263157895$ |
| | Multiplicative | MAE train = 20.26 | $\alpha = 0.263157894737$<br>$\beta = 0$<br>$\gamma = 0$<br>$\delta = 0.210526315789$ |

Table 4: Optimal parameters found for the two metrics for Taylor's DS model,using the initialization method (3.12). The number of parameter combinations explored was $20^4$ for both metrics.

# Bibliography

S Tom Au, Guang-Qin Ma, and Shu-Ngai Yeung. Automatic forecasting of double seasonal time series with applications on mobility network traffic prediction. In *2011 Joint Statistical Meetings, July*, 2011.

Peter J. Brockwell and Richard A. Davis. *Introduction to Time Series and Forecasting*, chapter 9. Springer, 2010.

Jake D Brutlag. Aberrant behavior detection in time series for network monitoring. In *LISA*, volume 14, pages 139–146, 2000.

Paulo Cortez, Miguel Rio, Miguel Rocha, and Pedro Sousa. Multi-scale internet traffic forecasting using neural networks and time series methods. *Expert Systems*, 29(2): 143–155, 2012.

Mine Cetinkaya-Rundel David M. Diez, Christopher D. Barr. *OpenIntro Statistics*, chapter 4: Foundations for Inference. CreateSpace Independent Publishing Platform, 2nd edition, 2012.

Ted Dunning and Ellen Friedman. *Practical Machine Learning: A new look at anomaly detection*. O'Reilly Media, 1st edition, 2014.

Peter Flach. *Machine Learning: the art and science of algorithms that make sense of data*. Cambridge University Press, 1st edition, 2012.

Sarah Gelper, Roland Fried, and Christophe Croux. Robust forecasting with exponential and holt–winters smoothing. *Journal of forecasting*, 29(3):285–300, 2010.

Paul Goodwin. The holt-winters approach to exponential smoothing: 50 years old and going strong. *ResearchGate*, 2010.

Rohitha Goonatilake, Susantha Herath, and Ajantha Herath. Probabilistic models for anomaly detection based on usage of network traffic. *Journal of Information Engineering and Applications*, 3(9):28–40, 2013.

Phillip G Gould, Anne B Koehler, Farshid Vahid-Araghi, Ralph D Snyder, J Keith Ord, and Rob J Hyndman. Forecasting time-series with multiple seasonal patterns.

Rob J Hyndman. Measuring forecast accuracy. 2014.

Rob J Hyndman and George Athanasopoulos. *Forecasting: principles and practice*, chapter 7: Exponential smoothing. OTexts, 2013.

Rob J Hyndman and Anne B Koehler. Another look at measures of forecast accuracy. *International journal of forecasting*, 22(4):679–688, 2006.

Rob J Hyndman, Anne B Koehler, Ralph D Snyder, and Simone Grose. A state space framework for automatic forecasting using exponential smoothing methods. *International Journal of Forecasting*, 18(3):439–454, 2002a.

Rob J Hyndman, Anne B Koehler, Ralph D Snyder, and Simone Grose. A state space framework for automatic forecasting using exponential smoothing methods. *International Journal of Forecasting*, 18(3):439–454, 2002b.

Nur Adilah Abd Jalil, Maizah Hura Ahmad, and Norizan Mohamed. Electricity load demand forecasting using exponential smoothing methods. *World Applied Sciences Journal*, 22(11):1540–1543, 2013.

Prajakta S Kalekar. Time series forecasting using holt-winters exponential smoothing. *Kanwal Rekhi School of Information Technology*, 4329008:1–13, 2004.

Spyros Makridakis and Michele Hibon. The m3-competition: results, conclusions and implications. *International Journal of Forecasting*, 16:451–476, 2000.

Wes McKinney. *Python for Data Analysis.* O'Reilly Media, 1st edition, 2012.

Gerhard Münz and Georg Carle. Application of forecasting techniques and control charts for traffic anomaly detection. In *Proc. 19th ITC Specialist Seminar on Network Usage and Traffic, Berlin, Germany*, 2008.

E Monteiro Pena, M de Assis, and M Lemes Proença. Anomaly detection using forecasting methods arima and hwds. In *proceeding of XXXII International Conference of The Chilean Computer Science Society (SCCC)*, 2013.

Mihir Rajopadhye, Mounir Ben Ghalia, Paul P Wang, Timothy Baker, and Craig V Eister. Forecasting uncertain hotel room demand. *Information sciences*, 132(1):1–11, 2001.

Spyros G. Makridakis Rob J Hyndman and Steven C. Wheelwright. *Forecasting: methods and applications*, chapter 4: Exponential smoothing methods. Willie, 3rd edition, 1997.

Vasilios A Siris and Fotini Papagalou. Application of anomaly detection algorithms for detecting syn flooding attacks. In *Global Telecommunications Conference, 2004. GLOBECOM'04. IEEE*, volume 4, pages 2050–2054. IEEE, 2004.

Reinaldo Castro Souza, Mônica Barros, and CVC Miranda. Short term load forecasting using double seasonal exponential smoothing and interventions to account for holidays and temperature effects. *TLAIO II-2 do Taller Latino Iberoamericano de Investigación de Operaciones. Acapulco, México*, pages 1–8, 2007.

Maciej Szmit and Anna Szmit. Usage of modified holt-winters method in the anomaly detection of network traffic: Case studies. *Journal of Computer Networks and Communications*, 2012, 2012.

Maciej Szmit, Slawomir Adamus, Sebastian Bugala, and Anna Szmit. Implementation of brutlag's algorithm in anomaly detection 3.0. In *FedCSIS*, pages 685–691, 2012.

James W Taylor. Short-term electricity demand forecasting using double seasonal exponential smoothing. *Journal of the Operational Research Society*, 54(8):799–805, 2003.

James W Taylor. An evaluation of methods for very short-term load forecasting using minute-by-minute british data. *International Journal of Forecasting*, 24:645–658, 2008.

James W Taylor. Exponentially weighted methods for forecasting intraday time series with multiple seasonal cycles. *International Journal of Forecasting*, 26(4):627–646, 2010a.

James W Taylor. Triple seasonal methods for short-term electricity demand forecasting. *European Journal of Operational Research*, 204(1):139–152, 2010b.

Amy Ward, Peter Glynn, and Kathy Richardson. Internet service performance failure detection. *ACM SIGMETRICS Performance Evaluation Review*, 26(3):38–43, 1998.

Li Wei, Nitin Kumar, Venkata Nishanth Lolla, Eamonn J Keogh, Stefano Lonardi, and Chotirat (Ann) Ratanamahatana. Assumption-free anomaly detection in time series. In *SSDBM*, volume 5, pages 237–242, 2005.

Nong Ye, Qiang Chen, and Connie M Borror. Ewma forecast of normal system activity for computer intrusion detection. *Reliability, IEEE Transactions on*, 53(4):557–566, 2004.