Extending Web Analytics with Prediction Models to Detect Anomalies in Distributed Web Applications

Master-Thesis von Anupma Raj aus India
Tag der Einreichung:

1. Gutachten: Prof. Dr.-Ing. Johannes Fuernkranz
2. Gutachten: Eneldo Loza Mencia

TECHNISCHE UNIVERSITÄT DARMSTADT

Department of Computer Science
Knowledge Engineering Group

Extending Web Analytics with Prediction Models to Detect Anomalies in Distributed Web Applications

Vorgelegte Master-Thesis von Anupma Raj aus India

1. Gutachten: Prof. Dr.-Ing. Johannes Fuernkranz
2. Gutachten: Eneldo Loza Mencia

Tag der Einreichung:

Erklärung zur Master-Thesis

Hiermit versichere ich, die vorliegende Master-Thesis ohne Hilfe Dritter nur mit den angegebenen Quellen und Hilfsmitteln angefertigt zu haben. Alle Stellen, die aus Quellen entnommen wurden, sind als solche kenntlich gemacht. Diese Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen.

Darmstadt, den 1st September 2016

(Anupma Raj)

Abstract

This aim of this thesis was to develop an automated anomaly detection system for typical web analytics data which could accurately and efficiently detect anomalies, report them through alerts within the time constraints and be easily incorporated into typical web analytics software.

Statistical techniques detect anomalies by relying on their standard deviation distance from the mean to trigger preconfigured anomaly detection thresholds specific to different types of data distributions. However, using these proves lacking in situations where data values are anomalous with respect to their position in the time series, but do not have large enough standard deviation distances from the mean of the whole time series. Another approach of anomaly detection is to train prediction algorithms with historical data to make a forecast, and identify an anomaly when it exceeds acceptable Mean Absolute Percentage Error levels when compared with the forecast. In comparison to statistical anomaly detection methods, prediction algorithms can be more resource expensive due to more complex computations.

This thesis proposes a solution which combines both approaches to overcome their respective drawbacks. The resulting automated anomaly detection system showed 100% accuracy in anomaly identification when tested against nine sample datasets collected from a web page which was tracked for over eight months. Thereafter, upon examining the impact of the length of training data on the forecasts made by the prediction algorithm, it was determined that the highest prediction accuracy was obtained when the shortest training data of two weeks was used. When the impact of anomalies in the training data upon the prediction accuracy were examined, results showed improvement in accuracy only during some tests and hence, proved inconclusive in determining any correlation.

Contents

List of Figures

List of Tables

1 Introduction

Today's world that is increasingly dependent on internet technology. There a major shift from traditional application provisioning to cloud-based Software-as-a-Service (SaaS) solutions and the importance of understanding and optimizing web application usage becomes of increasing importance. In order to better the quality of web applications and the content that they present, it becomes critical to gather complete, consistent and current information about web application usage. Appropriately analysing such data unearths a plethora of information that may support decision making and strategy formulation by identifying aspects of a web applications which successfully meet goals and those which don't. Examples are determining how successful a marketing campaign is, whether certain links in a web application are malfunctioning or detect which content works and which does not. The benefits of such knowledge are applicable in diverse fields, including IT operations, IT security, search engine optimization, marketing or advertising.

The process of measuring and collecting such usage data of web applications or web sites is known as web analytics [1]. There are two major approaches: log file analysis and page tagging [2]. Both have their advantages and disadvantages, and a choice of tracking method can be made based on which is more suitable to serve a specific use case. Log file analysis is limited to analysis of server-side log files, which mainly collect normal requests of clients to the server. Everything that is happening on client-side that does not result in a request to the server cannot be logged, and thus cannot be analysed. The page tagging approach adds insights about information and events that can only be detected on the client-side, by embedding additional scripts to web pages that execute on client-side and send back the desired information and/or events. Most of the popular web analytics software such as Google Analytics, Piwik, Open Web Analytics and Adobe Analytics [3, 4] track web usage data through page tagging. The data which has been collected through page tagging is usually presented to users of web analytics software via pre-configured reports that can be consumed easily. More advanced solutions also allow the users to individually query the collected raw data dynamically, and thus provide a means to deal with specialized custom information requirements.

Although web analytics solutions usually provide users with the ability to browse and analyse historical data a posteriori, they usually lack means to check the data automatically for the occurrence of special events and situations that a user wants to be informed about in a timely fashion, without the need of continuously checking the data manually. Examples for this are e.g. the detection of web traffic spikes that require action from operations, or occurrence of unusual access patterns that might indicate a security breach. This thesis will develop such an automated anomaly detection method that can easily be added to typical web analytics solutions, is able to accurately detect deviations in web usage data, and can be implemented in an efficient manner.

This chapter is structured into three sections. The first section provides the motivation of this thesis. The second section gives an insight into the problem statement which this thesis aims to solve. The third section explains the structure of the rest of the thesis.

1.1 Motivation

Web analytics software collect very valuable data. Understanding and appropriately analysing this data is critical for extracting the maximum beneficial knowledge possible. While web analytics software usually display the collected data in pre-configured reports, manual effort is required to observe these reports and conclude information from them. In addition to the resources required to manually and consistently observe these reports, the conclusions formed are also influenced by human error and bias and therefore, might not always be completely correct. Moreover, if there is a need for a timely reaction

to the occurrence of special events, a highly desirable solution would be one which could automatically analyse data, detecting special events and notify interested stakeholders.

Web analytics software collect diverse data in the form of different features, and some of these features can be considered to be universally important and interesting to common users of web analytics software. One such feature is the number of page views that a web page receives. The number of page views can give insights on when a web page is viewed, how many times it is viewed and upon analysis can deduce the correlation between these. This correlation helps to conclude whether the number of page views is increasing or decreasing over time (trend) and whether there are any repeating patterns (e.g. lower access rates after working hours or during the weekend). If one can model expected behaviour of this feature through these conclusions, it becomes possible to identify situations when this feature is not behaving as expected, i.e. behaving abnormally. Issuing an alert after the detection of such a situation allows the manual inspection of this anomaly and consequently, decisions can be made to handle the effects of this anomaly.

1.2 Problem Statement

This thesis aims to develop an automated anomaly detection system for typical web analytics data that is able to accurately detect anomalies, can be efficiently implemented and easily added to typical web analytics software. The developed solution should notify interested stakeholders about the occurrence of an anomaly automatically and in a timely fashion. While in the following this thesis focuses on page views for illustration purposes of typical web analytics data, the application of the anomaly detection system presented in this thesis is not limited to page views. The approach adopted to detect anomalies is based on getting a statistical understanding of the typical number of page views that occur at different times of the day and on different days of the week, and then recognizing when page views deviate from the expected (i.e. predicted) behaviour.

1.3 Thesis Structure

In chapter 2, literature relevant to web analytics software, anomaly detection techniques, prediction algorithms and forecasting accuracy evaluation methods is explored. In chapter 3, the concept of this thesis solution is developed by making the necessary design decisions such as selection of feature and best anomaly detection technique, prediction algorithm and forecasting accuracy evaluation method which are used with different patterns of feature data. Chapter 4 introduces the example web analytics software which is used for the purpose of obtaining real data and testing the ease of integration of this thesis solution. The implementation of the discussed concept in R and Java is also explained. The evaluation of the thesis solution is carried out in chapter 5 by testing its forecasting accuracy for 9 sample datasets. The impact of different lengths of historical data and the presence of anomalies in the historical data is examined. Moreover, the limitations of this thesis solution are discussed. Chapter 6 presents a conclusion to this thesis and proposes relevant future research work.

2 Literature Survey

The previous chapter introduced the motivation behind this thesis work and formulated the problem statement within the scope of this thesis.

This chapter discusses the literature relevant to this thesis. The first section discusses the purpose and types of web analytics software and the different methods of collecting data for it. Out of the many diverse features in this data, some are considered to be universally important and interesting to common users of web analytics software, and are time series. Therefore, the second section discusses the different components of time series and the difference between univariate, bivariate and multivariate time series. The third section discusses existing anomaly detection techniques that are applicable to time series data depending on whether the data is normally distributed or non-normally distributed. This section also discusses how statistical anomaly detection techniques employ thresholds based on the z-score of a data point, and examines how this is calculated. A discussion of the advantages and disadvantages of each anomaly detection technique allows the identification of promising approaches for each data distribution. Finally, this section pinpoints the situations in which these promising anomaly detection techniques may be found lacking. As a potential means to overcome such situations, the fourth section discusses the use of prediction algorithms to identify anomalies. This includes a study of existing prediction algorithms for time series, the advantages and disadvantages of each and identification of the most promising algorithms. Moreover, prediction accuracy measurement methods are discussed which identify the best methods of comparing the predictions made through these algorithms with actual data, so that anomalies can be identified through this comparison. Finally, the sixth section discusses existing methods which could be employed to evaluate the forecasting accuracy of the final thesis solution.

2.1 Web Analytics

The measurement and collection of web data unearths a plethora of information, analysis of which allows us to understand and optimize web usage. This knowledge is powerfully beneficial in the several fields, including IT operations, IT security, search engine optimization, marketing and advertising.

Web Analytics is the process of measuring and collecting such web data [1, 5] and is accomplished in two ways: log file analysis and page tagging [6].

2.1.1 Log File Analysis versus Page Tagging

The following section states the definitions of log file analysis and page tagging and displays the differences between them.

Log File Analysis

Log file analysis is a process whereby log files collected from the web server are parsed to extract information about page requests such as visitor IP address or requested page so that this information can be analysed [7].

Page Tagging

Page tagging is a process whereby a JavaScript code snippet is embedded within the code of the web page which accumulates information about web page usage, visitors and other client side events. When JavaScript is disabled at the clients end, sometimes an invisible image is inserted into a web page as a

fall back method of obtaining some minimal information about the web application usage such as when and how many times the web page was accessed.

Comparison of Log File Analysis and Page Tagging

Table 1 shows a comparison between the log file analysis and page tagging methods and discusses the advantages and disadvantages of both.

Table 1: Comparison of Log File Analysis and Page Tagging

Log File Analysis		**Page Tagging**	
Advantages	**Disadvantages**	**Advantages**	**Disadvantages**
No extra code needs to be embedded within the code of the web page.	Cannot track information about user activities on the client side as only requests issued by the client are logged. An example of user activity on the client side is mouse movement or clicks.	Along with having access to additional information about activity on the client side, it is also possible to instrument code to detect customized events.	Extra code needs to be embedded within the code of the web page. This causes more bandwidth to be used each time the page loads. Moreover, it is harder to change analytics tools as the embedded code would have to be changed.
	Caching drastically affects the data collection as web servers don't recognize a cache load as a new event which needs to be recorded.	High speed of data reporting.	If JavaScript or tracking is disabled, or older browsers are used, data cannot be collected.
	Server has to be specifically configured to assign cookies to visitors, eg. session handling will have to be turned on, even if it is not required, to be able to track users through a whole session.	Page tagging usually automates the process of assigning cookies to visitors.	Search engine spiders statistics are not included in the data collected as they often do not run JavaScript elements.
Data is stored on your private server, which is useful in situations where the data is confidential and it is important that there is no third-party access.	Data is stored on a local server. This becomes expensive as there is an initial server set up cost, high IT overhead of storing and archiving data, providing maintenance in the form of updates and security patches.	Less expensive than log file analysis.	Difficult to change vendors because your data is usually stored on a foreign server.
		Useful method for companies that do not run their own web servers or do not have access to the raw log files for their site.	

Evaluation

Ultimately, the method choice depends on which of the advantages and disadvantages suit the website owner best. There is also the possibility to use a hybrid data collection method, i.e. a combination of log file analysis and page tagging.

Giants in the field of web analytics, like Google Analytics, Adobe Analytics, Open Web Analytics, Data Workbench, Analyzer, Quantcast, Woopra and Yahoo! Web Analytics all use page tagging by embedding JavaScript snippets into the web page code. Companies like AWStats and Webalizer use log file analysis. An example of a company which uses a hybrid implementation of both methods is the open source web analytics software Piwik.

2.1.2 Type of Analytics

There are three categories of analytics [8–10], and these are also applicable to web analytics.

The first is 'descriptive analytics', also known as reporting analytics, which employs a set of tools and technologies to analyse historical data and thereafter identify patterns and trends within the data. This is done through visualization of the data, reporting and trend analysis. This is the most common analytics method used by companies [11].

The second category is 'predictive analytics', which determines probabilities and trends that are likely to occur in the future. Predictive analytics used modelling, machine learning and data mining concepts to find relations within the data that may not be identifiable through descriptive analytics alone and thereafter extrapolate these relationships to make predictions.

The third category is 'prescriptive analytics', which uses the predictions made by predictive analytics of possible probabilities and trends that can occur and suggests decision options to meet business objectives. Prescriptive analytics performs deterministic and stochastic optimizations with the goal of providing a decision or recommendation for a specific action.

2.2 Web Analytics Data: Time Series and its Components

Within the many features that web analytics measures, several occur sequentially through time. A time series is defined as a collection of observations made sequentially through time [12, 13]. Examples of such features are page views and visitors.

A time series can be decomposed into trend, seasonality and error components [14].

Trend

A trend exists when "there is a long-term increase or decrease in the data" [15]. This could be a linear or non-linear component of the time series which changes over time but does not repeat.

Seasonality

Hyndman defines seasonality as a "pattern which exists within a time series when it is influenced by seasonal factors" [15]. Such stable patterns, which repeat over a known and fixed period of time, can be caused by factors such as weather, vacation or holidays [16, 17]. Seasonality can be by the week, month or quarter year.

Error

The error component accounts for random changes within the time series which are unlikely to repeat. These are not a part of the trend or seasonality of the time series.

2.3 Anomaly Detection For Time Series

The definition of an anomaly or outlier is largely dependent on the data itself and the detection method used. While there is no single, universally accepted definition of an outlier, there are several which are generic enough to address different data types and detection methods. Barnet and Lewis define an anomaly as “an outlying observation that deviates markedly from other members of the sample in which it occurs” [18]. Hawkins defines an outlier as “an observation that deviates so much from other observations as to arouse suspicion that it was generated by a different mechanism” [19]. Johnson defines an outlier as “an observation in a data set which appears to be inconsistent with the remainder of that set of data” [20, 21].

Not all outliers are errors or noise, as is usually assumed. When an outlier occurs repeatedly at the same phase in a season, it could be an important part of the dataset. Ignoring such a value could lead to inaccurate modelling and imprecise results [22–24]. So apart from being used during preprocessing to remove anomalous data from the dataset [25], outlier detection is also used in the areas of intrusion detection, fraud detection, fault detection, voting irregularity analysis, data cleansing, severe weather prediction, geographic information system, health monitoring, amongst others [18, 24, 26–28].

An anomaly detection system models a normal system, network or program activity and thereafter uses this to identify activity which deviates from the model as a possible anomaly [29]. Anomaly detection can be of interest both for the importance that the anomaly itself presents, or for preventing anomaly contamination during data modelling [30]. For the purpose of this thesis, anomaly detection is used for the purpose of identifying the importance of the anomaly itself.

2.3.1 Univariate, Bivariate and Multi-Variate Anomalies

There are three types of anomalies- univariate, bivariate and multivariate. Univariate anomalies are those that occur within a single variable or attribute. Bivariate anomalies are those which occur between two variables. And multivariate anomalies occur within multiple variables. [31, 32]

For the purpose of this thesis, the prediction algorithms and anomaly detection will be performed on each feature individually, therefore we will be using univariate anomaly detection methods as there is only one variable involved.

2.3.2 Anomaly Detection Categories

The methods of identifying anomalies can be based on the concepts of statistics or data mining.

Statistical Anomaly Detection

Statistical methods observe data activity and generate profiles which represent normal behaviour. Two profiles need to be maintained to identify anomalies- the present profile and the historic profile. Events observed in real time are used to update the present profile. By comparing this present profile with a historic profile, an anomaly score is calculated. Upon comparison with pre-set thresholds, an anomaly generates an alert if one of the thresholds is triggered.

There are several advantages to using statistical anomaly detection methods such as not needing prior knowledge of every type of failure or anomaly. Moreover, they can accurately identify anomalies that typically occur over longer time durations, instead of just single occurrence anomalies.

There are also some disadvantages of statistical anomaly detection methods. With malicious intent, it is possible to train a statistical scheme to accept abnormal behaviour as normal. Moreover, pinpointing the right thresholds can be difficult. Setting them too strict may cause false alarms and setting them too lax may ignore situations that should cause alarms.

Machine Learning Anomaly Detection

In contrast to statistical methods, machine learning methods develop a model which continually improves performance by inculcating previous data behaviour.

Examples of machine learning methods include Bayesian models, Markov models, support vector machines, neural networks, clustering or principal component analysis, amongst others. The key to using machine learning concepts for anomaly detection is to precisely identify which method models the data accurately, as all these methods have certain assumptions about the data. Choosing an inaccurate model will result in an inaccurate anomaly detection system.

Each method has its respective advantages and disadvantages. But generally, machine learning methods offer superior accuracy of prediction at the cost of high computation and expensive resources [29]. Hence, using such techniques for operations which have to be completed within strict, short frames of time is not scalable.

2.3.3 Data Distributions

Choosing the correct anomaly detection technique or prediction algorithm for a certain dataset depends on how the data is distributed. The section below discusses the different types of data distributions.

Gaussian Distributions

In statistics and probability, Gaussian distribution [33, 34], also known as normal distribution, is a continuous probability distribution. The probability density of the normal distribution defined by the following formula [35, 36]:

$$P(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-(x-\mu)^2/2\sigma^2}$$

Where μ is mean or expectation of the distribution, as well as the median and mode, σ is the standard deviation and σ^2 is the variance.

The Gaussian distribution is the only absolutely continuous distribution whose cumulants, i.e set of quantities that provide an alternative to the moments of the distribution, are zero after the first two [37, 38].

Standard normal distribution is the simplest type of normal distribution when $\mu = 0$ and $\sigma = 1$. The probability density function is given below [35, 36]:

$$\phi(x) = \frac{e^{-\frac{1}{2}x^2}}{\sqrt{2\pi}},$$

All the other types of normal distributions are obtained by multiplying the domain of standard normal distribution by the standard deviation and thereafter translating it by its the mean value. The probability density function is given below [35]:

$$f(x \mid \mu, \sigma) = \frac{1}{\sigma}\phi\left(\frac{x-\mu}{\sigma}\right)$$

68.27% of the data values in a Gaussian distributions are between one standard deviation from the mean. The second standard deviation from the mean includes 95.45% of data values. The third standard deviation from the mean includes 99.73% of the data values. These data value distributions are given by the formulae below [39]:

$$\Pr(\mu - \sigma \leq x \leq \mu + \sigma) \approx 0.6827$$

$$\Pr(\mu - 2\sigma \leq x \leq \mu + 2\sigma) \approx 0.9545$$

$$\Pr(\mu - 3\sigma \leq x \leq \mu + 3\sigma) \approx 0.9973$$

Non- Gaussian Distributions

Data distribution which does not follow the 'bell curve' characteristic to normal distributions is called Non-Gaussian Distribution.

There are many diverse reasons for data to have non-Gaussian distribution. Data which can only take positive values such as weight or height data cannot be normally distributed. Distribution also gets skewed when the data has many extreme outliers. Furthermore, when data is a result of multiple processes, it displays non-normal distribution. An example would be a dataset displaying website views, in which both the weekday views and the weekend views are normally distributed by themselves but when combined, they don't collectively display a Normal distributed.

It is also possible that a different type of distribution is due to the specific data being analysed. For instance, the Weibull distribution [40] is related to the survival times of a product. Log-normal distribution [41] is displayed when the data is related to measuring lengths. Exponential distribution [42] is displayed when the data tracks growth. Other examples of distribution patterns include Poisson and binomial distributions.

2.3.4 Z-Score: Number of Standard Deviations from the Mean

The statistical anomaly detection methods for both Gaussian and non-Gaussian distributions which will be discussed below rely on a the distance of a data point in standard deviations from the mean of the dataset within which that point resides, called the z-score [43]. In the formula below used to calculate the z-score, X is the value of the data point, μ is the mean and σ is the standard deviation.

$$z = \frac{x-\mu}{\sigma}$$

2.3.5 Anomaly Detection through Thresholds

An anomaly score is calculated by comparing present data behaviour with historic data behaviour. The training phase of an algorithm is when normal behaviour of data is established and anomaly score

thresholds are determined which differentiate between normal and anomalous data values [28]. Users should be able to configure the thresholds to adjust the alerting sensitivity of the system to suit their use case. Detection of anomalies and subsequent alerting occurs when the anomaly score surpasses the threshold criteria [44].

There is a fine balance that needs to be maintained between the number of false positives and the anomaly detection accuracy. Failure to suppress false alarms not only impacts system accuracy, but can also cause system failure in cases where the burden of false alarms on the system is too high [45]. Moreover, anomaly detection in certain fields must satisfy a strict bound on the number of false positives allowed in an effective anomaly detection system. For instance, Axelsson [46] states that for an intrusion detection system to be effective, it should not surpass 1 false alarm for every 100,000 events.

2.3.6 Anomaly Detection Methods for Gaussian Distributions

There are several statistical methods that can be used for identifying anomalies in Gaussian data which are dependent on the inherent properties of normally distributed data. For example, mean and standard deviation calculations depend on the inherent assumption that the dataset is symmetric around its mean and non-zero over the entire real line.

Three Sigma Rule

Statistical anomaly detection uses the normal distribution of data values to determine outliers through the three-sigma rule [47], which concludes that 99.73% of data values lie within three standard deviations of the mean [39].

Tukey's Boxplot

J. W. Tukey introduced the Boxplot [48–50], which is a standardized method of displaying data distribution based on the minima, first quadrant, median, third quadrant, and maxima of a dataset. The boxplot calculates robust values of the mean and standard deviation by using the values of the first and third quadrants.

Using boxplot on symmetrically distributed data to identify anomalies is a common approach [51]. This method is more robust against extreme data values when compared to methods that depend on standard deviation or mean calculations [29]. Moreover, Tukey's method may not be appropriate for a small sample size [52]. In Tukey's boxplot, the Inter Quartile Range (IQR) is defined as the distance between the lower and upper quadrants. The inner fences are located at a distance of 1.5 IQR below the first quadrant and above the third quadrant. The outer fences are located at a distance of 3 IQR below the first quadrant and above the third quadrant. A data point which lies between the inner and outer fences can be identified as a mild outlier and a data point which lies beyond the outer fences can be recognized as an extreme outlier.

An example boxplot constructed on a normally distributed dataset is shown in Figure 1. There is a data point which lies outside the outer fences, and can clearly be identified as an outlier.

2.3.7 Anomaly Detection Methods for Non-Gaussian Distributions

Non-Gaussian datasets require anomaly detection methods that are different from those applied upon Gaussian Distributions, as they do not share the same inherent properties. It is important to use the

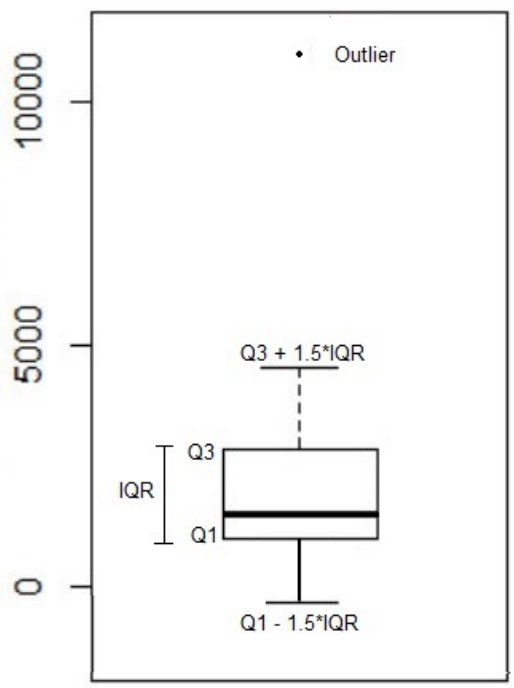

Figure 1: Boxplot of Normally Distributed Dataset with An Outlier

correct methods to avoid high false alarm rates. In statistics, there are several algorithms that can identify outliers for Non-Normally distributed datasets.

Theorem 1: Markov's Inequality

Markov's inequality theorem calculates the upper bound for the probability of a variable's function being greater than or equal to some positive constant. The function generalizes to any random variable that takes non-negative values. [53].

Hence, if X is a non-negative random variable and a > 0, then the probability $\mathbb{P}$ is equal to:

$$\mathbb{P}(X \geq a) \leq \frac{\mathbb{E}(X)}{a}$$

However, the bound 'a' is quite weak as it can only increase linearly. Moreover, the restriction of having only non-negative values is limiting. Therefore, we find Chebyshev's inequality theorem , which is a corollary of Markov's inequality theorem but it removes this restriction of non-negativity.

Theorem 2: Chebyshev's Inequality

Chebyshev's inequality theorem guarantees that a minimum of just 75 percent of values within a dataset must lie within two standard deviations of the mean and 89 percent within three standard deviations [54–56].

It is a corollary of Markov's inequality theorem, with the added advantage of not assuming the non-negativity of the random variable. Moreover, the denominator 'a' rises quadratically, instead of linearly.

Chebyshev's inequality guarantees that $\frac{1}{a^2}$ is the maximum number of the distribution's values which can be more than 'a' standard deviations away from the mean.

If X is a random variable with a finite expected value of $\mathbb{E}(X)$, then for any a > 0,

$$\mathbb{P}(|X - \mathbb{E}(X)| \geq a) \leq \frac{Var(X)}{a^2}$$

The bound can only decrease polynomially. Var(X) is the variance of X, calculated by the following formula:

$$Var(X) = \mathbb{E}[(X - \mathbb{E}(X))^2]$$

Table 2 shows the probability distributions of values for a non-Gaussian distribution per 'a' values.

k	Min. Percentage within 'a' Standard Deviations from Mean(%)	Max. Percentage beyond 'a' Standard Deviations from Mean (%)
1	0	100
$\sqrt{2}$	50	50
1.5	55.56	44.44
2	75	25
3	88.8889	11.1111
4	93.75	6.25
5	96	4
6	97.2222	2.7778
7	97.9592	2.0408
8	98.4375	1.5625
9	98.7654	1.2346
10	99	1

Table 2: Chebyshev's Distribution

Theorem 3: Chernoff's Bound

Chernoff's bound [57] is a theorem which gives exponentially decreasing bounds on tail distributions of sums of independent random variables. This contrasts with Chebyshev's inequality, in which the bound can only decrease polynomially. The bound is much sharper than in both Markov's inequality and Chebyshev's inequality.

Chernoff's bound can be found in the additive and multiplicative forms. The original additive form, also known as Chernoff-Hoeffding theorem, bounds the absolute error.

Chernoff's multiplicative form is much more practical and bounds the error relative to the mean. It is defined by the formula below. Let $X_1, X_2, ..., X_n$ be a sequence of independent random variables from 0 to 1, where $P(X_i = 1) = p$. Let $X = \sum X_i$ and $u \geq E[X]$. The sums expected value is denoted by $\mu = E[X]$. Then for any $\delta > 0$,

$$P_r(X > (1+\delta)\mu) < [\frac{e^{\delta}}{(1+\delta)^{1+\delta_s}}]^{\mu}$$

However, the Chernoff bound requires the variates to be independent.

2.3.8 Evaluation of Anomaly Detection Methods

Out of all the statistical methods discussed for anomaly detection for Gaussian and non-Gaussian distributions, each have their specific advantages and disadvantages.

Both the three sigma rule calculated via z-score and Tukey's boxplot are promising methods for anomaly detection for Gaussian distributions. They are simple and effective. Testing data with both could help determine which is a better method for Gaussian distributions.

In non-Gaussian distributions, Chebyshev's inequality is superior to Markov's which only allows the bound to rise linearly. Chernoff's bound is even more promising because it allows the bounds to rise exponentially. Out of the additive and multiplicative forms of Chernoff's bound, the multiplicative is much more practical as it bounds the error relative to the mean. However, Chernoff's bound requires the variates to be independent, and neither Markov nor the Chebyshev inequalities require this property. Therefore, Chebyshev's inequality seems to be the most promising method of anomaly detection for non-normal distributions when the variates are not independent.

However, the problem with all these methods discussed is that because they rely on standard deviation, the detection of an anomaly can be inaccurate in some situations. Consider the following situation- a web page is being tracked that has extremely a low number of page views during weekends. Even if one of these weekend data points value is twofold or threefold, it would still be less than the value of a data point from a weekday, as the number of page views during the weekdays are much higher. Therefore, this data point would fall within an acceptable number of standard deviations from the mean and be considered a non-anomalous value. However, this would be misidentification because if the value is twofold or threefold the expectation for that data point, then it is obviously not displaying normal behaviour and should be recognized as an anomaly.

2.4 Prediction Algorithms for Time Series

Extensive research has been done on prediction methods that can be applied upon time series, but these are specific to the nature of time series, i.e whether the time series is univariate or multivariate. Chatfield [58] defines an univariate prediction method as one "where the prediction depends only on the present and past values of the single variable being forecast. It is possible that this series is augmented by a function of time, such as a trend". Furthermore, he defines a multivariate prediction method as one "where the prediction depends on more that one time series variable".

For the purpose of this thesis, we will be focusing on univariate prediction methods as only one isolated feature will be used for making the prediction. An example of such a feature would be forecasting the number of visitors within the next hour based on the number of visitors over the past two weeks.

Traditional univariate predictive methods include ARIMA models (also individually considering auto-regression and moving average), exponential smoothing and Bayesian forecasts [59**?** , 60]. Moving average, ARIMA and exponential smoothing are simpler to understand and implement due to their linear nature, i.e. forecast values are constrained to be linear functions of past data.

Makridakis et al. [61] performed an extensive comparison of 24 univariate prediction methods using data from 1001 time series. The data samples were selected to cover a wide spectrum of possibilities,

including diverse sources and starting and ending dates. Although the authors specify that prediction algorithm performance differs depending upon the accuracy criteria used, they determine Holt-Winters and Parzen as the two prediction methods which exhibit the highest degree of consistency among the different accuracy criterion out of the 24 prediction methods. It was also emphasized that the choice of prediction method is largely dependent on the seasonality of data and the time horizon of forecasting, for instance Holt-Winters displayed the most superior accuracy for short term forecasts, i.e. forecasts of one period or less.

Gooijer et al. [62] researched 25 years of univariate time series forecasting methods and observed that ARIMA and triple exponential smoothing display highest forecast accuracy when forecasts were made by the hour and week respectively. After analysis of the state of ar, Gardner [63] concludes that Holt-Winters triple exponential smoothing performs better than ARIMA for seasonal data, but they are both promising approaches. Fildes [64], Rescher [65] and Chatfield et al. [64] further support the superior accuracy of the exponential smoothing and ARIMA methods compared to other prediction methods.

This thesis does not comprehensively explore and compare all the possible univariate prediction methods for time series data, but instead focuses on the two which seem most promising according to the aforementioned literature for univariate time series. The first method to be discussed is ARIMA models, which inculcates auto-regression and moving average.The second method to be discussed is exponential smoothing, which can be in single, double and triple form.

2.4.1 ARIMA

ARIMA modelling integrates both the auto-regression process and the moving average process as discussed below.

Auto-regression Process

Most time series consists of data which is serially dependent, so each data point is a linear combination of previous data points and a random error component. Auto-regression requires that the time series must be stationary. Brockwell et al. [13] define stationarity in terms of a consistent mean and variance through time as the joint probability distribution also remains consistent through time. It is defined by the formula given below. $\varphi_1, \ldots, \varphi_p$... are parameters, c is a constant, and the random variable ε_t is white noise.

$$X_t = c + \sum_{i=1}^{p} \varphi_i X_{t-i} + \varepsilon_t$$

Moving Average Process

Elements in a time series can also be impacted by past errors, and the moving average process accounts for this even though auto-regression cannot. Each data point is a linear combination of previous errors and a random error component. In the formula for calculating moving average below, $\theta_1, ..., \theta_q$ are the parameters of the model. μ is the expectation of X_t (often assumed to equal 0). White noise and errors are displayed as $\varepsilon_t, \varepsilon_{t-1}, ...$

$$X_t = \mu + \varepsilon_t + \sum_{i=1}^{q} \theta_i \varepsilon_{t-i}$$

ARMA model

The auto-regression moving average model was first popularized by Box and Jenkins [66]. The model, referred to as ARMA(p,q), depicts auto-regressive degree in p and the moving average degree in q. It is calculated using the Box and Jenkins approach with the following formula which combines the previous two formulae.

$$X_t = c + \varepsilon_t + \sum_{i=1}^{p} \varphi_i X_{t-i} + \sum_{i=1}^{q} \theta_i \varepsilon_{t-i}.$$

ARIMA

Auto-regressive integrated moving average (ARIMA) model is a result of the generalization of the ARMA model. The ARIMA model integrates auto-regression and moving average explained in the previous sections with differencing, which replaces data values with the difference between their values and the previous values. ARIMA models also make an inherent assumption of stationarity, which is inherited from the auto-regression process this is integrated in ARIMA.

For an ARIMA(p,d,q) model with drift $\delta/(1-\sum \phi i)$ and L as the lag operator,

$$(1-\sum_{i=1}^{p} \phi_i L^i)(1-L)^d X_t = \delta + \left(1+\sum_{i=1}^{q} \theta_i L^i\right) \varepsilon_t$$

Box and Jenkins [66] developed a practical approach to building ARIMA models, which performs model identification, parameter estimation and diagnostic checking. These steps are iteratively repeated until the successful construction of a satisfactory model.

Model identification depends on the theoretical autocorrelation properties of an ARIMA series. It becomes possible to identify possible ARIMA models by matching empirical and theoretical autocorrelation patterns. Sometimes, to achieve stationarity, data transformation is needed in this step.

Parameter estimation follows model identification and aims to reduce overall error through non-linear optimization techniques.

After parameters are estimated, the step of diagnostic checking examines statistics and plots of the residuals to test the accuracy of the model with past data. This step identifies whether there is a need for a new model to be identified according to the accuracy, in which case, the cycle is repeated again starting from model identification step.

Auto-ARIMA

The first method proposed for automating parameter selection by Hanna and Rissanen [67] involved fitting a long auto-regressive model to the data, and then computing the likelihood of potential models through several standard regressions. Gomez and Maravall [68] extended this research to account for multiplicative seasonality and then automated the procedure to calculate the parameters with the lowest Bayesian information criterion (BIC). In more recent literature, Forecast Pro [69] automates the ARIMA function with high accuracy. Another automated method was introduced in 2000 is called AutoBox [70].

Hyndman et al. [71] introduced an automated ARIMA method which is one of the most widely used which used the Akaike Information Criterion (AIC). The generic formula for AIC is given below where L is the maximum value of the likelihood function, k is the number of estimated parameters in the model,

p represents the auto-regression order and q represents the moving average order for non-seasonal part of the series. P and Q represent the auto-regression order and the moving average order for seasonal part of the series respectively.

$$AIC = -2log(L) + 2(p + q + P + Q + k)$$

Hyndman and Khandakar's algorithm [15] for automatic ARIMA modelling uses KPSS tests to determine the differencing order. A KPSS test is named after Kwiatkowski, Phillips, Schmidt and Shin tests, and is used for testing if an observable time series is stationary around a deterministic trend [72]. Thereafter, the seasonal differencing order is deduced with OCSB tests [73], when not specified.

Thereafter, P and Q values are calculated by minimizing AICc after differencing the data. Instead of testing every parameter combination, this algorithm works stepwise to determine the best model.

2.4.2 Exponential Smoothing

Exponential smoothing is used for smoothing time series data by assigning exponentially decreasing weights to the data points over time. There are three types of exponential smoothing- Single, Double and Holt-Winters triple exponential smoothing.

Single Exponential Smoothing

This is the simplest form of exponential smoothing. It is explained by the formula below, where α is the smoothing factor and takes values between 0 and 1.

$$s_t = \alpha \cdot x_t + (1 - \alpha) \cdot s_{t-1}$$

When the smoothing factor is closer to 0, older data points are given more importance while making the prediction. When the smoothing factor is closer to 1, recent data points are given more importance while making the prediction. It's possible to use statistical techniques to automatically deduce the smoothing factor.

Alternatively, a statistical technique such as least squares may be used to optimize the value of the smoothing factor. Single exponential smoothing is technically equivalent to an ARIMA(0,1,1) model and provides no better performance. Single Exponential Smoothing cannot handle any trend or seasonality in the time series data.

Double Exponential Smoothing

Double exponential is designed to handle a trend within the time series, which single exponential smoothing cannot do. x_t represents raw data, and starts at time t = 0. s_t represents the smoothed value for time t, and b_t is the best estimate of the trend at time t. The output of the algorithm is written as $F_t + m$, which estimates x at time t+m, m>0 based on the raw data up to time t. Double exponential smoothing is given by the formulas

α is the smoothing factor and must have a value between 0 and 1 and β is the trend factor and must also have a value between 0 and 1. Double exponential smoothing's formulae are given below.

$$s_1 = x_1$$

$$b_1 = x_1 - x_0$$

When time t > 2,

$$s_t = \alpha x_t + (1-\alpha)(s_{t-1} + b_{t-1})$$

$$b_t = \beta(s_t - s_{t-1}) + (1-\beta)b_{t-1}$$

To make a forecast of future values,

$$F_{t+m} = s_t + mb_t$$

One of the drawbacks of double exponential smoothing is that it accounts only for trend in time series data and not seasonality.

Holt-Winters Triple Exponential Smoothing

Holt-Winters Triple Exponential Smoothing is able to handle both trend and seasonality in time series data. This method calculates a trend line and seasonal indices which assign weight to data depending on their position in the season. Triple Exponential Smoothing can be additive or multiplicative.

Similar to the formulae for Double Exponential Smoothing, s_t represents the smoothed value and the output of the algorithm is written as F_t+m. The seasonal change is of length L b_t represents the sequence of best estimates for the linear trend that are superimposed on the seasonal changes and c_t represents the sequence of seasonal correction factors. α is the smoothing factor, β is the trend factor nd γ is the seasonality factor. α, β and γ must have a value between 0 and 1. Holt-Winters requires a minimum of two seasons of data to initialize a set of seasonal factors.

An additive model adds the Seasonality, Trend, and Error components. The formulae for additive seasonality are given below.

$$s_0 = x_0$$

$$s_t = \alpha(x_t - c_{t-L}) + (1-\alpha)(s_{t-1} + b_{t-1})$$

$$b_t = \beta(s_t - s_{t-1}) + (1-\beta)b_{t-1}$$

$$c_t = \gamma(x_t - s_{t-1} - b_{t-1}) + (1-\gamma)c_{t-L}$$

$$F_{t+m} = s_t + mb_t + c_{t-L+1+(m-1)modL}$$

A multiplicative model multiplies Seasonality with the Trend and Error components. The formulae for multiplicative seasonality are given below.

$$s_0 = x_0$$

$$s_t = \alpha\frac{x_t}{c_{t-L}} + (1-\alpha)(s_{t-1} + b_{t-1})$$

$$b_t = \beta(s_t - s_{t-1}) + (1-\beta)b_{t-1}$$

$$c_t = \gamma\frac{x_t}{s_t} + (1-\gamma)c_{t-L}$$

$$F_{t+m} = (s_t + mb_t)c_{t-L+1+(m-1)modL}$$

The general formula for the initial trend estimate b_0 is given by the formulae below.

$$b_0 = \frac{1}{L}\left(\frac{x_{L+1}-x_1}{L} + \frac{x_{L+2}-x_2}{L} + \ldots + \frac{x_{L+L}-x_L}{L}\right)$$

Seasonal indices are initialized using the formulae given below. Lets assume N to be the number of complete cycles present in your data.

$$c_i = \frac{1}{N}\sum_{j=1}^{N}\frac{x_{L(j-1)+i}}{A_j} \quad \forall i = 1, 2, \ldots, L$$

where

$$A_j = \frac{\sum_{i=1}^{L} x_{L(j-1)+i}}{L} \quad \forall j = 1, 2, \ldots, N$$

2.4.3 Evaluation of Prediction Algorithms

As per the literature survey, forecasting accuracy is affected by trend, seasonality and error in the time series. ARIMA and Exponential Smoothing are the two most promising methods for making forecasts for univariate time series data as they are capable of handling trend and seasonality. Out of the three forms of exponential smoothing, if the data displays no trend or seasonality, then single exponential smoothing may be employed. If the data display only trend, then double exponential smoothing may be employed. If the data employs both trend and seasonality then triple exponential smoothing, i.e. Holt-Winters may be employed.

Furthermore, relevant literature highlights the superiority of Holt-Winters predictions for short term forecasts, i.e. forecasts of one period or less.

Therefore, both ARIMA and the relevant form of exponential smoothing (depending on the presence of trend, seasonality and error) will be applied to the time series data to make forecasts. A comparison will then be made to determine which of the two methods has higher forecasting accuracy.

It is also worth mentioning that in comparison to statistical anomaly detection techniques, prediction algorithms are more resource expensive during as the computations are more extensive.

2.4.4 Prediction Accuracy Measurement Methods

Due to the importance of evaluating forecasting accuracy of a prediction algorithm, there is much discussion about different methods and their performance. Fildes et al. [74] further emphasized the impact that understanding accuracy of forecasting methods has on method selection. Finding the best possible accuracy measure depends highly on the purpose of forecasting, how important it is in making future business decisions and the exact needs and concerns that resulted in this forecasting.

This discussion started as early as 1984 when Mahmoud [75] presented a summary of different accuracy measuring methods. Makridikis et.al [61] identified Mean Absolute Percentage Error (MAPE),

Mean Square Error (MSE), Average Ranking (AR), Medians of absolute percentage errors (Md), and Percentage Better (PB) as the most common methods. Campbell [76], Fildes et al. [74] and Gooijer et al. [62] discussed the advantages and disadvantages of different methods and identified MSE and MAPE as promising approaches. Therefore, in continuation with the conclusions from the aforementioned literature, both these methods, along with their advantages and disadvantages, will be explored below.

Mean Square Error

Mean Square Error (MSE) squares and subsequently averages errors, and hence is a quadratic loss function. The implication of this simple method is that larger errors are given more importance than smaller ones. Hence, it is advantageous to use MSE when the weight of large errors is more impacting than smaller ones.

When X_t is the actual data at period t, F_t is the forecast made at period t, e_t is the forecast error at period t and m is the number of observations, the mean square error is calculated as follows:

$$MSE = \frac{\sum(X_t - F_t)^2}{m} = \frac{\sum(e_t)^2}{m}$$

There are some disadvantages to this simple method as well. Firstly, MSE is highly sensitive to extreme values. Secondly, Chatfield [64] and Armstrong et al. [77] correctly identified that MSE calculates absolute measures and is scale dependent, so it becomes difficult to make comparisons between different series.

Mean Absolute Percentage Error

The Mean Absolute Percentage Error (MAPE) expresses errors as a percentage of the data as a relative measure. MAPE is easy and intuitive in understanding the importance of errors and due to its relative nature, making comparisons between multiple methods becomes easy.

When X_t is the actual data at period t, F_t is the forecast made at period t, e_t is the forecast error at period t and m is the number of observations, it is calculated using the formula below.

$$MAPE = (\frac{\sum\left|\frac{X_t - F_t}{X_t}\right|}{m})100 = (\frac{\sum\left|\frac{e_t}{X_t}\right|}{m})100$$

MAPE has one obvious drawback which Makridakis et al. [61] recognized in that it gives equal errors regardless of whether the actual data, X_t, is larger or smaller than the forecast data,F_t. The difference in absolute percentage errors from this is highly problematic when the value of the actual data is close to zero and the value of the forecast data is high. Moreover, the MAPE is very sensitive to outliers and can be largely impacted in its value from them.

Evaluation of Prediction Accuracy Measurement Methods

Makridakis et al. [61] concluded that MSE is the best method in situations where a forecasting model has to be selected. However, for situations where an error evaluation has to be made of a single series or comparisons have to be made across several series, MAPE is the best method.

For the purpose of this thesis, a method is necessary which evaluates the accuracy of a forecasting method by studying error. Therefore, MAPE is the most suitable method for making such an error evaluation.

2.5 Evaluation of Forecasting Accuracy of Thesis Solution

Measurement of forecasting accuracy can be done by recognizing if an outcome is correctly inferred (true positive or true negative) or is a Type I (false positive) or Type II (false negative) error. In this thesis,

1. A true positive occurs when a data point is an anomaly and is categorized as an anomaly.
2. A true negative occurs when a data point is a non-anomaly and is categorized as a non-anomaly.
3. A false positive occurs when a data point which is a non-anomaly is categorized as an anomaly.
4. A false negative occurs when a data point which is an anomaly is categorized as a non-anomaly.

The false negatives rate is defined with the formula below

$$\text{False Negatives Rate} = \frac{\text{False Negatives}}{\text{True Positives} + \text{True Negatives} + \text{False Positives} + \text{False Negatives}} \times 100$$

The false positives rate is defined with the formula below

$$\text{False Positives Rate} = \frac{\text{False Positives}}{\text{True Positives} + \text{True Negatives} + \text{False Positives} + \text{False Negatives}} \times 100$$

Once each outcome has been classified into one of the four categories, the accuracy rate can be calculated with the formula below.

$$\text{Accuracy Rate} = \frac{\text{True Positives} + \text{True Negatives}}{\text{True Positives} + \text{True Negatives} + \text{False Positives} + \text{False Negatives}} \times 100$$

3 Concept

In the previous chapter, literature relevant to this thesis was discussed including the purpose and types of web analytics software and methods of data collection. The type of data was examined to identify it as univariate time series. Anomaly detection techniques were discussed and it was concluded that three sigma rule and Tukey's boxplot were promising approaches for normally distributed univariate time series data and Chebyshev's inequality theorem was a promising approach for non-normally distributed univariate time series data. However, disadvantages of using only standard deviation dependent methods for anomaly detection were examined. To overcome these, prediction algorithms were explored and according to the literature, ARIMA modelling and Holt-Winters were identified as promising prediction algorithms for time series data which displayed trend and seasonality and for making short term forecasts. MAPE was identified as the best prediction accuracy measure for error evaluation of a single series or made across several series and hence, was chosen for finding the deviation of the prediction from the actual value.

This chapter describes the step-wise development of the concept of this thesis solution. The first section discusses feature selection, where one feature is selected from web analytics software as an example of its typical data, to be used during the development and testing of this thesis solution. In the second section, this features' typical data is examined to identify its seasonality and distribution. The frequency of data collection is also determined. Moreover, this section discusses all the different patterns that this features' data can display and these data patterns are artificially generated using JMeter. The third section discusses an appropriate anomaly detection technique for the data distribution of the web analytics feature, which is then used to determine z-score thresholds that identify anomalies. In the fourth section, the different data patterns generated using JMeter are used to compare the forecasting performance of Holt-Winters against ARIMA to find the prediction algorithm that performs with higher accuracy. Thereafter, MAPE levels are determined which will be used to detect anomalies by comparing the forecast values from the chosen prediction algorithm with the actual data value. The fifth section discusses how this thesis' final solution will detect anomalies by combining the Chebyshev's inspired thresholds with MAPE level difference between forecasts and actual. Finally, this concept design is presented in the sixth section.

3.1 Feature Selection for Prediction Algorithm

In machine learning and pattern recognition, a feature is an individual measurable property of a phenomenon being observed [78]. Web analytics software measures multiple features, many of which are inter-dependent. For example, the visitors and visits feature is directly related to the number of page views, and both show similar seasonal usage patterns. The feature visits per visitor is also related to page views per visit.

Arguably, one of the features which is universally important to track is the number of page views to a web page. It is interesting for the common user of web analytics. Therefore, this thesis will be using the feature 'page views' for forecasting and consequent anomaly detection.

3.2 Page Views Data

In this section, typical page views data is examined to identify its seasonality and distribution. The frequency of data collection is also determined. Moreover, all the different patterns that page views data can display are discussed.

3.2.1 Seasonality of Data

Page views data displays strong periodicity for the same day of the week. For example, every Monday in consequent weeks displays the same levels of page views. Hence, the season of page views data is weekly.

3.2.2 Distribution of Page Views Data

To understand the distribution of typical page views data, a random page views data sample to a technical web page was collected from a sample web analytics software. The seasonality of the data is weekly.

The graph in Figure 2 shows the distribution of this sample dataset. The mean of the dataset is shown as a green vertical line, the yellow lines show the one standard deviation distance from the mean. The orange lines show two standard deviations distance from the mean. The red lines show three standard deviations from the mean. The blue line on the graph displays what a normal distribution would look like and is shown for comparison purposes.

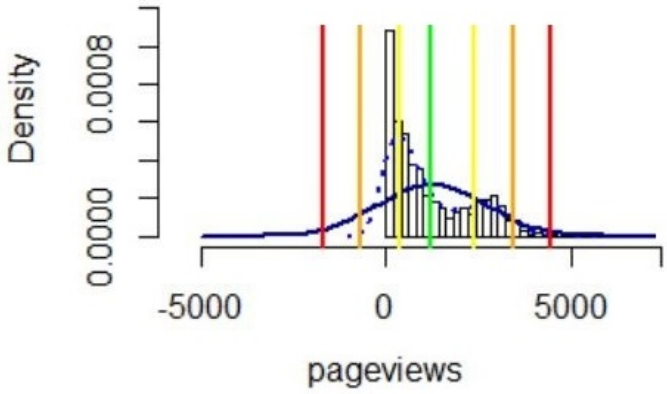

Figure 2: Distribution of Typical Page Views Data

As can be seen in Figure 2, page views data is not normally distributed. There are two reasons for this- firstly, weekly data combines weekdays data with weekend data. Both of these datasets are normally distributed on their own, but when they're combined the resulting dataset is no longer normally distributed. As can be seen through the two peaks in the graph, the resulting dataset has bimodal distribution instead of normal distribution. Secondly, the nature of the data is such that page views cannot be lower than 0, and hence, cannot take negative values. This explains the skewing of the data distribution after 0 on the x-axis.

Even though the graph shown and the discussion of reasons for non-normal distribution is for this specific dataset example, both these reasons hold true for all other page views data to this specific web page. For the purpose of this thesis, all the real page views data and samples are collected from the same web page. Therefore, the data used in this thesis will also show non-normal distribution and the anomaly detection techniques and prediction algorithms used should be those specific to data with non-normal distributions.

3.2.3 Frequency of Data Collection

Web analytics software inherently displays data at varying time granularities. Out of the possibilities of collecting data every minute, hour, day, week and month, for the purpose of this thesis, data will be

collected hourly. This is because hourly data is fine enough to identify minute data patterns and recurring unexpected behaviour without impacting efficiency of the prediction algorithm with too unnecessary fine data.

3.2.4 Page Views Data Patterns

When all the different data patterns that page views can display have been identified, it becomes possible to develop a thesis solution which is prepared to accurately detect anomalies if any such data pattern occurs and it also becomes possible to identify any limitations of this solution.

Ten different page views data patterns are identified in this section. These are- page views that are consistent everyday, continuously increasing, continuously decreasing, seasonally repeating, displaying big increases or big decreases, displaying momentary increase during a spike or momentary decrease during an outage.

Graphs displaying these different datasets are shown below. The number of page views are mapped on the y-axis and the time period, measured in seasons, is mapped on the x-axis.

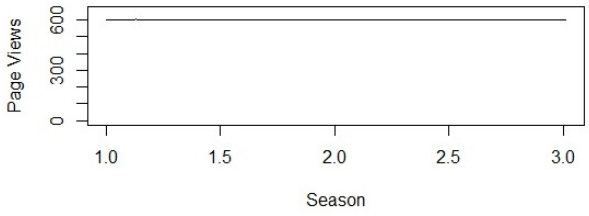

Figure 3: Page Views Dataset Type- Continuously Identical Data

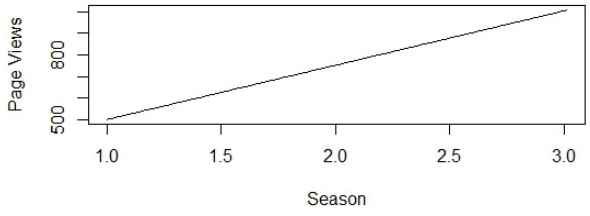

Figure 4: Page Views Dataset Type- Continuously Increasing Data

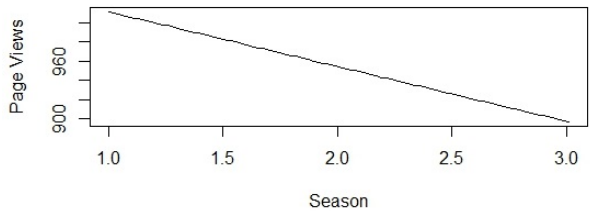

Figure 5: Page Views Dataset Type- Continuously Decreasing Data

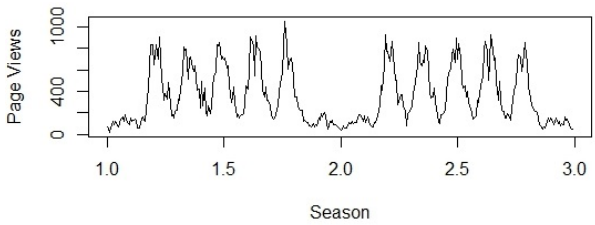

Figure 6: Page Views Dataset Type- Seasonally Repeating Data

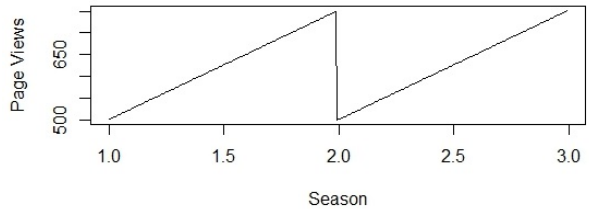

Figure 7: Page Views Dataset Type- Seasonally Repeating, Gradually Increasing Data

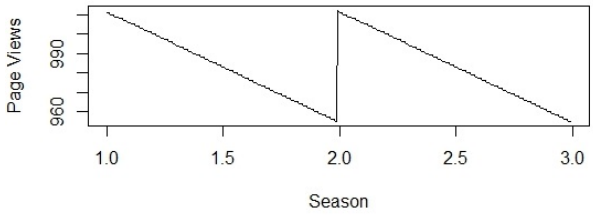

Figure 8: Page Views Dataset Type- Seasonally Repeating, Gradually Decreasing Data

Dataset 1: Continuously Identical Data

In the unlikely event of a web page receiving continuously similar or identical number of page views, the dataset hose graph is shown in Figure 3 would be useful to train the prediction algorithms.

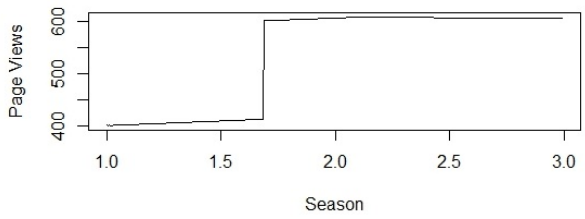

Figure 9: Page Views Dataset Type- Big Increase in Data

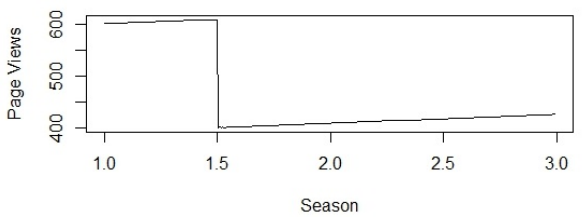

Figure 10: Page Views Dataset Type- Big Decrease in Data

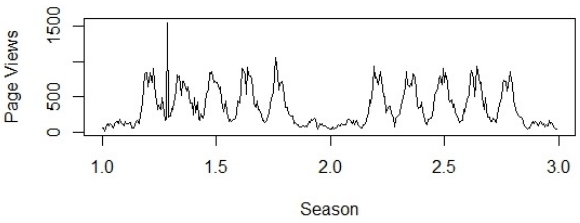

Figure 11: Page Views Dataset Type- Spike Data

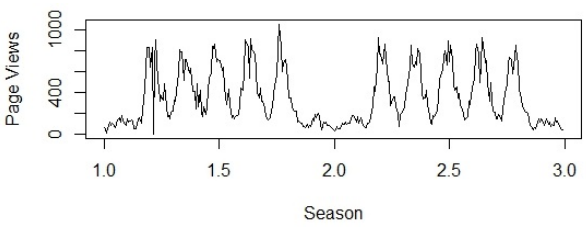

Figure 12: Page Views Dataset Type- Outage Data

Dataset 2: Continuously Increasing Data

The dataset displayed in Fig 4 a scenario where the number of page views are continuously and steadily increasing per season.

Dataset 3: Continuously Decreasing Data

The dataset displayed in Figure 5 displays a scenario where the number of page views is slowly and steadily decreasing per season.

Dataset 4: Seasonally Repeating Data

The dataset displayed in Figure 6 shows a scenario where the data shows a seasonally repeating pattern without any increasing or decreasing trend. For the dataset graphed below, the season is one week and the pattern of page views repeat themselves every season on the x-axis, i.e. 7 days.

Dataset 5: Seasonally Repeating, Gradually Increasing Data

The dataset displayed in Figure 7 shows a scenario where the pattern of page views repeats every season and number of page views increases continuously and gradually within the season. There is no trend in the dataset depicted below, as the long-term movement of page views is steady.

Dataset 6: Seasonally Repeating, Gradually Decreasing Data

The dataset displayed in Figure 8 shows a scenario where the pattern of page views repeats every season and number of page views decreases continuously and gradually within the season. There is no trend in the dataset depicted below, as the long-term movement of page views is steady.

Dataset 7: Big Increase in Data

The dataset displayed in Figure 9 shows a scenario where the normal pattern is disrupted with a large increase in the number of page views. The increased number of page views then becomes the new normal pattern.

Dataset 8: Big Decrease in Data

The dataset displayed in Figure 10 shows a scenario where the normal pattern is disrupted with a large decrease in the number of page views. The decreased number of page views then becomes the new normal pattern.

Dataset 9: Spike Data

The dataset displayed in Figure 11 shows a scenario in which there is a drastic, but fleeting increase in the number of page views. The dataset graphed below shows seasonally repeating data with a large spike between season 1 and 1.5 on the x-axis.

Dataset 10: Outage Data

The dataset displayed in Figure 12 shows a scenario in which there is a drastic, but fleeting decrease in the number of page views. The dataset graphed below shows seasonally repeating data with a large outage between season 1 and 1.5 on the x-axis.

3.3 Anomaly Detection in Time Series

It is necessary to understand which situations classify as an anomaly so that it becomes possible to identify those situations. The scope of an anomaly is obviously specific to the use case. A web page that tracks a patients heartbeat, might need to identify fluctuations of less than 5 heartbeats per minute as anomaly which needs alerting. However, if a web page is tracking weather conditions in Fahrenheit, its possible that even a 10 fahrenheit fluctuation might not be considered an anomaly.

For the purpose of this thesis, the selected feature is page views. The time series page views data will be used and the standard deviation will be calculated. If the new data point lies a certain number of standard deviations from of the mean, it will be identified as an anomaly. However, since anomalies can be of different severities, it is important to have different classifications of an anomaly to identify its severity.

3.3.1 Chebyshev Inspired Thresholds for Triggering Alerts

The severity of an anomaly and the level of alert that is consequently issued will depend on thresholds that state the number of standard deviations that the data point is from the mean. The initial thresholds will be inspired by values from the Chebyshev's inequality theorem, but these can also be configured by the user to adjust the alerting sensitivity specific to their use case. Applying only one set of thresholds to different use cases can lead to over alerting, which increases resource consumption and operational costs, or under alerting, which can be dangerous by missing out on real anomalies that require attention. Having multiple levels of alerting also allows for appropriate attention to be given to an anomaly as some anomalies may need immediate resolution while others could merely be a warning.

Levels of Alerts

There will be three levels of alerts- low level alert, medium level alert and high level alert.

Low level alerts serve as a warning to web page owners that the page views behaviour is not as was expected, and could potentially become alarming in the future. Following the data distribution according to Chebyshev's inequality theorem, a low level alert will be initially configured to trigger whenever a data value is 2 standard deviations from the mean, i.e. falls outside 75% of the data values.

Medium level alerts inform web page owners that the difference between the predicted page views and actual page views is vast enough that it should be looked into. As per Chebyshev's distribution, medium alerts will be initially configured to trigger when the data point is 3 standard deviations away from the mean, i.e. falls outside 88.89% of the data values.

High level alert inform web page owners that very unexpected behaviour is occurring and that it should be manually investigated immediately. As per Chebyshev's algorithm, the high level alert will be initially configured to trigger when the data point is 4 standard deviations away from the mean, i.e. falls outside 93.75% of the data values.

Only the user will be able to balance the sensitivity of the thresholds with the frequency of alerts according to their requirements. For this reason, the thresholds are configurable by the user.

3.4 Prediction Algorithms

As per the discussion in Chapter 3, conceptually the most promising algorithms to make predictions on the time series generated from tracking page views are ARIMA modelling and Holt-Winters Algorithm. In this section, both of these algorithms will be tested on the different types of Datasets discussed in Chapter 3.2 and a comparison will be made of the accuracy with which both algorithms performed prediction of page views.

3.4.1 Comparison of ARIMA and Holt-Winters Algorithm on Different Datasets

To compare the performance of ARIMA and Holt-Winters on the different datasets during this phase of the thesis, these algorithms were implemented in R where the hyper-parameter selection of both algorithms is automated. In the implementation phase in the next chapter, there is further discussion and elaboration of the R implementation.

ARIMA has three hyper-parameters: p which is the auto-regressive Order, d which is the degree of differencing and q which is the moving average order. The optimal value of these was calculated using a variation of the Hyndman and Khandakar algorithm [15] which combines unit root tests, minimization of the Akaike Information Criterion (AIC) and Maximum Likelihood Estimation (MLE) to obtain an ARIMA model.

As explained during the literature review, Holt-Winters has three hyper-parameters: 1. alpha which identifies the time series level, 2. beta which identifies the time series trend component and 3. gamma which identifies the time series seasonal component. These hyper-parameters can be optimally auto-generated using limited memory-BFGS algorithm [79–81]. Limited memory-BFGS is an optimization algorithm which uses an estimation to the inverse Hessian matrix to steer its search through a variable space using limited memory to approximate the Broyden-Fletcher-Goldfarb-Shanno (BFGS) algorithm [82, 83].

Dataset 1: Continuously Identical Data

Predictions made after automating the generation of optimal hyper-parameters for both algorithms are shown in Figure 13 and Figure 14 for Dataset 1.

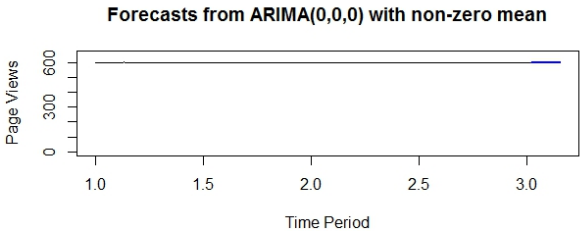

Figure 13: ARIMA Prediction- Continuously Identical Data

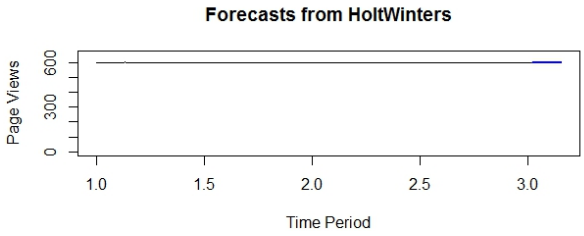

Figure 14: Holt-Winters Prediction- Continuously Identical Data

Dataset 2: Continuously Increasing Data

Predictions made after automating the generation of optimal hyper-parameters for both algorithms are shown in Figure 15 and Figure 16 for Dataset 2.

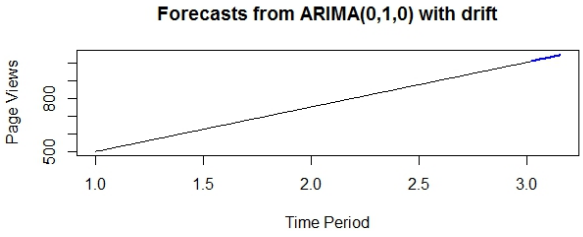

Figure 15: ARIMA Prediction- Continuously Increasing Data

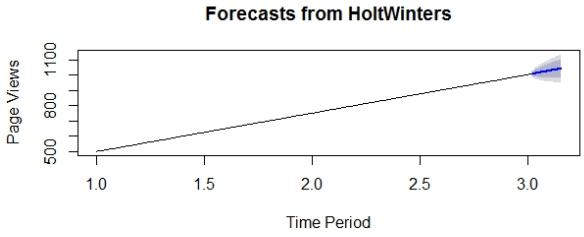

Figure 16: Holt-Winters Prediction- Continuously Increasing Data

Dataset 3: Continuously Decreasing Data

Predictions made after automating the generation of optimal hyper-parameters for both algorithms are shown in Figure 17 and Figure 18 for Dataset 3.

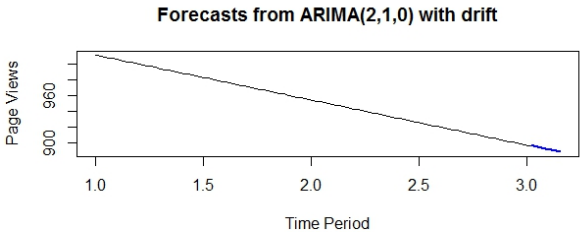

Figure 17: ARIMA Prediction- Continuously Decreasing Data

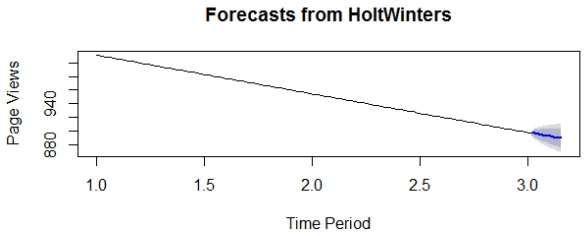

Figure 18: Holt-Winters Prediction- Continuously Decreasing Data

Dataset 4: Seasonally Repeating Data

Predictions made after automating the generation of optimal hyper-parameters for both algorithms are shown in Figure 19 and Figure 20 for Dataset 4.

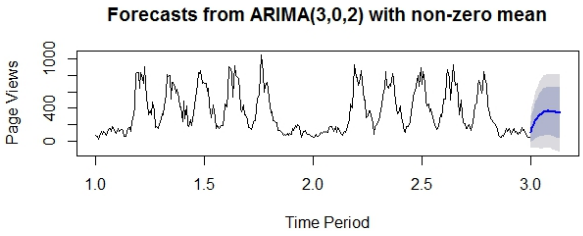

Figure 19: ARIMA Prediction- Seasonally Repeating Data

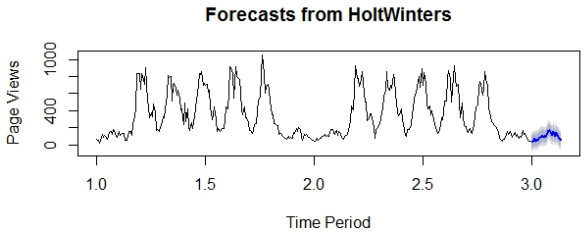

Figure 20: Holt-Winters Prediction- Seasonally Repeating Data

Dataset 5: Seasonally Repeating, Gradually Increasing Data

Predictions made after automating the generation of optimal hyper-parameters for both algorithms are shown in Figure 21 and Figure 22 for Dataset 5.

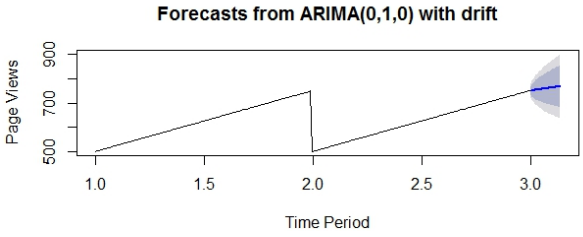

Figure 21: ARIMA Prediction- Seasonally Repeating, Gradually Increasing Data

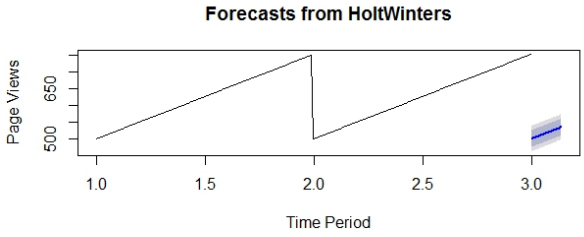

Figure 22: Holt-Winters Prediction- Seasonally Repeating, Gradually Increasing Data

Dataset 6: Seasonally Repeating, Gradually Decreasing Data

Predictions made after automating the generation of optimal hyper-parameters for both algorithms are shown in Figure 23 and Figure 24 for Dataset 6.

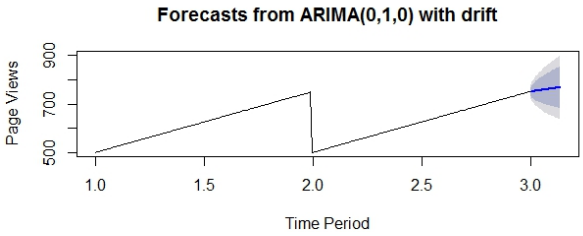

Figure 23: ARIMA Prediction- Seasonally Repeating, Gradually Increasing Data

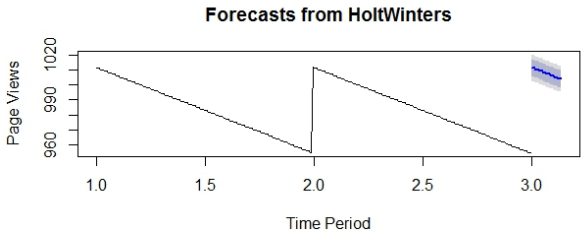

Figure 24: Holt-Winters Prediction- Seasonally Repeating, Gradually Decreasing Data

Dataset 7: Big Increase in Data

Predictions made after automating the generation of optimal hyper-parameters for both algorithms are shown in Figure 25 and Figure 26 for Dataset 7.

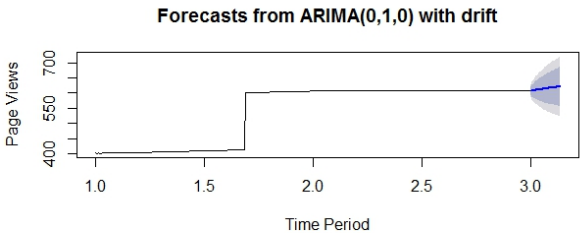

Figure 25: ARIMA Prediction- Big Increase in Data

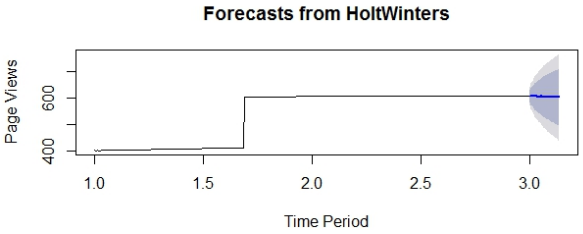

Figure 26: Holt-Winters Prediction- Big Increase in Data

Dataset 8: Big Decrease in Data

Predictions made after automating the generation of optimal hyper-parameters for both algorithms are shown in Figure 27 and Figure 28 for Dataset 8.

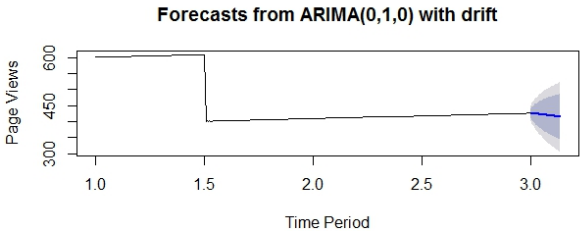

Figure 27: ARIMA Prediction- Big Decrease in Data

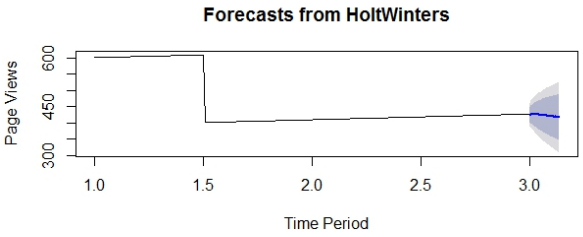

Figure 28: Holt-Winters Prediction- Big Decrease in Data

Dataset 9: Spike Data

Predictions made after automating the generation of optimal hyper-parameters for both algorithms are shown in Figure 29 and Figure 30 for Dataset 9.

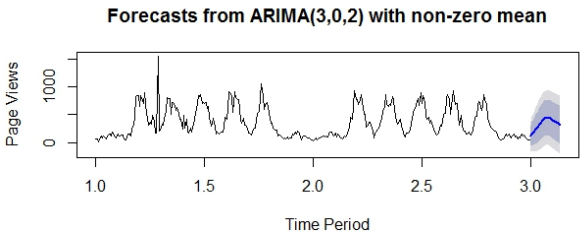

Figure 29: ARIMA Prediction- Spike Data

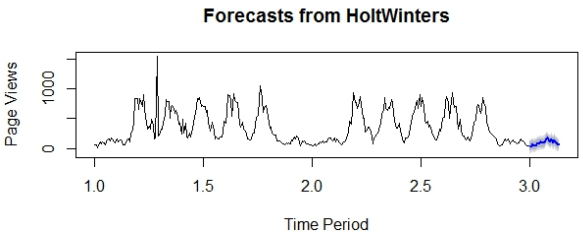

Figure 30: Holt-Winters Prediction-Spike Data

Dataset 10: Outage Data

Predictions made after automating the generation of optimal hyper-parameters for both algorithms are shown in Figure 31 and Figure 32 for Dataset 10.

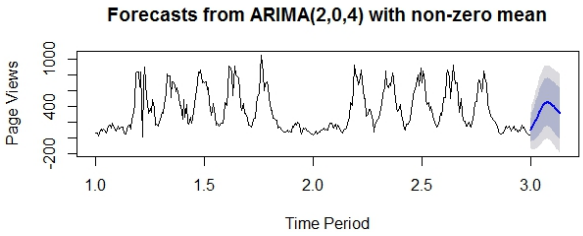

Figure 31: ARIMA Prediction- Outage Data

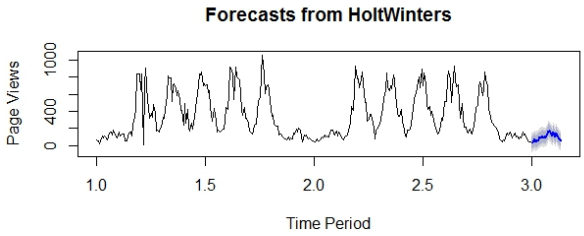

Figure 32: Holt-Winters Prediction- Outage Data

Evaluation of Forecasting Accuracy of Prediction Algorithms

Table 3 below compares the accuracy of the predictions made by ARIMA modelling and Holt-Winters algorithm by comparing the error percentage of predictions made by both for different types of datasets.

Overall, Holt-Winters displays over 10% more accuracy than ARIMA modelling in making predictions for all the datasets. The biggest difference in absolute prediction error, when Holt-Winters is 49% more accurate than ARIMA, is for the predictions of seasonally repeating gradually increasing dataset. Prediction errors are also substantially different in the spike dataset, with an absolute percentage difference of 18.9%. Also worth discussing is the difference in predictions for the seasonally repeating dataset, as this is the most natural and realistic dataset. Holt-Winters makes a more accurate prediction, with 7.6% more accuracy than ARIMA.

It is clear from the performance of Holt-Winters prediction on the different datasets that it models the data better and makes more accurate predictions than ARIMA. Therefore, from here on, Holt-Winters algorithm will be used for the purpose of this thesis.

3.4.2 MAPE Levels for Triggering Alerts

As discussed, after studying page view behaviour and patterns from the training data, a prediction will be made about the expected behaviour using Holt-Winters. When the actual behaviour is compared to the prediction that was made, an anomaly will identify situations where the difference between the two situations surpasses a pre-configured threshold. An example would be if the actual page views for the next hour are 15% more or less than the predicted value of page views for the next hour, then trigger an alert.

Levels of Alerts

Clearly, it is important to determine the difference between the forecast value and the actual value. This can be calculated with the MAPE between the two values. For the purpose of this thesis, if the MAPE between the forecast value and the actual value of the data point value is over 100%, then the actual data point will be considered a low level anomaly. If the MAPE between the forecast value and the actual value of the data point value is over 150%, then the actual data point will be considered a medium level

Table 3: MAPE Comparison of ARIMA and Holt-Winters on Different Datasets

Dataset Type	Actual page views for the Next Hour	ARIMA		HOLT-WINTERS	
		Predicted Value For Next Hour	Percentage Error	Predicted Value For Next Hour	Percentage Error
Continuously Identical Data	602	602.0000	0.0000	601.9948	0.0009
Continuously Increasing Data	1010	1009.5000	0.0495	1009.0000	0.0990
Continuously Decreasing Data	896	896.0001	0.0000	895.9993	-0.0001
Seasonally Repeating Data	69	111.0969	61.0100	32.1649	-53.3842
Seasonally Repeating, Gradually Increasing Data	503	751.7463	49.4525	501.0227	-0.3931
Seasonally Repeating, Gradually Decreasing Data	1010	954.8328	-5.4620	1010.9950	0.09852
Big Increase in Data	427	607.6119	42.2979	606.7635	42.0992
Big Decrease in Data	607	426.4776	-29.7400	424.1725	-30.1199
Spike Data	69	119.9619	73.8578	31.1037	-54.9222
Outage Data	69	104.3746	51.2676	34.1900	-50.4493
Mean Percentage Error				24.2733	-14.697
MAPE				24.2733	-14.697

anomaly. And finally, if the MAPE between the forecast value and the actual value of the data point value is over 200%, then the actual data point will be considered a high level anomaly.

3.5 Combining Anomaly Detection and Prediction Algorithms

For a robust solution for finding any anomalies in page views behaviour, a combination of anomaly detection technique with the prediction algorithm output needs to be implemented. To determine whether a data point is an anomaly, this thesis solution will first calculate the standard deviation of that data point with its historical data. If the data point is enough number of standard deviations away from the mean to trigger any one of the three alerting thresholds, i.e. the data point will be identified as a low level anomaly if it is more than 2 standard deviations away from the mean, or as a medium level anomaly if it is more than 3 standard deviations away from the mean or as a high level anomaly if it is more than 4 standard deviations away from the mean.

If the data point lies at a distance of 2 standard deviations or less from the mean and does not trigger any of the three thresholds, then its MAPE will be calculated. The process of calculating the MAPE involved using historical data values before the data point for training the Holt-Winters prediction algorithm, which will then make a forecast of what the value of that data point should be. A comparison is

made between the forecast value and the actual value for that data point and the MAPE is calculated. If the MAPE is over 100%, then the actual data point will be considered a low level anomaly. If the MAPE is over 150%, then the actual data point will be considered a medium level anomaly. Or finally, if the MAPE is over 200%, then the actual data point will be considered a high level anomaly.

3.6 Concept Design

The conceptual decisions made in this chapter now allow for the generation of a complete concept design. Figure 33 shows the collection of page views data from the web analytics software and how it is processed in order to determine anomalous behaviour.

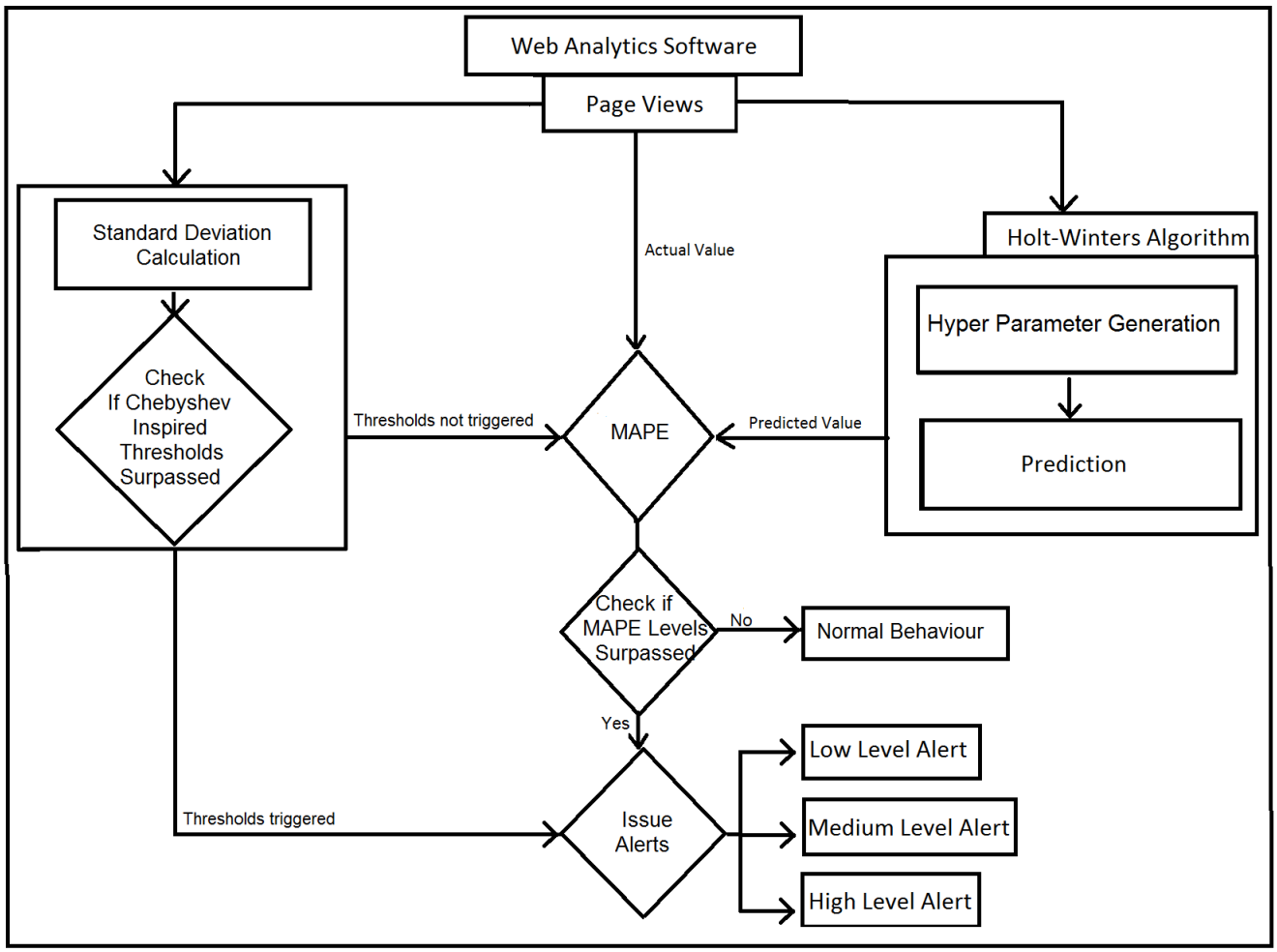

Figure 33: Concept Design of the Thesis Solution

4 System Implementation

The previous chapter described the step-wise development of the concept of this thesis solution. The feature 'page views' was selected as an example of typical web analytics data because it is universally considered important and interesting by common web analytics software users. Page views data was examined and its seasonality was determined to be weekly. Because it combined two normally distributed datasets (i.e. data from the weekdays and data from the weekends), its distribution was determined to be bimodal and hence, non-normal. Ten different types of patterns which page views data could display were discussed and artificially generated using JMeter. Thereafter, Chebyshev's inequality theorem was determined as the favourable anomaly detection technique because of the non-normal distribution of data, and was used to determine anomaly thresholds. Thereafter, the different data patterns generated using JMeter were used to compare the forecasting performance of Holt-Winters against ARIMA, and Holt-Winters made forecasts with higher average accuracy, which was calculated using MAPE. Thereafter, it was determined that the thesis' final solution will develop an anomaly detection system which depends on both Chebyshev's inspired thresholds and MAPE level difference between forecasts made by Holt-Winters and actual data value. Finally, a concept design diagram which encapsulated all these decisions was presented.

This chapter discusses the implementation details of the concept design formulated earlier. To obtain real web analytics data and to test integration of the developed concept, there is a need for an illustrative web analytics software. The first section introduces SAP Web Analytics (SWA), which is used as an example of a typical web analytics software. As SWA tracks page views and currently has no anomaly detection alerting method, it is an appropriate system for testing the integration of the developed concept. When the accuracy of forecasts made by Holt-Winters and ARIMA had to be compared to determine the best prediction algorithm in chapter 3, there was a need for a fast paced development environment where these two algorithms could be efficiently implemented. The second section discusses R, which is used as the programming language and development environment for the initial concept development because it already has both these algorithms in its default forecast package and is user friendly and fast while generating graphs which aid in the analysis of results. R's forecast.Arima and forecast. HoltWinters functions are explored. Methods of automating the identification of optimal model paramteres and hyper-parameters respectively are discussed. The third section emphasizes the need to implement the final concept in Java so that it can be integrated in SWA, the web analytics software being used as an example in this thesis, whose backend is written Java and identifies Java as the preferred language for any "extensions" to be added to SWA to allow easy integration with the existing web application stack. Firstly, the REST calls made to SWA API to obtain page views data are discussed along with the configurations file, which holds the configuration settings required to make successful calls to the API. Thereafter, the received data is discussed along with how it is processed. The method for calculating z-score is then discussed so that the Chebyshev inspired thresholds can use it to identify anomalies. The implementation of Holt-Winters algorithm is discussed, including its hyper-parameter generation using cross-validation and the implementation of both the additive and multiplicative forms after the calculation of the initial trend, level and seasonal indices. All the open-source libraries used are discussed. Thereafter, the comparison of the Holt-Winters forecast with the actual data value to calculate the MAPE is shown. The MAPE levels are determined which identify anomalies and are integrated with the Chebyshev inspired thresholds to form the final thesis solution - an automated anomaly detection system. The alerts generated from the automated anomaly detection system are then displayed. Finally, this section concludes with an architecture diagram of the Java implementation.

4.1 SAP Web Analytics

SAP Web Analytics (SWA) is used as an example web analytics software to obtain real web analytics data and test the integration of this thesis concept. SWA, like other web analytics software, aims to understand how users use a web application by tracking page loads, click events, or custom events. SWA runs on SAP SE's propriety HANA Cloud Platform, and uses the page tagging technique of embedding a JavaScript snippet into the code of the web page being tracked.

SWA has a series of predefined reports including new and recurring visitors, device used, browser language, operating system, geographic location, click-through paths, heat map of popular positions on the web page and page views, the feature selected to represent typical web analytics data in this thesis. Moreover, as there is currently no anomaly detection alerting system in SWA, it is an appropriate system for testing the integration of the developed concept.

Figure 34 a screen shot of the SWA landing page. The menu on the left displays the different preconfigured reports, which have been grouped into six categories- behaviour, technology, geography, social, search and clickmap.

Figure 34: SWA- Landing Page

The sample page views report in Figure 35 gives information about how many page views there are for the website and how many of them are unique page views. This report allows the identification of behavioural patterns in the usage of the website. In the graph below, the page views report displays data for one week and the data collection is set to be hourly. The data points in blue display the page views per hour and the data points in green display the number of unique visitors per hour.

As per the users convenience, the report can be displayed as a line chart, horizontal and vertical bar chart or a table view. The data for the selected time range and granularity of data collection can be downloaded as a CSV or Excel file.

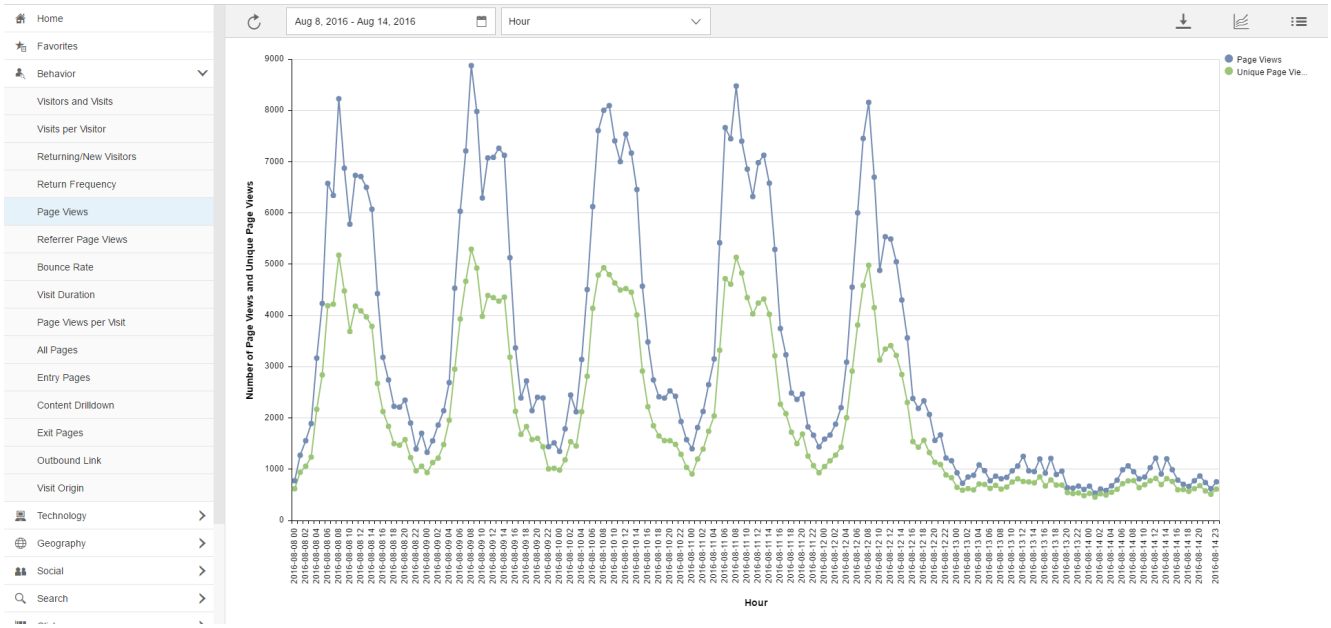

Figure 35: SWA- Page Views Report

4.2 R Implementation

In chapter 2, theoretical comparisons were made between several prediction algorithms to determine which were worth further investigating for the purpose of this thesis. The prediction algorithms were narrowed down to Holt-Winters and ARIMA on the basis of the literature review. During the concept development phase, the forecasting accuracy of these two algorithms needed to be compared on different data patterns to determine the one with highest average performance. Hence, an environment was required where this comparison could be made.

R is a powerful open-source language for statistical computing and graphics. It is an easy choice of programming language and development environment to compare the accuracy of forecasts made with ARIMA and Holt-Winters because it already has both these algorithms in its default forecast package [84, 85], resulting in fast paced development and prototyping. Moreover, R is user friendly and fast in generating graphs which aid in the analysis of results.

For this thesis, the R version used is 0.99.893. The version of the aforementioned forecast package is 7.1 which includes implementations of both ARIMA as forecast.Arima function and and Holt-Winters as forecast.HoltWinters function. Both these functions are written by Rob J Hyndman [86]. The implementations of these functions are discussed below.

4.2.1 forecast.Arima

The forecast.Arima function is used on univariate ARIMA models and returns forecasts and related information. The object whose forecast has to be made is usually the result of a call to R's $arima()$, $auto.arima()$, $ar()$, $arfima()$ or $fracdiff()$ functions. In this thesis, the $auto.arima()$ function was used to automate the identification of optimal ARIMA model parameters to generate this object. Manually determining model parameters can be time consuming and potentially an inaccurate method. auto.arima calculates the first order differencing through KPSS tests and seasonal order differencing through OCSB tests. First order and seasonal values of the parameters p and q are determined through step-wise selection.

The page views data is read from a CSV file, downloaded directly from SWA, using the $read.csv()$ function and stored into a time series object using the $ts()$ function. The time series object is then passed as an argument of $auto.arima()$ function. Thereafter, the resulting object and the number of periods of forecasts to be made are passed as arguments of the $forecast.Arima()$ function. The output elements of forecast function are shown in Table 4 and these also apply to the forecast.Arima function.

Table 4: Output Elements of forecast Function

Element Name	Description
summary	Summary of the forecasting results
plot	Plot of the forecasts and prediction intervals
residuals	Residuals from the fitted model, calculated by subtraction the original time series from the fitted values.
fitted	Fitted values, one-step forecasts.
model	A list containing information about the fitted model
method	The name of the forecasting method as a character string
mean	Point forecasts as a time series.
lower	Lower limits for prediction intervals
upper	Upper limits for prediction intervals.
level	The confidence values associated with the prediction intervals
x	Original time series.

4.2.2 forecast.HoltWinters

The forecast.HoltWinters function is used on univariate Holt-Winters models and returns forecasts and related information. The object whose forecast has to be made is usually the result of a call to R's $HoltWinters()$ function.

The page views data is read from a CSV file, downloaded directly from SWA, using the $read.csv()$ function and stored into a time series object using the $ts()$ function. The time series object is passed as an argument of $HoltWinters()$ function to generate a Holt-Winters model. Thereafter, this model and the number of periods of forecasts to be made are passed as arguments of the $forecast.HoltWinters()$ function. Within this function, the hyper-parameters can either be manually specified, or automatically calculated if set to $NULL$ by minimizing the squared one-step prediction error. The alpha parameter signifies trend. The beta parameter signifies trend and can be set to $FALSE$ indicating lack of a trend. The gamma parameter signifies seasonality and can be set to $NULL$ indicating lack of seasonality. The hyper-parameter values range from 0 to 1. Values close to zero indicate that relatively little weight is placed on the most recent observations while making forecasts and vice versa.

The output elements of forecast function are shown in Table 4 and these also apply to the $forecast.HoltWinters()$ function.

4.3 Java Implementation

SWA, the example web analytics software being used for this thesis, has its backend written in Java and identifies Java as the preferred language for any "extensions" to be added to SWA to allow easy integration with the existing web application stack. Therefore, this thesis solution was implemented in Java.

The Java implementation is a command line application that serves as a prototype to demonstrate the likelihood of this thesis solution being able to integrate in a real web analytics application and in principle, enable timely alerts when anomalies are detected. This section discusses the REST calls

made to SWA API to obtain page views data and the processing of the response data. Thereafter,the method for calculating z-score is discussed so that the Chebyshev inspired thresholds can use it to identify anomalies. The implementation of Holt-Winters algorithm is discussed, including its hyper-parameter generation using cross-validation and the implementation of both the additive and multiplicative forms after the calculation of the initial trend, level and seasonal indices. All the open-source libraries used are discussed. Thereafter, the comparison of the Holt-Winters forecast with the actual data value to calculate the MAPE is shown. The MAPE levels are determined which identify anomalies and are integrated with the Chebyshev inspired thresholds to form the final thesis solution- an automated anomaly detection system. The alerts generated from the automated anomaly detection system are then displayed. Finally, this section concludes with an architecture diagram of the Java implementation.

The project was implemented using the software stated in Table 5.

Table 5: Java Implementation - Software and Technologies Used

System Property	Description
Java	Version 1.8.0_102
Eclipse	Version: Mars.2 Release (4.5.2)
SAP Web Analytics (SWA)	Beta Version- Backend written in Java and Frontend in Javascript.

The Java project consisted of 3 packages with different classes and these have been briefly explained in Table 6. Kingly note that '*' signifies "com.sap.webanalytics." string.

4.3.1 Obtaining SWA Page Views Data and Processing It

The first step in implementing this thesis solution is obtaining the page views data from SWA so that it can be processed. This is done by sending an HTTP GET request to the SWA API as shown in the Java code below.

```
String urlPath = System.getProperty(Configuration.SETTINGS_SWA_USERNAME) +
    "/report/behaviorPageViews?from=" + getOldDate(14) +"&to=" + getOldDate(0) +
    "&timeAccuracy=hour";
URL url = new URL(urlPath);
HttpsURLConnection uc = (HttpsURLConnection) url.openConnection();
uc.setRequestMethod("GET");
String userPassword = System.getProperty(Configuration.SETTINGS_SWA_USERNAME) + ":" +
    System.getProperty(Configuration.SETTINGS_SWA_PASSWORD);
uc.setRequestProperty("Authorization", "Basic " +
    Base64.encodeBase64String(userPassword.getBytes(userPassword));
uc.addRequestProperty("User-Agent", Configuration.USER_AGENT);
uc.connect();
```

The *urlPath* variable is passed with the GET request and it states the SWA API endpoint, the report being accessed (i.e. PageViews), the time duration for when the data should be accessed (i.e. all the PageViews data from 14 days ago up to the current date) and the frequency of data collection (i.e. hourly). The authorization related variables refer to the SWA username and password for accessing the API. Table 7 shows the values of these variables and others which can be found inside a configuration file. The HTTP GET request to the SWA API returns the page views data as a JSON object. This JSON object is processed in a class called *GetPageViews*, which uses the *ResponseHelper*() function to input it into an ArrayList.

Table 6: Description of Packages and classes in Java Project

Package	Class and Description	Description
*api.prediction.utils	HTTPRequestUtil	This class executes the HTTP GET request. It opens the connection, formulates the request and executes it.
*api .prediction.configurations	Configuration	File containing the values of the variables necessary to make the GET Request and the definition of thresholds and MAPE levels for detecting anomalies.
	ResponseHelper	Declares ResponseHelper, used to handle JSON response.
*api.prediction	Main	This class calls functions to implement the entire concept design of this thesis solution.
	GetPageViews	This class sets up the GET request with all its paramters, converts the JSON response into an ArrayList
	CalculateZScore	This class calculates the z-score of the ArrayList of page views data.
	ThresholdDependentAlert	This class compares the calculated z-score with the thresholds setup in the configuration file and issues an alert accordingly.
	HoltWintersHyperParamters	This class uses the ArrayList of page views data, performs cross-validation and calculates optimal alpha, beta and gamma parameters for Holt-Winters.
	HoltWinterAlgorithm	This class executes the Holt-Winters algorithm by calculating initial level, trend, seasonal indices and executing additive and multiplicative forms. It returns an ArrayList of forecast values.
	MAPEDependentAlert	This class uses the last value from the ArrayList of page views data and compares it with the forecast values ArrayList to calculate the MAPE levels. If MAPE levels specified in the configuration file are surpassed, an alert is issued.

4.3.2 Anomaly Detection Through Thresholds

This ArrayList is then passed to $CalculateZScore()$ function which calculates the distance of the last data value in standard deviations from the mean of the page views. The calculated z-score is passed to the $ThresholdDependentAlert()$ function which compares the calculated z-score with the thresholds stated in the configuration file to determine if an anomaly is detected. If any one of the thresholds are triggered, an alert is displayed in the command line and the program execution ends.

Table 7: Configuration File Variables

Configuration Variable	Value
USER_AGENT	"Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.0)"
SETTINGS_SWA_REPORTS	"swa_report_api"
DEFAULT_SWA_REPORTS_URL	"https://sapwebanalytics.hana.ondemand.com/warp/api/v1"
DEFAULT_HHTP_PROXY_HOST	"proxy.wdf.sap.corp"
DEFAULT_HHTP_PROXY_PORT	"8080"
DEFAULT_SKIP_SSL_CHECKING	"true"
SETTINGS_SWA_USERNAME	"THESIS_PREDICTION_USER"
SETTINGS_SWA_PASSWORD	"********"
PERIOD	168
M	1
HIGH_ZSCORE_THRESHOLD	3.2
MED_ZSCORE_THRESHOLD	2.8
LOW_ZSCORE_THRESHOLD	2.5
HIGH_MAPE_LEVEL	200
MED_MAPE_LEVEL	150
LOW_MAPE_LEVEL	100

4.3.3 Anomaly Detection Through MAPE Levels

If none of the z-score thresholds are triggered, this thesis solution then implements Holt-Winters prediction algorithm and compares the forecast with the actual value for the last hour to calculate the MAPE difference.

The ArrayList containing page views data is passed to the $HoltWintersParamters()$ function where the optimal hyper-parameters: alpha, beta and gamma are determined using cross-validation, which is a model validation technique. Although, there are several methods for determining the optimal hyper-parameters, one possible method is cross-validation, which is well suited to applications which aim to perform forecasting and is also often used to estimate the accuracy of a forecasting model. When applied successfully, cross-validation can reduce the negative impact of over-fitting a model and can deduce how accurately a training model will make forecasts on an unknown data set. In the $CalculateHyperParamters()$ function, all the possible combinations of the hyper parameters (i.e. all the alpha, beta and gamma values from 0 to 1, with 0.1 increments) are used to make forecasts, and the hyper-parameter combination which minimizes the error in forecast is selected. However, testing all the possible combinations of hyper-parameters negatively impacts the performance of this thesis solution and slows it down.

Once the hyper-parameters have been determined, Holt-Winters forecasts must be made. There is one open-source implementation of Holt-Winters algorithm by Nishant Chandra [87] in Java. However, his implementation only supports the multiplicative form. There is also a need for implementing the additive form of Holt-Winters because the trend of a time series is prone to changing forms and can change from multiplicative to additive or vice versa. Therefore, in this thesis solution, the additive form was built upon Chandra's multiplicative form implementation.

For the additive form, the calculation of the initial level is shown in the code below, where *period* of the data is one week, i.e. 168 data values. The initial level is stored in a Double variable.

```
for (int i = 0; i < period ; i++){
sumofpageviewsArrayList =+ pageviewsArrayList.get(i);
}
```

```
initiallevel = sumofpageviewsArrayList/period;
```

For the additive form, the calculation of the initial trend is shown in the code below, which is stored in a Double variable.

```
for (int i = 0; i < period; i++) {
sumofpageviewsArrayList += (pageviewsArrayList.get(period + i) -
    pageviewsArrayList.get(i));
}
initialtrend = sumofpageviewsArrayList / (period * period);
```

For the additive form, the calculation of the initial seasonal indices is shown in the code below. The Double variable called L represents the lag operator. The seasonal indices are stored in an ArrayList of Doubles.

```
for (int i = 0; i < period; i++) {
seasonalIndices[i] = seasonalIndices[i] - L ;
}
```

Finally, the code below displays the implementation of the formulae for additive Holt-Winters forecasting. The seasonal values are represented as St, the trend values as Bt and the level values as It. The forecasts, Ft are thereafter calculated.

```
// Overall smoothing calculations
if ((i - period) >= 0) {
St[i] = alpha * (y.get(i) - It[i - period]) + (1.0 - alpha)* (St[i - 1] + Bt[i - 1]);
} else {
St[i] = alpha * y.get(i) + (1.0 - alpha) * (St[i - 1] + Bt[i - 1]);
}

// Trend smoothing calculations
Bt[i] = gamma * (St[i] - St[i - 1]) + (1 - gamma) * Bt[i - 1];

// Seasonal smoothing calculations
if ((i - period) >= 0) {
It[i] = beta * (y.get(i) - It[i - period]) + (1.0 - beta) * It[i - period];
}

// Forecast calculations
if (((i + m) >= period)) {
Ft.set((i+m), (St[i] + (m * Bt[i]) + It[i - period + m]));
}
```

This function outputs the forecast in a Double ArrayList. The forecast made can be compared with the actual value to deduce the MAPE, and if it surpassed the levels specified in the $Configuration$ file, then an alert was issued on the command line. A screenshot of the output of the implementation of this thesis solution in Java is displayed in Figure 36.

The screenshot 36 displays the predicted value and the actual value and calculated that neither the z-score dependent thresholds, nor the MAPE levels were surpassed. Therefore, no alerts are issued.

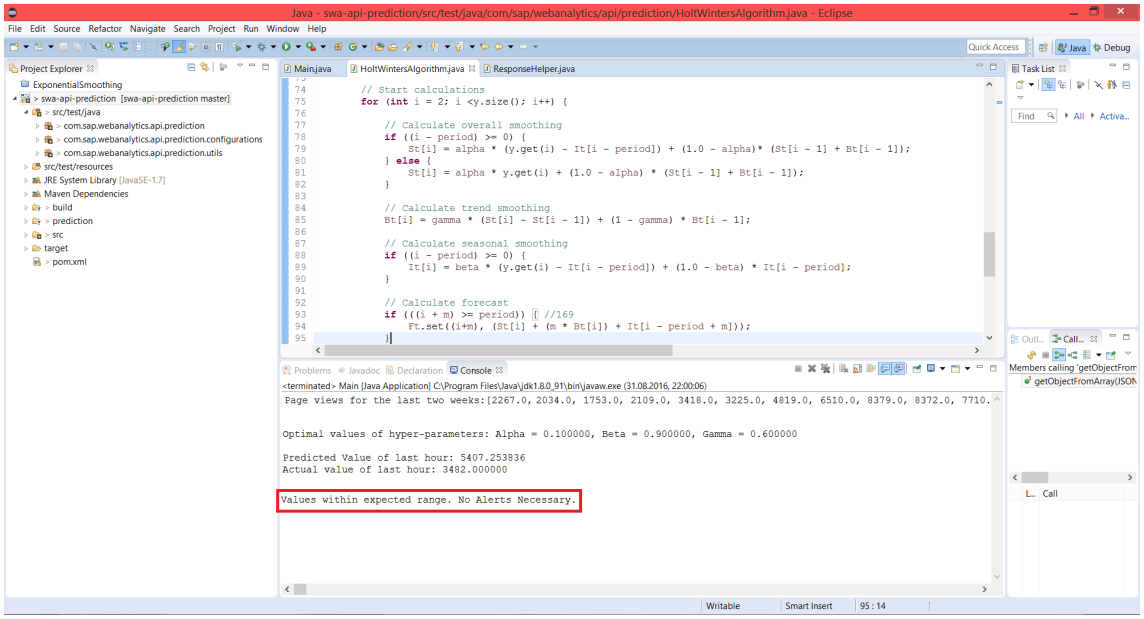

Figure 36: Java Implementation Output Screenshot

4.3.4 Thesis Solution Architecture

In conclusion, the Figure 37 displays the architecture of this thesis solution. Each box represents a class implemented in the Java implementation.

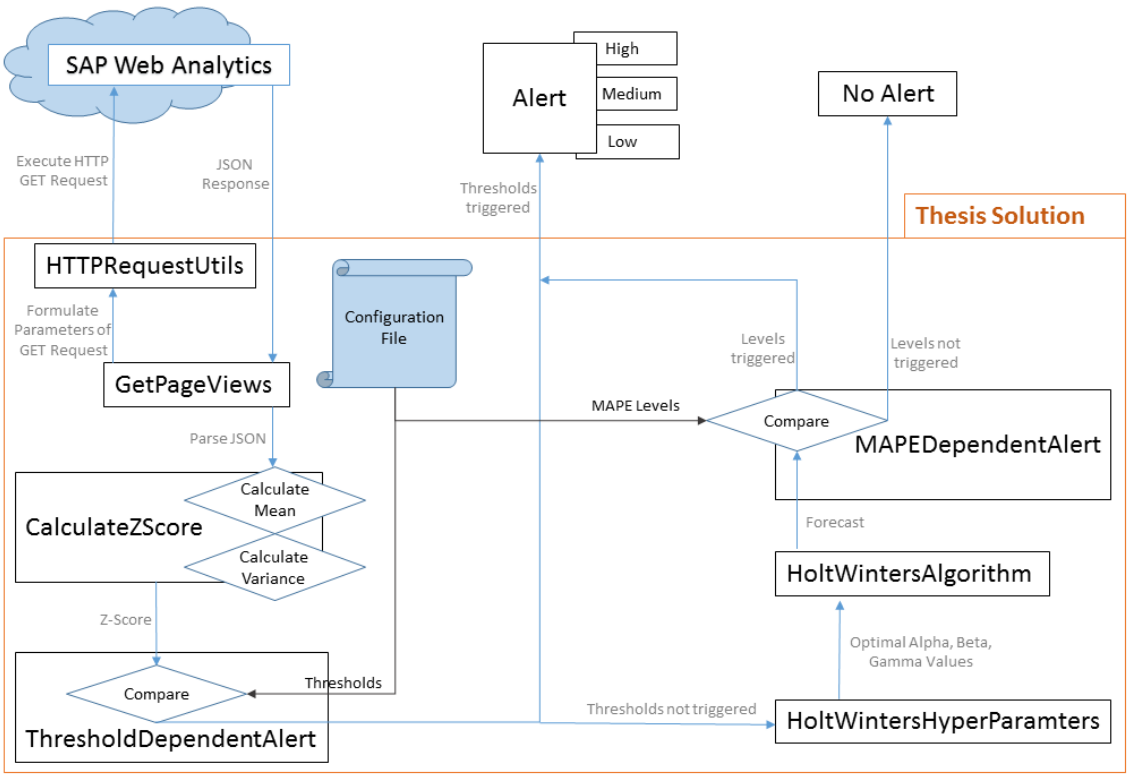

Figure 37: Thesis Solution Architecture

5 Evaluation of Thesis Solution

The previous chapter discussed the use of SWA as an example of a web analytics software to obtain real web analytics data and to test integration of the developed concept. R was used as the programming language and development environment for the initial concept development because it already had both ARIMA and Holt-Winters algorithms in its default forecast package, and was user friendly and fast while generating graphs which aided in the analysis of results. The auto.arima and HoltWinters functions were discussed, as well as the methods for automating the identification of optimal hyper-parameters of these algorithms. Implementation of the final thesis concept in Java was done because SWA's backend is in Java and whose backend is written Java and it identifies Java as the preferred language for any "extensions" to be added to SWA to allow easy integration with the existing web application stack. The Java implementation was explained which included the REST calls to the API to obtain data, processing of that data, calculation of z-score and implementation of Chebyshev inspired thresholds, calculation of the Holt-Winters hyper-parameters through cross-validation, implementation of the additive and multiplicative forms and calculation of the initial level, trend and seasonal indices. Thereafter, the MAPE was calculated by comparing the Holt-Winters forecast with the actual data value and MAPE levels were determined to identify anomalies. These were integrated with the Chebyshev inspired thresholds to form the final thesis solution - an automated anomaly detection system which issues timely and efficient alerts. An architecture diagram of the Java implementation was also displayed.

In this chapter, the R implementation is evaluated, which used page views data collected from tracking an internal SAP web page. The reason the R implementation was used for evaluation is because there is some scope for improvement in the precision of the Java implementation, such as the hyper-parameter generation function which only tests hyper-parameter values at 0.1 increment intervals. Therefore, the final hyper-parameter selection in the Java solution is not as precise as the ones that R calculates automatically as it checks hyper-parameter values in higher precision. Moreover, the R implementation is faster in executing and generating graphs and figures easily.

The specific web page whose page views are tracked is selected for evaluation in comparison to others because it displays diverse naturally occurring data patterns in the page views. This solution was first tested on nine random data points from the page views collected from one website spanning 8 months of tracked page views data. This thesis solution calculates the z-scores for detecting anomalies through Chebyshev inspired thresholds, and uses two weeks of each data points historical data to train Holt-Winters prediction algorithm. The accuracy rate of the final solution on these nine data points is then calculated. Thereafter, the impact of the amount of historical data used to train the prediction algorithm on the accuracy of the forecast is explored by training the prediction algorithm for the same nine data points with four, six and eight weeks of historical data as well. A MAPE comparison is then made of the accuracy of the forecasts when different lengths of historical data are used for training Holt-Winters. Finally, the impact of anomalies in training data is explored and a comparison is made between the MAPE of forecasts made with anomalies and without anomalies. Finally, limitations of this thesis solution will be discussed.

5.1 Evaluation Method

The season of this data remains weekly and the data granularity is hourly, i.e. the shortest data sample had a length of 336 data points, i.e. 2 seasons of hourly data. The actual value of the 337^{th} data point was manually inspected to determine whether it is an anomaly. Nine specifically chosen samples of data were collected from SWA. These samples were chosen to emphasize the importance of simultaneously using both the MAPE and Chebyshev inspired, z-score dependent thresholds as judgement criterion for issuing alerts.

The forecast value of page views for the next hour was compared to the actual page views value of the next hour to determine whether it was correctly identified, i.e. if the dataset was of two seasons (336 data points), then the actual value of the 337^{th} data point was manually determined to be an anomaly or not and thereafter was compared with the alert made with this thesis solution for the 337^{th} data point. Correct or incorrect identification of anomalies by this thesis solutions classified the alerting output as a true positive, true negative, false positive or false negative.

The minimum amount of training data required for Holt-Winters to begin making forecasts is two weeks (336 data points). It is important to first test the accuracy of forecasts on the shortest time length of a dataset so that one is aware of the performance of this solution in the 'worst case' scenario of having the least amount of training data. Therefore the nine datasets were collected over a time period of two weeks.

However, it is also critical to understand the impact of the dataset length on forecasting accuracy and to determine whether the shortest dataset length is actually the 'worst case' situation compared to longer dataset lengths, because these two answers allows the identification of situations that this solution is best suited to and its limitations. Therefore, the same nine data point forecasts which were made with only two weeks of historical data, were tested with four, six and eight weeks of historical data. The MAPE of these forecasts was compared with the MAPE of the forecasts made with two week long datasets to understand the impact of dataset length on forecasting accuracy and which dataset length is best for highest accuracy.

5.2 Acceptable Accuracy Rate, False Positives Rate and False Negatives Rate

Efficiency of the alerting system is critical for the success of this thesis concept. There is a fine balance which needs to be achieved between the sensitivity of the thresholds and the frequency of alerts. Overly sensitive thresholds can cause false positive identification of anomalies [88]. The consequent alerts will have to be manually investigated and therefore, will cause a significant increase in operational costs. On the other hand, if the thresholds are set to be very coarse, they may miss out on identifying actual anomalies or slight anomalies which could potentially upgrade to severe anomalies.

For SWA, SAP's performance requirements are:

1. Accuracy rate $\geq 99\%$
2. False negatives rate $\leq 0.05\%$
3. False positives rate $\leq 0.50\%$

5.3 Sample Datasets Collected Over Two Weeks

The nine datasets collected over the minimum possible time period of two weeks, to be used for testing the accuracy of this thesis solution, are displayed below. The 337^{th} data point within each dataset is encircled in red. The encircled data point is the actual value which will be compared to a forecast value made for the same data point using this thesis solution.

5.3.1 Dataset 1

Dataset 1, shown in Figure 38, is from the first few weeks of data collection of this web page, which was when this web page was first introduced on the SAP website. Just by observing Figure 38, there is clear instability in the data pattern, varying trend over two seasons and anomalies (in the form of spikes) on

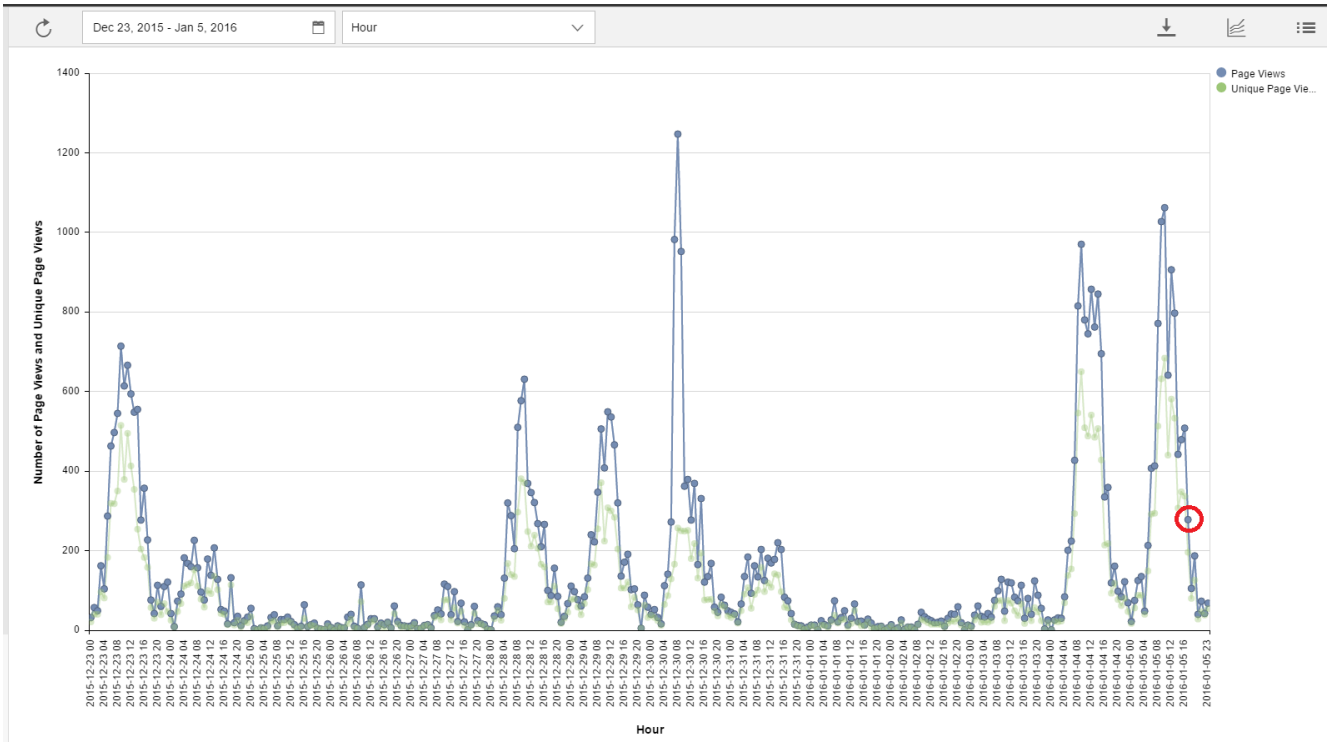

Figure 38: Sample Datasets Collected Over Two Weeks- Dataset 1

the 30th of December 2016. This irregularity in the pattern of page views is only natural as this web page is newly introduced on the website, and the trend coincides with its popularity. The anomalous spikes on the 30th of December coincide with a link that was posted on the SAP homepage which directly connects to this web page. Hence, the increased traffic over this day. The Chebyshev inspired, z-score dependent thresholds are triggered which satisfy one of the conditions for issuing an alert. Therefore this data point is recognized as a high alert anomaly despite the MAPE being under 100%.

Table 8: Forecasting And Alerting Accuracy with Two Weeks Data- Dataset 1

	Predicted Value of Page views for the Next Hour	Actual Value of Page views for the Next Hour	Classification of Actual Value as Anomaly or Normal Behaviour- Manually Identified	Classification of Actual Value as Anomaly or Normal Behaviour- As Per this Solution	Evaluation of Forecasting and Alerting Accuracy of this Solution	Absolute Percentage Error
Dataset 1	511.8099	278	Normal Behaviour	High Alert	True Positive	45.68296

This thesis solution makes a forecast for the 337th data point, which highly deviates from the actual value for that data point. Therefore, it is identified as an anomaly and an high alert is issued. The calculations are shown in Table 8. Due to the instability of the training data and that fact that this data point is part of a season where there is a steep increasing trend compared to its previous season, this data point has been correctly identified as an anomaly by this thesis solution. In this situation, when the page views pattern is changing so much, it is clearly difficult to accurately forecast the value of page views for the next hour.

5.3.2 Dataset 2

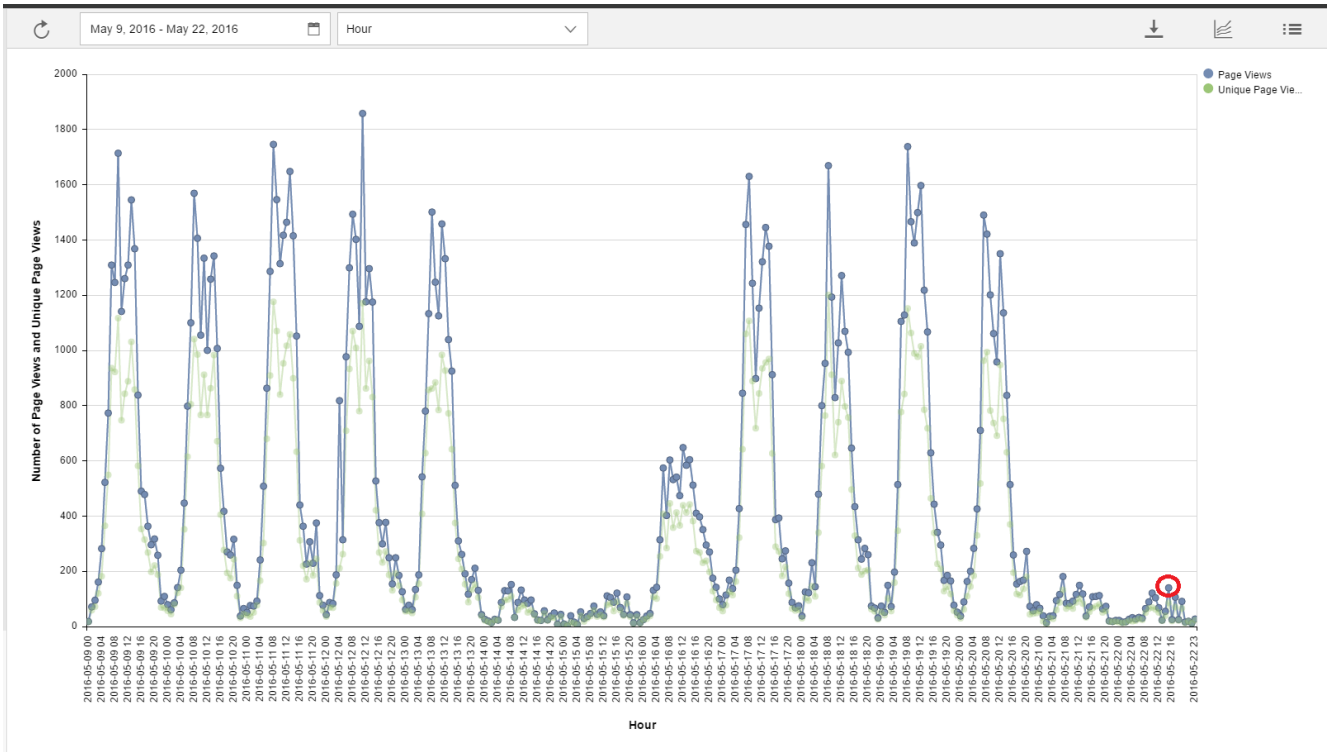

Figure 39: Sample Datasets Collected Over Two Weeks- Dataset 2

Dataset 2, shown in Figure 39 displays two weeks of data in the month of May 2016 and there is a clear distinction between the two seasons displayed on the graph. Each weekday displays high traffic and Saturday and Sunday display negligible amounts of traffic, which makes sense as this is SAP's internal web page and is usually accessed during working days, i.e. weekdays. Comparing the first days of both seasons shows that there is a drop in the traffic on the first day of the second season, i.e. on the 5^{th} of May 2016. This is because this was a public holiday in Germany, where most of the traffic originates from. This explains the drop of page views to almost half. However, this thesis solution accepts this drop in traffic as 'normal' due to the inherent assumption that the training data used is always clean and free of anomalies and abnormal behaviour. Hence, neither the Chebyshev inspired, z-score dependent thresholds are triggered and nor is the MAPE over 100%. Therefore this data point is not recognized as an anomaly.

Table 9: Forecasting And Alerting Accuracy with Two Weeks Data- Dataset 2

	Predicted Value of Page views for the Next Hour	Actual Value of Page views for the Next Hour	Classification of Actual Value as Anomaly or Normal Behaviour- Manually Identified	Classification of Actual Value as Anomaly or Normal Behaviour- As Per this Solution	Evaluation of Forecasting and Alerting Accuracy of this Solution	Absolute Percentage Error
Dataset 2	91.29641	140	Normal Behaviour	No Alert	True Negative	53.34667

The forecast is made for a data point on the 22nd of May, which is a Sunday. The page views for this data point are low when compared to the others because this data point occurs during the weekend. This thesis solution correctly identifies this data point as 'normal' behaviour and does not raise any alerts. The MAPE is high despite this data point not being an anomaly because the absolute value of the data point is quite low (in comparison to the other data point values in this dataset). So even a slight difference in the page views of the actual value from the predicted will highly impact the MAPE. The calculations are shown in Table 9.

5.3.3 Dataset 3

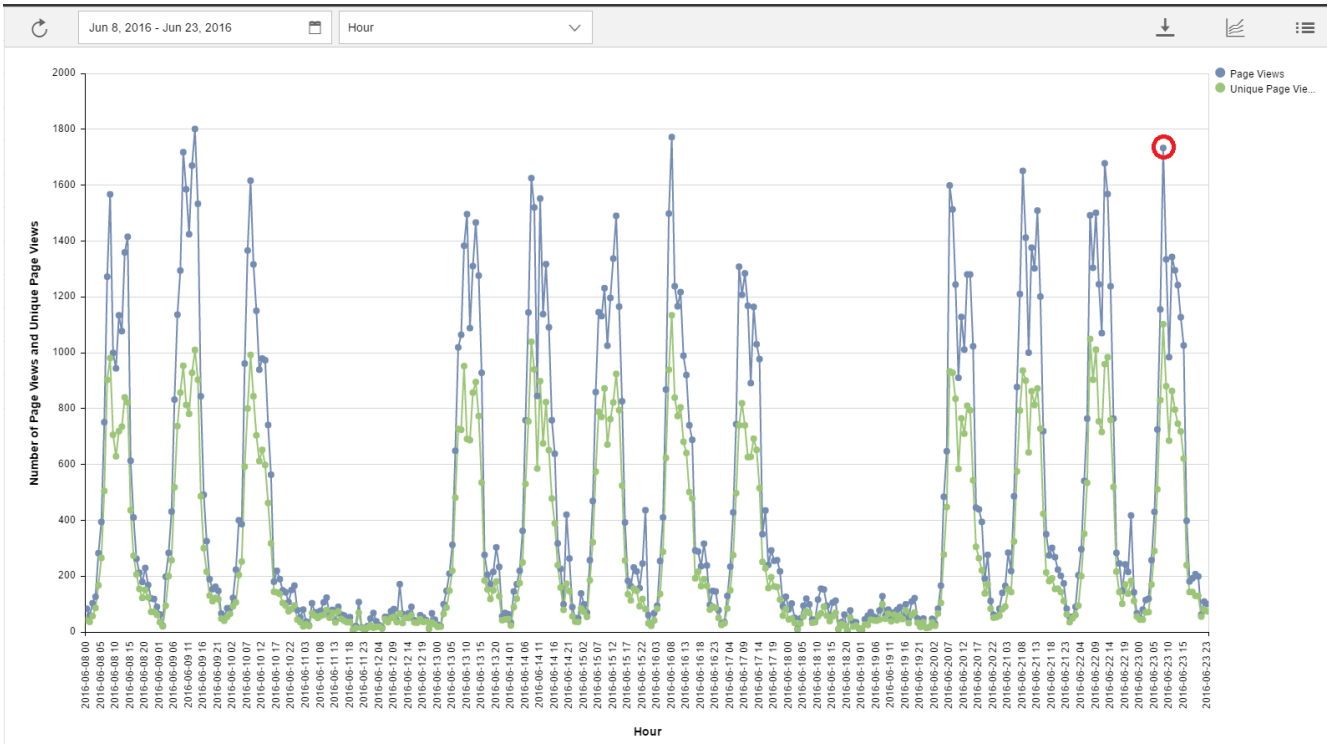

Figure 40: Sample Datasets Collected Over Two Weeks- Dataset 3

Dataset 3, shown in Figure 40, displays two weeks of data in the month of June 2016. When the first season is observed, the fourth day of the season, i.e. Thursday, is clearly higher than the peaks of the rest of the working days within this season. As this is part of the training data, which is assumed to be clean, this value is also assumed to be clean. When this thesis solution makes a forecast for the page views for this data point in the next season, the value for the Thursday is expected to be higher than the values for the rest of the working days. This is why the forecast made is higher than the actual number of page views. The resulting 'low alert' can mean either of two things- first, that the Thursday values from the first season are actually anomalies and were incorrectly fed into the training data as normal, which caused an incorrect alert to be issued. Second, that a slight rise in page views every Thursday is normal and expected and because this rise did not occur in the second season, this data point has correctly been identified as a low level alert. For the purposes of consistency within this thesis, it is assumed that the training set is clean and that there is supposed to be a rise in page views every Thursday compared to other working days. Therefore, this thesis solution correctly issues an alert. The calculations are shown in Table 10.

Table 10: Forecasting And Alerting Accuracy with Two Weeks Data- Dataset 3

	Predicted Value of Page views for the Next Hour	Actual Value of Page views for the Next Hour	Classification of Actual Value as Anomaly or Normal Behaviour- Manually Identified	Classification of Actual Value as Anomaly or Normal Behaviour- As Per this Solution	Evaluation of Forecasting and Alerting Accuracy of this Solution	Absolute Percentage Error
Dataset 3	1784.597	1733	Anomaly	Low Alert	True Positive	2.891226

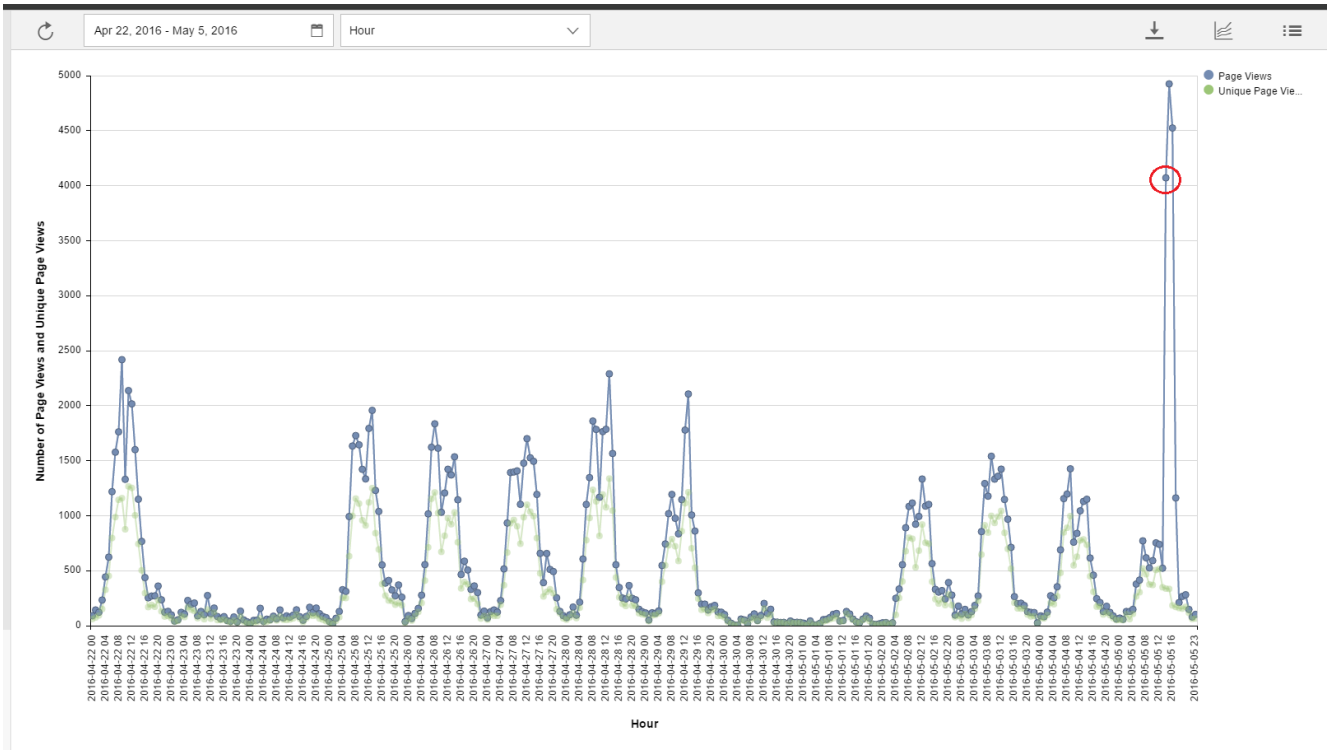

Figure 41: Sample Datasets Collected Over Two Weeks- Dataset 4

5.3.4 Dataset 4

Dataset 4, shown in Figure 41, displays two weeks of data between the months of April and May 2016. Observing just this two-week window of data, there appears to be a slight decreasing trend from one season to the next. The encircled data point in red is a very clear anomaly. Along with the two neighbouring data points, these spikes are not a part of the regular page views pattern according to the training data fed into the prediction algorithm. This data point is correctly identified as a high alert anomaly which can also be seen through the very large MAPE. The calculations are shown in Table 11.

5.3.5 Dataset 5

Dataset 5, shown in Figure 42, displays two weeks of data between the months of January and February 2016. Observing just this two-week window of data, the data values every season for the second day, i.e. Tuesday, are lower on average and display a decreasing trend. Therefore, the predicted value is much

Table 11: Forecasting And Alerting Accuracy with Two Weeks Data- Dataset 4

	Predicted Value of Page views for the Next Hour	Actual Value of Page views for the Next Hour	Classification of Actual Value as Anomaly or Normal Behaviour- Manually Identified	Classification of Actual Value as Anomaly or Normal Behaviour- As Per this Solution	Evaluation of Forecasting and Alerting Accuracy of this Solution	Absolute Percentage Error
Dataset 4	8.538859	4071	Anomaly	High Alert	True Positive	47576.16

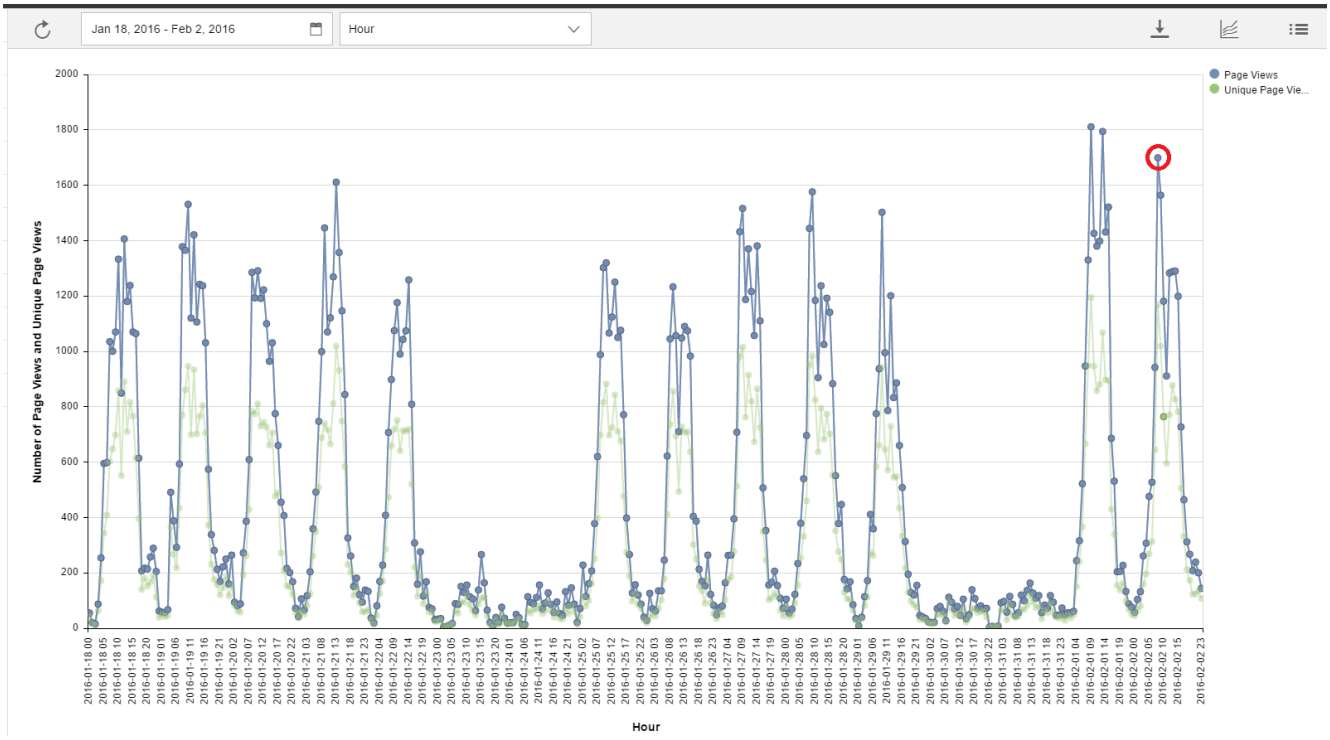

Figure 42: Sample Datasets Collected Over Two Weeks- Dataset 5

lower than the actual value and this data point is correctly identified as an anomaly with a high alert.The calculations are shown in Table 12.

Table 12: Forecasting And Alerting Accuracy with Two Weeks Data- Dataset 5

	Predicted Value of Page views for the Next Hour	Actual Value of Page views for the Next Hour	Classification of Actual Value as Anomaly or Normal Behaviour- Manually Identified	Classification of Actual Value as Anomaly or Normal Behaviour- As Per this Solution	Evaluation of Forecasting and Alerting Accuracy of this Solution	Absolute Percentage Error
Dataset 5	1344.746	1699	Anomaly	high level	True Positive	26.34353

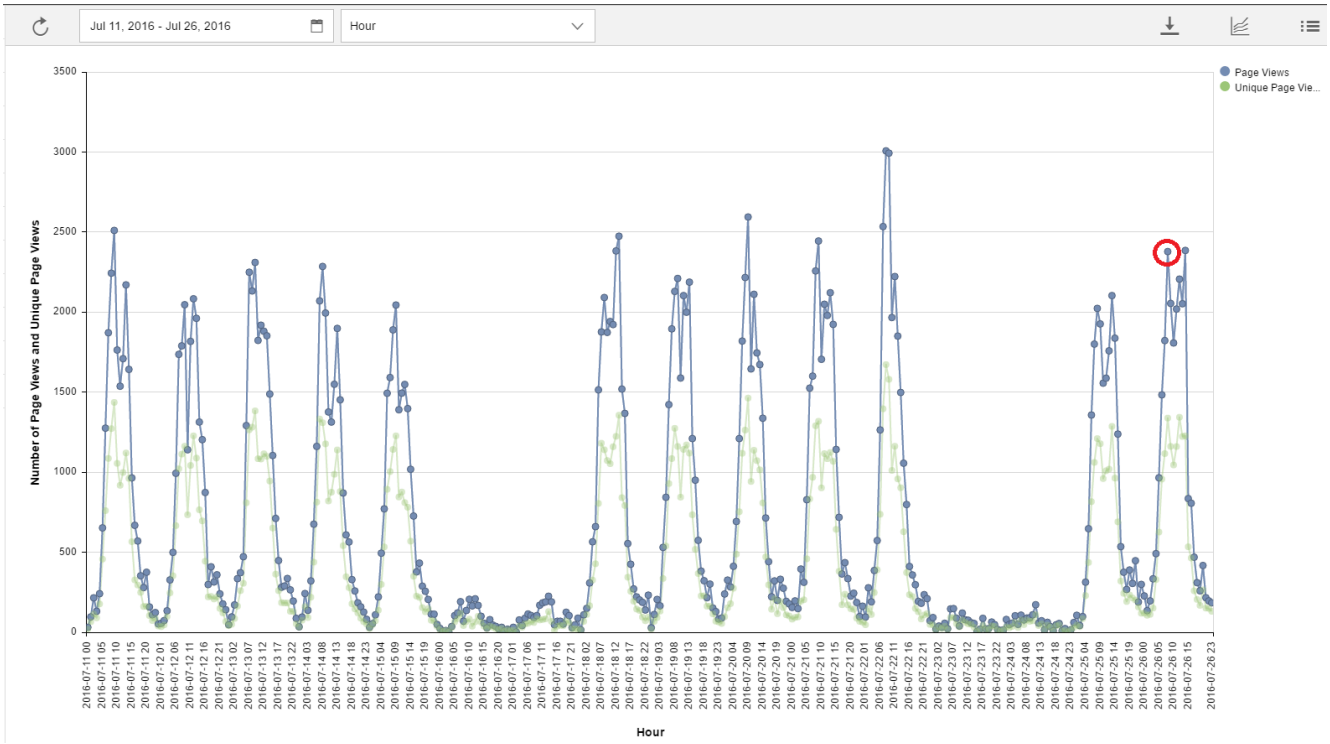

Figure 43: Sample Datasets Collected Over Two Weeks- Dataset 6

5.3.6 Dataset 6

Dataset 6, shown in Figure 43, displays two weeks of data for the month of July 2016. Only visually observing this data graph would make it difficult to identify any anomalous behaviour with a brief glance. However, this dataset is a perfect example of why there is a need to automate this anomaly detection and alerting process. On careful inspection, it can be observed that within every season, the data values for the second day of the season are lower than the first. For the data point that was forecast, the prediction was made so that it would also be lower than the values of the first day of its season. However, the actual value was higher. So despite the low MAPE of around 10%, this data point is correctly identified as a medium level anomaly. The calculations are shown in Table 13.

Table 13: Forecasting And Alerting Accuracy with Two Weeks Data- Dataset 6

	Predicted Value of Page views for the Next Hour	Actual Value of Page views for the Next Hour	Classification of Actual Value as Anomaly or Normal Behaviour- Manually Identified	Classification of Actual Value as Anomaly or Normal Behaviour- As Per this Solution	Evaluation of Forecasting and Alerting Accuracy of this Solution	Absolute Percentage Error
Dataset 6	2154.387	2378	Anomaly	Medium Alert	True Positive	10.37943

5.3.7 Dataset 7

Dataset 7, shown in Figure 44, displays two weeks of data for the months of May and June 2016. Observing the graph shows several data points which could be anomalies. However, keeping with the

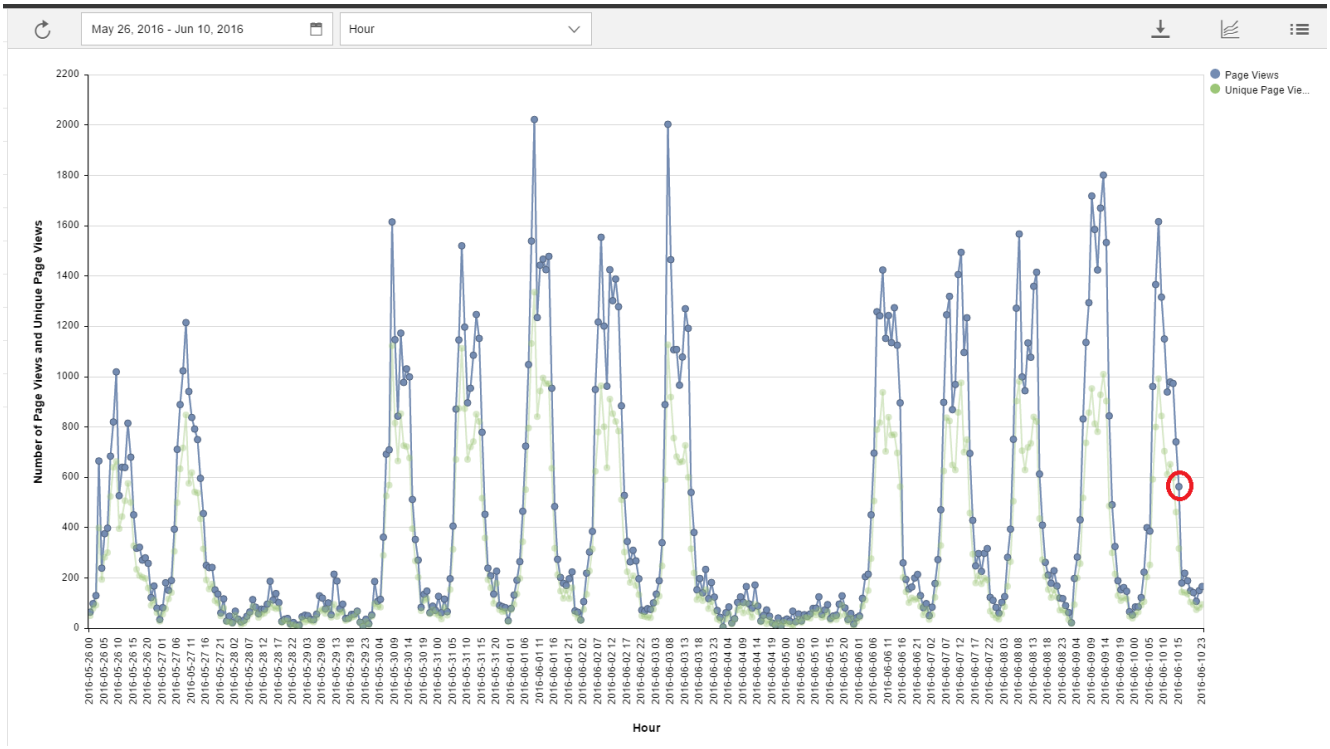

Figure 44: Sample Datasets Collected Over Two Weeks- Dataset 7

assumption that the training data is clean, these data points are also assumed to be clean and normal. The prediction made for the data point keeps with the page views pattern within this two week window. The forecast value is very close to the actual value and therefore, the data point is not identified as an anomaly. The calculations are shown in Table 14.

Table 14: Forecasting And Alerting Accuracy with Two Weeks Data- Dataset 7

	Predicted Value of Page views for the Next Hour	Actual Value of Page views for the Next Hour	Classification of Actual Value as Anomaly or Normal Behaviour- Manually Identified	Classification of Actual Value as Anomaly or Normal Behaviour- As Per this Solution	Evaluation of Forecasting and Alerting Accuracy of this Solution	Absolute Percentage Error
Dataset 7	539.0523	563	Normal Behaviour	No Alert	True Negative	4.442555

5.3.8 Dataset 8

Dataset 8, shown in Figure 45, displays two weeks of data for the months of March and April 2016. There is an obvious decrease in the number of page views on 28th March 2016, Monday of the first season, because it was another public holiday. However, the prediction made for the last day of the season, Sunday, is not influenced by this change in page views pattern because the prediction being made is for a data point on Sunday.

This is a very important example as it displays the need for this thesis solution to use both Chebyshev inspired, z-score dependent thresholds and MAPE calculations to make an alert. For this exact dataset,

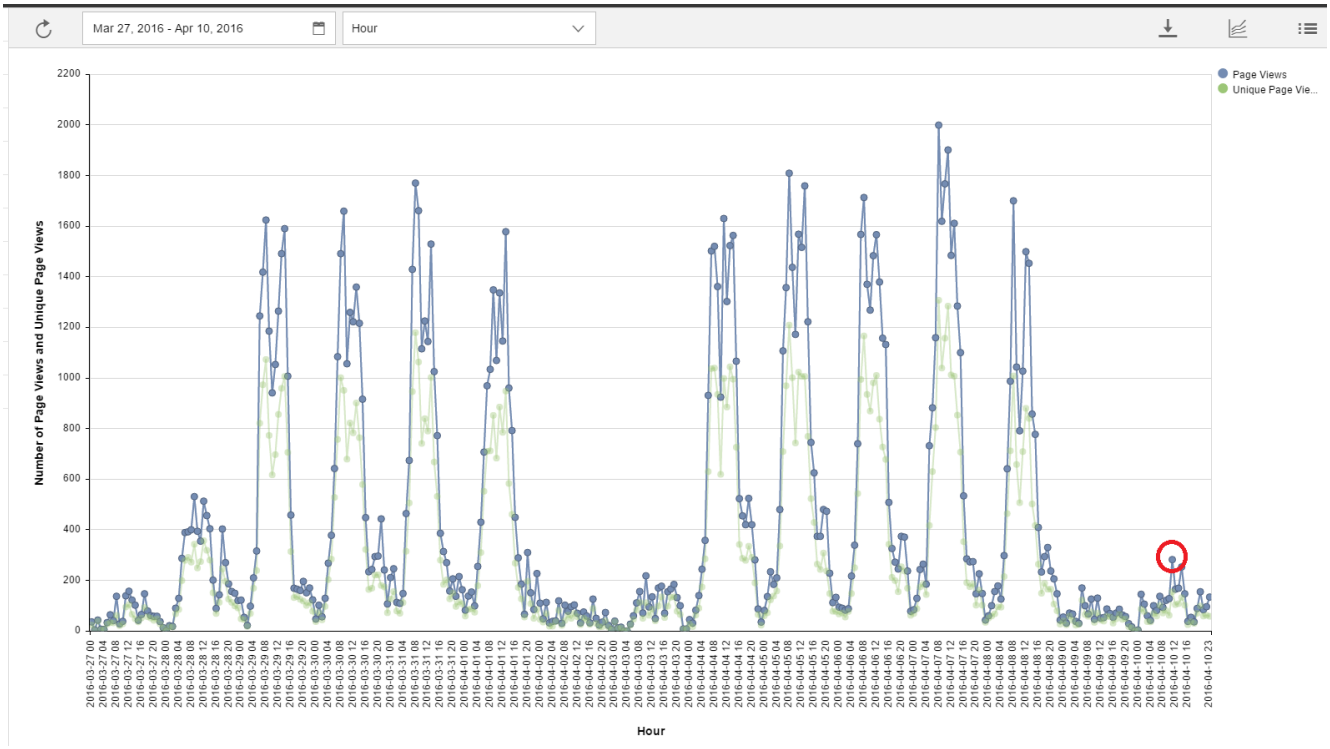

Figure 45: Sample Datasets Collected Over Two Weeks- Dataset 8

when only standard deviation is used to identify whether this data point is an anomaly or not, this data point is not identified as a anomaly! This is because despite the fact that the data points value is more than it should be for a Sunday afternoon, it falls well within the distribution of 3 standard deviations from the mean. So it is not recognized as a value that should raise any alarm. However, since this data is a times series in which the data point values are highly dependent on the time of their occurrence, it is important to address that at this time on the time series, the value for this data point should be lower. Therefore, when using only z-score dependent thresholds would have failed to recognize this data point as an anomaly, using the MAPE allowed the correct identification of this point as an anomaly. The calculations are shown in Table 15.

Table 15: Forecasting And Alerting Accuracy with Two Weeks Data- Dataset 8

	Predicted Value of Page views for the Next Hour	Actual Value of Page views for the Next Hour	Classification of Actual Value as Anomaly or Normal Behaviour- Manually Identified	Classification of Actual Value as Anomaly or Normal Behaviour- As Per this Solution	Evaluation of Forecasting and Alerting Accuracy of this Solution	Absolute Percentage Error
Dataset 8	117.5493	282	Normal Behaviour	Low Alert	True Positive	139.8994

5.3.9 Dataset 9

Dataset 9, shown in Figure 46, displays two weeks of data for the months of June and July 2016. It can be observed from the graph that clearly the season for which the data value is forecast is higher than the values of the previous seasons. Therefore, the prediction made on the basis of the two weeks of historical

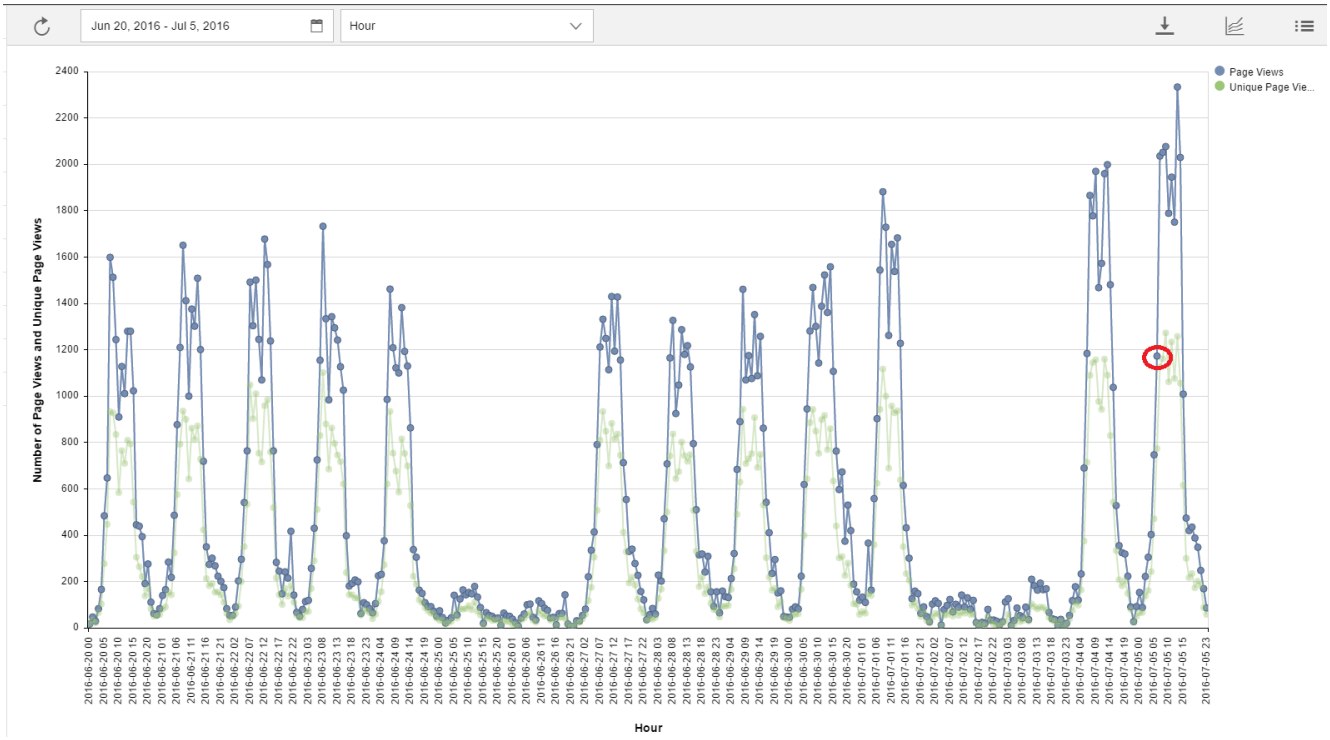

Figure 46: Sample Datasets Collected Over Two Weeks- Dataset 9

data is lower than the actual value of the data point. Despite the MAPE being below the 100% which is required to trigger an alert, the other condition for issuing an alert is met. The Chebyshev inspired, z-score dependent thresholds are triggered and this data point is recognized as a medium alert anomaly. The calculations are shown in Table 16.

Table 16: Forecasting And Alerting Accuracy with Two Weeks Data- Dataset 9

	Predicted Value of Page views for the Next Hour	Actual Value of Page views for the Next Hour	Classification of Actual Value as Anomaly or Normal Behaviour- Manually Identified	Classification of Actual Value as Anomaly or Normal Behaviour- As Per this Solution	Evaluation of Forecasting and Alerting Accuracy of this Solution	Absolute Percentage Error
Dataset 9	945.2566	1173	Anomaly	Medium Alert	True Positive	24.09329

5.4 Forecasting Accuracy for Datasets Collected Over Two Weeks

The testing of this thesis solution on nine different datasets of length two weeks showed good results. Seven of the datasets were true positives and identified anomalies correctly, and two of the datasets were true negatives and identified normal behaviour correctly. Despite these promising forecasting accuracy results of using only two weeks of training data, it is critical to understand the impact of the dataset length on forecasting accuracy and explore different lengths of datasets. Moreover, it is also essential to determine the best length of training data to achieve highest possible forecasting accuracy.

To address these two requirements, the same nine data point forecasts which were made with only two weeks of historical data, were tested with four, six and eight weeks of historical data. The graphs showing

the data points marked against (up to) eight weeks of historical data are shown in the Appendices. The MAPE of each forecast was compared with the MAPE of the forecasts made with only two week long training data to understand the impact of dataset length on forecasting accuracy and which dataset length is best for highest accuracy.

Detailed results discussed for each dataset can be found in Figure 19 in the Appendix. The false negatives rate, false positives rate and accuracy rate are calculated below.

The false negatives rate is calculated below.

$$\text{False Negatives Rate} = \frac{0}{6+3+0+0} \times 100 = 0\%$$

The false positives rate is calculated below.

$$\text{False Positives Rate} = \frac{0}{6+3+0+0} \times 100 = 0\%$$

The accuracy rate is calculated below.

$$\text{Accuracy Rate} = \frac{6+3}{6+3+0+0} \times 100 = 100\%$$

The accuracy rate is 100%, the false positives rate is 0% and the false negatives rate is 0%. Thus, SAP's requirements for accuracy rate, false positives rate and false negatives rate are met.

5.5 Forecasting Accuracy for Datasets Collected Over Four Weeks

As Dataset 1 and Dataset 5 were collected in the initial weeks of data collection for this web page, it was not possible to get four weeks worth of historical data for them.

Out of the seven data points, only five data points were correctly identified. The remaining two data points issued alerts incorrectly not because of the thresholds, but because the MAPE was over 100% in both cases and hence, the second condition necessary for triggering alerts was met. They were both False Positives.

The increase in MAPE is due to the inaccuracy of the forecast. In the first case, Dataset 2, observing the four week graph clearly shows that three very large anomalies are included in the second season of the dataset. These anomalies are accepted as normal data under the assumption of a clean training set and highly influence the forecast being made, thus explaining the inaccurate prediction. Similarly, in the second case, Dataset 8, there is also an anomaly in the second season which is accepted as normal data under the inherent assumption of a clean training dataset. Therefore, this comparison highlights the importance of a clean training set and its obvious negative impact on forecasting. Having extra false positives would require manual investigation of these anomalies, which would use resources and increase operation costs unnecessarily.

It is interesting to observe that Dataset 3 is identified as Medium Alert and has a greater MAPE when the historical data used is four weeks instead of being identified as low alert when the historical data was two weeks. Similarly, Dataset 9 issued a low alert with four weeks of historical data in comparison to the Medium alert which was issued with two weeks of historical data despite close MAPE's.

Detailed results discussed for each dataset can be found in Figure 20 in the Appendix. The false negatives rate, false positives rate and accuracy rate are calculated below.

The false negatives rate is calculated below.

$$\text{False Negatives Rate} = \frac{0}{4+1+2+0} \times 100 = 0\%$$

The false positives rate is calculated below.

$$\text{False Positives Rate} = \frac{2}{4+1+2+0} \times 100 = 28.57\%$$

The accuracy rate is calculated below.

$$\text{Accuracy Rate} = \frac{4+1}{4+1+2+0} \times 100 = 71.43\%$$

The accuracy rate is 71.43%, the false positives rate is 28.57% and the false negatives rate is 0%. SAP's requirements for accuracy rate and false positives rate are not met. However, the requirements for false negatives rate are met.

5.6 Forecasting Accuracy for Datasets Collected Over Six Weeks

As Dataset 1 and Dataset 5 were collected in the initial weeks of data collection for this web page, it was not possible to get six weeks worth of historical data for them.

Out of the seven data points, only four data points were correctly identified. The accuracy of forecasting is worse than using two weeks or four weeks of historical data to make forecasts. However, what is particularly worrying is the lack of an alert for Dataset 9, i.e. a false negative. Over-alerting can be costly, but under-alerting might let serious anomalies pass which could cause problems and also defeat the purpose of this entire thesis.

Referencing the explanation in the previous section, the first and second case of inaccurate anomaly identification, i.e. Dataset 2 and Dataset 8 respectively, are attributed to the anomalies in the training dataset which negatively impact the forecasts made. Similarly, in the third case of Dataset 9, observing the graph displays large anomalies in the second season of the training dataset. Again, these anomalies clearly have a large negative influence on the forecast made.

Using six weeks of historical data clearly shows poor accuracy as the larger the historical dataset becomes, the more important it is to ensure that it is clean. The presence of anomalies highly impacts the forecasts made.

Detailed results discussed for each dataset can be found in Figure 21 in the Appendix. The false negatives rate, false positives rate and accuracy rate are calculated below.

The false negatives rate is calculated below.

$$\text{False Negatives Rate} = \frac{1}{3+1+2+1} \times 100 = 14.29\%$$

The false positives rate is calculated below.

$$\text{False Positives Rate} = \frac{2}{3+1+2+1} \times 100 = 28.57\%$$

The accuracy rate is calculated below.

$$\text{Accuracy Rate} = \frac{3+1}{3+1+2+1} \times 100 = 57.14\%$$

The accuracy rate is 57.14%, the false positives rate is 28.57% and the false negatives rate is 14.29%. SAP's requirements for accuracy rate, false positives rate and false negatives rate are not met.

5.7 Forecasting Accuracy for Datasets Collected Over Eight Weeks

As Dataset 1 and Dataset 5 were collected in the initial weeks of data collection for this web page, it was not possible to get eight weeks worth of historical data for them.

Out of the seven data points, five data points were correctly identified. The accuracy of forecasting is worse than using two weeks of historical data to make forecasts but better than using four weeks or six weeks of historical data. The false positive identification of an anomaly in Dataset 2 will require manual investigation and hence, will utilize unnecessary resources and increase operation costs. However, even more worrying is the lack of an alert for Dataset 9, i.e. a false negative. Allowing an actual anomaly to pass could lead to problems and also defeat the purpose of this entire thesis.

Referencing the explanation in the previous sections, the first case of inaccurate anomaly identification, i.e. Dataset 2, is attributed to the anomalies in the training dataset which negatively impact the forecasts made. Similarly, in the second case, i.e. Dataset 9, observing the graph displays large anomalies in the second season of the training dataset. Clearly the presence of anomalies within the training set has a large negative influence on the forecast being made.

Detailed results discussed for each dataset can be found in Figure 22 in the Appendix. The false negatives rate, false positives rate and accuracy rate are calculated below.

The false negatives rate is calculated below.

$$\text{False Negatives Rate} = \frac{1}{3+2+1+1} \times 100 = 14.29\%$$

The false positives rate is calculated below.

$$\text{False Positives Rate} = \frac{1}{3+2+1+1} \times 100 = 14.29\%$$

The accuracy rate is calculated below.

$$\text{Accuracy Rate} = \frac{3+2}{3+2+1+1} \times 100 = 71.43\%$$

The accuracy rate is 71.43%, the false positives rate is 14.29% and the false negatives rate is 14.29%. SAP's requirements for accuracy rate, false positives rate and false negatives rate are not met.

5.8 MAPE Comparison of Forecasting Accuracy

Out of the nine specifically chosen datasets that this evaluation is performed on, only three of the datasets are used to make a forecast for a non-anomalous data point. Hence, these are the datasets whose MAPE needs to be discussed to understand the accuracy of forecasting made when the data point to be predict is a 'normal' value.

The table below shows the MAPE vales of Datasets 2, 7 and 8, which were the non-anomalous datasets. The average MAPE is calculated for each dataset and then the error in forecasts made with diverse lengths of historical data is compared.

Clearly, the best average MAPE is from the datasets which only used two weeks of historical data to train the prediction algorithm before making a forecast. However, a very important point to note is that

Table 17: MAPE Comparison of Data Collected for Different Time Lengths With Anomalies in the Training Data

	MAPE in Forecast			
	2 Weeks	**4 Weeks**	**6 Weeks**	**8 Weeks**
Dataset 2	53.3467	1560.834	421.6488	143.6615
Dataset 7	4.4426	49.3441	61.7258	254.6147
Dataset 8	139.899	163.1127	104.2188	52.4256
Average MAPE	65.8962	591.0969	195.8645	150.2339

in all three of these datasets, when only two weeks of training data is used, there are no anomalies included in that window. In contrast, the four, six and eight weeks of historical data include anomalies in the training data while making forecasts.

To make a fair MAPE comparison, Datasets 2, 7 and 8 would have to be cleaned of anomalies and then used to make forecasts with four, six and eight weeks of historical data.

5.9 MAPE Comparison of Forecasting Accuracy After Removing Anomalies in the Training Data

As highlighted in the previous section, making a fair MAPE comparison requires that all the datasets are clean and anomaly-free. In this section, the anomalies will be removed from Datasets 2, 7 and 8 and replaced with their predicted values (generated from using two weeks worth of data points before them as training data).

The MAPE results for this fair comparison are shown below.

Table 18: MAPE Comparison of Data Collected for Different Time Lengths Without Anomalies in the Training Data

	MAPE in Forecast			
	2 Weeks	**4 Weeks**	**6 Weeks**	**8 Weeks**
Dataset 2	53.34667	562.8047	904.4592	199.0241
Dataset 7	4.4426	49.3441	68.0024	154.3929
Dataset 8	139.8994	132.9365	109.3309	61.0886
Average MAPE	65.8962	248.3617	360.5975	138.1686

5.9.1 Influence of Time Length of Collection Data On MAPE

After removing the anomalies from the data used for training the prediction algorithm, the best accuracy of forecasting is still from using only two weeks of historical data to train the prediction algorithm. When forecasts were made for datasets 2, 7 and 9 using four weeks of anomaly-free historical data, the MAPE was 248.36%, almost half of the MAPE when the four weeks of training data had anomalies. Similarly, eight weeks of historical data also showed more than 10% MAPE improvement when the anomalies were removed from the dataset. However, for six weeks of training data, the MAPE increased by over 150% when the anomalies were removed from the dataset. Because of the different impact of removing anomalies in the datasets, it becomes impossible to generalize whether removing anomalies from the dataset is advantageous or not.

The highest MAPE accuracy is when only two weeks of training data is used compared to longer historical training datasets, and one possible explanation could be that the popularity of a web page seldom remains constant. There are several situations that influence the page views patterns of a web

page such as new advertising campaigns, web page mention by a celebrity or the release of a competitor web page. Since such actions reflect immediately in the page views, using very old historical data for training prediction algorithms can be more of a hindrance than help to the prediction accuracy. Through this evaluation chapter and the extensive comparisons made, it is obvious that for this use case, the best forecasting results are obtained when the most recent data is used for training the prediction algorithm, i.e. only two weeks of historical data.

5.10 Limitations of this Thesis Solution

The quality of training data directly limits the performance of any machine learning algorithm or statistical tool. Therefore, the first limitation of this thesis solution is the inherent assumption that only clean data, i.e. data that is free of any anomalies or irregular behaviour, is used to train the prediction algorithm. Evaluation results suggest that if the data used for training the prediction algorithm has anomalies, the accuracy of the forecasts made is negatively impacted. The extent of this impact was beyond the scope of this thesis.

There is a minimum Holt-Winters training period. The algorithm requires a minimum of two seasons of historical training data to make a forecast and this is a limitation for the final thesis solution which cannot be executed completely until two seasons have passed so that Holt-Winters can make predictions. Until these two seasons of historic data is collected, the anomaly detection system needs to solely rely on the z-score, which can be disadvantageous in situations where an data point is not distant enough from the mean in standard deviations to trigger thresholds, but is in fact an anomaly.

While diverse page views datasets were used for the concept development, testing and evaluation of this thesis solution, another limitation is that all the page views data is from a single web page. Therefore, all the generalized conclusions that were drawn through the different data points in the evaluation are limited by the fact that their data comes from the same source.

6 Conclusion and Future Work

This aim of this thesis was to develop an automated method of anomaly detection in typical web analytics data that is able to accurately and efficiently detect anomalies, report them through alerts in a timely manner and be easily incorporated in typical web analytics solutions.

Out of the various features within web analytics data, some are considered to be universally important and interesting to common users. One such feature is the number of 'page views' that a web page receives, and was selected as the feature to illustrate the concept of this thesis. Because the data collected about page views is time series, several different methods of detecting anomalies within time series and the advantages and disadvantages of each were explored during the literature survey. As the page views data displays non-Gaussian distribution and page views is not an independent feature, Chebyshev's inequality theorem was concluded to be best suited for anomaly detection for this type of data. Chebyshev's inequality theorem calculates the standard deviation of the data and puts bounds on the percentage of data points which can exist within a certain number of standard deviations from the mean. In this thesis solution, thresholds were initially configured to match Chebyshev's inequality theorem, but can be manually configured to suit the SWA use case and prevent over or under alerting. Low level anomalies were identified as data points which were a distance of over 2.5 standard deviations from the mean. Medium level anomalies were identified as data points which were a distance of over 2.8 standard deviations from the mean. High level anomalies were identified as data points which were a distance of over 3.2 standard deviations from the mean.

However, as the data point values for page views are strongly correlated to the time when they occur, relying just on standard deviation to identify anomalies can become an inaccurate solution. Consider the following situation - if the web page being tracked has extremely low number of page views during weekends, even if one of these values is twofold or threefold it would still be less than the number of page views during weekdays. Therefore, it is possible that this data point would fall within an acceptable number of standard deviations from the mean and be considered a non-anomalous value. However, this would be misidentification because if the value is twofold or threefold the expectation for that data point, then it is obviously not displaying normal behaviour and should be recognized as an anomaly.

To handle such situations, it was concluded that the MAPE between the expected number of page views and the actual number of page views could be used as a contributing factor in anomaly detection. The expected number of page views is calculated using a prediction algorithm and during the literature survey, several prediction algorithms and the advantages and disadvantages of each were discussed. As the univariate time series data displays both trend and seasonality, and the forecast being made was for a short term, i.e. one period or less, Holt-Winters algorithm showed the best prediction accuracy while forecasting page views for the next hour for different pattern types of datasets. The forecast made for the next hour was then compared with the actual value that occurred in that hour to calculate the MAPE. If the actual value of a data point varied from the forecast by an average MAPE of over 100% then the actual data point was identified as a low level. Similarly if the average MAPE was over 150% then the data point was identified as a medium level anomaly or if the average MAPE was over 200%, then the data point was identified as a high level anomaly. These MAPE levels are the default values and work well with the current data from SWA, but these can be configured by users to suit their specific use case.

The final solution that was implemented relies on both Chebyshev inspired, z-score dependent thresholds and MAPE to automatically identify a data point as an anomaly and issue an alert. If the standard deviation of a data point is not far enough from the mean to trigger a threshold, the MAPE for that data point is calculated by comparing the actual data point value with its forecast value. If the MAPE is higher than a user defined acceptable percentage, then that data point will be identified as anomaly and an alert will be issued, otherwise the data point will not be recognized as an anomaly. This solution, when

implemented in Java and run on a standard laptop (Windows 8 operating system, Intel(R) Core(TM) i7-4900MQ CPU @ 2.80GHz, 16.0 GM RAM), required 5.85 seconds to complete execution and issue an anomaly and was easily integrated with SWA, the example web analytics software used to demonstrate this solution. Although the Java implementation is just a small prototype without much optimization, the execution time that was already achieved makes it very likely that the solution can be integrated in a real web analytics application and perform continuous checks of web analytics data in the background (on a much more powerful server machine), i.e. this thesis solution should in principle enable timely reactions based on detected events.

This solution was first tested on nine random data points from the page views collected from one website spanning 8 months of tracked page views data. The page views data was collected from a real SAP internal web page. Two weeks of each data points historical data was used to train the prediction algorithm. Two weeks of historical data is the least amount of data which the Holt-Winters prediction algorithm requires to start making forecasts. Hence, the accuracy of this thesis solution was tested for potentially the worst case situation, where only two weeks of data would be available. In each of the nine cases with two weeks of historical data, the outcome was correctly identified, indicating 100% statistical accuracy, which is calculated based on the number of true positives, true negatives, false positives and false negatives. Out of the nine data points tested, six data points were anomalies. Some of these anomalies were obvious and could be easily identified just my manual observation of the graphed data and others appeared more subtle in the graphed data and could potentially have been missed without careful manual inspection and analysis. Moreover one of the anomalous data points had a value within the acceptable number of standard deviations from the mean, and therefore did not trigger any z-score dependent thresholds, but was identified as anomaly due to its large average MAPE value which surpassed the acceptable percentage.

Thereafter the impact of the amount of historical data used to train the prediction algorithm on the accuracy of the forecast was explored. Forecasts were made for the same nine aforementioned data points by expanding the historical data used to train the prediction algorithms to four, six and then eight weeks. However, two of these data points did not have enough historical data for these tests. Therefore, comparisons were made between the MAPE for the remaining seven data points when four, six and then eight weeks of historical data were used for making their forecasts. Surprisingly, the highest accuracy and lowest MAPE of forecasting was obtained when only two weeks of historical data was used to train the prediction algorithm with the accuracy rate being 100% and average MAPE being 65.89%. Moreover, the false positives rate and the false negatives rate were both 0%. Hence, all of the accuracy requirements were satisfied. When four weeks of historical data was used, the accuracy rate dropped to 71.43% and there was an extremely high jump in MAPE, rising to an astonishing 591.09%. Even though the false negatives rate was 0% and met the requirements, the false positives rate was 28.57%. Overall, using four weeks of historical data did not satisfy the requirements. When six weeks of historical data was used, the accuracy rate dropped further to 57.14%, but there was a significantly lower MAPE of 195.87% from the forecasts made with four weeks of historical data. The false negatives rate was 14.29% and the false positives rate was 28.57%. Overall, using six weeks of historical data did not satisfy the requirements. When forecasts were made using eight weeks of historical data, the accuracy improved to 71.43% and the MAPE continued to reduce to 150.24%. The false positives rate and false negatives rate were both 14.29%. However, even though the overall accuracy of forecasts made with eight weeks of historical data is better than forecasts made with four or six weeks of historical data, SAP's requirements were still not met. Therefore, forecasts made with two weeks of historical data fared best. However, the two weeks of historical data for each data point did not contain anomalies. In contrast, the four, six and eight weeks of historical data used for the forecasting of the same data points did contain (sometimes several) anomalies.

Further tests were conducted to determine whether this discrepancy in MAPE between forecasts made for two, four, six and eight weeks of data was due to the presence of anomalies in only some of the training datasets. The anomalies within the historical data of these data points were manually removed. When forecasts were made after removing the anomalies from historical data of four, six and eight weeks, the MAPE improved for forecasts made with four and eight weeks of historical data but did not improve for forecasts made with six weeks of historical data. Since the removal of anomalies in the training sets improved the MAPE of two datasets but negatively impacted one dataset, these results prove inconclusive in determining any correlation between the removal of anomalies from the training datasets and the forecasting accuracy. If anomalies are to be removed from the training dataset, it must be ensured that such data point values will continue to be anomalies in the future seasons as well, which is impossible to do. Removing an anomaly that is the first occurrence of a new pattern can be more of a hindrance than help in bettering forecasting accuracy.

6.1 Future Work

The results obtained through this thesis are promising but there are areas that could be further improved in this solution.

Firstly, the Java implementation of the Holt-Winters algorithm currently finds the optimal parameters for Holt-Winters by iteratively checking and then comparing the accuracy of forecasts made with every possible permutation and combination of the hyper-parameters. This is obviously resource expensive and slows down the performance of this solution. Moreover, the hyper-parameters values are only checked in increments of 0.1 and this will negatively impact the precision of optimal hyper-parameter selection. A possible future work could be to find a better way of deducing the optimal hyper-parameters.

Secondly, the results proved inconclusive in determining the correlation between removal of anomalies from the training datasets and the accuracy of forecasts made using those training datasets. More research needs to be done to determine the existence and extent of any such correlation.

Moreover, there is an inherent assumption that only clean data, i.e. data that is free of any anomalies or irregular behaviour, is used to train the prediction algorithm. Although evaluation done in this thesis suggests that anomalies have a negative impact on the forecasting accuracy, this evaluation was done only with a few samples of data. Research needs to be done to determine the extent of this impact.

Finally, the first condition for issuing alerts is the triggering of thresholds which are z-score dependent. The default values for these thresholds are defined by Chebyshev's inequality theorem but there is an option to manually configure these values as well, as was done for the purpose of this thesis. Manual configuration involved several tests and an understanding of the use case (in this case SWA). However, there is research that indicates that thresholds can be automatically optimized using different algorithms and methods. Kreugel et al. [28] suggest using cross-validation to find optimal thresholds. Ghafouri et al. [88] introduced an adaptive threshold strategy which first determines an optimal threshold independent of time. Thereafter a time dependent threshold is calculated using a polynomial time algorithm and it associates a cost with threshold changes. Laszka et al. [89] also explored different methods of finding optimal thresholds. Hence, a possible future work could be to explore the current literature relevant to automating the process of determining optimal thresholds and implementing the method which is found to be most suitable.

Appendices

Forecasting Accuracy for Datasets Collected Over Two Weeks - Results

Table 19: Summary of Forecasting And Alerting Accuracy with Two Weeks Data

	Predicted Value of Page views for the Next Hour	Actual Value of Page views for the Next Hour	Classification of Actual Value as Anomaly or Normal Behaviour- Manually Identified	Classification of Actual Value as Anomaly or Normal Behaviour- As Per this Solution	Evaluation of Forecasting and Alerting Accuracy of this Solution	Absolute Percentage Error
Dataset 1	511.8099	278	Anomaly	High Alert	True Positive	45.68296
Dataset 2	91.29641	140	Normal Behaviour	No Alert	True Negative	53.34667
Dataset 3	1784.597	1733	Anomaly	Low Alert	True Positive	2.891226
Dataset 4	8.538859	4071	Anomaly	High Alert	True Positive	47576.16
Dataset 5	1344.746	1699	Anomaly	high level	True Positive	26.34353
Dataset 6	2154.387	2378	Anomaly	Medium Alert	True Positive	10.37943
Dataset 7	539.0523	563	Normal Behaviour	No Alert	True Negative	4.442555
Dataset 8	117.5493	282	Normal Behaviour	No Alert	True Negative	139.8994
Dataset 9	945.2566	1173	Anomaly	Medium Alert	True Positive	24.09329

Forecasting Accuracy for Datasets Collected Over Four Weeks - Results

Forecasting Accuracy for Datasets Collected Over Six Weeks - Results

Forecasting Accuracy for Datasets Collected Over Eight Weeks - Results

Sample Datasets Collected Over Four, Six and Eight Weeks Out of the nine data points, two are collected in the first few weeks of the introduction of this web page on SAP's website. These are Dataset 1 and Dataset 5. Therefore there is no more than two weeks of historical data for both of these and they are not in the graphs below.

The datasets show the seven data points (encircled in red) which were forecast during the Evaluation chapter. These are Dataset 2, 3, 4, 6, 7, 8 and 9. The graphs below show these data points with their two, four, six and eight weeks of historical data respectively, marked with arrows at the bottom of each graph.

Table 20: Summary of Forecasting And Alerting Accuracy with Four Weeks Data

	Predicted Value of Page views for the Next Hour	Actual Value of Page views for the Next Hour	Classification of Actual Value as Anomaly or Normal Behaviour- Manually Identified	Classification of Actual Value as Anomaly or Normal Behaviour- As Per this Solution	Evaluation of Forecasting and Alerting Accuracy of this Solution	Absolute Percentage Error
Dataset 1	-	278	Normal Behaviour	-	-	-
Dataset 2	8.429499	140	Normal Behaviour	Medium Alert	False Positive	1560.834
Dataset 3	1528.257	1733	Anomaly	Medium Alert	True Positive	13.39716
Dataset 4	217.6775	4071	Anomaly	High Alert	True Positive	1770.198
Dataset 5	-	1699	Anomaly	-	-	-
Dataset 6	2134.65	2378	Anomaly	Medium Alert	True Positive	11.40001
Dataset 7	376.9817	563	Normal Behaviour	No Alert	True Negative	49.34411
Dataset 8	107.1784	282	Normal Behaviour	Low Alert	False Positive	163.1127
Dataset 9	931.6882	1173	Anomaly	Low Alert	True Positive	25.90049

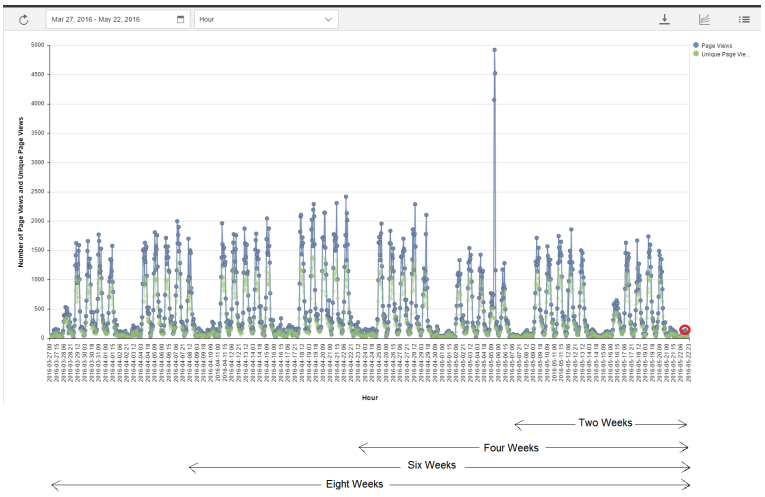

Figure 47: Sample Datasets Collected Over Eight Weeks- Dataset 2

Table 21: Summary of Forecasting And Alerting Accuracy with Six Weeks Data

	Predicted Value of Page views for the Next Hour	Actual Value of Page views for the Next Hour	Classification of Actual Value as Anomaly or Normal Behaviour- Manually Identified	Classification of Actual Value as Anomaly or Normal Behaviour- As Per this Solution	Evaluation of Forecasting and Alerting Accuracy of this Solution	Absolute Percentage Error
Dataset 1	-	278	Normal Behaviour	-	-	-
Dataset 2	-43.52573	140	Normal Behaviour	High Alert	False Positive	421.6488
Dataset 3	1454.038	1733	Anomaly	Medium Alert	True Positive	19.18534
Dataset 4	112.9432	4071	Anomaly	High Alert	True Positive	3504.66
Dataset 5	-	1699	Anomaly	-	-	-
Dataset 6	1958.479	2378	Anomaly	Medium Alert	True Positive	21.42078
Dataset 7	348.1202	563	Normal Behaviour	No Alert	True Negative	61.72576
Dataset 8	138.0872	282	Normal Behaviour	Low Alert	False Positive	104.2188
Dataset 9	973.0944	1173	Anomaly	No Alert	False Negative	20.54328

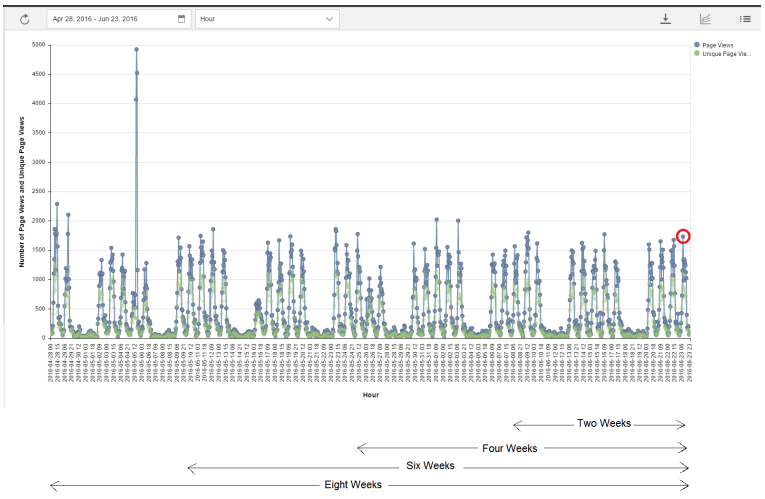

Figure 48: Sample Datasets Collected Over Eight Weeks- Dataset 3

Table 22: Summary of Forecasting And Alerting Accuracy with Eight Weeks Data

	Predicted Value of Page views for the Next Hour	Actual Value of Page views for the Next Hour	Classification of Actual Value as Anomaly or Normal Behaviour- Manually Identified	Classification of Actual Value as Anomaly or Normal Behaviour- As Per this Solution	Evaluation of Forecasting and Alerting Accuracy of this Solution	Absolute Percentage Error
Dataset 1	-	278	Normal Behaviour	-	-	-
Dataset 2	57.45676	140	Normal Behaviour	Low Alert	False Positive	143.6615
Dataset 3	1559.032	1733	Anomaly	Low Alert	True Positive	11.15873
Dataset 4	238.1507	4071	Anomaly	High Alert	True Positive	1609.422
Dataset 5	-	1699	Anomaly	-	-	-
Dataset 6	1938.593	2378	Anomaly	Medium Alert	True Positive	22.66627
Dataset 7	158.7639	563	Normal Behaviour	No Alert	True Negative	254.6147
Dataset 8	185.0083	282	Normal Behaviour	No Alert	True Negative	52.42561
Dataset 9	1038.739	1173	Anomaly	No Alert	False Negative	12.9254

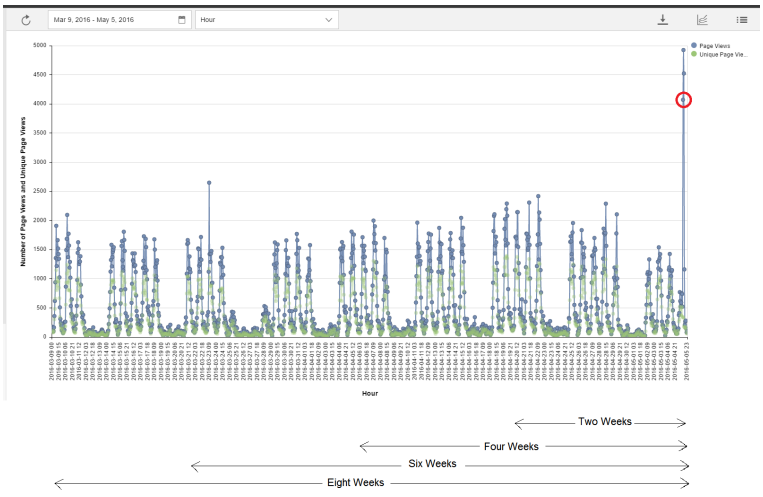

Figure 49: Sample Datasets Collected Over Eight Weeks- Dataset 4

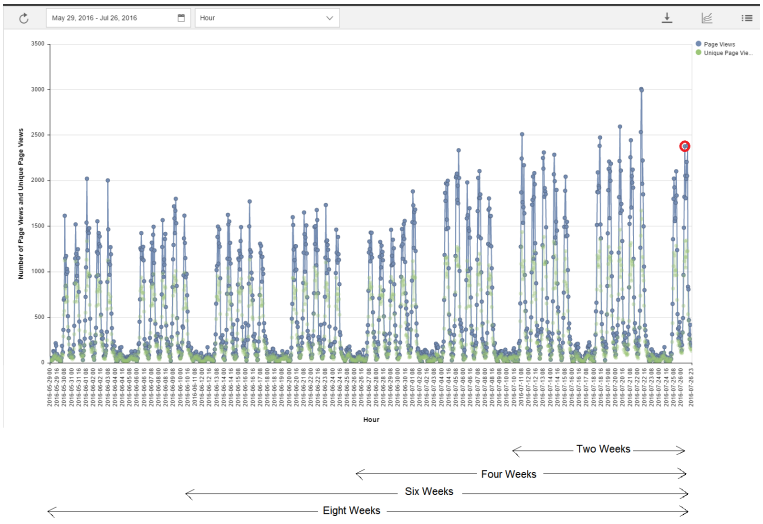

Figure 50: Sample Datasets Collected Over Eight Weeks- Dataset 6

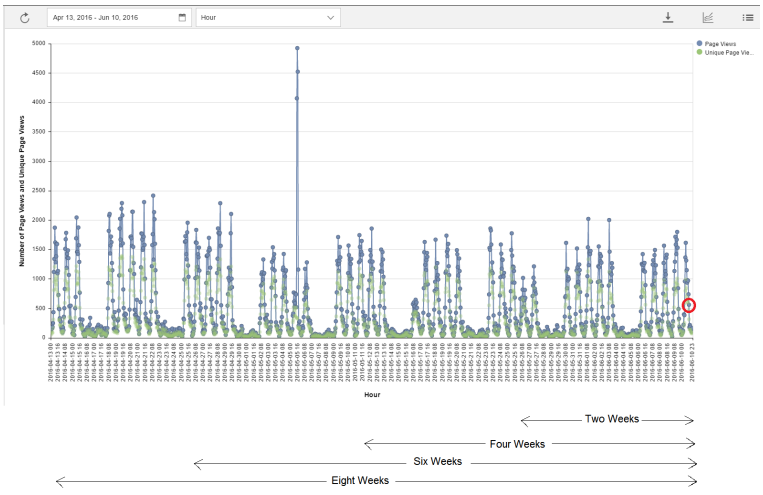

Figure 51: Sample Datasets Collected Over Eight Weeks- Dataset 7

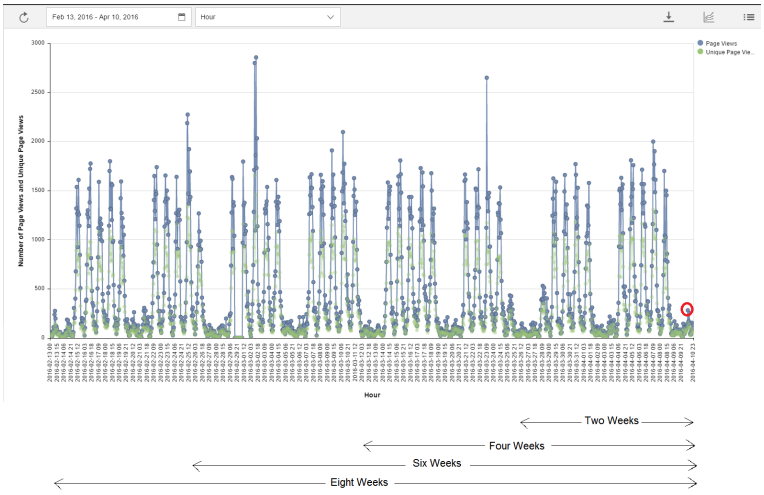

Figure 52: Sample Datasets Collected Over Eight Weeks- Dataset 8

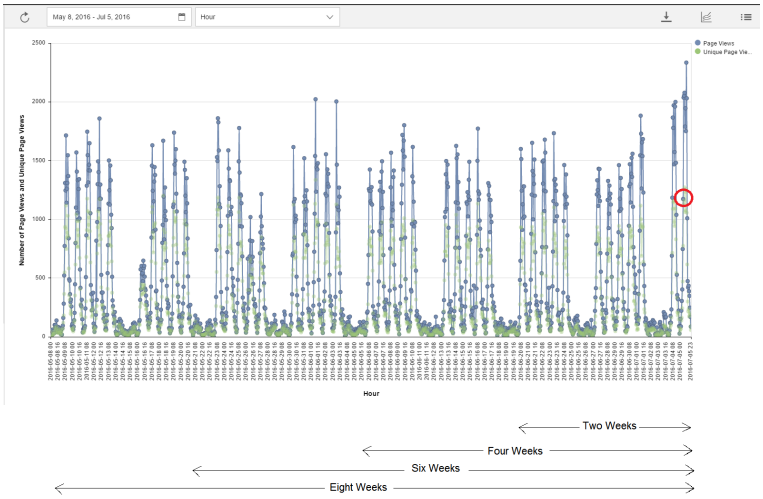

Figure 53: Sample Datasets Collected Over Eight Weeks- Dataset 9

References

[1] Andrew Phippen, L Sheppard, and Steven Furnell. A practical evaluation of web analytics. *Internet Research*, 14(4):284–293, 2004.

[2] Bernard J Jansen. *Handbook of research on web log analysis*. IGI Global, 2008.

[3] Tracking Code Overview how does google analytics collect data. `https://developers.google.com/analytics/resources/concepts/gaConceptsTrackingOverview`, note = Accessed: 2016-07-26.

[4] List of web analytics software. `https://en.wikipedia.org/wiki/List_of_web_analytics_software`,note = Accessed: 2016-07-26.

[5] C Wyatt and T Axelson. Web analytics-using emetrics to guide marketing strategies on the web. clark university-information technology services, 2011.

[6] Danielle Booth and Bernard J Jansen. A review of methodologies for analyzing websites. *IGI Global*, pages 141–62, 2009.

[7] Bernard J Jansen. Understanding user-web interactions via web analytics. *Synthesis Lectures on Information Concepts, Retrieval, and Services*, 1(1):1–102, 2009.

[8] Sa-kwang Song, Do-Heon Jeong, Jinhyung Kim, Myunggwon Hwang, Jangwon Gim, and Hanming Jung. Research advising system based on prescriptive analytics. In *Future Information Technology*, pages 569–574. Springer, 2014.

[9] Peter J Haas, Paul P Maglio, Patricia G Selinger, and Wang Chiew Tan. Data is dead... without what-if models. *PVLDB*, 4(12):1486–1489, 2011.

[10] Ramesh Sharda, Daniel Adomako Asamoah, and Natraj Ponna. Business analytics: Research and teaching perspectives. In *Information Technology Interfaces (ITI), Proceedings of the ITI 2013 35th International Conference on*, pages 19–27. IEEE, 2013.

[11] CK Praseeda and BL Shivakumar. A review of trends and technologies in business analytics. *International Journal of Advanced Research in Computer Science*, 5(8), 2014.

[12] Chris Chatfield. *The analysis of time series: an introduction*. CRC press, 2016.

[13] Peter J Brockwell and Richard A Davis. *Time series: theory and methods*. Springer Science & Business Media, 2013.

[14] Genshiro Kitagawa and Will Gersch. A smoothness priors–state space modeling of time series with trend and seasonality. *Journal of the American Statistical Association*, 79(386):378–389, 1984.

[15] RJ Hyndman and Y Khandakar. Automatic time series forecasting: The forecast package for r 7. 2008. *URL: https://www. jstatsoft. org/article/view/v027i03 [accessed 2016-02-24][WebCite Cache]*, 2007.

[16] Carroll Croarkin and Paul Tobias. Nist/sematech e-handbook of statistical methods. *NIST/SEMATECH, July. Available online: http://www. itl. nist. gov/div898/handbook*, 2006.

[17] Andrew C Harvey. *Forecasting, structural time series models and the Kalman filter*. Cambridge university press, 1990.

[18] V Bamnett and T Lewis. Outliers in statistical data. 1994.

[19] Douglas M Hawkins. *Identification of outliers*, volume 11. Springer, 1980.

[20] Irad Ben-Gal. Outlier detection. In *Data mining and knowledge discovery handbook*, pages 131–146. Springer, 2005.

[21] Tamraparni Dasu and Theodore Johnson. *Exploratory data mining and data cleaning*, volume 479. John Wiley & Sons, 2003.

[22] Bernard Rosner. On the detection of many outliers. *Technometrics*, 17(2):221–227, 1975.

[23] Simon Hawkins, Hongxing He, Graham Williams, and Rohan Baxter. Outlier detection using replicator neural networks. In *International Conference on Data Warehousing and Knowledge Discovery*, pages 170–180. Springer, 2002.

[24] Hancong Liu, Sirish Shah, and Wei Jiang. On-line outlier detection and data cleaning. *Computers & chemical engineering*, 28(9):1635–1647, 2004.

[25] Michael R Smith and Tony Martinez. Improving classification accuracy by identifying and removing instances that should be misclassified. In *Neural Networks (IJCNN), The 2011 International Joint Conference on*, pages 2690–2697. IEEE, 2011.

[26] Wenke Lee and Dong Xiang. Information-theoretic measures for anomaly detection. In *Security and Privacy, 2001. S&P 2001. Proceedings. 2001 IEEE Symposium on*, pages 130–143. IEEE, 2001.

[27] Varun Chandola, Arindam Banerjee, and Vipin Kumar. Anomaly detection: A survey. *ACM computing surveys (CSUR)*, 41(3):15, 2009.

[28] Christopher Kruegel and Giovanni Vigna. Anomaly detection of web-based attacks. In *Proceedings of the 10th ACM conference on Computer and communications security*, pages 251–261. ACM, 2003.

[29] Animesh Patcha and Jung-Min Park. An overview of anomaly detection techniques: Existing solutions and latest technological trends. *Computer networks*, 51(12):3448–3470, 2007.

[30] Bernard Rosner. Percentage points for a generalized esd many-outlier procedure. *Technometrics*, 25(2):165–172, 1983.

[31] Edwin M Knox and Raymond T Ng. Algorithms for mining distancebased outliers in large datasets. In *Proceedings of the International Conference on Very Large Data Bases*, pages 392–403. Citeseer, 1998.

[32] Univariate and multivariate outliers. `http://www.statisticssolutions.com/univariate-and-multivariate-outliers/`, note = Accessed: 2016-08-03.

[33] Christopher M Bishop. Pattern recognition. *Machine Learning*, 128, 2006.

[34] Venkatadri Seshadri. *The inverse Gaussian distribution: a case study in exponential families*. Oxford University Press, 1993.

[35] Stephen M Stigler. A modest proposal: a new standard for the normal. *The American Statistician*, 36(2):102, 1982.

[36] George Marsaglia et al. Evaluating the normal distribution.

[37] Thomas M Cover and Joy A Thomas. *Elements of information theory*. John Wiley & Sons, 2012.

[38] Sung Y Park and Anil K Bera. Maximum entropy autoregressive conditional heteroskedasticity model. *Journal of Econometrics*, 150(2):219–230, 2009.

[39] Leonard J Kazmier. *Schaum's outline of theory and problems of business statistics*. McGraw Hill Professional, 2003.

[40] Toshio Nakagawa and Shunji Osaki. The discrete weibull distribution. *IEEE Transactions on Reliability*, 24(5):300–301, 1975.

[41] John Aitchison and James Alexander Campbell Brown. *The lognormal distribution.*, volume 5. CUP Archive, 1976.

[42] Benjamin Epstein. The exponential distribution and its role in life testing. Technical report, DTIC Document, 1958.

[43] Erwin Kreyszig. *Advanced engineering mathematics*. John Wiley & Sons, 2010.

[44] Pedro Garcia-Teodoro, J Diaz-Verdejo, Gabriel Maciá-Fernández, and Enrique Vázquez. Anomaly-based network intrusion detection: Techniques, systems and challenges. *computers & security*, 28 (1):18–28, 2009.

[45] David Newman, Joel Snyder, and Rodney Thayer. Crying wolf: False alarms hide attacks. *Network World*, 24, 2002.

[46] Stefan Axelsson. The base-rate fallacy and its implications for the difficulty of intrusion detection. In *Proceedings of the 6th ACM Conference on Computer and Communications Security*, pages 1–7. ACM, 1999.

[47] Friedrich Pukelsheim. The three sigma rule. *The American Statistician*, 48(2):88–91, 1994.

[48] John W Tukey. Exploratory data analysis. 1977.

[49] David C Hoaglin, Frederick Mosteller, and John Wilder Tukey. *Understanding robust and exploratory data analysis*, volume 3. Wiley New York, 1983.

[50] Jorma Laurikkala, Martti Juhola, Erna Kentala, N Lavrac, S Miksch, and B Kavsek. Informal identification of outliers in medical data. In *Fifth International Workshop on Intelligent Data Analysis in Medicine and Pharmacology*, pages 20–24, 2000.

[51] Songwon Seo. A review and comparison of methods for detecting outliers in univariate data sets. 2002.

[52] Fatemeh Eghbali Moghaddam. Estimation of pyrrolizidine alkaloids in native and invasive weed species of the netherlands using reflectance spectroscopy. 2010.

[53] Wiesław Pawłucki and Wiesław Pleśniak. Markov's inequality andcâĹđ functions on sets with polynomial cusps. *Mathematische Annalen*, 275(3):467–480, 1986.

[54] Hans P Heinig and Lech Maligranda. Chebyshev inequality in function spaces. *Real Analysis Exchange*, 1991.

[55] Peter J Huber. The behavior of maximum likelihood estimates under nonstandard conditions. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, pages 221–233, 1967.

[56] John G Saw, Mark CK Yang, and Tse Chin Mo. Chebyshev inequality with estimated mean and variance. *The American Statistician*, 38(2):130–132, 1984.

[57] Chernoff Bound. Probability of error, equivocation, and the. *IEEE Transactions on Information Theory*, 16(4), 1970.

[58] Chris Chatfield. *Time-series forecasting*. CRC Press, 2000.

[59] Paul Newbold and Clive WJ Granger. Experience with forecasting univariate time series and the combination of forecasts. *Journal of the Royal Statistical Society. Series A (General)*, pages 131–165, 1974.

[60] G Peter Zhang. Time series forecasting using a hybrid arima and neural network model. *Neurocomputing*, 50:159–175, 2003.

[61] Spyros Makridakis, A Andersen, Robert Carbone, Robert Fildes, Michele Hibon, Rudolf Lewandowski, Joseph Newton, Emanuel Parzen, and Robert Winkler. The accuracy of extrapolation (time series) methods: Results of a forecasting competition. *Journal of forecasting*, 1(2): 111–153, 1982.

[62] Jan G De Gooijer and Rob J Hyndman. 25 years of time series forecasting. *International journal of forecasting*, 22(3):443–473, 2006.

[63] Everette S Gardner. Exponential smoothing: The state of the art. *Journal of forecasting*, 4(1):1–28, 1985.

[64] Chris Chatfield and Mohammad Yar. Holt-winters forecasting: some practical issues. *The Statistician*, pages 129–140, 1988.

[65] Nicholas Rescher. *Predicting the future: An introduction to the theory of forecasting*. SUNY press, 1998.

[66] George EP Box, Gwilym M Jenkins, Gregory C Reinsel, and Greta M Ljung. *Time series analysis: forecasting and control*. John Wiley & Sons, 1970.

[67] Edward J Hannan and Jorma Rissanen. Recursive estimation of mixed autoregressive-moving average order. *Biometrika*, 69(1):81–94, 1982.

[68] Víctor Gómez, Agustín Maravall, et al. Automatic modeling methods for univariate series. Technical report, Banco de Espa a, 1998.

[69] Spyros Makridakis and Michele Hibon. The m3-competition: results, conclusions and implications. *International journal of forecasting*, 16(4):451–476, 2000.

[70] David Reilly. The autobox system. *International Journal of Forecasting*, 16(4):531–533, 2000.

[71] 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, 2002.

[72] Denis Kwiatkowski, Peter CB Phillips, Peter Schmidt, and Yongcheol Shin. Testing the null hypothesis of stationarity against the alternative of a unit root: How sure are we that economic time series have a unit root? *Journal of econometrics*, 54(1-3):159–178, 1992.

[73] Denise R Osborn, Alice PL Chui, Jeremy P Smith, and Chris R Birchenhall. Seasonality and the order of integration for consumption. *Oxford Bulletin of Economics and Statistics*, 50(4):361–377, 1988.

[74] Robert Fildes and Spyros Makridakis. The impact of empirical accuracy studies on time series analysis and forecasting. *International Statistical Review/Revue Internationale de Statistique*, pages 289–308, 1995.

[75] Essam Mahmoud. Accuracy in forecasting: A survey. *Journal of Forecasting*, 3(2):139–159, 1984.

[76] Paul R Campbell. *Evaluating forecast error in state population projections using Census 2000 counts*. Population Division, US Bureau of the Census, 2002.

[77] J Scott Armstrong and Fred Collopy. Error measures for generalizing about forecasting methods: Empirical comparisons. *International journal of forecasting*, 8(1):69–80, 1992.

[78] Yuichiro Anzai. *Pattern Recognition & Machine Learning*. 2012.

[79] José D Bermúdez, Jose V Segura, and Enriqueta Vercher. Holt–winters forecasting: an alternative formulation applied to uk air passenger data. *Journal of Applied Statistics*, 34(9):1075–1090, 2007.

[80] Juan Félix San-Juan, Montserrat San-Martín, Iván Pérez, and Rosario López. Hybrid perturbation methods based on statistical time series models. *Advances in Space Research*, 57(8):1641–1651, 2016.

[81] R Open Source Code forecast package. `https://github.com/robjhyndman/forecast`, note = Accessed: 2016-07-28.

[82] Dong C Liu and Jorge Nocedal. On the limited memory bfgs method for large scale optimization. *Mathematical programming*, 45(1-3):503–528, 1989.

[83] Stephen G Nash and Jorge Nocedal. A numerical study of the limited memory bfgs method and the truncated-newton method for large scale optimization. *SIAM Journal on Optimization*, 1(3): 358–372, 1991.

[84] R Core Team. R: A language and environment for statistical computing. 2016. URL `https://www.R-project.org/`.

[85] Rob J Hyndman and George Athanasopoulos. *Forecasting: principles and practice*. OTexts, 2014.

[86] R forecast package, howpublished = `https://cran.r-project.org/web/packages/forecast/forecast.pdf`, note = accessed: 2016-08-16.

[87] Nishant Chandra. Holt-winters triple exponential smoothing algorithm. `https://github.com/cmdrkeene/holt_winters`, 2011.

[88] Amin Ghafouri, Waseem Abbas, Aron Laszka, Yevgeniy Vorobeychik, and Xenofon Koutsoukos. Optimal thresholds for anomaly-based intrusion detection in dynamical environments. *arXiv preprint arXiv:1606.06707*, 2016.

[89] Aron Laszka, Waseem Abbas, S Shankar Sastry, Yevgeniy Vorobeychik, and Xenofon Koutsoukos. Optimal thresholds for intrusion detection systems. In *Proceedings of the Symposium and Bootcamp on the Science of Security*, pages 72–81. ACM, 2016.