Developing event identification methods for structured and unstructured data streams

Nora Alkhamees

A thesis submitted for the degree of
Doctor of Philosophy

School of Computer Science and Electronic Engineering

University of Essex

July 2019

Abstract

Data, now more than ever before, are continuously being generated in huge volumes, and at rapid speed. Data may originate from various sources, for instance: sensor readings, financial transactions, social networks, etc.. A data stream is a continuous sequence of data arriving in almost real-time and often at a high speed.

In this thesis, we are interested in benefiting from the availability of such data and developing methods for detecting the occurrence of events from data streams, such as a text stream and a price time-series stream. Hence, we have explored event identification from structured and unstructured data streams in the domain of finance.

We employ the Directional Change (DC) approach to high frequency time-series streams to identify significant price transitions (i.e. events). DC is an event-based approach for summarizing price movements based on a fixed, a-priori threshold. We propose a dynamic threshold definition method, which replaces the fixed threshold and is appropriate for markets that operate over specific opening and closing times. A dynamic threshold provides more flexibility and extends the DC approach allowing the identification of price changes in continuously changing environments.

With the proliferation of social media data reporting on all aspects of human activity, being able to automatically identify events is becoming increasingly important. We present a framework for detecting the occurring events on a daily basis, via social network streams. We develop and extend a Frequent Pattern Mining method by proposing a dynamic support definition method to replace the fixed support. As the number of text posts streamed each day is not fixed, a dynamic support, can adapt to the nature of data streams and can improve the identification of events.

Finally, we explore whether we can bring together the insights from the time-series stream and the social network stream to understand if events as identified from both streams can be correlated.

Acknowledgements

All praises are to Allah, the almighty, for giving me the opportunity, strength, and endurance to complete my PhD thesis. I am truly grateful to everyone who so generously supported me and inspired me, to everyone who contributed to the work presented in this thesis.

I would like to express my deepest and sincere gratitude to my supervisor Prof Maria Fasli for her continuous support, motivation, and immense knowledge throughout my PhD journey. Her guidance has helped me in all the time of research and writing of this thesis, I have learnt from her how to be focused, dedicated, and an independent person. I could not have imagined having a better supervisor and mentor for my PhD study.

Huge thanks to my amazing sisters Hend and Alhanouf, amazing brothers Fahd and Turki, and amazing nieces Lama and Maha for their unconditional love, encouragements, and endless support. Nawaf my beloved nephew and flatmate, thank you for everything you have done for me.

I would like to thank all my colleagues and friends for their kind support and care. Very special and deep thanks to Arwa Basbrin, Enas Jambi, Fatima Alshahrani, Huda Altisan, Manal Alghannam, Samah Felemban, and Thabat Thabit for being always there for me when I needed them, and for all the great times we spent together. Asma Alzaid and Nouf Bindris thank you for all the long discussions we have been through, and for the lovely weekends we spent together in Colchester, Bristol, and London. My friends and co-workers, Monira Aloud and Nora Almuzaini, thank you for your ongoing support, and for the lovely meetings we always have whenever I am back home. Special thanks to Monira for sharing her PhD experience so willingly with me.

Last but not least, I am so thankful to my parents for simply everything. To my caring father for being the number one believer in me and my abilities, and for supporting me in all my pursuits. For all the discussions and interests he showed to my research just to help me think out loud. To my loving mother for her endless prays, and the boosting voice-notes she constantly sent which always lift me up and pushed me towards hard work. My parents are the most important people in my world, and I dedicate this thesis to them with loads of love.

Contents

List of Figures

List of Tables

List of Algorithms

Chapter 1

Introduction

The amount of data generated, nowadays, is extraordinary. This includes data produced from sensor readings, financial markets, social networks, web logs, click streams, etc.. In general, one cannot reason using such data streams easily, as they are of an unbounded size and their data elements arrive at an irregular rate [7]. Furthermore, it is not possible to backtrack over the past arrived data elements or review and keep track of the entire history. Data streams are infinite continuous data feeds, ordered by time, and generated at high speeds [8].

Data streams can be considered as one of the main sources of what is referred to as Big Data [9]. The concept of Big Data is commonly used to describe huge amounts of heterogeneous data coming from different sources, which can be analysed using special types of tools in order to obtain the advantage and insight of that data. In 2011, Gartner [10] defined Big Data as: high *volume* "increasing size of data", high *velocity* "increasing rate at which data is produced", and high *variety* "increasing range of data formats including structured (e.g., relational databases), semi structured (e.g., XML documents) and unstructured data (e.g., emails)", and in 2012 Gartner with others [11], expanded the definition to include high *Veracity* "increasing uncertainty to origins of data".

There has been an exponential growth in the generation of unstructured data, and furthermore, the amount of unstructured data has exceeded the amount of structured data [12]. In addition, in 2016 it was estimated that 80% of the world's generated data is unstructured[1]. Sources of unstructured data include: web pages, emails, images, medical records, mobile content, social media contents, etc.. Unstructured data do not typically have a defined data model due to the heterogeneous sources of the data, and such data may also not be easy to process using available tools [13]. In this context, text mining is an extension of data mining but for text, and is conducted to reveal value from unstructured, textual data [14].

Revealing insights from streams of data is becoming of an increasing interest, examples of such streaming analysis are: using sensors to detect pollution levels [15–17], discover influenza areas [18, 19], and capture the state of a city's traffic [20–23]. Other examples would be using social networks to spot disasters such as earthquakes [24–26] and to sense the effect of public mood [5, 27, 28].

In this thesis, we are interested in developing and extending event identification methods for structured and unstructured data streams. We want to focus on the methods that deal with the changing nature of the data streams. Instead of considering the dynamic and changing nature of data streams as a challenge and as a barrier to face, in this research, we will try and see this as an advantage and strength, and develop event identification methods that can accommodate and function in such data streams to provide better and more effective analysis. Thus, we seek to adapt to the changing nature of those streams, where the stream volume is extremely high, and the arrival rate of its data elements is unpredictable in advance. Particularly, we want to develop event detection methods for high frequency time-series data streams and social networks data streams (i.e. text streams). We will also try to explore whether the insights derived from the two streams can be correlated and brought together.

1.1 Research Aim and Objectives

In this thesis, we are interested in developing methods and techniques by which to detect the occurrence of events from various types of data streams, typically a text stream and a high frequency time-series data stream. We intend to focus on methods that deal with the changing nature of the data streams. Hence, the main aim is to study current state of the art event detection methods for structured and unstructured data and extend and develop them to deal with streams of data. Guided by the above aim we look to achieve the following core objectives: event detection from unstructured data streams, event detection from structured data streams, and cross-reference across various data streams.

[1] https://www.ibm.com/blogs/watson/2016/05/biggest-data-challenges-might-not-even-know/

1.1.1 Event detection from unstructured data streams

From a social network stream (Twitter) consisting of unstructured data, we aim to collect text posts relating to a certain major event and seek to develop methods that detect the occurrence of topics/events within that major event. An event in SN is defined in [29] as “an occurrence causing change in the volume of text data that discusses the associated topic at a specific time, and often associated with entities such as people and location”. Twitter is an online social network platform where users can post and interact using messages known as tweets. Twitter is a popular microblogging website, in 2013, twitter was described as the “the SMS of the Internet” [30]. Furthermore, in 2016, it was named as the largest source for breaking news after receiving 40 million tweets in a single day —the US presidential election day in 2016 [31]. Therefore, it was considered as the source of data for our investigated text stream rather than any other microblogging sites.

We intend to develop a method for identifying the occurrence of daily events in a social network text stream (Twitter), using a Frequent Pattern Mining (FPM) [32] method and a dynamic support value instead of a fixed given value. We want to see if the topics which are detected using the proposed dynamic support can better fit with the changing nature of text streams or not.

1.1.2 Event detection from structured data streams

From a price time-series data stream consisting of structured data, we are interested in developing methods to detect events (i.e. significant price fluctuations) once they occur. We want to study the financial market using a High Frequency Data (HFD) stream. HFD are referred to as extremely large amount of financial transactions at daily frequencies or even finer time scale [33].

In financial markets which operate over certain opening and closing times, we aim to develop a method for detecting the occuring events using a Directional Change (DC) [34, 35] approach and a dynamic threshold to replace the fixed given one. In DC, an event is identified in price time-series data streams once a significant price change is spotted (i.e. price change between two points satisfies a given threshold value).

We seek to find out if a dynamic threshold that is daily defined can lead to the more effective detecting of events than a fixed threshold value that is used throughout the whole stream (i.e. does not change).

Furthermore, we wish to explore the application of the dynamic threshold as part of a trading strategy and compare this with other trading strategies that have been developed based on the DC approach such as [36–38]. Hence, we want to find out if a trading strategy based on a dynamic threshold that is defined daily is more profitable than a fixed threshold or not.

1.1.3 Cross-reference across various data streams

Finally, we seek to cross-reference over data streams and identify correlations between the identified events. We want to bring insights from both streams together, and explore the relation between the events identified from them. Hence, we aim to put together the events identified from both streams, process them, and then look for correlation between them.

We intend to draw inferences from the two streams in order to disclose hidden relationships. We will try to identify correlations between the detected events from different data streams (structured and unstructured data streams), and see how and in what way they may affect each other. Thus, we will explore and further investigate the relationships between various data streams to see what links can be found, if any.

In addition, we want to find out if there is a correlation between events identified from the social network stream (Twitter), and those obtained from the high frequency time-series data stream. In particular to determine whether regional or global events are more correlated with stock market price changes.

1.2 Contributions

The work presented in this thesis contributes to the development of event identification methods from streams of data, both structured and unstructured. We have chosen the domain of finance as the domain of application as this is an area that is data rich and structured data can be obtained for study and experimentation. In particular, this thesis develops and extends event detection methods to be able to deal with the dynamic and changing nature of data streams. The main contributions are as follows:

1. In the text data stream (i.e. unstructured stream) analysis and event identification, we are using the FPM approach on a social network stream (Twitter) in

order to identify the occurring topics/events. Previous work on text streams using the FPM approach to obtain the occurring frequent patterns (i.e. events) require determining the number of terms to be selected in advance [39–41], in other words, they were using a fixed support value. In contrast, we think that in changing and dynamic environments such as text data streams, it is difficult to specify the suitable number for items (i.e. terms in text streams) to be retrieved in advance. This is because the number of items received is not fixed and may vary from one batch to another.

Thus, we propose a new method to define the support value (i.e. threshold) dynamically for every window-batch separately, to better cope with the nature of data streams. Using a dynamic defined support value to identify events from text data streams, addresses the problem of having fixed support values (Chapter 3).

2. In relation to the high frequency time-series data stream (i.e. structured data stream) analysis and event identification, we use the DC event approach to identify the financial market price fluctuations (i.e. events). An event in the DC approach is detected if the price change between two points exceeds/is below the given threshold value. A fixed threshold has always been the case when using the DC approach [6, 35–38, 42–51]

   We improve the operational performance of the DC approach by introducing a novel method for dynamically defining the threshold on a daily basis instead of setting it as a fixed value. Using dynamic threshold values, events of different magnitudes can be detected, which is not applicable with fixed thresholds (Chapter 4).

3. The DC approach has been used to define and use trading strategies in different markets. One of the issues in designing a strategy is finding the most profitable threshold value [37, 38, 51, 52]. Thus, we introduce a trading strategy (named the DT-TS) based on the DC approach and the dynamic defined threshold value so as to further evaluate the usefulness of the method for defining the daily dynamic threshold (Chapter 5).

4. Finally, we perform a study to explore the relationship between the different data streams, one originating from social networks (unstructured data) and the other originating from the stock market (structured data). We explore the relation

between the events identified from both streams and try to cross-reference between them. This is especially important in supporting decision making in domains such as the financial markets (Chapter 7).

1.3 Thesis Structure

This thesis is structured into eight main chapters.

Chapter 2 provides a literature review and discusses the related work, to show what has been done in the area to be studied. It consists of seven main sections, starting with a section on the semantic web, which shows the role and importance of the semantic web in relation to the data available on the World Wide Web (WWW). This is followed by the stream reasoning section, where we define a data stream, and compare between traditional reasoning and stream reasoning. In the next section, we present the topic detection methods, which are the document-pivot method, the feature-pivot method, and the probabilistic method. The fourth section reviews some attempts at event identification in relation to Twitter. After that, we describe the FPM method, and illustrate attempts at using the FPM for detecting events occurring on Twitter. In section six we explain the DC approach, which is an approach for summarizing price movements in financial streams. Section seven highlights some work linking and connecting financial market data with Twitter data.

Chapter 3 describes the developed method for detecting events/topics from a social network data stream (text stream). In more details, we show how the FPM method was extended and developed to cope with the nature of data streams by introducing a dynamic support definition method to replace the fixed given one for detecting the occurring topics/events. Then we present the experimental work, which starts by showing how the data (i.e. text posts) were collected, then how the dynamic support value was set and how the experiments were carried out to identify topics/events, and finally, we discuss and analyse the findings. We conclude the chapter by evaluating the performance of our topics detection framework and discuss the evaluation results.

Chapter 4, describes the developed method for detecting events (i.e. price transitions) from high frequency time-series data streams. In more details, we show how the DC approach was employed to the price time-series data streams along with describing the dynamic threshold definition method. In addition, a comparison between the detected DC events using the dynamic threshold and different fixed thresholds is conducted, and finally a discussion and analyses of the findings is presented.

After that, in the fifth chapter, we propose a trading strategy, named the Dynamic Threshold-Trading Strategy (DT-TS), based on the DC approach and the daily dynamic defined threshold. We evaluate the performance of DT-TS against various fixed threshold values and different trading strategies. Furthermore, in Chapter 6, we explore the functionality of that trading strategy (i.e. DT-TS) on a lower frequency data stream to see whether it performs equally well in that circumstance. We discuss and analyse the performance of both data streams (the high and lower frequency data streams).

In Chapter 7, we explore whether we can bring insights from the two streams together (the text stream and the price time-series data stream) and correlate over them. We apply the Correlation Coefficient test and the Granger Causality method in order to put together and draw inferences across events detected from both streams.

Finally, in Chapter 8, the thesis concludes with a summary and some conclusions regarding the work presented and the findings, and the planned future work is outlined.

1.4 Publications

The following publications were produced as a result of the work undertaken as part of this PhD project:

1. N. Alkhamees and M. Fasli, “Event detection from time-series streams using directional change and dynamic thresholds,” 2017 IEEE International Conference on Big Data (Big Data), Boston, MA, 2017, pp. 1882-1891. doi: 10.1109/BigData.2017.8258133

2. N. Alkhamees and M. Fasli, “An exploration of the directional change based trading strategy with dynamic thresholds on variable frequency data streams,” 2017 International Conference on the Frontiers and Advances in Data Science (FADS), Xi’an, 2017, pp. 108-113. doi: 10.1109/FADS.2017.8253207

3. N. Alkhamees and M. Fasli, “A Directional Change Based Trading Strategy with Dynamic Thresholds,” 2017 IEEE International Conference on Data Science and Advanced Analytics (DSAA), Tokyo, 2017, pp. 283-292. doi: 10.1109/DSAA.2017.48

4. N. Alkhamees and M. Fasli, "Event detection from social network streams using frequent pattern mining with dynamic support values," 2016 IEEE International Conference on Big Data (Big Data), Washington, DC, 2016, pp. 1670-1679. doi: 10.1109/BigData.2016.7840781

Chapter 2

Literature Review

While attempting to develop methods for event detection from various types of data streams, in this chapter we present a background and a literature review of related topics. The literature review chapter covers seven main topics, in sequence: the Semantic Web; stream reasoning; topic detection methods; event identification from Twitter; Frequent Pattern Mining (FPM); the Directional Change approach; and finally, linking financial market data and Twitter data.

2.1 Semantic Web

The development of the Semantic Web is promising to revolutionise the World Wide Web (WWW) and its use by providing content which can be 'understood' by computers. This has opened the door more widely to the possibility of taking full advantage of everything that is available on the web. In [53], the Semantic Web is defined as an extension of the current Web in which information is given well-defined meanings, to better enabling computers and people to work in cooperation.

Generally, to be able to reason with data from the Web, there must be a data model to represent the data and an ontology to be followed which allows communication with that data. An ontology is a defined abstract model of a domain from a particular point of view/perspective, it describes the objects (terms) and their relations and can be used for providing a common understanding of the underlying domain of application. It is the key to machine-processable data on the Semantic Web [54].

The Resource Description Framework (RDF) [55] is a data model and is the foundation upon which the Semantic Web has been built. It is used for data representation and exchange on the Web. An RDF graph contains subject- predicate- object triples.

The Web Ontology Language (OWL) [56] is the language defined for representing ontologies in the Semantic Web; it was developed by the World Wide Web Consortium (W3C) Web Ontology Working Group. It is used to process and analyse the contents of the Web, not just to present it. An OWL ontology can be stored or exchanged when specified using the RDF or Extensible Markup Language (XML) syntaxes, respectively. An OWL ontology consists of a head, a class and properties.

The Semmantically-Interlinked Online Communities (SIOC) [1] is an ontology designed for use on the Web to describe user communities (message boards, wikis, and blogs); it depends on RDF and the OWL ontology, and was developed by the Science Foundation, Ireland. An overview of SIOC is shown in Figure 2.1. Later in this chapter, we will show some efforts using the SIOC ontology with Twitter data.

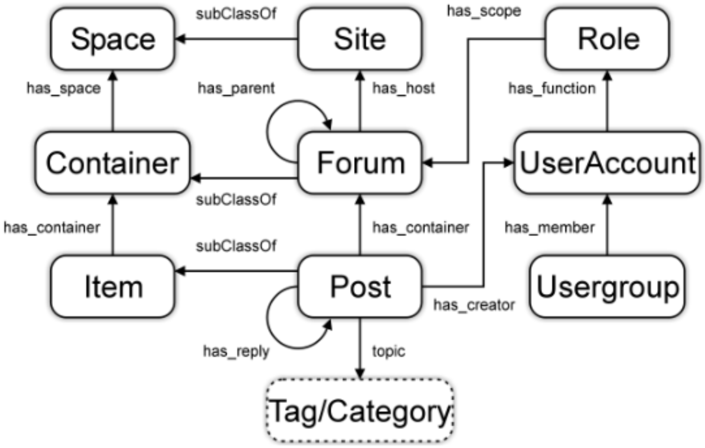


Figure 2.1: SIOC Overview (Image taken from [1])

Several studies including, but not limited to, [57–62] show that by using ontologies one can "reason" and understand the meaning of web pages, processes and analyse them more effectively.

2.2 Stream Reasoning

A data stream is a continuous sequence or flow of data over time produced in real-time or near real-time and often at high velocities [8]. Data streams produce a huge amount of dynamic data, and this can come from different sources: sensors, website click streams, tick prices, social networks, etc. In addition this data is either implicitly ordered by arrival time or explicitly by timestamp [63].

Examples of such streams are produced by real-time traffic monitoring systems where sensors are distributed among a city to capture information about traffic jams and road congestion, such in [20, 21, 64, 65]. This is of interest not only to citizens (in order to avoid congested roads), but it is also of interest to city managers, who can reason with those streams to better manage and further plan the city. In addition, the streams are of interest to event planners who also can reason using these streams – to sense the impact of current events as well as to choose better venues for future events. Another example of a data stream is that of share prices streamed at high frequency [66–68]. HFD are referred to as observations taken at fine time intervals. In finance, HFD refers to observations taken daily or at even finer time scales [33].

Due to the compelling need to reason using such streams in real-time or near real-time, stream reasoning was first explored by Della Valle et al. in 2009 in [7]. They commented that stream reasoning was "an unexplored, yet high impact, research area"; such reasoning is applied in real-time to noisy data streams, so as to support decision making. Table 2.1 shows a comparison between traditional reasoning and stream reasoning, showing how the reasoning is performed and when it starts and terminates.

Traditional Reasoning	**Stream Reasoning**
Analysis applied to all available data.	Hard to apply to all data; hence only applied to parts of it.
Processing starts when a query is fired and ends when answer is found or all data has been scanned.	Continuous processing is required; it does not end.

Table 2.1: Traditional VS Stream Reasoning

Stream processing systems can be categorised as Data Stream Management System (DSMS), which were developed by the database community, or as the Complex Event Processing (CEP) systems, which was developed by the event based research community [69].

The DSMS [70] inherits the relational data model and its expressive query language is adapted from the Data Base Management System (DBMS). The Continuous Query Language (CQL) is used with DSMS; here queries run continuously, and can be expressed using the SPARQL language [71]. In DBMS, data are saved in tables and users can submit queries to retrieve it. On the other hand, in DSMS users submit queries which are run continuously as new data arrives. The Large Knowledge Collider project (LarKC) is a web-scale reasoning platform started in 2008 which adopts DSMS functionality, and was built as part of the EU 7th framework project LarKC. Developers can implement their plugins and deploy them to the LarKC platform [72].

The CEP system is commonly used in event driven environments where a time element must be associated with streams; its main principle is continuous processing carried out to detect the occurrence of events [73]. Instead of queries, rules are defined which specify what constitutes the occurrence of an event; this is done via an Event Processing Language (EPL) or can be expressed using the TESLA language [74]. Esper is a system for complex event processing, it is open source and available in the Java and C# programming languages; it was developed by EsperTech in 2006 [75]. Next, we show some efforts using stream reasoning systems with social networks data streams.

2.2.1 Stream Reasoning with Social Networks Data

In [76], the authors used the data available on social networks in order to find out about the impact of city-scale events. They were interested in finding out about the popularity of events happening in a city. They presented a Streaming Linked Data (SLD) framework to collect data streams, analyse them, and finally to show the results they obtained on a dashboard. They used RDF as their data model and SIOC as the ontology to represent their data; for this study, they considered the London Olympic Games 2012 and the Milano Design Week (MDW) 2013 as the two social city scale events to be focused on.

For the London 2012 Olympic Games they wanted to detect events which were taking place in one of the three main Olympic venues: the Olympic Stadium, the Aquatic Centre, or the Water Polo Arena. To validate the events that they found mentioned in the data streams, they used the Olympic Games calendar. They were able to use a stream of three million tweets produced in the period between the 25th July and the 13th August 2012, across London. A continuously active C-SPARQL query which counted the number of tweets posted from an area every 15 minutes was "fired". It looked for bursts, taking the view that bursts are signals that an event is occurring, and it used a sliding window which "slid" every 1 minute. A burst was said to have occurred if the number of tweets from a location within a 15 minute period was greater than the average number of tweets collected in the last 2 hours + double the standard deviation. Moreover, if a burst was found from public transportation near the venue, and then was found in areas outside the venue, and finally was found in the venue itself, then it was posited that an event had been found.

For the MDW 2013, the authors of [76] wanted to detect events and analyse the crowd's opinions and sentiments concerning these events. During that week they collected tweets in real-time; they looked for all tweets posted from Milan city as well as tweets posted from world-wide locations that mentioned any of the 300 words in the keyword list related to MDW 2013; they collected a total of 107,044,478 tweets. The posts in the RDF streams were each associated with a value ranging from [-1, 1]; this value represented the sentiment of each tweet. Sentiment measurement was performed using a dictionary based sentiment classifier. A query was run every 15 minutes to count tweets and isolate the ones with a positive impression score in the range [0.3,1] and also tweets with a negative impression score in the range [-1,-0.3]. The in-between scores indicate neutrality. The total number of tweets broken down into positive, negative and neutral rating were calculated every 15 minutes.

From both of the cases discussed these researchers were able to detect the popularity of events from the numbers of tweets posted from particular locations. Many tweets were missed because users did not enable the location property – although they were, perhaps, actually at the event venue. Also, this research was limited in terms of reasoning because it only considers the number of tweets posted from a location and ignores the tweet text. If the tweet text had been included in the reasoning, this would perhaps have provided even better event popularity sensing because then only tweets related to the event in question would have been included.

Another study, [2], was conducted to gain insights from people's opinions posted on social media; in this case, this was done in order to rate restaurants and coffee shops in the Insadong district, Seoul, Korea. The authors built an application offering recommendations related to Point Of Interest (POI)s across a particular district of Seoul city, based on public opinion as discovered from social media. They used two types of data: the social media streams (Twitter); and static descriptions of POIs collected from the websites and portals of 319 restaurants, resulting in a geo referenced knowledge base providing 44 attributes for each restaurant (name, image, address, specialities, etc.). From Twitter they collected 200 million tweets sourced over three years (4 Feb 2008 to 23 November 2010). The vast majority of tweets from that district were not related to its restaurants; indeed it was found that only 109390 tweets were concerned with about 245 restaurants. They designed the BOTTARI ontology (see Figure 2.2) which extends the SIOC ontology by defining TwitterUser as a special case of UserAccount and Tweet as special case of Post. The most distinct thing about the BOTTARI ontology is the addition of the talksAbout property: this can indicate a user talking about something positively, negatively or neutrally. The stream reasoning resulted in RDF streams which indicated positive, negative, and neutral ratings for each restaurant. The reasoning activity was based on using the LarKC platform and depended on the BOTTARI ontology. To calculate the number of recommendations, four C-SPARQL queries were applied: the first one counts the number of positive ratings for each POI, daily; the second aggregates the result of query 1 for a whole week; the third computes the aggregation from query 2 for a month; and finally the fourth further aggregates query 3 for a whole year.

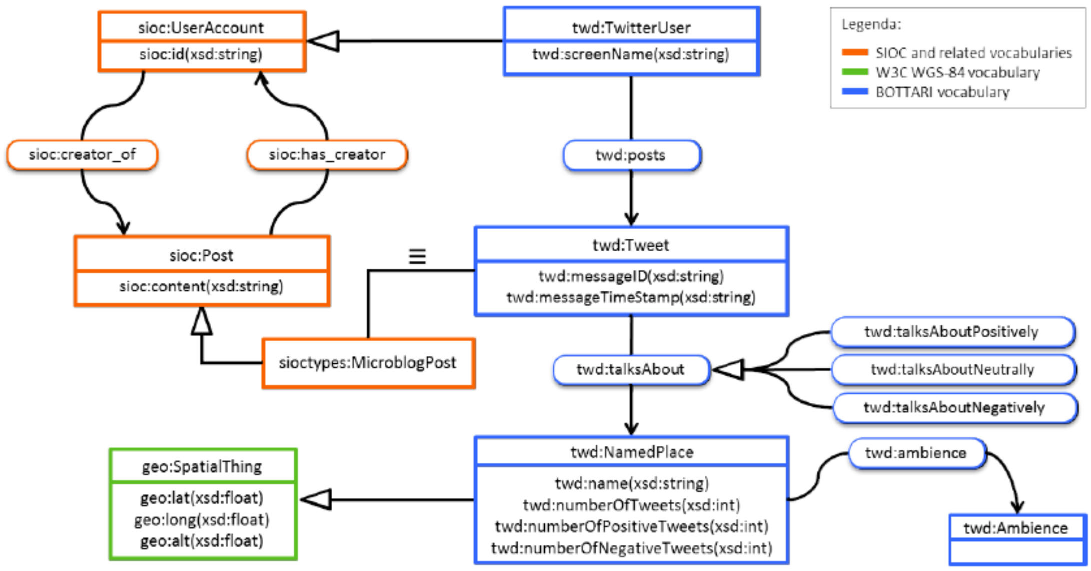


Figure 2.2: The BOTTARI Ontology (Image taken from [2] page 4)

In this research, the reasoning which was applied depended on the BOTTARI ontology definition which was developed especially for it, and as they were looking only for tweets related to Insadong restaurants, the number of tweets examined was small and so the streaming did not flood the system at all times. This means that the reasoning was not put under the kind of pressure that is usual in this type of investigation and the queries which were run did not experience the volume of tweets which could have degraded the aggregation task.

A study in [77], sought to discover the popularity of particular social events which occurred in a city. In relation to this, the researchers proposed an event-tweet pair coefficient as a metric for measuring event popularity. The dataset that was used for the experiment consisted of two parts. First, a data-set collected from websites and portals reporting on social events occurring across London which contained, for each event, the title, the description, the date and time, the performer, the location, and the type of event. The second part of the dataset was the collection of tweets posted on Twitter from the 6th March to the 11th April 2013. The total number of the social events collected was 10033 and the total number of streamed tweets was over 4 million. Streaming, in this research, was performed by applying CEP principles. The event stream processing was performed using the Esper [75] software package. For each event in the social event list, they aimed to detect all tweets posted between the beginning and the end time of that event. Moreover, for every event-tweet pair an Association Coefficient method (AC) was performed (see Equation 2.1) to calculate the degree of association between the event and the tweet.

$$AC = 0.5 * P + 0.25 * W + 0.125 * L + 0.125 * B \tag{2.1}$$

where P, B, and L are Boolean variables (equal to either 0 or 1). In more detail: if the event's performer is mentioned in the tweet's text, then $P = 1$, otherwise $P = 0$; if the event location is mentioned in the tweet's text, then $L = 1$, $L = 0$ otherwise; and if a brief description of the performer is mentioned in the tweet's text, then $B = 1$, if not then $B = 0$. The value of W on the other hand is equal to the number of words of the event title which also appear in the tweet's text, divided by the total number of words in the event title. The weights assigned to the metrics were based on common sense.

The period examined for this experiment was short, less than five weeks; we believe that given this, the study and its results cannot necessarily be generalized and serve as a foundation for future research into detecting events from Twitter. Also the processing used is only feasible when dealing with certain kinds of events (e.g., social events) which are known to be happening in advance, because in order to detect events the methods depend on the availability of the social event data-set.

Another, recent, study on social data analytics proposed the Social Set Analysis (SSA) approach, which is more tied to the sociology of associations, the mathematics of set theory, and advanced visual analytics [78]. The concepts were demonstrated by showing the way in which people's actions on social media reflect real world events. More precisely, these researchers related user interaction on a social network (Facebook) to real world events (before events, during events, and after events). This was presented via a Social Set Visualizer, which is an interactive visual analytics dashboard.

The garment industry in Bangladesh was chosen as a case study. They started by looking for events which were associated with garment factory accidents in Bangladesh; after this, news reports relevant to these, published via the traditional news media, were manually collected. Subsequently, using the SODATO tool [79], they retrieved the Facebook wall archives of the companies which were most frequently mentioned in these media reports (Benetton, Calvin Klein, Carrefour, H&M, JC Penny, Mango, Primark, Walmart, Zara, PVH, and E.C. Ingles). Once this was done, they designed and developed a "Social Set Visualizer" dashboard in order, mainly, to visualize a time-line of Bangladeshi garment factory accidents and Facebook Wall Activities.

In general, most research which has been applied to reasoning using social networks streams has mainly focused on post volumes rather than the text of the posts. However, we believe that the post text is also important and can provide further insights; at the same time, looking at the texts of posts could help in eliminating unrelated posts. Social media are filled with spam, advertisements, bot accounts that publish large volumes of posts, and internet memes [80]. As a result, in our study we will try to consider both the contents of the posts and the volume of the posts when reasoning to identify events.

2.3 Topic Detection Methods

Topic detection methods are generally classified under three major categories [81]: document-pivot methods, feature-pivot methods, and probabilistic topic methods.

2.3.1 Document-Pivot

In document-pivot methods, a topic is represented as a set of related documents. Typically such methods compute a similarity measure between either pairs of documents or between each document and a cluster. If the similarity measure exceeds a certain threshold value, then the document is added to the cluster, and if not, a new cluster is created.

Document-pivot approaches mainly differ in the ways in which they calculate the similarity between a pair of documents or between a document and a cluster. For instance, in [82] the similarity calculations were based on the Term Frequency-Inverse

Document Frequency (TF-IDF) weighting scheme. In general, the TF-IDF [83] scheme counts the frequency of a word (i.e. term) in a specific document compared to the inverse proportion of that word over the entire document corpus. It is used to evaluate how important a word is to a document in a corpus. Given a document corpus or collection D, a word w, and an individual document $d \in D$. The TF-IDF calculation is shown in Equation 2.2, where $f_{w,d}$ is the number of times w appears in d, $|D|$ is the size of the corpus, and $f_{w,D}$ is the number of documents in which w appears in corpus D [83, 84].

$$w_d = f_{w,d} * log(|D|/f_{w,D}) \tag{2.2}$$

So in [82], the similarity was calculated by comparing the TF-IDF score of the incoming tweet with the TF-IDF score of the first tweet in each cluster, along with the TF-IDF score for the most common words in that cluster. This comparison results in either the adding of that tweet to the best matching cluster or to the creation of a new cluster. Petrovic et al. in [85] aimed at detecting the first document discussing a topic in a large corpus, which is called the First Story Detection (FSD) approach [86]. Conventionally the FSD identifies a new story if the incoming document's similarity with other clusters is lower than a certain threshold.

In [85], they proposed a modification of the FSD approach which involves the use of Locality Sensitive Hashing (LSH), which is able to find the best matching/most-similar document (the nearest neighbour in a vector space) in a faster way. Another approach, proposed in [87], uses a graphical model named Location-Time Constrained Topic (LTT), which identifies the content, time and location of each social post. As a result, a post is represented as a probability distribution and the similarity between two posts is calculated based on the distance between their distributions.

As stated in [88], a general issue with document-pivot approaches, when used with social media streams, is that not all documents (i.e. posts) are related to relevant topics (e.g., memes and spam are generally not). In addition, document-pivot methods are not always scalable with respect to large amounts of data, as this requires batch processing.

2.3.2 Feature-Pivot

The feature-pivot approach mainly focuses on grouping terms based on their occurrence as representing a topic. It is a two-step approach, starting with selecting targeted terms based on their frequency or burstiness, then clustering terms based on some inter-term similarities.

Feature-pivot approaches mainly differ with respect to the term selection criteria. For instance, in [89], term selection was based on an “energy” measure for each term. The term’s “energy” was calculated from both the term’s frequency and the importance of the user who made the post. Depending on its “energy”, a term was clustered using a graph based algorithm; the results of this process were used to detect events. In [90], the term selection was based on “bursty” terms (i.e. terms with a frequency higher than usual). Then bursty keywords were grouped using a greedy search to discover the emerging trends. Finally a trend analysis was applied in order to find keywords that were not necessarily bursty.

An alternative, segment based, term selection procedure was adopted in [91]. It was based on bursty segments (multi-words segments) rather than bursty single terms. Another approach, used in [92], is Event Detection with Clustering of Wavelet-based Signals (EDCoW), which selects terms by applying a wavelet analysis based on the frequency of terms then clusters the selected terms based on a modularity-based graph partitioning technique for representing an event.

The Frequent Pattern Mining (FPM) method, [32], has been used in feature-pivot approaches to measure the co-occurrence of n terms instead of the co-occurrence of pairs of terms [93]. Some studies which use FPM to detect topics are: [4, 39–41] (these approaches will be discussed further in the coming sections). In this thesis, we present a dynamic way to create a term selection criteria for a text stream; this dynamic method is based on the size of a sliding window and the frequency of the terms within it.

2.3.3 Probabilistic Topic Model

The probabilistic topic model treats the topic detection issue as a probability inference problem. A topic is represented as a distribution between both terms and documents.

A survey of probabilistic topic models in [94] showed that the most well-known probabilistic topic model is the Latent Dirichlet Allocation (LDA) [95], whereby documents are represented as mixtures of latent topics. The learning and interpretation in LDA is performed, quite naturally, using variational Bayes [96] and other approaches including Gibbs sampling [97]. Furthermore, supervised versions of LDA were proposed in [98], and in [99], with application to Twitter. Another probabilistic topic model is Probabilistic Latent Semantic Indexing (PLSI) [100].

2.4 Event Identification from Twitter

In the current social network era almost everyone has now become a virtual broadcaster sharing an incredible number of messages, especially in this age of smart phones and the availability of mobile networks. Furthermore, the 2011 McKinsey survey [101] on the ages of social network users showed a 7% increase in the number of users within the age range 25-34, and a 22% increase in the number of users within the age range 35-45, even more surprising is the increase in ages 55-65 has been 90%. Although not all data generated through social networks is useful, we can benefit from the availability of the data in social networks across many areas and domains including, health, government and business.

Social networks are defined in [102] as web based sites that offer users the opportunity to build their own (public or private) profiles, in order to share posts and navigate though different connections belonging to other users. Twitter is a social networking service; it was launched in July 2006. A member in Twitter can post messages (tweets), follow accounts they are interested in, and other users can follow them in return — with their permission, if they have a private account, or without, if they have a public one. On your time line, you can view the tweets posted by you and the people you follow in real time. A tweet is a 140 character (in Nov 2017, the limit was doubled) message that a user creates to share what is happening, their opinions, and so on. In a tweet, you can include a hashtag (this # symbol proceeding a word or topic) to show that this tweet belongs to or discusses a certain topic. If you click that hashtag you will see all the tweets posted containing that hashtag. Also in a tweet, you can mention a specific user by using the symbol @username. Moreover you have the options to re-tweet or like a tweet which appears on your time line. The Twitter website is www.twitter.com.

For [86], an event is defined as a real world occurrence of something when it is associated with a real-world time-period and place. More specifically, the kind of event we are concerned with, in the context of social media in [103], is defined as the occurrence of something in the real world, which in turns initiates a discussion related to that event by different users just after it has occurred, or sometimes, in anticipation of its occurrence. Another definition is found in [29], wherein an event is defined as "An occurrence causing change in the volume of text data that discusses the associated topic at a specific time. This occurrence is characterized by topic and time, and often associated with entities such as people and location". Event identification is the process of looking for such events.

From the investigations conducted in [104, 105], it can be seen that Twitter can, essentially, be adopted as a medium for event detection. In more detail, Petrovic et al. in [104] compared Twitter with traditional newswires. They found that major events were covered by both newswires and by Twitter. However Twitter took the lead when it came to reporting on small or local events along with sports, political or business events. In [105], the authors examined how Facebook, Google Plus and Twitter report breaking news. They found that all the social media streams report similar events; however Twitter takes the lead in providing timely news, as compared to other Social Media streams.

One of the earliest attempts at detecting references to target events (such as an earthquake) in Twitter streams in real-time was in [25]. These authors considered Twitter users to be sensors who feel the occurrence of an earthquake, and the posted tweets as sensor readings. So they produced an algorithm to monitor tweets and to detect posts related to a target event. To classify tweets as either positive (referring to the target event) or negative (not related to the target event) they built a classifying model, using the Support Vector Machine (SVM) algorithm, and trained it on a set of tweets containing both positive and negative examples — so that it could, later, automatically classify tweets. For each tweet they assigned values to the attributes A, B, and C (see below), and then from these calculated the probability that the tweet referred to the target event.

A= the number of words in the tweet and the position of query words.

B= the word id determined for each word in a tweet.

C= words before and after query word.

They performed an experiment on a Twitter stream sourced from Japan. The purpose was to find target events using the Twitter stream search API; this was applied every second with the following query words {earthquake, shaking}. For each retrieved tweet they found A, B, and C (as above); then they calculated the probability of the tweet referring to an earthquake event that had occurred, $Poccur(t)$, as per Equation

2.3. Where $n_0(1 - e^{-\lambda(t+1)})/(1 - e^{-\lambda})$ is the number of sensors at time t, and $\lambda = 0.34$.

$$Poccur(t) = 1 - P_f^{n_0(1-e^{-\lambda(t+1)})/(1-e^{-\lambda})} \tag{2.3}$$

If $(Poccur(t))$>0.95, then it was considered that an earthquake event had been detected via the Twitter stream, and so an email was sent to registered users. As a result of this work, email notifications were sent faster, and before the Japan Metrological Agency (JMA) could do so — that is from the body responsible for broadcasting earthquake events in Japan.

TwitterMonitor, in [90], was another contribution to this field. This is a system that performs trend detection on Twitter streams. It identifies the emerging topics (i.e., the trends) via a three step approach. This starts with finding the bursty keywords, then it proceeds to the discovery of emerging trends; this is performed (second step) by grouping the related bursty keywords via a greedy search strategy. Lastly a trend analysis is performed whereby tweets related to the detected trend are linked together in order to discover further keywords associated with that trend (that are not necessarily bursty). For each trend a chart is produced showing its popularity, and this is updated for as long as the trend is popular.

Another study is [91], proposed a segment-based event detection system for tweets. Basically tweet-segments are used instead of unigrams for the purpose of detecting events; this requires that a tweet-segment may consist of one or more tweet words. Then event-segments are formed by finding bursty tweet-segments; these are identified by applying a Gaussian distribution based on a predefined fixed time-window along with a consideration of the user frequency, which means looking at the number of users tweeting about a certain segment. Candidate events are found by using a clustering algorithm to group event-related segments. Finally Wikipedia [1] is used as a knowledge base to filter the detected events.

TwiCal was a system created by the authors of [106] which extracted events from Twitter and categorized them into an open calendar. It extracted a 4-tuple representation of events from Twitter, which comprised the attributes: named entity, event phrase, calendar date, and event type.

These authors started by targeting tweets which mentioned temporal keywords (today, tomorrow, next week, month names, etc.), and in this way collected 100 million

[1] https://en.wikipedia.org/wiki/Main_Page

tweets in a corpus. From the collected tweets they extracted named entities using Natural Language Processing (NLP) tools specially trained on tweets [107]. After that, to find event phrases, which could potentially occur using various different parts of speech, they applied their Part Of Speech Tagging (POS) tagger which was built using dictionaries of event terms gathered from WordNet [108]. Finally, the extracting of temporal expressions was achieved using TempEx [109] which takes as input a reference date such as for instance "next Friday", "tomorrow", or "August 21", plus some text, and also an indication of the part of speech of the latter — from their tagger. Classifying and ranking the extracted events was performed according to the frequency of tweets appearing with the same named entity and date (the higher the frequency, the more significant the event). In addition, events had to be tied to unique days, which means not appearing in most calendar days, this was done to avoid insignificant events.

A subsequent work which also extracted events was the work by Zhou et al. [110]. These authors wanted to improve the TwiCal [106] process, which extracts events by looking for representations of named entities, event phrases, calendar dates, event types, and locations. Each of these items of data was stored as part of a tuple.

The four-value tuple adopted was (y,d,k,l), where y stands for named entity, d for date, l for location, and k for event related keywords. First they performed a pre-processing step to assign values to these elements within the tuples. Hence, to recognize time expressions in order to fulfil the d tuple, they used SU-Time [111] method which is a temporal expression recognizer. This takes, as input, text such as "next week", or "tomorrow" plus the text's posting date and then outputs a more expressive date.

Extracting named entities (the y tuple value) from tweets was based on news articles published at (roughly) the same time that the tweets under examination were posted. Using the Stanford named entity recognizer [112], they looked for named entities in news articles. They decided to do this because tweets suffer from spelling mistakes and abbreviations, besides a study carried out by Petrovic et. al. in [104] found that events mentioned in tweets are also found in news articles at around the same time period. So they created a dictionary containing the recognized named entities from news articles, and used that dictionary to extract location named entities l and non-location named entities y from tweets. Finally, to find event related keywords k they used a POS tagger trained on tweets by [113], in which only words tagged as noun, verb, or adjective are taken account of. Stemming was applied to these words as well, and words appearing less than three times were removed.

The output of the pre-processing step *(y,d,k,l)* was used to find events via their proposed Latent Event Model (LEM). In the LEM model, an event is represented as a joint distribution of named entities, dates, locations and keywords. This representation encourages events sharing the same named entities and keywords, and appearing at the same time and location to be combined as the same event. They applied an experiment on a data set containing 2468 tweets already associated with 21 events in advance, from [104]. With their method of extracting events, they outperformed TwiCal [106], which they considered their benchmark, by over 7%.

A study by [114] was dedicated to identifying the content which was most relevant to events from Twitter in real-time. Their aim was to use the TF-IDF algorithm on documents, rather than on single tweets, in order to construct vectors. As the TF-IDF performs better on long paragraphs rather than short, noisy sentences (i.e. tweets).

They wanted to benefit from the use of hashtags (#) in tweets to identify events and the content relevant to events. Thus, they used the Twitter stream API to retrieve tweets and these tweets were then sliced according to their time of posting into frames. In every frame, all hashtags which appeared more than five times were considered potentially relevant hashtags, afterwards a hashtag based document was created for every such hashtag.

Next, they used the Twitter Search API to retrieve all the tweets which included at least one potentially relevant hashtag within a certain time period. All tweets related to any one of these hashtags are placed in a document corresponding to the hashtag. Subsequently, they stemmed every document and then analysed it by removing stop words, repeated characters, twitter notations, and URLs. The remaining words were tokenized on a unigram basis using Mahout. These tokens were the document based vectors and were then used to find event relevant content. The union of all these vectors created the hashtag-based vector.

A more recent attempt at event detection, described in [115], identifies bursty topics from Twitter using a sketch-based topic detection model. Bursty topics are detected via a two-step method: first, a “sketch” data-set maintaining the total number of all tweets, the occurrence of each word and the occurrence of each word pair was constructed. Bursty topics were identified using the sketch data-set. Second, to cope with scalability but maintain topic quality, a hashing reduction technique was applied.

A recent and detailed survey on research into event detection from Twitter streams can be found in [103].

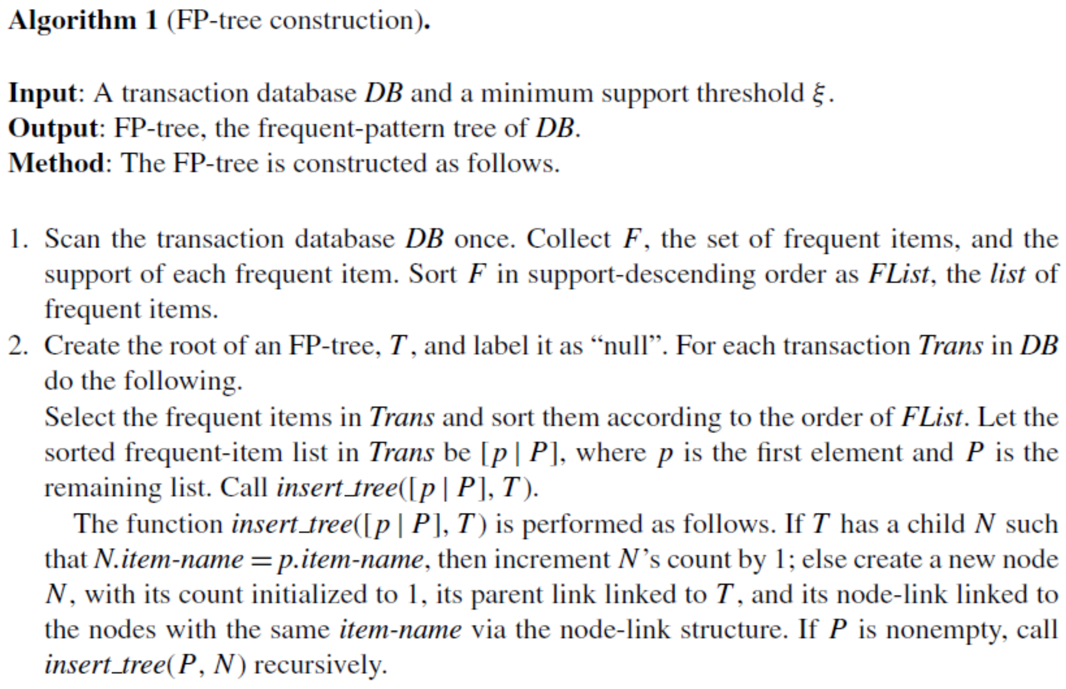
2.5 Frequent Pattern Mining

Frequent patterns are sets of items or transactions that occur in a dataset with a frequency of no less than a predefined threshold value, which is referred to as the *minimum support* [93]. In general, Frequent Pattern Mining (FPM) searches for repeatedly occurring relationships in a given dataset. The *minimum support* is a predefined value that is related to the frequency of occurrence of patterns. A pattern is said to be frequent if its support (occurrence frequency) is no less than the predefined minimum support value. An itemset or a transaction is formed of a collection of one or more items. An example of an itemset that may appear frequently in a supermarket database is milk, bread, eggs, where the items are milk, bread, and eggs. Frequent patterns have played an essential role in finding correlations, mining associations, and many more data analysis tasks.

FPM was first proposed for market basket analysis research by Agrawal et al. [116]. In that application, it was used to analyse customer shopping baskets in order to find associations between the items that they had bought. The FPM method introduces three basic frequent itemset mining methodologies: [32] a-priori [117], FP-Growth [118], and Eclat [119].

FP-Growth is a frequent itemset mining method which requires fewer numbers of database scans than the other techniques, necessitates no candidate generation, and works in a divide-and-conquer way. It is suitable for situations where there are very large numbers of database transactions and also for situations where there are relatively long patterns [3].

Generally, FP-Growth is a 2-step strategy, starting with the construction of a compact tree (the FP-tree) that is then mined to find frequent patterns (the mining is step 2): thus eliminating the need to mine the whole dataset. Specifically, step 1 starts by scanning the given dataset to find frequent items (only items satisfying the minimum support value criterion are retained), along with sorting the set of items or transactions in a descending order according to their support count. Then the FP-tree is built from a single root node, and such that there is a tree branch for each transaction in the dataset (this requires a 2nd dataset scan). Figure 2.3 shows the FP-tree construction process in more detail, wherein the dataset is only scanned twice in order to build the FP-tree. Step 2 consists of mining the constructed FP-tree to find the set of frequent patterns evident in the given dataset via the FP-tree (Figure 2.4 shows the overall picture regarding step 2). From the FP-tree, the mining starts with the least frequent item as the initial suffix of a pattern, and construct its conditional FP-tree. The conditional FP-tree consists of the set of prefix paths in the FP-tree co-occurring with that suffix. This results in the size of the dataset needing to be searched in order to find frequent patterns being reduced, only the conditional FP-tree need to be examined in order to find the frequent items related to the specific suffix. Recursively the same process is applied for each item until the most frequent item is reached. Concatenating the suffix pattern with the frequent patterns generated from its conditional FP-tree leads to the generation of frequent patterns.



Algorithm 1 (FP-tree construction)**.**

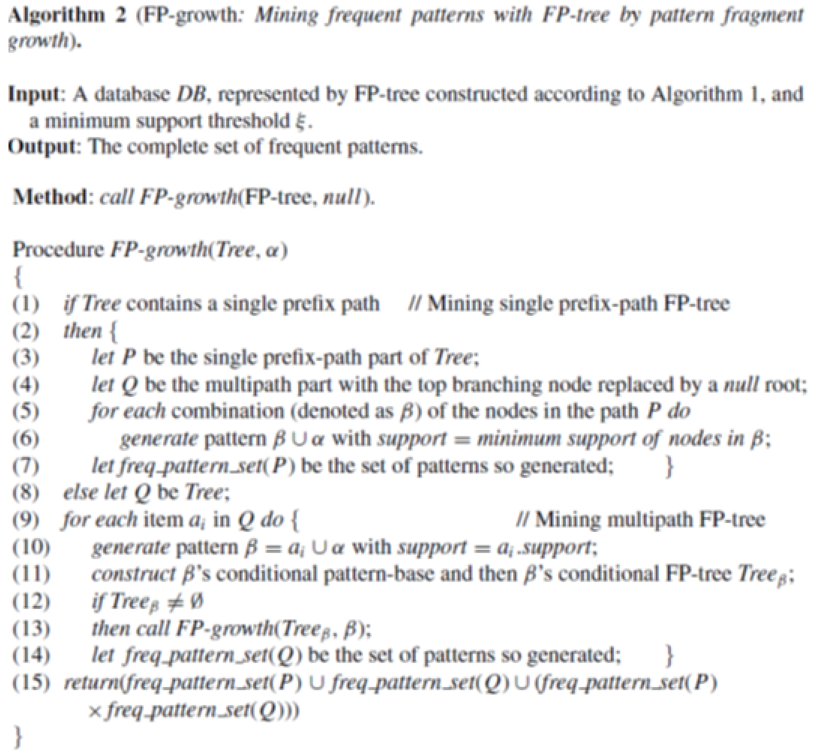
Input: A transaction database *DB* and a minimum support threshold ξ.
Output: FP-tree, the frequent-pattern tree of *DB*.
Method: The FP-tree is constructed as follows.

1. Scan the transaction database *DB* once. Collect F, the set of frequent items, and the support of each frequent item. Sort F in support-descending order as *FList*, the *list* of frequent items.
2. Create the root of an FP-tree, T, and label it as “null”. For each transaction *Trans* in *DB* do the following.
   Select the frequent items in *Trans* and sort them according to the order of *FList*. Let the sorted frequent-item list in *Trans* be $[p \mid P]$, where p is the first element and P is the remaining list. Call *insert_tree*($[p \mid P], T$).
   The function *insert_tree*($[p \mid P], T$) is performed as follows. If T has a child N such that *N.item-name* = *p.item-name*, then increment N's count by 1; else create a new node N, with its count initialized to 1, its parent link linked to T, and its node-link linked to the nodes with the same *item-name* via the node-link structure. If P is nonempty, call *insert_tree*(P, N) recursively.

Figure 2.3: FP-tree Construction (Algorithm taken from [3] page 58)

2.5.1 Frequent Pattern Mining from Online Data Streams

Online data streams share a number of features which are currently considered reasoning challenges. These data streams are of unknown size, their data arrival rate is irregular, and only a single scan, with no backtracking, is possible [120]. Examples of such streams are readings from sensors, Internet and web traffic, stock exchange data, etc. The traditional methods for data mining frequent itemsets in a static DB require a



Algorithm 2 (FP-growth: *Mining frequent patterns with FP-tree by pattern fragment growth*).

Input: A database *DB*, represented by FP-tree constructed according to Algorithm 1, and a minimum support threshold ξ.
Output: The complete set of frequent patterns.

Method: *call FP-growth*(FP-tree, *null*).

```
Procedure FP-growth(Tree, $\alpha$)
{
if Tree contains a single prefix path     // Mining single prefix-path FP-tree
then {
    let $P$ be the single prefix-path part of Tree;
    let $Q$ be the multipath part with the top branching node replaced by a null root;
    for each combination (denoted as $\beta$) of the nodes in the path $P$ do
        generate pattern $\beta \cup \alpha$ with support = minimum support of nodes in $\beta$;
    let freq_pattern_set($P$) be the set of patterns so generated;     }
else let $Q$ be Tree;
for each item $a_i$ in $Q$ do {                              // Mining multipath FP-tree
    generate pattern $\beta = a_i \cup \alpha$ with support = $a_i$.support;
    construct $\beta$'s conditional pattern-base and then $\beta$'s conditional FP-tree $Tree_\beta$;
    if $Tree_\beta \neq \emptyset$
    then call FP-growth($Tree_\beta$, $\beta$);
    let freq_pattern_set($Q$) be the set of patterns so generated;     }
return(freq_pattern_set($P$) $\cup$ freq_pattern_set($Q$) $\cup$ (freq_pattern_set($P$)
    $\times$ freq_pattern_set($Q$)))
}
```

Figure 2.4: The FP-tree Mining (Algorithm taken from [3] page 67)

number of DB scans. However this is not applicable to online data streams due to memory and computational constraints. Therefore traditional methods cannot be applied directly to data streams.

It is impossible to consider the whole data stream in a single scan, only a subset of it can be looked at. According to [121] mining frequent itemsets from data streams methods fall into one of the following categories: landmark [122], tilted-time window (fading) [123], or sliding window [124].

A landmark model considers all data from a specified point in time “a landmark” to the present time. Usually the start point is the beginning of the stream; and it treats all data equally within that period. A fading model works in the same way as the landmark model except it assigns different weights to different items of data. New data transactions are given higher weights than older ones, on the basis that it considers the latest data to be the most important. Finally the sliding window model uses a sliding window that slides over the data stream to find frequent itemsets. This window can either be a transaction based window, consisting of a fixed number of transactions, or a time based window of a fixed length of time.

Finding frequent itemsets in a datastream using the frequent pattern mining either depends on the complete arrival of transactions to form a batch (i.e. all transaction batches are of the same size), such efforts as [122–126], or depends on the timing aspect (e.g. an hour) to form a batch rather than the transaction count for finding frequent patterns, such efforts as [121, 127, 128]. A detailed review on mining frequent itemsets from online data Streams can be found in [129].

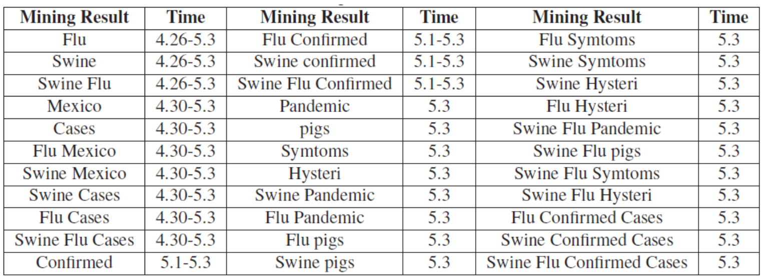
2.5.2 Event Identification from Twitter Stream Using FPM

Delaying the mining process until a batch is formed (which constitutes a fixed number of transactions) is not always feasible; this is the case when, for instance, identifying the frequent patterns from a Twitter stream. Thus mining depending on time stamps and regardless of the number of transactions which arrive makes more sense in these cases.

The first attempt at using Frequent Pattern Mining to detect topics from Twitter was in [4], where they used the Frequent Pattern Stream Mining Algorithm (FP-Stream) proposed by [123]. The FP-Stream algorithm uses the fading model to identify frequent patterns from a stream by constructing a frequent pattern tree to handle frequent and sub-frequent itemsets with a tilted-time window table for each frequent pattern that is able to answer to queries at multiple time granularities.

In order to be able to apply the FP-Stream algorithm, each tweet was treated as a transaction, and each word in the tweet was treated as an item. In addition, the FP-Stream tilted-time window table was updated to include a batch number, which corresponded to a batch for a certain time. This batch number was included because the researchers were looking for topics happening within certain time-frames, rather than based on complete arrival of data chunks. Two types of thresholds were used: the support (σ) and the error rate (ϵ). A pattern is frequent if its support is greater than σ, it is sub-frequent if its support is below σ but not below σ-ϵ, otherwise the pattern is infrequent. Both frequent and sub-frequent patterns were kept while the infrequent ones were dropped. Sub-frequent patterns were kept because they may become frequent later on.

An experiment was conducted by [4] on tweets collected from Twitter regarding the Swine Flu topic from 26 April till 3 May 2009. The tweets pre-processing phase was applied, which consisted of data cleaning and stop word removal. Each day's tweets (transactions) formed a batch; the size of a batch was not fixed. The support threshold was set on a fixed manner to $\sigma = 0.03$ and the error rate was set to $\epsilon = 0.001$. The frequent patterns found were considered Twitter hot topics; refer to Figure 2.5 for a table showing the hot topics found via the experiment.



Mining Result	Time	Mining Result	Time	Mining Result	Time
Flu	4.26-5.3	Flu Confirmed	5.1-5.3	Flu Symtoms	5.3
Swine	4.26-5.3	Swine confirmed	5.1-5.3	Swine Symtoms	5.3
Swine Flu	4.26-5.3	Swine Flu Confirmed	5.1-5.3	Swine Hysteri	5.3
Mexico	4.30-5.3	Pandemic	5.3	Flu Hysteri	5.3
Cases	4.30-5.3	pigs	5.3	Swine Flu Pandemic	5.3
Flu Mexico	4.30-5.3	Symtoms	5.3	Swine Flu pigs	5.3
Swine Mexico	4.30-5.3	Hysteri	5.3	Swine Flu Symtoms	5.3
Swine Cases	4.30-5.3	Swine Pandemic	5.3	Swine Flu Hysteri	5.3
Flu Cases	4.30-5.3	Flu Pandemic	5.3	Flu Confirmed Cases	5.3
Swine Flu Cases	4.30-5.3	Flu pigs	5.3	Swine Confirmed Cases	5.3
Confirmed	5.1-5.3	Swine pigs	5.3	Swine Flu Confirmed Cases	5.3

Figure 2.5: The Found Hot Topics in [4] (Table taken from [4] page 4)

A limitation in this research was the duration of the experiment, which was too short — 7 days only. In addition, every frequent topic was identified regardless of whether or not it was just an element of another, more comprehensive one; Figure 2.5 gives the topics identified. The hot topics found were rather general, especially when longer time periods were considered.

Another study, by Petkos et al. in [39], was also dedicated at detecting topics from Twitter. They applied a "softened" version of the FPM, namely the Soft Frequent Pattern Mining (SFPM) algorithm, which consists of three phases:

Phase 1 is term selection from the corpus under consideration. This is based on calculating the probability of a term occurring in the corpus at hand (a corpus of tweets related to the event under consideration) and the probability of it occurring in the reference corpus (a reference corpus is an independent corpus consisting of randomly collected tweets), as shown in Equation 2.4. Here, N_w is the number of times term w appears in the corpus, δ is a small constant (to regularize the probability estimate) set to 0.5 , and n is the number of term types which appear in the corpus.

$$P(w|corpus) = \frac{N_w + \delta}{(\sum_u^n N_u) + \delta n} \tag{2.4}$$

Afterwards, for every term in the studied corpus the ratio of the term occurring in both corpora is calculated by taking the probability of the term occurring in the studied corpus and dividing this by the probability of the term occurring in the reference corpus, as shown in Equation 2.5. Terms with higher ratios are selected as significant terms.

$$\frac{P(w|corpus_{new})}{P(w|corpus_{ref})} \tag{2.5}$$

Phase 2 is Co-occurrence-vector formation. After discovering what the significant terms are, in phase one, phase two is performed to detect topics. This is done by defining S, which contains the set of terms that is then forming a topic. S is greedy, and is created by adding terms that frequently occur with terms already in S. To calculate the similarities between a term, t, and the terms in S, a vector, D_s, was maintained for S and a vector D_t for the term t. The length of both vectors is n, where n is the number of documents (tweets) in the collection. The i^{th} element of D_s states the number of terms in S appearing in the i^{th} document, while the i^{th} element in D_t is a binary value indicating whether term t appears in the i^{th} document or not. S is expanded once the best match between the vector D_s and D_t is found. For the expansion to happen, the cosine similarity between the original term and its best match in S must exceed a certain threshold, otherwise expansion is stopped, as shown in Equation 2.6.

$$\theta(S) = 1 - \frac{1}{1 + exp((|S| - b)/c)} \tag{2.6}$$

This phase is applied for all selected terms from phase 1, each time S is initialised with term t. Refer to [39] for the full SFPM algorithm on how topics are found.

Phase 3 is the post-processing phase. In this phase the duplicated topics were removed by calculating the term similarity between each possible pair of topics, if this was greater than 75% then the smaller (less frequent) topic was removed.

An experiment was carried out in order to test the SFPM algorithm on three datasets relating to three different events (the U.S.A. Super Tuesday Primaries held on March 2012, the FA Cup Final held on May 2012, and the U.S.A. Elections held on November 2012). The method for evaluating the results was based on collecting topics manually for the three data sets using the mainstream media (the Wall Street Journal, CNN, Fox News, The Washington Post, the Guardian, and the Huffington Post). The total number of topics found in the Super Tuesday data-set was 22, for the FA Cup there were 13 topics, and for the U.S.A. Elections there were 64 topics. Also, for every topic a set of mandatory terms, optional terms and forbidden terms were identified. The test results showed that the SFPM algorithm performs well and it outperforms competing methods.

A drawback of the SFPM algorithm is that it tends to produce a great many topics — that are not necessarily significant or relevant. The authors of [39], when evaluating their identified topics, relied solely on the topic recall measure, and did not use the topic precision measure. Precision and recall, are the basic evaluation metrics used in information retrieval [130]. Precision measures relevance, while recall measures the completeness of results (we will discuss these metrics further, later in the thesis, when evaluating the topics we, ourselves, detected from text streams). In addition, applying phase three was a must, in this method, in order to get rid of duplicated topics. We believe that it is more practicable, in terms of processing resources, to stop the expansion process of S once it is discovered, at the time, that a duplication is occurring; this would make topic identification process faster and would reduce computational costs.

An improved version of SFPM [39] was presented in [41], where the data stream was split into windows, and the list of keywords used to refine Twitter stream were continuously refined to include new emerging keywords, if any.

A more recent study, [40], proposes a High Utility Pattern Clustering (HUPC) framework for detecting topics from micro-blog streams. This describes a two-step framework, starting with detecting representative High Utility Pattern (HUP)s from the micro-blog stream and then going on to grouping these pattern into topic clusters.

To identify the HUP patterns, a top-K HUP mining algorithm was applied, where K was a number of patterns which is specified in advance. They used the FP-Growth algorithm with a minimum support set to to zero or almost zero — in order to find all the frequent patterns. Then every pattern was scanned once more and was compared with all the other HUP patterns, in relation to a specified value (δ). This was in order to find out the overlap degree, which is a metric for being a HUP pattern, until k-patterns were found. Refer to [40] to view the top K-HUP mining algorithm.

Afterwards, clustering was applied in order to group related HUP patterns together in order to form a topic. The KNN process was used, based on the k^{th} nearest neighbour.

In this particular work they used a value of almost zero for the minimum support, as a result a huge number of patterns were found. To then restrict this, they selected the top 3000 most frequent terms from each day's tweet batch. In addition, they limited the number of required patterns to k-patterns, where k was determined in advance. This led to the problem that it was difficult to specify k, the required number of HUPs (i.e. number of required patterns) in advance.

2.6 Directional Change Approach

The Directional Change (DC) is an event-driven approach used for analysing financial time-series and summarizing price movements [34,35]. Traditional methods for identifying price movments and transitions in time-series data streams depends on physical time, such as by looking at daily closing prices. Physical time observations, which depends on a specific time unit (seconds, minutes, hours), fails to capture the full activity of price movements [34,131]. The DC, on the other hand, samples data at irregular time intervals using intrinsic time rather than a static physical time, which makes it able to picture significant points in price movements that the traditional physical time methods cannot. Intrinsic time was first used by [132], where they proposed a "transaction clock" based timing which ticks at every worldwide transaction. In the DC [34], intrinsic time is defined by directional change events (i.e. DC is an event-based-time view).

DC is based on a defined directional change threshold value, and yields two possible forms of events: a downturn event, and an upturn event. The period between a downturn event and the next upturn event is called a downward run, while an upward run is the period between an upturn event and the next downturn event [133].

In a downward run, the last low price, p_l, is continually updated to be the minimum of either the current market price *p(t)* or the last low price. Similarly, in an upward run, the last high price, p_h, is continually updated to be the maximum of either the current market price *p(t)* or the last high price. Initially the last low price, p_l, and the last high price, p_h, are both set to the initial market price *p(t_0)* [133].

In a downward run, an upturn event is said to have been found when the current price *p(t)* exceeds the last lowest price, p_l, by a given fixed threshold value, Δx_{dc}, see the formula in 2.7.

$$p(t) \geq p_l \times (1 + \Delta x_{dc}) \tag{2.7}$$

In an upward run, a downturn event is said to have been found when the current price *p(t)* is lower than the last highest price, p_h, by a given fixed threshold value, Δx_{dc}, see the formula in 2.8.

$$p(t) \leq p_h \times (1 - \Delta x_{dc}) \tag{2.8}$$

The starting point of a downturn event is called the downturn point (the point where the price last peaked), while the end point is called the downturn confirmation point (the point where the price has fallen by the fixed threshold from the downturn point).

On the other hand, the starting point of an upturn event is called an upturn point (the point where the price last reached a low), while the end point is called the upturn confirmation point (the point where the price has increased by the fixed threshold value from the price at the upturn point).

An Overshoot Event (OS) occurs at the end of the current directional change event and lasts until the beginning of the next directional change event. An OS can be of one of two types: a downward overshoot event following a downturn event or an upward overshoot event following an upturn event. Algorithm 1 from [6] defines the occurrence of DC and OS events in time T. Figure 2.6 shows the DC concepts illustrated via a graph — with reference to the FTSE 100 data stream. It shows a downturn event starting on the 1st Feb at 8:01 am (downturn point) and ending on 2nd Feb at 8:36 am (downturn confirmation point), then a downward OS event starting on 2nd Feb at 8:37 am and ending on the 11th Feb 9:30 (the beginning of the next upturn event).

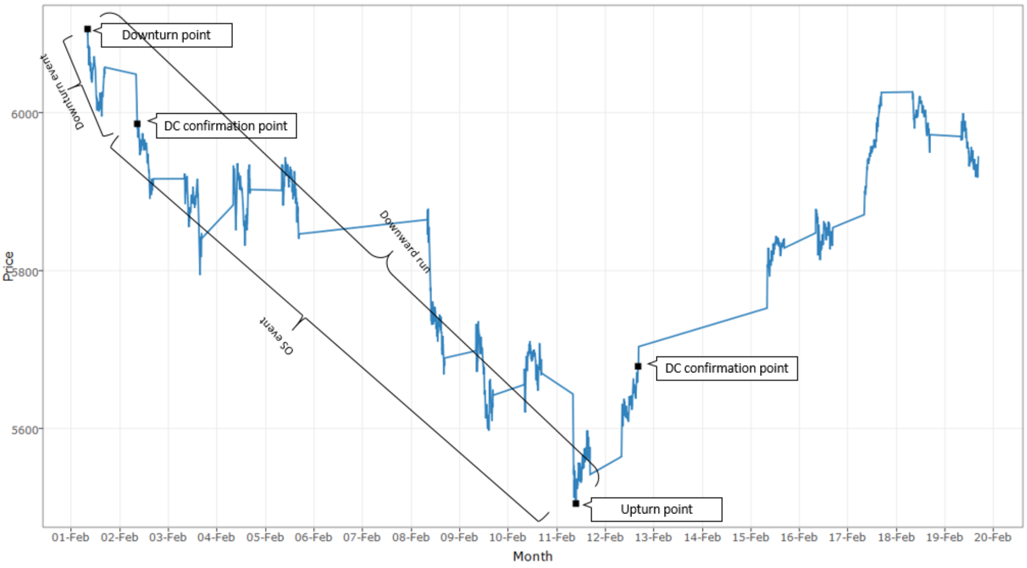


Figure 2.6: DC Concepts

The DC approach has shown its potential for studying and analysing financial time-series, in relation to the Foreign Exchange Market [6, 35, 42–46, 52, 133, 134], and has the potential to be applied to other market data [47–49]. Moreover, the DC approach has been included as an element in trading strategies [36, 37, 52, 134] and forecasting [44,46,47,50] studies. It has typically been applied with a fixed threshold, and a dynamic threshold may be more appropriate for markets whose nature is inherently dynamic.

Algorithm 1: Defining DC and OS events (Algorithm source [6])

```
Require: initialise variables ($DCevent$ is upturn event, $\Delta x_{DC}$ (Fixed ) $\geq 0$,
         $t_{DC,0} = t_{DC,1} = t_{OS,0} = t_{OS,1} = t$, $p_h = p_l = p(t)$ at time $t$)
if $DC$ $event$ $is$ $upturn$ $event$ then
    if $p(t) \leq p_h \times (1 - \Delta x_{DC})$ then
        $DCevent$ =downturn event
        $p_l = p(t)$
        $t_{DC,1} = t$ // End time for a downturn DC event
        $t_{OS,0} = t + 1$ // Start time for a downward OS event
    else
        if $p_h < p(t)$ then
            $p_h = p(t)$
            $t_{DC,0} = t$ //Start time for a downturn DC event
            $t_{OS,1} = t_1$ //End time for an upward OS event
        end
    end
else
    if $p(t) \geq p_l \times (1 + \Delta x_{DC})$ then
        $DCevent$ =upturn event
        $p_h = p(t)$
        $t_{DC,1} = t$ // End time for an upturn DC event
        $t_{OS,0} = t + 1$ // Start time for an upward OS event
    else
        if $p_l > p(t)$ then
            $p_l = p(t)$
            $t_{DC,0} = t$ //Start time for an upturn DC event
            $t_{OS,1} = t_1$ //End time for a downward OS event
        end
    end
end
```

2.7 Financial Market Data and Twitter

Several studies including, but not limited to [5, 135–146] have tried to make links between market data (e.g., stock prices) and tweets retrieved from Twitter. Some of this work has attempted to show that there is a correlation between stock prices and the posted tweets [5,142–145], while other studies have attempted to show that the sentiment of tweets may affect the stock prices and help in stock price prediction [5, 135–141].

Work has been undertaken by J Bollen, et al. in [5] to investigate whether public mood correlates with economic indicators. More precisely, they wanted to find out whether the mood of Twitter tweets can correlate with the prices included in the Dow Jones Industrial index.

Basically, they looked only at tweets which described the mood of the people who posted them (tweets containing for example: I feel, I am feeling, I'm feeling, I don't feel, etc.); tweets were collected day by day. Then the sentiment representing the public mood was evaluated by analysing the daily collected tweets using both the OpinionFinder [2] (OF) software (which measure mood in terms of whether it is positive or negative), and the Google Profile of Mood States (GPOMS) [147] lexicon, which measures the following six moods (calm, sure, kind, vital, alert, and happy).

The result of this step was the assigning of a sentiment measure to every day included in the experiment, so they were able to create a mood time-series using seven moods. Also they looked at the Dow Jones Industrial Average (DJIA) daily closing price from Yahoo Finance to help examine the correlation between the daily mood compounded from the tweets and the DJIA daily closing price. For this purpose, they used Granger causality; this states that if X causes Y, then changes in X will precede those in Y. It was then concluded that among the seven moods examined, the "calm" mood was more likely to represent changes in public mood and matched changes in DJIA values with a 3-4 day latency, see Figure 2.7.

[2] http://www.cs.pitt.edu/mpqa/opinionfinderrelease/

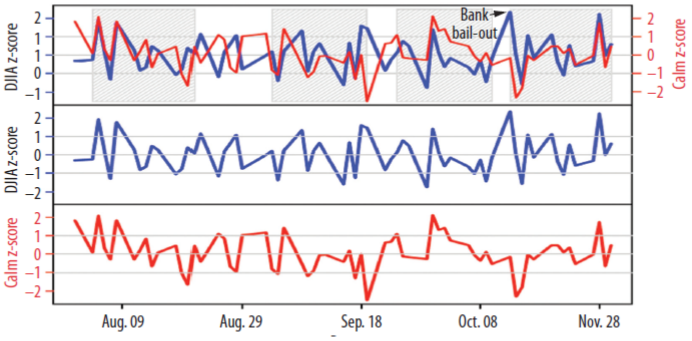


Figure 2.7: Correlation between Calm mood and DJIA prices (Image taken from [5] page 4)

Figure 2.7 shows 3 graphs, the red one is the "calm" time-series, the blue graph is the DJIA time-series, and the third one shows them both together, with the calm values on days (d-3) predicting the rise or fall in DJIA values on day (d).

A subsequent work, [135] (based on [5]), aimed at finding the correlation between public sentiment — inferred from Twitter — and market sentiment. Twitter data was used to predict the public mood, then the public mood along with the previous day's stock prices were used to predict the stock market movements. The Twitter stream was filtered such that only tweets that expressed feelings were looked at; they retrieved the DJIA prices from Yahoo! Finance. The public mood and the DJIA stock prices were fed into their proposed framework, which was a self-organizing fuzzy neural network model that learned to predict DJIA future prices.

Moreover they applied the Granger causality test [148] to find-out how predictive one indicator was of another after a specific time-lag. They found that with a 3 or 4 day time-lag, recognition of the calm and happy moods achieved an 75% prediction accuracy (in relation to the DJIA index).

Another study, [136], also looked at the effects of Twitter sentiment on stock price returns. The proposed procedure detect an event as a Twitter volume peak is identified; then the sentiment polarity was computed as being either positive or negative. Finally the event was related to stock prices. They collected tweets relating to certain stock companies — i.e., tweets containing the relevant cash-tag (e.g., "$NKE" for Nike) — and in terms of the market data they retrieved the stock company closing price. A low Pearson correlation and Granger causality was found across both time series for the investigated time period, but a significant dependence was found between Twitter sentiment and the abnormal returns which occurred during Twitter volume peaks.

In [145], the authors wanted to see if stock prices and the trading volumes are correlated with Twitter tweets. From Twitter they retrieved tweets relating to a particular company. A tweet found within a time interval was represented by an interaction graph, wherein nodes denoted tweets, users, URLs, and hashtags, while the relationship between nodes was expressed by edge labels such as retweet, authorship, and referencing. For stock data on the other hand they obtained selected stock's daily closing prices and trading volumes.

They found that a stock's trading volume was slightly more correlated with the number of connected components in its graph, rather than being correlated with the number of tweets. The stock price in contrast was not strongly correlated with any of the extracted features; it was only correlated with the number of connected components and only to some extent.

The majority of these research studies, which consider Twitter as an additional source of data when studying the financial markets, focus on tweets that mention the studied asset (a tweet is retrieved if the name of the stock is explicitly mentioned). In contrast, we are concerned with retrieving tweets that are related to a certain event (such as the GE 2015) that is not, necessarily, tied to the financial market.

2.8 Summary

In this chapter, we have discussed subjects related to analysing different types of data streams, also we have examined and analysed different related studies. We mainly focused on event detection methods from streams of data, typically a text stream (unstructured data stream) and a high frequency time-series data stream (structured data stream). In the following, we summarize the key findings in the literature:

Trying to identify events/topics from texts streams using the feature pivot method mainly depends on the targeted terms selection criteria. The FPM approach uses the support value to retain items (i.e. targeted terms) with frequency satisfying the support value in order to identify the frequent patterns (i.e. topics). A fixed support value has always been the case when using the FPM on streams of text data [4,39–41]. However, we think that there is a need for a dynamic support definition method to be introduced instead of a fixed support value to effectively cope with the dynamic nature of data streams.

Sampling high frequency time-series data streams depending on intrinsic time rather than fixed physical times is very crucial [34, 131]. The DC observes time-series data streams based on intrinsic time, and uses a threshold value as the basic determinant for identifying price movements (i.e. events). Using a fixed threshold has always been the case when employing the DC approach on high frequency time-series data streams [6, 36–38, 42–50, 134]. However, we think that it is more appropriate for the threshold value to be set in a dynamic manner rather than having it as a fixed a-priori one, so that to be able to detect events with different magnitudes.

Combining an additional data stream when analysing time-series data streams focus on streams that mention the studied assets or shares explicitly. We will look at social media text streams that are not related to financial markets. We are interested in developing methods for identifying events from different types of data streams: time-series streams/structured data streams and social media streams/unstructured data streams and see if we can identify any possible correlations between them.

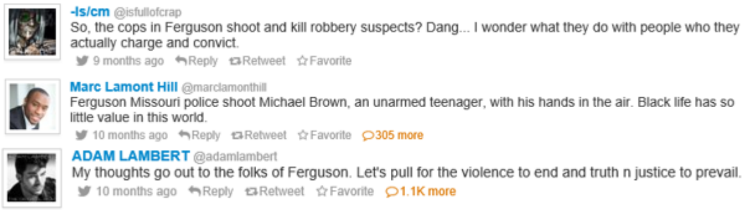
Chapter 3

Social Network Stream Analysis and Event Identification

3.1 Introduction

In this era of Social Media and Social Network, almost everyone has now become a virtual broadcaster, sharing an incredible number of messages. This has been facilitated by the widespread use of smart phones and the availability of mobile networks. The Pew Research Centre survey in 2015 on social network users [149] showed that three out of four people online use social networks. Even more surprisingly, it showed that it is not only young people who are increasingly using social networks, but that the number of people of age 65 and over who use these has more than tripled since 2010. We are interested in benefiting from the availability of data in Social Network (SN) streams to develop methods for detecting and identifying events.

Sometimes events spread faster on social media rather than via other forms of media, as they are widely discussed there. This is clearer with regards to unplanned events than already known and planned ones. Consider, for example, the shooting event that happened in Ferguson, Missouri, USA, which turned into a national and international news item with 3.6 million tweets posted from the 9th of August (the day of shooting) until the 17th of August. The first story emerged on Twitter before any other news channel reported it[1]. Furthermore, when tracking events published on social networks, one can sense people's reactions towards the event. In comparison, news wires are more formal and objective; refer to figures 3.1 and 3.2 for a comparison between tweets posted from news channels and those originating from public users regarding the Ferguson shooting event.

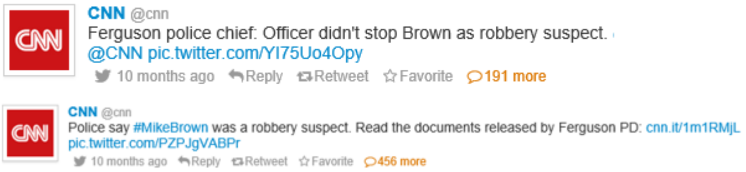


-Is/cm @isfullofcrap
So, the cops in Ferguson shoot and kill robbery suspects? Dang... I wonder what they do with people who they actually charge and convict.

Marc Lamont Hill @marclamonthill
Ferguson Missouri police shoot Michael Brown, an unarmed teenager, with his hands in the air. Black life has so little value in this world.

ADAM LAMBERT @adamlambert
My thoughts go out to the folks of Ferguson. Let's pull for the violence to end and truth n justice to prevail.

Figure 3.1: Some Ferguson Public Tweets



CNN @cnn
Ferguson police chief: Officer didn't stop Brown as robbery suspect. @CNN pic.twitter.com/YI75Uo4Opy

CNN @cnn
Police say #MikeBrown was a robbery suspect. Read the documents released by Ferguson PD: cnn.it/1m1RMjL pic.twitter.com/PZPJgVABPr

Figure 3.2: Some CNN Ferguson Tweets

An event is defined as a real world occurrence of something with an associated place and time period [86]. Event identification is the process of looking for events.

In this work, and using a SN stream (text stream), we want to develop methods for detecting the daily occurring events —if any. We aim to adopt the FPM [32] approach as being one that is capable of running on streams of data even more effectively when using a dynamic support instead of a fixed given one. In order to run any FPM algorithm, the support threshold, which is a metric for item retrieval, must be specified in advance. Any item with a frequency greater than or equal to the minimum support value is retrieved. Using a fixed support value on a data stream consisting of several windows (for instance, where each window is related to a single day's posts or messages) is not always effective. Such windows may be of different sizes, and hence, the fixed support value will function differently depending on each window size. Sometimes (for some windows) it will properly identify events, while at other times (on other windows representing different circumstances) it will not be able to identify "true" events.

Thus, using the FPM method with a dynamically defined support value, we want to identify the daily occurring frequent patterns which represent the detected events. Our premise is that a dynamic support that is defined daily can cope better with the nature of data streams (i.e. text streams, where the number of daily posts or messages is not fixed nor is it predetermined). An event is identified by tracking topics which

[1] http://www.pewresearch.org/fact-tank/2014/08/20/cable-twitter-picked-up-ferguson-story-at-a-similar-clip/

are widely discussed and frequently mentioned within the investigated SN text stream. Given the close link between events and topics, in the remainder of the thesis we use the two terms interchangeably when we refer to events being identified from SN streams.

Our motivation is to provide an event detection method that is capable of identifying events from a text stream regardless of the window size, using a dynamically defined threshold based on people's engagement with a major event (to replace the fixed one which is normally used). This framework can be utilised by any person or software that is interested in knowing the effects and the growth of major events — to discover the source topics/events.

Typically, from every day's window-batch (each one consisting of a day's text-posts) we have to dynamically define an appropriate support value, and then use that support value to detect the occurring events (i.e. topics). The support is a critical value, an accurate definition of it can lead to accurate and valid identified events. Moreover, a low support value may lead to mixed topics or events with too many selected terms, while on the other hand, with high support values fewer terms are selected, and this may lead to very generic topics or events. Thus each day, we aim to define the support value based on the window size and the occurrence and frequency of keywords.

The rest of this chapter is organized as follows. In the following section (section 3.2) we show our event detection framework and the support definition method; this is the core of the chapter. Section 3.3 begins by describing the collected text data, then presents the applied experimental work, and concludes with an evaluation of the detected events/topics using our framework and an alternative framework. Finally the chapter ends with a summary and conclusions in section 3.4.

3.2 Developing Methods for Event Identification

Before describing our framework for event identification, we first provide some definitions and clarifications. Let a SN stream S represent the text-posts present at a given time interval T. Let the current window-batch S_T, represent all the recent text-posts arriving in a fixed time interval (e.g. a single day), and so the text-posts that arrived within the previous time interval (i.e. previous day), will then belong to the previous window-batch, S_{T-1}. Thus, a SN stream S consisting of a sequence of window-batches arriving over time with unknown ending time is represented as $S = \{S_1, S_2, S_3, ..., S_{T-1}, S_T, ...\}$, where T is the current time interval.

Each text-post, s_j, belongs to a window-batch S_i, and is represented as a bag-of-words [150], where $S_i(i = 1, 2, ..., T)$ and $(j = 1, 2, ..., m)$, m is the number of text-posts in a window-batch S_i. In addition, a window-batch S_i size is not fixed, it may consist of any number of text-posts. Each window-batch S_i will have a dynamically defined support value Spp_i depending on that window-batch's received text-posts. The current window-batch, S_T, which has the dynamic support Spp_T may have no detected events at all $E_T = \{\varnothing\}$, or multiple N events $E_T = \{E_1, E_2, ..., E_N\}$.

In order to process data from a SN stream and identify events, a number of steps are required in our approach. These are illustrated in Figure 3.3. There are three main components to be applied on each window-batch: Support definition, Detection of Frequent Patterns (FPs), and Post-processing. Once a window-batch for the current time interval (S_T) is received (i.e. the text-posts posted during the current day are received) from the SN stream S, we want to dynamically set its support value (Spp_T). In the next step, and by using the support value (Spp_T), we want to employ the FPM algorithm in order to identify and detect the frequent patterns in each window-batch. Finally a post-processing step is carried out on the frequent patterns which have been found in order to represent the detected events more concisely and compactly.

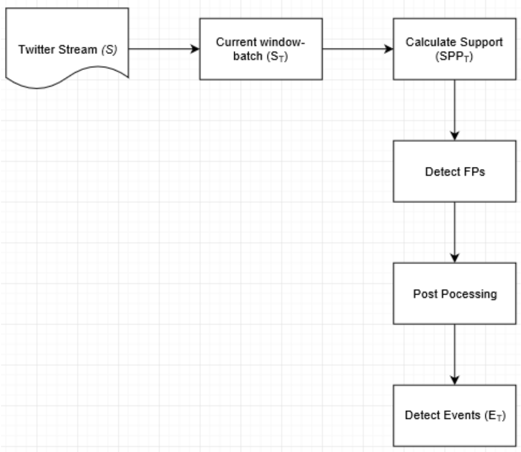


Figure 3.3: Event Detection Abstract Model

3.2.1 Dynamic Support Definition Method

The FPM is an approach for extracting and finding the frequent patterns that occur within a dataset. A pattern is said to be frequent if its support (frequency of occurrence) is greater than or equal to the predefined minimum support value [93].

When using the FPM on a text stream, the support is a metric used for term selection and retrieval. Terms with a frequency which satisfies the support value are retrieved while the others are ignored. Each day (for each window-batch S_i) a different support value (Spp_i) will be set, depending on that day's text-posts.

We seek to develop a dynamic support value that has the potential to allow for the detection of events. In particular, a low support value may lead to a very large number of detected topics or events (frequent patterns) with too many selected terms. While on the other hand with a high support value fewer terms are selected and this can lead to very generic and vague topics or events. Hence, we must aim to obtain support values that are proportional to and reflect the window size. This means a higher support value for big windows and a lower one for small sized windows.

Essentially, for each window-batch (S_i), the support value should be defined in a way that it can enhance the terms' selection criteria. Thus, in order to select the targeted terms, a number that is less than the number of distinct terms (in a window-batch the distinct terms (d_t) are the number of terms without repetition) is needed. If we divide the distinct number of terms (d_t) by the total number of terms (t_t stand for all terms with repetition), then we will get a number less than 1 unless the number of distinct terms are equal to the total number of terms ($d\_t = t\_t$ in this case, all terms are occurring only once without any repetition). As a result we define the Spp_T as in Equation 3.1. By Equation 3.1 a number that is less than the number of distinct terms is found for the current window-batch S_T.

$$Spp_T = d\_t \times (\frac{d\_t}{t\_t}) \tag{3.1}$$

Consider for example the number of $d\_t = 2000$ in two different windows S_1 and S_2, while the number of total terms (number of terms with repetition) $t\_t = 4000$ in S_1 and $t\_t = 5000$ in S_2, then using Equation 3.1 the $Spp_1 = 1000$ and $Spp_2 = 800$. Window S_2 was larger and with higher terms repetition than window S_1, however Spp_2 was lower than Spp_1. Although the number of total and distinct terms may provide some indication of frequency, this method fails to reflect or take into account the window size. Hence, this method for defining the support value was discarded.

From the previous support definition method, it was realized that the definition of the support value in a text stream should mainly focus on terms frequency, as it gives an impression on the window size and terms occurrence. So, we decided to calculate the average frequency of terms in each window batch S_i, as this will help in measuring the terms repetition size. Equation 3.2 shows how the average (mean) frequency of the terms for the current window-batch S_T is calculated, where d_t is the number of distinct terms in the current window S_T, and f_x is the frequency of each term belonging to the current window-batch S_T, where $x =, 1, 2, ..., d\_t$.

$$Avg(S_T) = \frac{1}{d\_t} \sum_{x=1}^{d\_t} f_x \tag{3.2}$$

In most windows the number of terms whose frequency exceeds the average frequency will be relatively large, and we are looking for patterns that are truly frequent and are as concise as possible. We can conclude that it is insufficient to use the average frequency of terms $Avg(S_T)$ found in a window-batch, on its own, as a metric for term retrieval, the support must be defined using additional information as well.

Next, consideration was given to using a window's median value $median(S_i)$ as an indicator conveying potentially relevant information that could be used as part of the support threshold definition. The median is a statistical value that separates higher values from lower ones, it is the middle value of a sorted list [151]. The median of a list of terms sorted according to term frequency is the middle frequency of that list. This also means that half of the terms are with frequency lower than the median and the other half are of frequency higher than the median. We considered this indicator because we were looking for support values that are greater than the average frequency and at the same time are proportional to the window size. A high median value is found in windows where terms occur with high frequencies, as the middle value of a list sorted according to such frequencies will still be high. While a lower median value is found when terms are less frequent, as the middle frequency of a list sorted according to such frequencies will be relatively low.

Thus, for the current window-batch S_T, a list (L_T) of distinct terms along with their frequencies (f_x) was sorted in descending order according to frequency $L_T = \{f_1, f_2, ..., f_H\}$, where f_H is the highest term frequency in window-batch S_T. The median for that window-batch $(median(S_T))$ was then the frequency of the middle item of that list ($f_{\frac{H}{2}}$ or $\frac{f_{\frac{H}{2}}+f_{(\frac{H}{2}+1)}}{2}$ depending on the number of items f_x in list L_T). Equation 3.3 shows this method for the calculation of the support for the current widow-batch, which multiplies the average frequency of the terms for the current window $Avg(S_T)$ by that window's median value $median(S_T)$.

$$Spp_T = Avg(S_T) \times median(S_T) \tag{3.3}$$

Using Equation 3.3 the average frequency of the terms in a window is multiplied by a number that represents the middle frequency value in that window. The middle frequency value is dependent on the size of the term frequencies in that window, i.e. the higher the terms frequencies the higher is that value.

An issue which needs to be taken into account when defining a dynamic support value is very small sized windows, in another words less active days (i.e. windows) in a text stream need more attention. Very small size windows will result in defined support values which are also very small. This in turn could lead to vague or insignificant detected events (as such a threshold is easily satisfied); this is especially a problem in relation to streams belonging to social networks because such are full of rumours and/or fake posts [152, 153]. Consider for example, a support value of 20, this means the terms present in any tweet which is retweeted at least 20 times satisfy the support value. A support value which is too low will lead to finding of frequent patterns which are vague and do not represent any event.

Therefore, a stricter support value is needed for less active days (window-batches with fewer text-posts) to avoid detecting spurious events. So, depending on the number of text-posts found in a window-batch (window-batch size), the support value will be defined by either Equation 3.4 or 3.5. Equation 3.5 is the small sized window equation, where we double the median value in order to have stricter support value. The determination of the window size as being large or small sized window was regarded as a 2-class big or small classification problem. Hence, a Logistic Regression model is built based on a training phase which is then applied on a testing phase, this is discussed later in the experimental work section. So, we define the support value depending on the size of the window as follows:

Large window-batch:

$$Spp_T = avg(S_T) \times median(S_T) \tag{3.4}$$

Small window-batch:

$$Spp_T = avg(S_T) \times (2 \times median(S_T)) \tag{3.5}$$

In Algorithm 2, we present the description of the Dynamic-FPM (D-FPM) algorithm adopted for the dynamic support value definition, which is the basis of our event detection framework from SN streams (i.e. text streams). It takes as an input the SN stream S, and return the support defined value for each window-batch. In lines 3 and 4, both the total number and the distinct number of terms are calculated in order to find the average and median values in lines 5 and 6. Depending on the window-batch size a suitable support definition method is chosen in lines 7-12. Each window-batch's defined support value is used for retrieving terms with frequencies greater than or equal to the support.

Algorithm 2: Dynamic Support Definition Method (the D-FPM)

```
   Input  : $SNstream$: Social Network stream
   Output: $Spp_T$: the support calculated value for the current window
1  for each incoming window $S_T$ in SNstream do
2      $No.tweets$= counttweets($S_T$)
3      $d\_t$= countdistinctterms($S_T$)
4      $t\_t$= countallterms($S_T$)
5      $avg(S_T)$ = calculateavg($d\_t$,$S_T$)
6      $median(S_T)$ = calculatemed($d\_t$,$S_T$)
7      $BigWindow$= ClassifyWindow($No.tweets$)
8      if $BigWindow$ then
9          $Spp_T = Avg(S_T) \times median(S_T)$
10     else
11         $Spp_T = Avg(S_T) \times (2 \times median(S_T))$
12     end
13     return $Spp_T$
14 end
```

3.3 Experimental Work

In the experimental work section, we want to employ a FPM method to text data streams using our dynamic support definition method (the D-FPM in Algorithm 2) in order to detect the occurrence of events (i.e. topics). Hence, this section begins by showing the text streams collection and preparation steps. After that, we present how the support values are dynamically defined for each window-batch in the text stream. In the following sub-section (Sub-section 3.3.3), we apply the FPM for each window-batch (S_i) using its defined dynamic support (Spp_i) in order to identify the occurring events. Next, we discuss and analyse the identified events from the investigated text streams. Finally, this section ends by evaluating the identified events using our event detection framework (the D-FPM) and an alternative framework (the SFPM) using Precision and Recall metrics.

3.3.1 Data Collection, Description and Preparation

We aim to develop a method to identify events that are occurring or taking place in a text stream. To focus the experimental work, rather than merely collecting data on any event at random, we decided to consider specific large scale events that would be unfolding over a period of time and which people would be talking about. Hence, we had two text streams, one related to the UK General Election 2015 (GE 2015) and the other related to the Greece Crisis 2015. The retrieved posts which formed the text streams, were placed in a MongoDB[2], which is a NOSQL database that is capable of handling unstructured data. We used the Eclipse IDE[3] for Java to implement our work.

From the Twitter stream and using the Twitter streaming API[4] which facilitates the extraction of tweets in real-time or near real-time, based on certain query parameters such as: certain keywords, certain locations, etc. We obtained more than one million tweets relating to the GE 2015 event; these were collected on weekdays from 9am-5pm (to be in-line with our time-series stream which is focused on stock prices) in the period from 15-4-2015 until 26-5-2015. We queried the Twitter API for the following terms (British Elections, GE2015, VoteGE15, GE15, General Elections) in order to collect tweets related to the GE 2015. In addition, we obtained more than 150k tweets collected in the period from 29-6-2015 until 16-7-2015 for the Greece crisis 2015 event using the following terms (greece crisis, greece bailout, greece referendum, grexit, greece eu, greece eurozone). Refer to Table 3.1 for the data streams collection summary, and to Appendix 1 for more details regarding the tweets collection process. As the GE 2015 stream was larger and had more windows than the other streams, we will focus mainly on this when we come to showing the details of the SN stream and snapshots of the analysis.

[2] https://docs.mongodb.com/

[3] https://www.eclipse.org/downloads/packages/

[4] https://dev.twitter.com/tags/streaming-api

Event	**No. of tweets**	**no. of windows**
General Election 2015	1131926	29
Greece Crisis 2015	151804	17

Table 3.1: Data collection summary

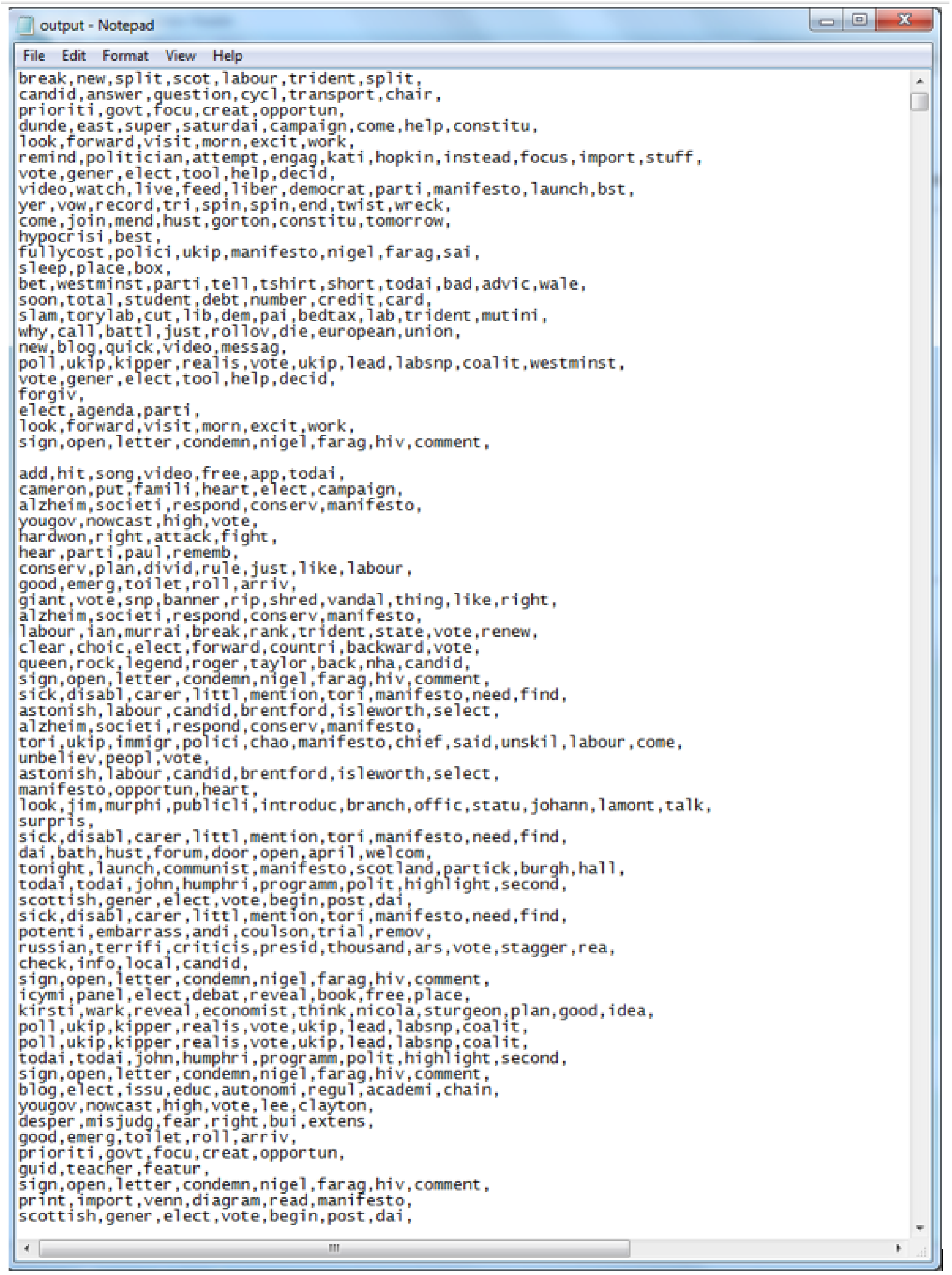
Thus, the first step was connecting to MongoDB, and the next was retrieving the tweets text along with the date and time the tweets were posted. After this we tokenized every tweet text to tokens, not only taking into account white space but also keeping in mind mentions @s, hashtags # and URLs [154]. Subsequently the pre-processing step was carried out: this begins by removing punctuations, mentions, hashtags #, URLs, and stop words [5]. Moreover some words in tweets may be written with repeated letters e.g., “yesssssss”; this is because Twitter is a social network platform that is used quite informally by public users, thus we had to remove excess repeated letters where they occurred within words. Following this, we applied the Porter stemming algorithm [155] by using the Apache Lucene Snowball Library [6] on those tokens in order to stem them and bring them back to their root words. As a result of this processing, a text file was created containing every tweet, each on an individual line with its tokens separated by commas. Figure 3.4 shows a snap shot of a text file containing some tokenized tweets.

As mentioned before, we are planning to use the FPM with the added ability to dynamically define the support value. Thus, before applying the FPM on the Twitter stream we had to choose between either the landmark, the fading, or the sliding window model for scanning the data stream [121]. We found that the window model was the one which most suited us because we are interested in finding topics that originate every single day (to be able to reason across the identified topics and changes in the stock prices) rather than topics occurring from the beginning of the stream till the current stream time point.

Therefore in the next step, we had to define the window size which will be sliding over the tweet tokens to find the occurrence of topics. Depending on the date on which a tweet was posted, it will belong to a particular window. The total number of tweets in each window is not fixed, for example in the GE 2015 data stream the total number of tweets on the first day of our tweet collection, which was 15th of April 2015, was 23836 tweets, while on day 15, the 6th of May 2015, there were 57348 tweets. Figure

[5] http://xpo6.com/list-of-english-stop-words/

[6] http://snowball.tartarus.org/download.html



```
break,new,split,scot,labour,trident,split,
candid,answer,question,cycl,transport,chair,
prioriti,govt,focu,creat,opportun,
dunde,east,super,saturdai,campaign,come,help,constitu,
look,forward,visit,morn,excit,work,
remind,politician,attempt,engag,kati,hopkin,instead,focus,import,stuff,
vote,gener,elect,tool,help,decid,
video,watch,live,feed,liber,democrat,parti,manifesto,launch,bst,
yer,vow,record,tri,spin,spin,end,twist,wreck,
come,join,mend,hust,gorton,constitu,tomorrow,
hypocrisi,best,
fullycost,polici,ukip,manifesto,nigel,farag,sai,
sleep,place,box,
bet,westminst,parti,tell,tshirt,short,todai,bad,advic,wale,
soon,total,student,debt,number,credit,card,
slam,torylab,cut,lib,dem,pai,bedtax,lab,trident,mutini,
why,call,battl,just,rollov,die,european,union,
new,blog,quick,video,messag,
poll,ukip,kipper,realis,vote,ukip,lead,labsnp,coalit,westminst,
vote,gener,elect,tool,help,decid,
forgiv,
elect,agenda,parti,
look,forward,visit,morn,excit,work,
sign,open,letter,condemn,nigel,farag,hiv,comment,

add,hit,song,video,free,app,todai,
cameron,put,famili,heart,elect,campaign,
alzheim,societi,respond,conserv,manifesto,
yougov,nowcast,high,vote,
hardwon,right,attack,fight,
hear,parti,paul,rememb,
conserv,plan,divid,rule,just,like,labour,
good,emerg,toilet,roll,arriv,
giant,vote,snp,banner,rip,shred,vandal,thing,like,right,
alzheim,societi,respond,conserv,manifesto,
labour,ian,murrai,break,rank,trident,state,vote,renew,
clear,choic,elect,forward,countri,backward,vote,
queen,rock,legend,roger,taylor,back,nha,candid,
sign,open,letter,condemn,nigel,farag,hiv,comment,
sick,disabl,carer,littl,mention,tori,manifesto,need,find,
astonish,labour,candid,brentford,isleworth,select,
alzheim,societi,respond,conserv,manifesto,
tori,ukip,immigr,polici,chao,manifesto,chief,said,unskil,labour,come,
unbeliev,peopl,vote,
astonish,labour,candid,brentford,isleworth,select,
manifesto,opportun,heart,
look,jim,murphi,publicli,introduc,branch,offic,statu,johann,lamont,talk,
surpris,
sick,disabl,carer,littl,mention,tori,manifesto,need,find,
dai,bath,hust,forum,door,open,april,welcom,
tonight,launch,communist,manifesto,scotland,partick,burgh,hall,
todai,todai,john,humphri,programm,polit,highlight,second,
scottish,gener,elect,vote,begin,post,dai,
sick,disabl,carer,littl,mention,tori,manifesto,need,find,
potenti,embarrass,andi,coulson,trial,remov,
russian,terrifi,criticis,presid,thousand,ars,vote,stagger,rea,
check,info,local,candid,
sign,open,letter,condemn,nigel,farag,hiv,comment,
icymi,panel,elect,debat,reveal,book,free,place,
kirsti,wark,reveal,economist,think,nicola,sturgeon,plan,good,idea,
poll,ukip,kipper,realis,vote,ukip,lead,labsnp,coalit,
poll,ukip,kipper,realis,vote,ukip,lead,labsnp,coalit,
todai,todai,john,humphri,programm,polit,highlight,second,
sign,open,letter,condemn,nigel,farag,hiv,comment,
blog,elect,issu,educ,autonomi,regul,academi,chain,
yougov,nowcast,high,vote,lee,clayton,
desper,misjudg,fear,right,bui,extens,
good,emerg,toilet,roll,arriv,
prioriti,govt,focu,creat,opportun,
guid,teacher,featur,
sign,open,letter,condemn,nigel,farag,hiv,comment,
print,import,venn,diagram,read,manifesto,
scottish,gener,elect,vote,begin,post,dai,
```

Figure 3.4: Example of Tokenized tweets from GE 2015 (each token sepreated by comma)

3.5 illustrates the required steps that are applied as a starting point to our work.

3.3.2 Setting the Dynamic Support Values

Here, we demonstrate how we applied Equations 3.4 and 3.5 to our SN text streams in order to define the support value for each window-batch S_i. We will also show the

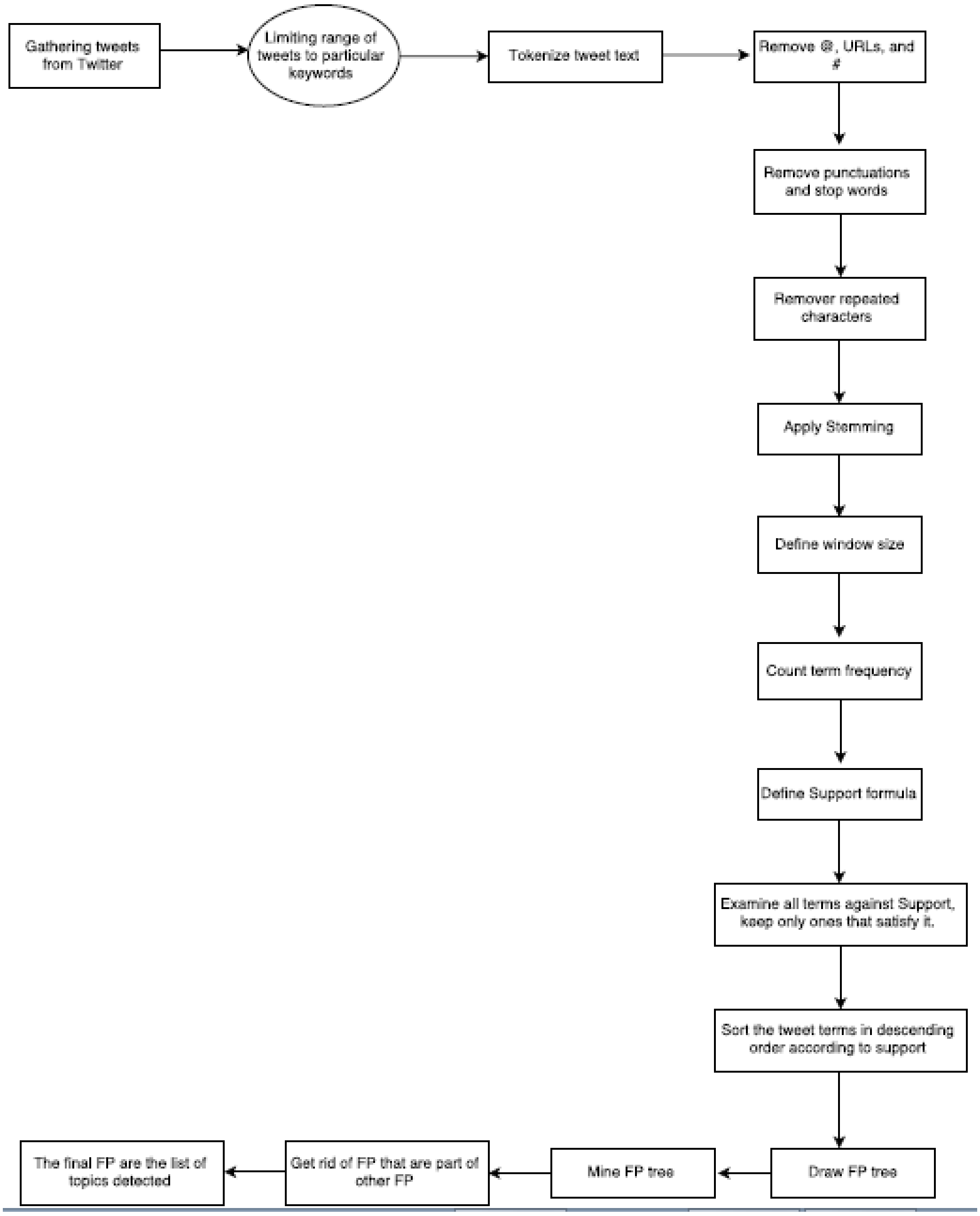

Gathering tweets from Twitter
Limiting range of tweets to particular keywords
Tokenize tweet text
Remove @, URLs, and #
Remove punctuations and stop words
Remover repeated characters
Apply Stemming
Define window size
Count term frequency
Define Support formula
Examine all terms against Support, keep only ones that satisfy it.
Sort the tweet terms in descending order according to support
Draw FP tree
Mine FP tree
Get rid of FP that are part of other FP
The final FP are the list of topics detected

Figure 3.5: Event identification from Twitter main steps

support values for all window-batches in the GE 2015 stream and the Greece crisis 2015 stream.

Initially, we simply counted the frequency of each term across all the text-posts which belonged to a particular window-batch; in other words, we counted the frequency of each term on a daily basis. We found that terms occurring only once (with a frequency of one) are almost equivalent to one-third of the total number of distinct terms in a window. Refer to Table 3.2 to see the number of distinct terms (we mean by distinct terms, all the terms without repetition) in each window-batch with and without terms with a frequency of one. As terms which appear only once will not help in finding the important and frequent patterns, they are removed as a preliminary step. Figure 3.6 shows the frequency value of some of the terms found in a single window-batch —displayed in a text file.

Refer to Figure 3.7 and to Table 3.3 to see the number of terms which exceeded the average frequency of the terms in their respective windows and the number of terms which exceeded their respective support values, in comparison to the total number of distinct terms in each window. Furthermore, Figure 3.8 provides a closer look at the comparison between the number of terms which exceed the average frequency of the terms in their respective windows and the number of terms which exceed the support value. The large difference between the number of terms exceeding the average frequency and the number of terms exceeding their support value can clearly be seen. The number of terms exceeding their support value is a lot fewer than the ones exceeding their respective average, for all the 29 window-batches from the GE 2015 stream. This means that the terms with frequencies exceeding their support value are indeed frequent and so are more important.

From the data stream which was related to the GE 2015, we realised that the frequent patterns which were found just a few days after the election (more precisely from Monday 13th of May onwards) were less related to events, but instead were concerned with contentious issues and focused on disseminating information of various kinds. The support values which were set in this period fell dramatically, as a result of the lower numbers of posted tweets. Consequently keyword co-occurrence in these windows was reduced as well.

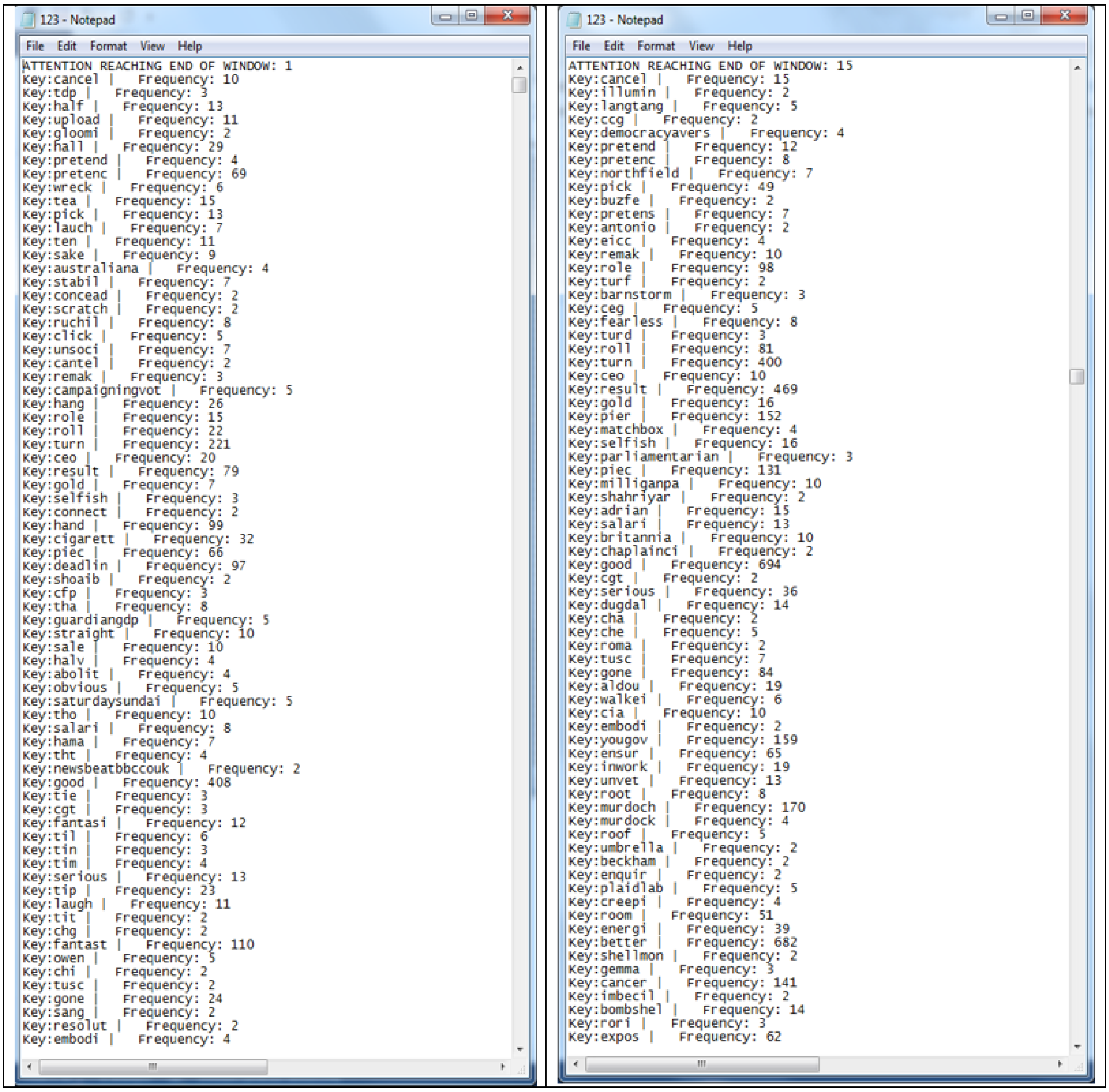
Table 3.4 presents the number of posted tweets, the number of keywords, and the support values set for the relevant period. It is clear that from window 20 onwards, the

Window no.	Distinct terms with frequency of 1	Distinct terms without frequency of 1
1	9973	6303
2	9906	6364
3	10231	6518
4	10484	6719
5	10930	7021
6	10273	6660
7	10114	6592
8	10189	6441
9	11463	7463
10	10919	7035
11	12783	8144
12	12793	8240
13	13106	8012
14	11294	6896
15	16834	10387
16	19543	12152
17	30445	18623
18	47752	28614
19	9154	4958
20	6609	3571
21	5508	3007
22	4714	2553
23	4403	2411
24	3581	1960
25	3054	1755
26	2610	1449
27	2579	1486
28	2815	1651
29	2023	1248

Table 3.2: Number of terms in each window with and without a frequency of 1 from GE 2015

number of text posts fell significantly and so did the support values. In window 26 for example, the support value was set to 21, this means that the terms belonging to any tweet which was retweeted 21 times or more satisfied the support threshold. As a result, the stricter version of the support definition method, given in Equation 3.5, had to be used instead.

A window is considered a small sized window in the GE 2015 stream, if the number of posted tweets in it is less than one-third of the average number of total tweets in the stream. One-third of the average was set as the determinant for a small sized window, because when comparing the detected, purported events with news related events found in the WWW, we noticed the events detected in those windows were insignificant and



```
ATTENTION REACHING END OF WINDOW: 1
Key:cancel |   Frequency: 10
Key:tdp |   Frequency: 3
Key:half |   Frequency: 13
Key:upload |   Frequency: 11
Key:gloomi |   Frequency: 2
Key:hall |   Frequency: 29
Key:pretend |   Frequency: 4
Key:pretenc |   Frequency: 69
Key:wreck |   Frequency: 6
Key:tea |   Frequency: 15
Key:pick |   Frequency: 13
Key:lauch |   Frequency: 7
Key:ten |   Frequency: 11
Key:sake |   Frequency: 9
Key:australiana |   Frequency: 4
Key:stabil |   Frequency: 7
Key:concead |   Frequency: 2
Key:scratch |   Frequency: 2
Key:ruchil |   Frequency: 8
Key:click |   Frequency: 5
Key:unsoci |   Frequency: 7
Key:cantel |   Frequency: 2
Key:remak |   Frequency: 3
Key:campaigningvot |   Frequency: 5
Key:hang |   Frequency: 26
Key:role |   Frequency: 15
Key:roll |   Frequency: 22
Key:turn |   Frequency: 221
Key:ceo |   Frequency: 20
Key:result |   Frequency: 79
Key:gold |   Frequency: 7
Key:selfish |   Frequency: 3
Key:connect |   Frequency: 2
Key:hand |   Frequency: 99
Key:cigarett |   Frequency: 32
Key:piec |   Frequency: 66
Key:deadlin |   Frequency: 97
Key:shoaib |   Frequency: 2
Key:cfp |   Frequency: 3
Key:tha |   Frequency: 8
Key:guardiangdp |   Frequency: 5
Key:straight |   Frequency: 10
Key:sale |   Frequency: 10
Key:halv |   Frequency: 4
Key:abolit |   Frequency: 4
Key:obvious |   Frequency: 5
Key:saturdaysundai |   Frequency: 5
Key:tho |   Frequency: 10
Key:salari |   Frequency: 8
Key:hama |   Frequency: 7
Key:tht |   Frequency: 4
Key:newsbeatbbccouk |   Frequency: 2
Key:good |   Frequency: 408
Key:tie |   Frequency: 3
Key:cgt |   Frequency: 3
Key:fantasi |   Frequency: 12
Key:til |   Frequency: 6
Key:tin |   Frequency: 3
Key:tim |   Frequency: 4
Key:serious |   Frequency: 13
Key:tip |   Frequency: 23
Key:laugh |   Frequency: 11
Key:tit |   Frequency: 2
Key:chg |   Frequency: 2
Key:fantast |   Frequency: 110
Key:owen |   Frequency: 5
Key:chi |   Frequency: 2
Key:tusc |   Frequency: 2
Key:gone |   Frequency: 24
Key:sang |   Frequency: 2
Key:resolut |   Frequency: 2
Key:embodi |   Frequency: 4
```

```
ATTENTION REACHING END OF WINDOW: 15
Key:cancel |   Frequency: 15
Key:illumin |   Frequency: 2
Key:langtang |   Frequency: 5
Key:ccg |   Frequency: 2
Key:democracyavers |   Frequency: 4
Key:pretend |   Frequency: 12
Key:pretenc |   Frequency: 8
Key:northfield |   Frequency: 7
Key:pick |   Frequency: 49
Key:buzfe |   Frequency: 2
Key:pretens |   Frequency: 7
Key:antonio |   Frequency: 2
Key:eicc |   Frequency: 4
Key:remak |   Frequency: 10
Key:role |   Frequency: 98
Key:turf |   Frequency: 2
Key:barnstorm |   Frequency: 3
Key:ceg |   Frequency: 5
Key:fearless |   Frequency: 8
Key:turd |   Frequency: 3
Key:roll |   Frequency: 81
Key:turn |   Frequency: 400
Key:ceo |   Frequency: 10
Key:result |   Frequency: 469
Key:gold |   Frequency: 16
Key:pier |   Frequency: 152
Key:matchbox |   Frequency: 4
Key:selfish |   Frequency: 16
Key:parliamentarian |   Frequency: 3
Key:piec |   Frequency: 131
Key:milliganpa |   Frequency: 10
Key:shahriyar |   Frequency: 2
Key:adrian |   Frequency: 15
Key:salari |   Frequency: 13
Key:britannia |   Frequency: 10
Key:chaplainci |   Frequency: 2
Key:good |   Frequency: 694
Key:cgt |   Frequency: 2
Key:serious |   Frequency: 36
Key:dugdal |   Frequency: 14
Key:cha |   Frequency: 2
Key:che |   Frequency: 5
Key:roma |   Frequency: 2
Key:tusc |   Frequency: 7
Key:gone |   Frequency: 84
Key:aldou |   Frequency: 19
Key:walkei |   Frequency: 6
Key:cia |   Frequency: 10
Key:embodi |   Frequency: 2
Key:yougov |   Frequency: 159
Key:ensur |   Frequency: 65
Key:inwork |   Frequency: 19
Key:unvet |   Frequency: 13
Key:root |   Frequency: 8
Key:murdoch |   Frequency: 170
Key:murdock |   Frequency: 4
Key:roof |   Frequency: 5
Key:umbrella |   Frequency: 2
Key:beckham |   Frequency: 2
Key:enquir |   Frequency: 2
Key:plaidlab |   Frequency: 5
Key:creepi |   Frequency: 4
Key:room |   Frequency: 51
Key:energi |   Frequency: 39
Key:better |   Frequency: 682
Key:shellmon |   Frequency: 2
Key:gemma |   Frequency: 3
Key:cancer |   Frequency: 141
Key:imbecil |   Frequency: 2
Key:bombshel |   Frequency: 14
Key:rori |   Frequency: 3
Key:expos |   Frequency: 62
```

Figure 3.6: Tweet Keys and Values from the GE 2015 Stream

less related to actual events. Thus, windows with a number of tweets in the range defined by this value require the strict support method (Equation 3.5). For the GE 2015 stream, a window with less than 11k tweets was considered a small sized window. So windows 20 to 29 are considered small sized windows and require the small window support definition method (Equation 3.5).

Table 3.5 demonstrates a summary of the GE 2015 stream data analysis for all of the 29 window-batches, showing the number of tweets, the total number of terms (all terms with repetition), the total number of distinct terms (terms without repetition), the average frequency of the terms, the median, and the support values after applying the adjustment for small sized window-batches. After applying the relatively low support value restriction, all the set support values reflect their window-batch size, and this, in

window No.	# words	#words ≥ Avg frequency	#words ≥ Spp
1	6303	1131	194
2	6364	1144	237
3	6518	1146	207
4	6719	1100	191
5	7021	1237	210
6	6660	1345	203
7	6592	1258	188
8	6441	1328	185
9	7463	1369	240
10	7035	1341	216
11	8144	1306	250
12	8240	1416	246
13	8012	1300	296
14	6896	1255	284
15	10387	1655	337
16	12152	1800	364
17	18623	2178	533
18	28614	2714	724
19	4958	864	179
20	3571	621	181
21	3007	546	118
22	2553	546	112
23	2411	502	117
24	1960	440	134
25	1755	417	70
26	1449	349	110
27	1486	344	101
28	1651	325	87
29	1248	266	76

Table 3.3: Number of terms exceeding the Avg_i and the Spp_i for each window-batch in GE 2015 stream

turn, will help to retrieve only targeted and potentially significant terms. The defined support value for window 18, which is the biggest window of 388966 tweets containing (28614) distinct terms, is equal to 544, whereas the support value for window 26, which is the smallest window of only 1607 tweets containing (1449) distinct terms, is 41. For window 10, which is a medium sized window of 27023 tweets containing (7035) distinct terms, the support is value 151.

When event detection is applied in real-time or in near real-time, the decision concerning which support calculation to use cannot be made, as the size of the stream is unknown in advance. Thus we regard the window size allocation and decision as a two-class (large or small) classification problem. The decision as to whether any particular

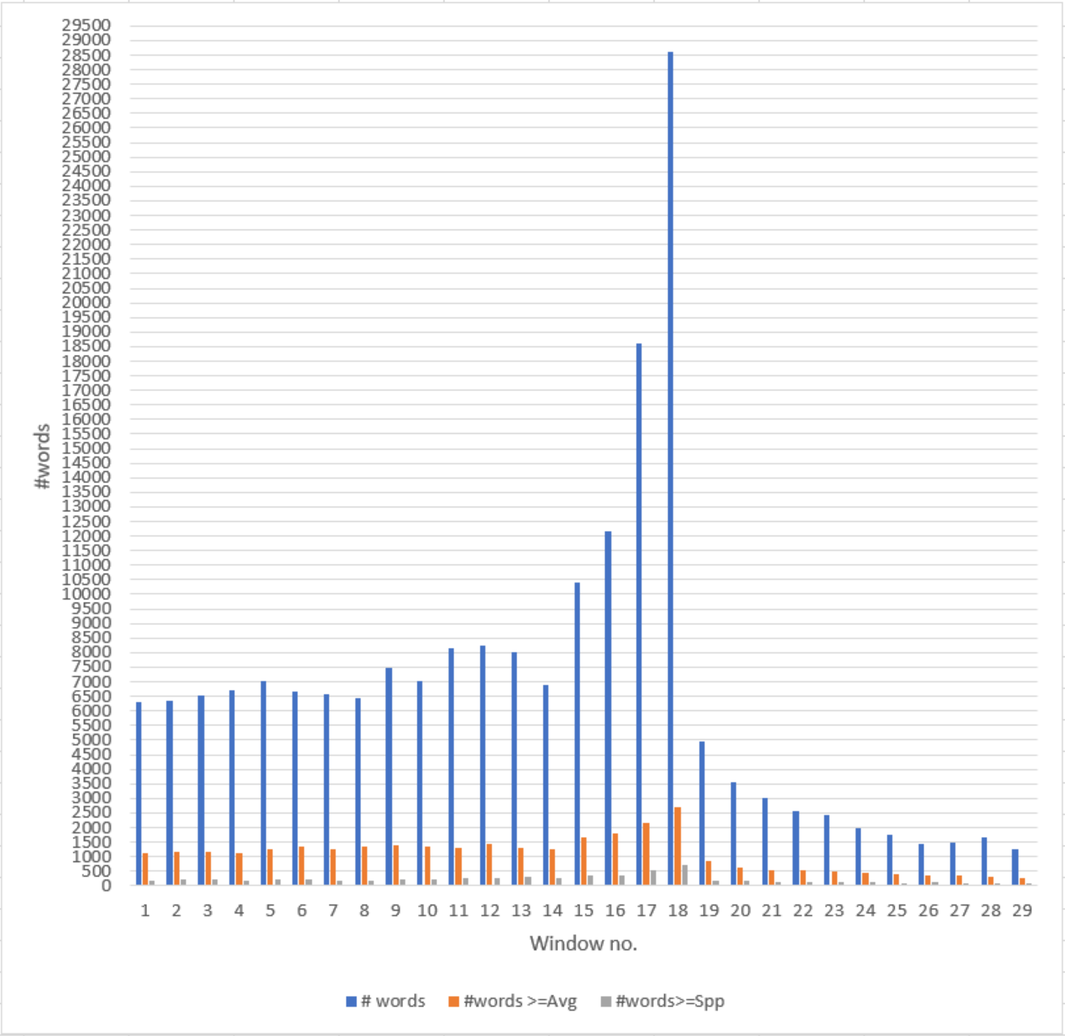


Figure 3.7: Number of terms exceeding the average and the support compared to total number of terms from GE 2015

upcoming window-batch should be considered a large or small sized window was made using the Logistic Regression (LR) model [156, 157]. The LR is a machine learning algorithm widely used for binary classification. Essentially, it is a statistical modelling method that estimates the probability of a binary response (outcome) based on one or more predictor (independent) variables.

Logistic regression classification models have been used in many areas and domains such as: spam detection [158], fraud detection [159–162], disease detection [163, 164], etc. The LR model is shown in Equation 3.6, where β_0 is the intercept and $\beta_1, \beta_2, .., \beta_n$ are coefficients associated with the independent variables $x_1, x_2, .., x_n$.

$$\frac{1}{1+e^{-(\beta_0+\sum_{i=1}^{n}\beta_i x_i)}} \tag{3.6}$$

We applied the LR to construct a prediction model that is able to classify each

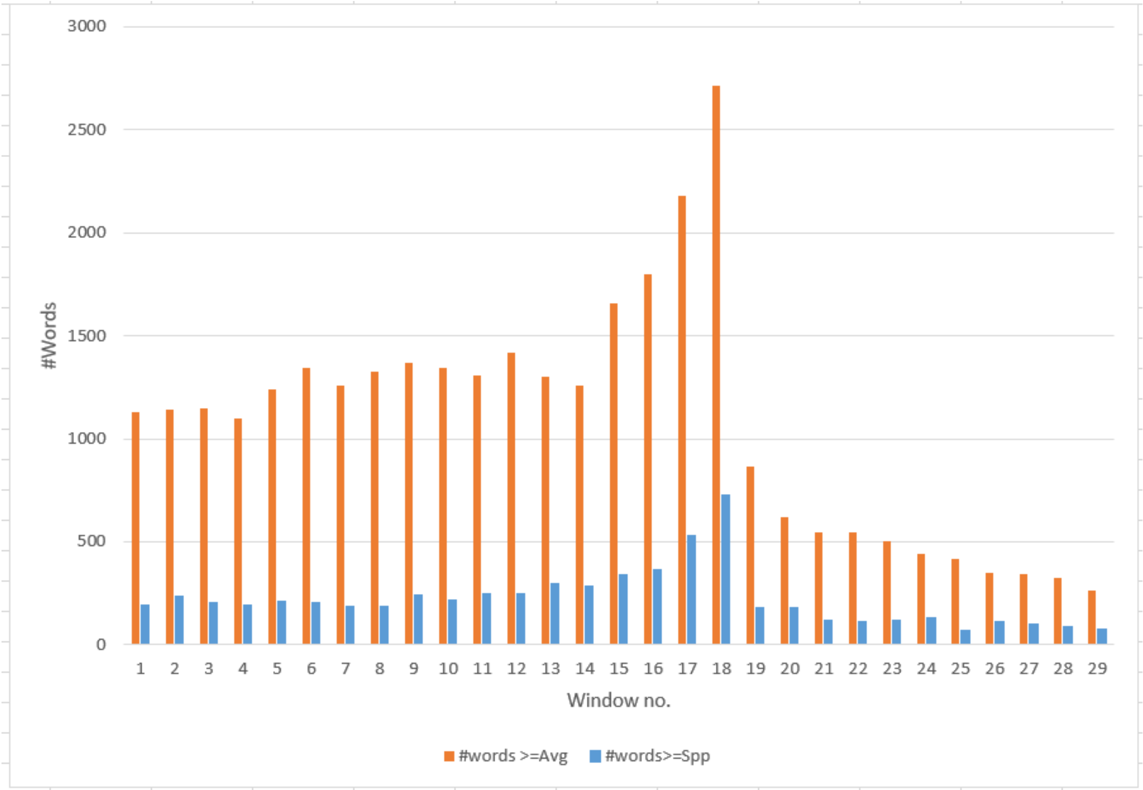


Figure 3.8: Number of terms exceeding the average compared to number of terms exceeding the support value from GE 2015

incoming window-batch in terms of it being either large or small. We have the number of tweets streamed in a window as the independent variable which determines the size of the window (either large or small). Building a classification model requires a labelled training dataset containing both large and small windows. However we wanted to avoid labelling the training set (window-batches) manually. Thus, since we noticed that the detected events from the GE 2015 stream were insignificant, we realized that the number of tweets in those windows was less than one-third of the average number of total tweets in the stream. As a result, a window was considered a small sized window in the training set (from the GE 2015 stream) if the number of posted tweets is less than one-third of the average number of total tweets in the stream. Table 3.6 illustrates the training dataset for building the LR classification model. It shows, for each window-batch, the number of tweets and the window size, where a value of 1 means a large size window and the value of 0 means a small size window.

Our logistic regression model for deciding whether an incoming window-batch is large or not is shown in Equation 3.7, where P is the probability of the window being large. These probabilities must then be transformed into binary values in order to actually make a prediction. The intercept for the $\beta_0 = -72.55$ and the coefficient for

W no.	Tweets	All-words	distinct-terms	Avg	Median	Spp
1	23836	151971	6303	24.11	6	145
2	24294	150417	6364	23.64	5	118
3	27210	174002	6518	26.7	6	160
4	33622	215318	6719	32.05	6	192
5	27759	188474	7021	26.84	6	161
6	23427	151191	6660	22.7	6	136
7	22511	144524	6592	21.92	6	132
8	22032	141516	6441	21.97	6	132
9	29957	193169	7463	25.88	6	155
10	27023	176840	7035	25.14	6	151
11	35546	238658	8144	29.3	6	176
12	34281	217300	8240	26.37	6	158
13	34410	227812	8012	28.43	5	142
14	27052	172363	6896	24.99	5	125
15	57348	347061	10387	33.41	6	200
16	73571	466232	12152	38.37	6	230
17	171829	998267	18623	53.6	6	322
18	388966	2595697	28614	90.71	6	544
19	13648	88096	4958	17.77	5	89
20	7942	47426	3571	13.28	4	53
21	5545	35827	3007	11.91	4	48
22	4081	24199	2553	9.48	4	38
23	3744	21982	2411	9.12	4	36
24	2429	15194	1960	7.75	3	23
25	2262	12920	1755	7.36	4	29
26	1607	10126	1449	6.99	3	21
27	2046	13349	1486	8.98	3	27
28	2215	15308	1651	9.27	3	28
29	1699	11242	1248	9.01	3	27

Table 3.4: Support Values Before the Strict Support Definition was Introduced from GE 2015

$No.tweets = 0.0583$ (these values were set from the training dataset). Figure 3.9 shows the logistic regression model graph when applied on the Greece Crisis stream to classify its window-batches as being of either large or small sized window's.

$$P = \frac{1}{1 + e^{-(-72.550 + 0.0538 \times No.tweets_{(T)})}} \tag{3.7}$$

So for the Greece Crisis stream, we explored 17 windows in the period from 29/6/2015-16/7/2015. Table 3.7 shows the Greece Crisis stream data analysis for all of these 17 window-batches, showing the number of tweets, the total number of words (all words including repetitions), the total number of distinct words (words excluding repetitions), the average frequency of the terms, the median, and the support values. Windows 6 and

W#	Tweets	All-words	distinct-words	AVG	Median	Spp
1	23836	151971	6303	24.11	6	145
2	24294	150417	6364	23.64	5	118
3	27210	174002	6518	26.7	6	160
4	33622	215318	6719	32.05	6	192
5	27759	188474	7021	26.84	6	161
6	23427	151191	6660	22.7	6	136
7	22511	144524	6592	21.92	6	132
8	22032	141516	6441	21.97	6	132
9	29957	193169	7463	25.88	6	155
10	27023	176840	7035	25.14	6	151
11	35546	238658	8144	29.3	6	176
12	34281	217300	8240	26.37	6	158
13	34410	227812	8012	28.43	5	142
14	27052	172363	6896	24.99	5	125
15	57347	347061	10387	33.41	6	200
16	73571	466232	12152	38.37	6	230
17	171829	998267	18623	53.6	6	322
18	388966	2595697	28614	90.71	6	544
19	13648	88096	4958	17.77	5	89
20	7942	47426	3571	13.28	4	106
21	5545	35827	3007	11.91	4	95
22	4081	24199	2553	9.48	4	76
23	3744	21982	2411	9.12	4	72
24	2429	15194	1960	7.75	3	46
25	2262	12920	1755	7.36	4	59
26	1607	10126	1449	6.99	3	41
27	2046	13349	1486	8.98	3	53
28	2215	15308	1651	9.27	3	56
29	1699	11242	1248	9.01	3	54

Table 3.5: GE 2015 Stream Data Analysis

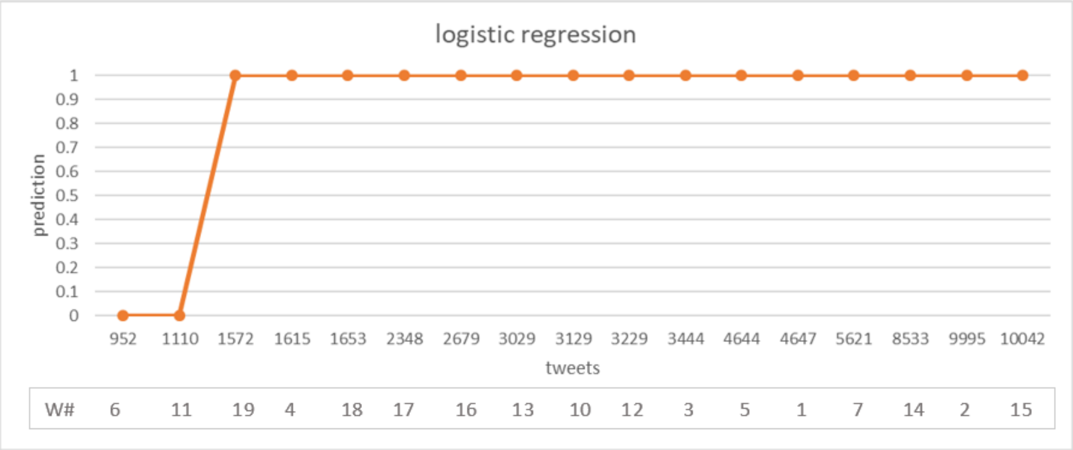


Figure 3.9: Greece Crisis Stream Logistic Regression Model

W#	No. Tweets	Window size
1	23836	1
2	24294	1
3	27210	1
4	33622	1
5	27759	1
6	23427	1
7	22511	1
8	22032	1
9	29957	1
10	27023	1
11	35546	1
12	34281	1
13	34410	1
14	27052	1
15	57347	1
16	73571	1
17	171829	1
18	388966	1
19	13648	1
20	7942	0
21	5545	0
22	4081	0
23	3744	0
24	2429	0
25	2262	0
26	1607	0
27	2046	0
28	2215	0
29	1699	0

Table 3.6: LR Training Dataset from the GE 2015 Stream

11 were classified as small sized windows with our LR classification model and required the use of Equation 3.5 for calculating their support.

3.3.3 Applying Frequent Pattern Mining (FPM)

We employ the FP-Growth [3] algorithm using our dynamically defined support value in order to find the frequent patterns from each window-batch. FP-Growth is basically a 2-step approach, which starts by constructing an FP-tree, and then proceeds to mine and traverse through that tree to find the frequent patterns. These patterns are assumed to be the topics or events discussed on that particular day. FP-Growth was chosen from the other FPM algorithms (Apriori and Eclat) because it best suits our data stream, it requires fewer stream scans, and no candidate patterns are generated. The text-posts

W#	Tweets	All-words	Distinct-words	AVG	Median	Spp
1	9294	57582	4429	13	4	52
2	19989	137878	6746	20.44	4	81
3	6888	46625	3226	14.45	4	57
4	3229	19735	1898	10.4	3	31
5	9287	62927	4304	14.62	4	58
6	1903	11317	1473	7.68	3	48
7	11241	76276	4775	15.97	4	64
10	6258	38105	2674	14.25	4	57
11	2219	15012	1535	9.78	3	60
12	6458	46488	3307	14.06	4	56
13	6058	39571	3215	12.31	4	49
14	17065	126024	6993	18.02	4	72
15	20083	142991	8328	17.17	4	68
16	5357	34957	3047	11.47	4	46
17	4695	32267	2675	12.06	3	36
18	3306	22127	2026	10.92	3	33
19	3143	19870	1857	10.7	3	32

Table 3.7: Greece Crisis Stream Data Analysis

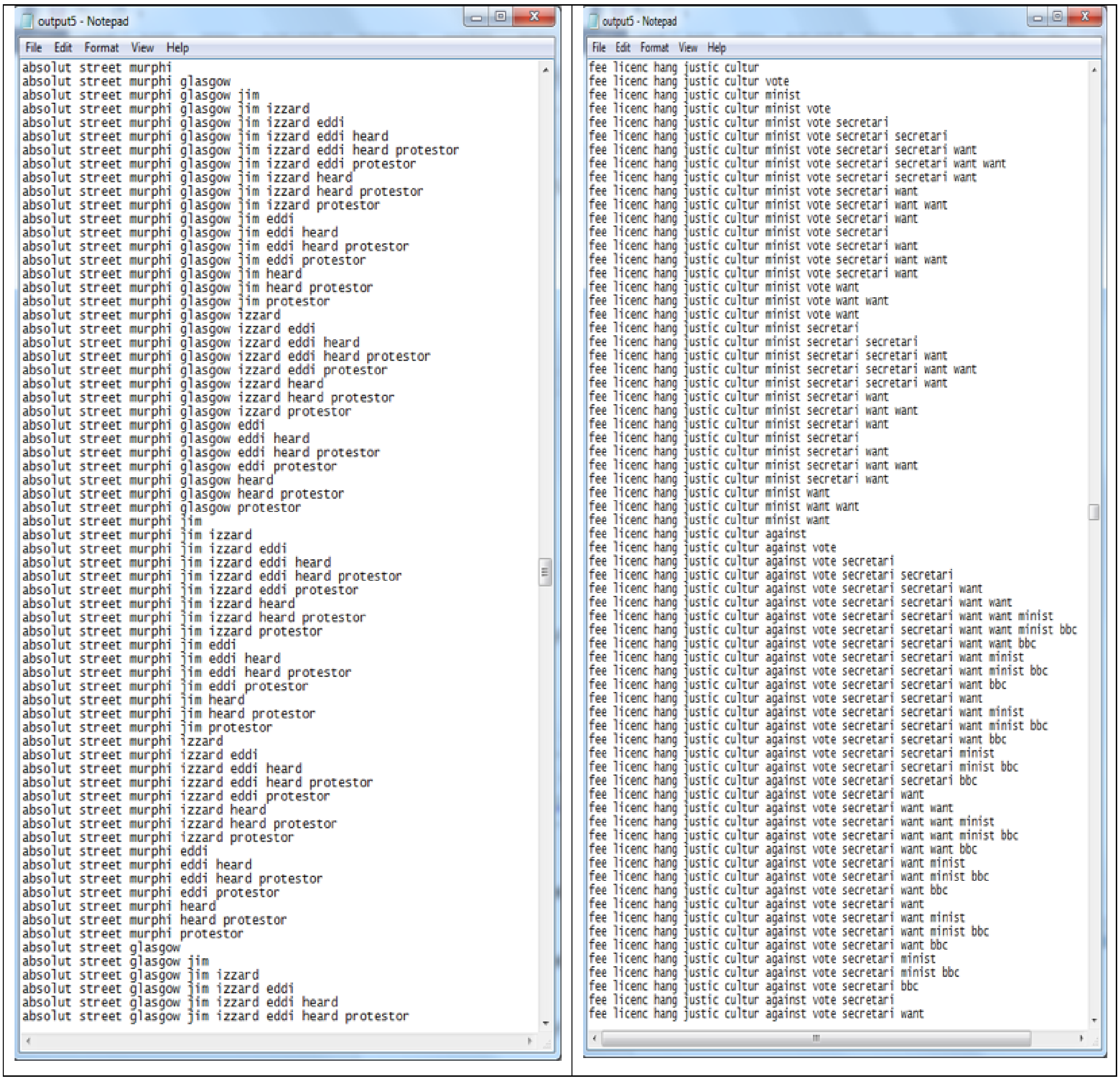
(tweets) belonging to a window-batch are assumed to be the transactions or itemsets, and the text-post terms are assumed to be the items.

We applied the FP-Growth for strings algorithm using SPMF: an open source data mining Java library [165]. Snapshots of the frequent patterns found are shown for some windows in figure 3.10.

The frequent patterns which were found after applying the FP-Growth algorithm with our dynamically defined support values were partially duplicated (see figure 3.10), and some others were irrelevant patterns. We are not interested in finding all the frequent patterns from our SN stream, instead we only want to identify patterns which may represent events.

In addition, some studies have shown that large number of tweets associated with major events may be irrelevant or misinforming. A study of the Ferguson unrest in 2014, in [152], showed that almost 25% of the posted tweets were in fact concerned with rumours. Another study in [153] considered the Hurricane Sandy event in 2012; this study showed that more than 10k unique tweets out of 1.7m tweets contained fake images. It was also found that 68% of the dissemination of fake tweets was via retweets, and 30 out of a total of 10,215 users were the source of 90% of the fake retweets.

As a result, we needed to consider a further post processing step in order to discover more useful and compact patterns. Accordingly, only patterns belonging to large



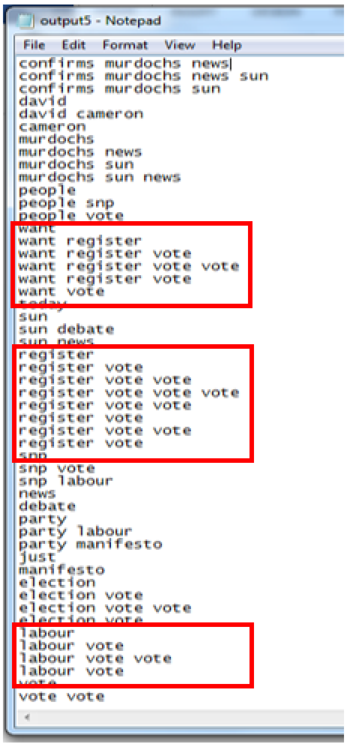
```
absolut street murphi
absolut street murphi glasgow
absolut street murphi glasgow jim
absolut street murphi glasgow jim izzard
absolut street murphi glasgow jim izzard eddi
absolut street murphi glasgow jim izzard eddi heard
absolut street murphi glasgow jim izzard eddi heard protestor
absolut street murphi glasgow jim izzard eddi protestor
absolut street murphi glasgow jim izzard heard
absolut street murphi glasgow jim izzard heard protestor
absolut street murphi glasgow jim izzard protestor
absolut street murphi glasgow jim eddi
absolut street murphi glasgow jim eddi heard
absolut street murphi glasgow jim eddi heard protestor
absolut street murphi glasgow jim eddi protestor
absolut street murphi glasgow jim heard
absolut street murphi glasgow jim heard protestor
absolut street murphi glasgow jim protestor
absolut street murphi glasgow izzard
absolut street murphi glasgow izzard eddi
absolut street murphi glasgow izzard eddi heard
absolut street murphi glasgow izzard eddi heard protestor
absolut street murphi glasgow izzard eddi protestor
absolut street murphi glasgow izzard heard
absolut street murphi glasgow izzard heard protestor
absolut street murphi glasgow izzard protestor
absolut street murphi glasgow eddi
absolut street murphi glasgow eddi heard
absolut street murphi glasgow eddi heard protestor
absolut street murphi glasgow eddi protestor
absolut street murphi glasgow heard
absolut street murphi glasgow heard protestor
absolut street murphi glasgow protestor
absolut street murphi jim
absolut street murphi jim izzard
absolut street murphi jim izzard eddi
absolut street murphi jim izzard eddi heard
absolut street murphi jim izzard eddi heard protestor
absolut street murphi jim izzard eddi protestor
absolut street murphi jim izzard heard
absolut street murphi jim izzard heard protestor
absolut street murphi jim izzard protestor
absolut street murphi jim eddi
absolut street murphi jim eddi heard
absolut street murphi jim eddi heard protestor
absolut street murphi jim eddi protestor
absolut street murphi jim heard
absolut street murphi jim heard protestor
absolut street murphi jim protestor
absolut street murphi izzard
absolut street murphi izzard eddi
absolut street murphi izzard eddi heard
absolut street murphi izzard eddi heard protestor
absolut street murphi izzard eddi protestor
absolut street murphi izzard heard
absolut street murphi izzard heard protestor
absolut street murphi izzard protestor
absolut street murphi eddi
absolut street murphi eddi heard
absolut street murphi eddi heard protestor
absolut street murphi eddi protestor
absolut street murphi heard
absolut street murphi heard protestor
absolut street murphi protestor
absolut street glasgow
absolut street glasgow jim
absolut street glasgow jim izzard
absolut street glasgow jim izzard eddi
absolut street glasgow jim izzard eddi heard
absolut street glasgow jim izzard eddi heard protestor
```

```
fee licenc hang justic cultur
fee licenc hang justic cultur vote
fee licenc hang justic cultur minist
fee licenc hang justic cultur minist vote
fee licenc hang justic cultur minist vote secretari
fee licenc hang justic cultur minist vote secretari secretari
fee licenc hang justic cultur minist vote secretari secretari want
fee licenc hang justic cultur minist vote secretari secretari want want
fee licenc hang justic cultur minist vote secretari secretari want
fee licenc hang justic cultur minist vote secretari want
fee licenc hang justic cultur minist vote secretari want want
fee licenc hang justic cultur minist vote secretari want
fee licenc hang justic cultur minist vote secretari
fee licenc hang justic cultur minist vote secretari want
fee licenc hang justic cultur minist vote secretari want want
fee licenc hang justic cultur minist vote secretari want
fee licenc hang justic cultur minist vote want
fee licenc hang justic cultur minist vote want want
fee licenc hang justic cultur minist vote want
fee licenc hang justic cultur minist secretari
fee licenc hang justic cultur minist secretari secretari
fee licenc hang justic cultur minist secretari secretari want
fee licenc hang justic cultur minist secretari secretari want want
fee licenc hang justic cultur minist secretari secretari want
fee licenc hang justic cultur minist secretari want
fee licenc hang justic cultur minist secretari want want
fee licenc hang justic cultur minist secretari want
fee licenc hang justic cultur minist secretari
fee licenc hang justic cultur minist secretari want
fee licenc hang justic cultur minist secretari want want
fee licenc hang justic cultur minist secretari want
fee licenc hang justic cultur minist want
fee licenc hang justic cultur minist want want
fee licenc hang justic cultur minist want
fee licenc hang justic cultur against
fee licenc hang justic cultur against vote
fee licenc hang justic cultur against vote secretari
fee licenc hang justic cultur against vote secretari secretari
fee licenc hang justic cultur against vote secretari secretari want
fee licenc hang justic cultur against vote secretari secretari want want
fee licenc hang justic cultur against vote secretari secretari want want minist
fee licenc hang justic cultur against vote secretari secretari want want minist bbc
fee licenc hang justic cultur against vote secretari secretari want want bbc
fee licenc hang justic cultur against vote secretari secretari want minist
fee licenc hang justic cultur against vote secretari secretari want minist bbc
fee licenc hang justic cultur against vote secretari secretari want bbc
fee licenc hang justic cultur against vote secretari secretari want
fee licenc hang justic cultur against vote secretari secretari want minist
fee licenc hang justic cultur against vote secretari secretari want minist bbc
fee licenc hang justic cultur against vote secretari secretari want bbc
fee licenc hang justic cultur against vote secretari secretari minist
fee licenc hang justic cultur against vote secretari secretari minist bbc
fee licenc hang justic cultur against vote secretari secretari bbc
fee licenc hang justic cultur against vote secretari want
fee licenc hang justic cultur against vote secretari want want
fee licenc hang justic cultur against vote secretari want want minist
fee licenc hang justic cultur against vote secretari want want minist bbc
fee licenc hang justic cultur against vote secretari want want bbc
fee licenc hang justic cultur against vote secretari want minist
fee licenc hang justic cultur against vote secretari want minist bbc
fee licenc hang justic cultur against vote secretari want bbc
fee licenc hang justic cultur against vote secretari want
fee licenc hang justic cultur against vote secretari want minist
fee licenc hang justic cultur against vote secretari want minist bbc
fee licenc hang justic cultur against vote secretari want bbc
fee licenc hang justic cultur against vote secretari minist
fee licenc hang justic cultur against vote secretari minist bbc
fee licenc hang justic cultur against vote secretari bbc
fee licenc hang justic cultur against vote secretari
fee licenc hang justic cultur against vote secretari want
```

Figure 3.10: Frequent Patterns Snapshot

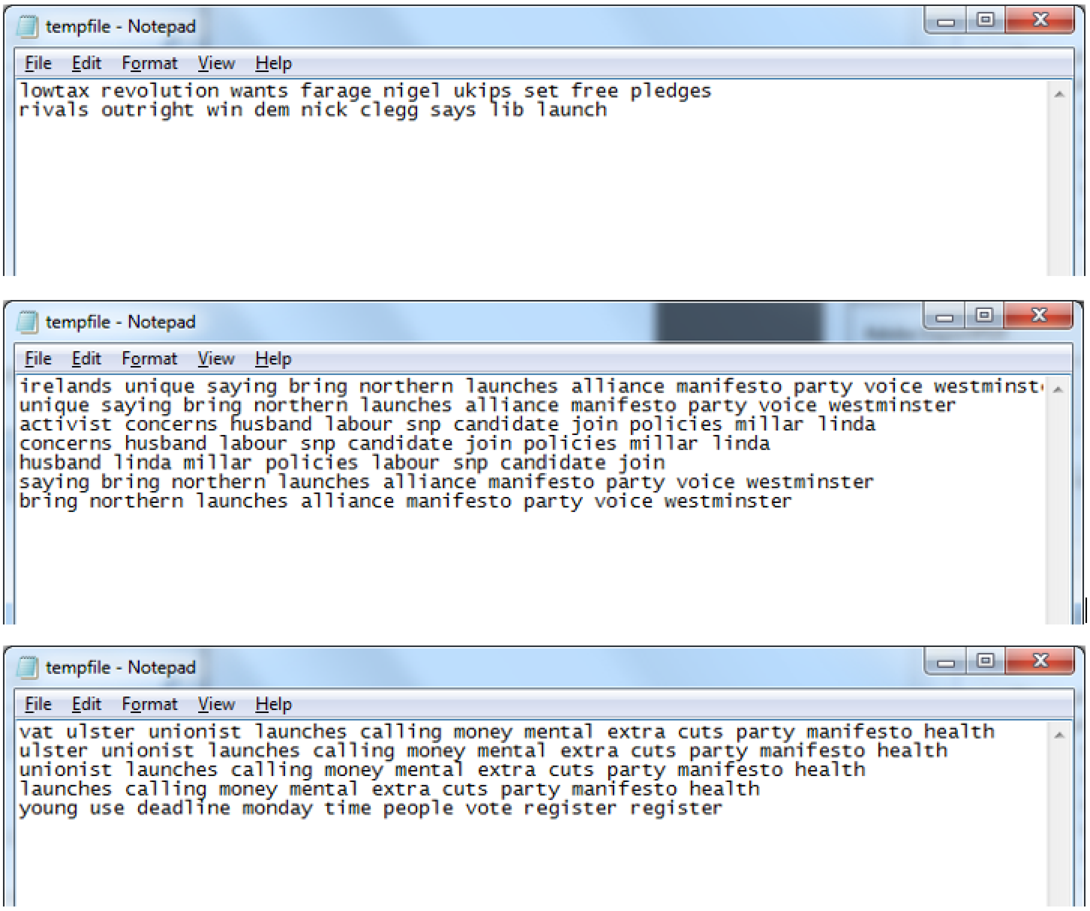
branches are preserved; a branch is said to be large if its size exceeds the set support value for that window. We chose the support value as a determinant for a branch retention, because we needed a threshold value, and again we can not have it as a fixed value. Hence, if the branch size exceeds the support value then it is kept, otherwise it is ignored. Applying this branch size restriction helped in avoiding non-relevant and meaningless patterns which in most cases belong to small sized branches. Figure 3.11 shows examples of some non-relevant patterns.

Subsequently, we looked for the longest pattern within each retained large branch and omitted all the subset ones, in order to avoid repetition. Snapshots of the frequent patterns which were found in three different windows (after both the branch size constraint and the patterns subset omission were implemented) are shown in figure 3.12.



```
confirms murdochs news
confirms murdochs news sun
confirms murdochs sun
david
david cameron
cameron
murdochs
murdochs news
murdochs sun
murdochs sun news
people
people snp
people vote
want
want register
want register vote
want register vote vote
want register vote
want vote
today
sun
sun debate
sun news
register
register vote
register vote vote
register vote vote vote
register vote vote
register vote
register vote vote
register vote
snp
snp vote
snp labour
news
debate
party
party labour
party manifesto
just
manifesto
election
election vote
election vote vote
election vote
labour
labour vote
labour vote vote
labour vote
vote
vote vote
```

Figure 3.11: Non-relevant Patterns



```
lowtax revolution wants farage nigel ukips set free pledges
rivals outright win dem nick clegg says lib launch
```

```
irelands unique saying bring northern launches alliance manifesto party voice westminst
unique saying bring northern launches alliance manifesto party voice westminster
activist concerns husband labour snp candidate join policies millar linda
concerns husband labour snp candidate join policies millar linda
husband linda millar policies labour snp candidate join
saying bring northern launches alliance manifesto party voice westminster
bring northern launches alliance manifesto party voice westminster
```

```
vat ulster unionist launches calling money mental extra cuts party manifesto health
ulster unionist launches calling money mental extra cuts party manifesto health
unionist launches calling money mental extra cuts party manifesto health
launches calling money mental extra cuts party manifesto health
young use deadline monday time people vote register register
```

Figure 3.12: Snapshots of Some Windows Frequent Patterns

As some FPs from Figure 3.12 were related to the same subject and discuss the same topic, similarity metrics need to be added in order to choose a representative pattern rather than all of them. Cosine similarity [166] and Jaccard coefficient [167] have been applied to keep only the longest pattern and discard the other similar ones; the similarity threshold was set to 75%. So, if two patterns are at least 75% similar to each other the longer pattern is kept while the other one is ignored. A relatively high similarity threshold (75%) was set in order to limit that process to patterns that are truly related to each other. In addition, other studies on Twitter stream have also used that percentage for similarity retrieval [39, 41]. The cosine similarity equation is shown in Equation 3.8, and the Jaccard coefficient equation is shown in Equation 3.9, where A and B are the two frequent patterns being compared.

$$COS(\Theta_{A,B}) = \frac{\overrightarrow{A} \,.\, \overrightarrow{B}}{||\overrightarrow{A}||||\overrightarrow{B}||} \tag{3.8}$$

$$Jaccard(A,B) = \frac{|A \cap B|}{|A \cup B|} \tag{3.9}$$

See Table 3.8 for a comparison between the two similarity metrics, the cosine similarity and the Jaccard coefficient, applied on the patterns found in a number of the investigated windows-batches. Generally, the cosine similarity was less strict and yields better results; thus it was adopted in our work.

3.3.4 Discussion

Using our event detection framework from SN streams (text streams), which is based on the dynamic support definition method (the D-FPM), we can detect events/topics occurring every day, if there are any. The goal is not to identify all the daily occurring FPs from every window-batch, but instead, to detect and find the FPs which may relate to an event. Looking for the events/topics which may emerge each day is challenging; we essentially depend on only a single window-batch to define its support value and to identify and detect events if any.

Evaluating the events/topics which were found was done manually. We looked for news headlines published on the WWW on the same day that an event (a frequent pattern) was found and compared the two. We created a gold-standard list from the BBC News Headlines, in Appendix 2, of news articles published during the GE 2015

		Cosine Similarity	Jaccard Coefficient
W1	$Spp_1 = 145$	lowtax revolution wants farage nigel ukips set free pledges	lowtax revolution wants farage nigel ukips set free pledges
		rivals outright win dem nick clegg says lib launch	rivals outright win dem nick clegg says lib launch
W2	$Spp_3 = 118$	irelands unique saying bring northern launches alliance manifesto party voice westminster	irelands unique saying bring northern launches alliance manifesto party voice westminster
		activist concerns husband labour snp candidate join policies millar linda	activist concerns husband labour snp candidate join policies millar linda
			bring northern launches alliance manifesto party voice westminster
W3	$Spp_3 = 160$	vat ulster unionist launches calling money mental extra cuts party manifesto health	vat ulster unionist launches calling money mental extra cuts party manifesto health
		young use deadline monday time people vote register register	young use deadline monday time people vote register register
			launches calling money mental extra cuts party manifesto health

Table 3.8: Similarity Metrics Comparison

period. If a detected event or topic keywords matched some news headline keywords and was found on the same day that the corresponding news item was published, then the indicated event was said to be a true event, and to be a false or spurious event otherwise. According to [39, 41] there is yet no official baseline method to evaluate the identified topics against. Previous works [39, 41] have used a manually created ground truth as an evaluation metric for their identified topics.

Accordingly, we created a gold-standard list to compare our identified events against; also we used the SFPM algorithm [39] to evaluate our identified topics. We will evaluate both the identification technique used along with the results achieved.

The frequent patterns which were found from the GE 2015 stream were related to news headlines and were found on the same day that the associated news headline was published on the WWW. This was so, except in regard to window 18, which was one day after the date of the election; on that day the lingering effect of the elections results was still showing. The detected events still seemed to be associated with the aftermath of the elections, and often concerned with topics that people has been discussing during the elections like the NHS, disability allowances etc. Table 3.9 shows the detected events in the GE 2015 period, where each window is shown with its detected FPs along with the associated news headline.

In the Greece crisis 2015 stream, all events detected using our event detection framework along with the LR model did match news headlines, Table 3.10 presents the detected events with the matching news headlines. Furthermore, the nature of the Greece Crisis event was different from the GE 2015 event. This was clear from the high number of detected events for some windows, where events were unsustainable, successive and fast. In fact, non English identified FPs were eliminated.

It is worth mentioning that some of the identified frequent patterns from the Twitter stream which occurred after major announcements (such as for instance, those which occurred after Greece officially announcing not being able to pay back the IMF by the agreed deadline), reflected people's reaction toward what had been announced as this was more frequently discussed than the event itself (especially on Twitter; because it is a free online social network platform, where users communicate and interact through their posted tweets). Table 3.11 shows some examples of the found FPs from the Greece Crisis stream, where we show the date, the identified FP, a Tweet sample, and the matching news article. The identified FPs matched news headlines although they

Window no.	Frequent pattern	News headline
1	lowtax, revolution wants farage nigel ukips set free pledges rivals outright win dem nick clegg says lib launch	UKIP would make working people better through a ”low-tax revolution”, Nigel Farage has said as he launched his party’s manifesto. http://www.bbc.co.uk/news/election-2015-32312687 Nick Clegg has said no party will win an outright election victory and warned voters they face a choice between the Lib Dems, the SNP and UKIP over who holds the balance of power. (launching his manifesto). http://www.bbc.co.uk/news/election-2015-32311736
2	ireland unique saying bring northern launches alliance manifesto party voice Westminster activist concerns husband labour snp candidate join policies millar linda	The Alliance Party will bring a ”unique” voice to Westminster, ensuring the people of Northern Ireland are heard. http://www.bbc.co.uk/news/election-2015-northern-ireland-32334446 Former Labour candidate Linda Millar and activist husband join SNP after concerns over policies. www.thenational.scot/.../former-labour-candidate-linda-millar-and-activist-husband-jo
3	vat ulster unionist launches calling money mental extra cuts party manifesto health young use deadline monday time people vote register register	The Ulster Unionist Party outlines tax reductions and extra money for mental. http://www.bbc.co.uk/news/election-2015-northern-ireland-32348636 Election 2015: Have you registered to vote? https://www.bbc.co.uk/news/election-2015-32267376 More people register to vote ’than ever before (news on mon 21/4) https://www.bbc.co.uk/news/election-2015-32401218
4	No found frequent patterns	
5	rattled badly obviously polls sir snp john says tories majors speech	SNP says Sir John Major’s speech ’very foolish’. This attitude doesn’t respect democracy and is completely wrong. The Tories are obviously badly rattled by the polls. http://www.theguardian.com/politics/live/2015/apr/21/election-2015-live-labour-john-major-blackmail-snp-nicola-sturgeon-ed-miliband?page=with%3Ablock-55361517e4b046f7a16b5992
6	No found frequent patterns	
7	underestimates aware ifs says cuts miliband lab scottish labours gets excited	Nicola Sturgeon has also been responding to the IFS report on Twitter. “Before Scottish Lab gets excited about IFS, they should be aware that Ed Miliband says it underestimates Labour’s cuts”. http://www.theguardian.com/politics/live/2015/apr/23/election-2015-live-ifs-verdict-labour-conservatives-liberal-democrats-snp-tax-and-spending-plans?page=with%3Ablock-553900cbe4b0401b98b64bdd
8	No found frequent patterns	
9	breaking dems news lib snp labour poll scotland new tns	BREAKING NEWS: New TNS Scotland poll (SNP 57, Labour 1, Lib Dem 1). http://www.telegraph.co.uk/news/general-election-2015/11566036/Poll-Labour-reduced-to-one-seat-in-Scotland.html
10	No found frequent patterns	
11	forget votes win win scotland seats working polls lets stronger elections enough labour people says jim murphy need change east end years glasgow correct forms voting candidates east labours postal major hull missing including cockup	Poll: SNP on course for clean sweep in Scotland. http://www.telegraph.co.uk/news/politics/SNP/11570881/polls.html Again we stand here tonight proud to be Labour. With Labour change is coming to the east end of Glasgow, East end of Manchester, Liverpool,Cardiff, Edinburgh. http://www.scottishlabour.org.uk/blog/entry/no-more-waiting.-no-more-wasted-lives.-jim-murphy#sthash.SvcIUJfo.dpuf A major election cock-up in Hull. The council has sent out a large batch of postal ballot papers in the Hull East constituency which have omitted the names of the Labour candidate Karl Turner and the Green candidate Sarah Walpole. http://news.channel4.com/election2015/04/29/
12	running stop vote snp snp time sun sun Scottish country	The Sun and its sister paper, the Scottish Sun, have endorsed different parties in the general election. http://www.bbc.co.uk/news/election-2015-scotland-32523804
13	shortest history suicide note scottish political says let miliband tories work	Miliband said. “If the price of having a Labour government was coalition or a deal with the Scottish National Party, it’s not going to happen.”. http://www.telegraph.co.uk/news/general-election-2015/politics-blog/11576757/Ed-Milibands-SNP-lie-could-damage-him-more-than-Nick-Cleggs-tuition-fee-promise.html
14	condemns violent aggressive event protestors izzard eddie campaign labour glasgow	Scottish Labour leader Jim Murphy and comedian Eddie Izzard were heckled by opponents during general election campaigning in Glasgow. http://www.bbc.co.uk/news/election-2015-scotland-32581803
15	No found frequent patterns	
16	pinned throttled wall galloways supporters told george bradford fucking jew	A burly Asian man in a black suit and sunglasses rushes up and grabs me round the neck, pinning me to a low perimeter wall. “Get out, you fucking Jew,” he shouts. http://www.politico.eu/article/galloway-bradford-elections-uk-ge2015/
17	No found frequent patterns	
18	teacher time poor sick old disabled unemployed immigrant student unless alive nurse doctor disability suffer family kind congrats lose job get tory voters hope sick better vulnerable mourning environment rip today day nhs rights human national welfare state	
19	fee licence rid equalities secretary secretary wants minister voted bbc against culture justice hanging	BBC licence fee in doubt as John Whittingdale is named culture secretary. https://www.theguardian.com/media/2015/may/11/john-whittingdale-culture-secretary-bbc-charter-renewal
20	fee equalities secretary secretary voted wants minister bbc against	The BBC will be forced to slash its drama output if the licence fee is cut, according to the man who commissioned hits like Poldark, Sherlock, Call the Midwife and Wolf Hall. http://www.bbc.co.uk/news/entertainment-arts-32702746
21	enquiries reports make kent following seat police south fraud electoral thanet	Kent Police investigate allegations of election fraud after Ukip leader Nigel Farage’s general election defeat in South Thanet. https://www.telegraph.co.uk/news/politics/nigel-farage/11602620/Police-probe-allegation-of-electoral-fraud-in-Thanet-South.html
22-29	No found frequent patterns	

Table 3.9: GE 2015 Stream Detected Events and Matching News Headlines

W#	Frequent Patterns	News Headlines
1	sit greece bailout campaign indiegogo crashed money crowdfunding raise popular, multiplying bills apparently greece crisis debt got control huge losing, bbc appeal rejects news eurozone debt debt crisis bailout greece, advised ert greek referendum tsipras called lenders dragasakis accept tells deputy offer suggests, ministers finance extend continue greece bailout eurogroup eurozone talks refuses, bid alexis makes lastminute tsipras new crisis greece	Greek bail out crowdfunding campaign crashes Indiegogo after raising 250,000 in ONE DAY News article Here Greece debt crisis: Athens fails to repay IMF as bailout runs out, News Article Here Eurozone rejects bailout appeal. News Article Here Greek Deputy PM Dragasakis advised Tsipras to accept lenders' offer & suggests referendum could be called off News Article Here Eurozone finance ministers have rejected a Greek government call to extend its bailout News Article Here Greece say a last-minute offer was made by creditors on Monday night. Mr Tsipras appealed to Greeks to reject the creditors' proposals. News Article Here
2	rank greece greece crisis tsipras seeking outside scapegoats berlin blasts, washington post slams sunday door europe greece referendum talks, takeover soars ignores greece crisis talks asx asx aus aus	Greece crisis: Berlin accuses Tsipras of seeking scapegoats outside own ranks, News Article Here, E.U. slams the door on talks with Greece before Sunday referendum News Article Here ASX Australian shares soars on takeover talks, ignores Greece Crisis,News Article Here
3	takeover soars ignores greece crisis talks asx asx aus, twice forgotten origins blame think make crisis greeces, time bluff called germany ahead vote bailout greeces, greece referendum vote lnl europe proud decent adams talking phillip, greece greece vote bailout talks talks eurozone chiefs rule ruled	The ASX is on track for a third day of gains. News Article Here The forgotten origins of Greece's crisis will make you think twice about who's to blame, News Article Here How Germany Called Greece's Bluff Ahead of Bailout Vote. News Article Here Yanis Varoufakis backed his prime minister's recommendation of a 'No' vote and repeated the assertion by Alexis Tsipras that the government may resign if the result goes the other way. News Article Here, Eurozone rules out talks until after referendum. News Article Here
4	reject alexis voters greece referendum bailout tsipras blackmail, shattered greece euro long quite happy crucified limp manages, devaluing greece euro economic example fear following recovery icelands leaving defaulting, eurocrats greek euro care care people best failed project march greece referendum vote support massive today paris, updates thousands referendum ahead live europe sunday rally solidarity	Greece debt crisis: Tsipras urges 'No' to 'blackmail' News Article Here EU is quite happy to see Greece crucified as long as the shattered Euro manages to limp on. The EU does NOT care about what is best for Greece., News Article Here, Greece should have no fear of leaving Euro, defaulting, devaluing & following Iceland's example of economic recovery (express.com). News Article Here, All those running the EU care about is their project. They care not for the best wishes of the Greek people, but simply for sustaining, News Article Here, Thousands of protesters in European cities are rallying in solidarity with Greece. News Article Here
5	NO frequent patterns	
6	inspired greece give give chance campaign case saying feels, breaking introduce parliament head currency european new wins says todays greece referendum vote, sunday greece referendum vote bailout greeks greeks athens voting high stakes	A tweet by Mark Rufallo encouraging the All we are saying is give Greece a chance. @Indiegogo campaign in case anyone feels inspired to give those 3s:Greece bailout crowdfunding,News Article Here Greek referendum: No campaign storms to victory with 61.31% of the vote,News Article Here Greece voters overwhelmingly rejected the latest bailout package from European creditors in Sunday's referendum, News Article Here
9	worked referendum greeks tsipras say wait amazing berlin steps strategy	Greek referendum: smart response from Tsipras, but triumph may be brief, News Article Here
10	harsher demands voted terms offer nears zone just oxi euro greece grexit	Greece debt crisis: Athens accepts harsh austerity as bailout deal nears, News Article Here
11	wrapup promising hikes sends tax reform plan greece new, ministers finance make tomorrow major decision decision eurozone greece bailout plan, tolerating grexit verhofstadt ttip push prevent alde banker order mobster moldovan	Greece sends reform plan to EU promising new tax hikes. News Article Here, EUROZONE FINANCE MINISTERS will make a 'major' decision tomorrow on the latest proposals from Athens on a fresh bailout plan, News Article Here, Guy Verhofstadt and ALDE are tolerating a Moldovan mobster banker in order to prevent a Grexit and push TTIP., News Article Here
12	sources talks demand zone ministers bailout euro greece, mulling european source fiveyear plan germany grexit temporary	Euro zone ministers demand Greece do more before bailout talks. News Article Here, Germany mulling five-year 'temporary Grexit' plan. News Article Here
13	sidelines euro merkel summit tsipras hollande meeting meeting tusk eus suspends, discussion financial latest exit greek summit greece crisis euro temporary leaders, tricky efficiently modern run greece eurogroup wednesday wants country basically, amt eurogroup document timeout total proposals reform needed reprofling grexit debt debt temporary offer offer new media restructure stuff freaking, partnership greece crisis bank dead obama insider cuba beltway brazilian grateful opens	'Greek compromise proposed' by Tsipras, Hollande and Merkel. News Article Here Greece debt crisis: EU summit cancelled as talks over a third bailout deal for Greece continue. News Article Here Basically the Eurogroup wants Greece to become a modern, efficiently run country by Wednesday. Could be extremely tricky and near impossible. News Article Here Here's the full 4pg eurogroup document on #Greece, inc "time-out", total amt needed (82-6bn) & reform proposals. News Article Here EU leaders have reached agreement that paves the way to a third Greek bailout, if Athens parliament approves tough austerity measures News Article Here Brazil has angrily attacked last week's £1.6bn International Monetary Fund payment to Greece on the same day as a report from the fund identified a new £9.6bn black hole in Greek News Article Here
14	parallel secret ious drachma plans reveals varoufakiscurrency grexit, grexit grexit varoufakis day reveals age planb send neolithic, pla bailout deal greeces syriza against leftist leftistgroup party considering voting lawmakers ruling source, radicals clash opposed faces syriza tsipras eurozone bailout greece, seeks backing bbc news tsipras eurozone debt crisis deal greece, gets debt crisis crisis greece greece greece greece greece bailout bailout deal, play grexit currency alan digital yong say read role	Varoufakis reveals his secret plans: Parallel currency IOUs, Grexit, Drachma. News Article Here Varoufakis 2011: Grexit will send us back to Neolithic Age.. 2day he reveals Grexit was his Plan B! News Article Here Source tells CNBC that the ECB will next consider Greek ELA (funding level for Greek banks) on Thursday. A leftist group of lawmakers in Greece's ruling Syriza party may vote against.., News Article Here Tsipras faces clash with Syriza radicals opposed to eurozone bailout for Greece. News Article Here, Tsipras seeks backing for eurozone deal. News Article Here, Deal reached after marathon all-night summit. News Article Here, Will Digital Currency play a role in Grexit? Read what Alan Yong has to say. (Tweet 13 July) News Article Here
15	plans secret far report needs relief imf debt Greece, water conditions camp moria migrants unbearable hundreds near sleep toilets, controversial bailout debt debt greeces european banks unpunished barely	Greece needs debt relief far beyond EU plans - secret IMF report, News Article Here, Greece forgotten crisis: Lesbos on verge of 'catastrophe' as 1,000 refugees arrive ashore daily, on Twitter 14 July on news 17 July, News Article Here, The Problem of Greece is Not Only a Tragedy: It is a Lie.,It is less than the debt of European banks whose "bailout" in 2007-8 was barely controversial and unpunished., News Article Here
16	linebyline bailout deal varoufakis yanis greeces just completely takedown trashed, tsip greece bailout vote vote way minister parliamentary awaits paving prime test alexis, negotiator greek vote plan new tonight reforms case ready negative	Yanis Varoufakis just trashed Greece's bailout deal with a line-by-line takedown, News Article Here Greece Awaits Crucial Parliamentary Vote Paving Way for Bailout, News Article Here In case of negative vote of Greek reforms plan tonight, the EU is ready with a new negotiator. (Tweet) MPs to vote on crucial reforms News Article Here
17	hour greece like text baes latenight message screaming fight brussels, reopen lifeline cash ecb new banks greek debt crisis greece	Screaming fight between Greece and EU ends in late night text message,News Article Here New ECB cash lifeline could reopen Greek banks.News Article Here
18	mps vote talks debt crisis german bailout greece	German MPs vote 'yes' to bailout talks. News Article Here
19	banking greece bailout reforms hurdle approves deal clearing measures involve overhaul	Greek parliament approves bailout measures as Syriza fragments. News Article Here

Table 3.10: Greece Crisis stream Identified Events with matching News Headlines

did not share many keywords with them. It was possible to associate them to news headlines as the frequent patterns found were manually evaluated. In general, this was only noticed after major announcements, as a reaction to what had been just announced or what had just happened (people were still debating about the impact of what had just been announced). In the GE 2015 stream this effect was in evidence after the announcement that the Conservatives had won the GE 2015 elections, and in terms of the Greece Crisis stream, after the announcement that Greece were not going to pay the IMF by the agreed deadline, and when the Eurogroup announced a list of reforms that Greece must meet in order to remain in the EU.

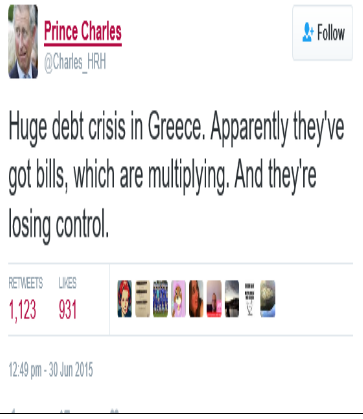
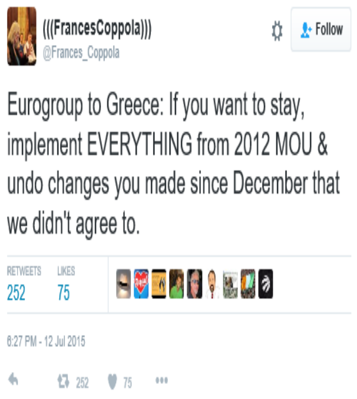
Date	FP	Tweet Sample	Matching News Article
30 June	multiplying bills apparently greece crisis debt got control huge losing	 Prince Charles @Charles_HRH Huge debt crisis in Greece. Apparently they've got bills, which are multiplying. And they're losing control. RETWEETS 1,123 LIKES 931 12:49 pm - 30 Jun 2015 	Greece has officially said that it won't pay back the money it owes to the IMF by the midnight deadline. News Article Here
12 July	undo december greece eurogroup want implement stay changes mou	 (((FrancesCoppola))) @Frances_Coppola Eurogroup to Greece: If you want to stay, implement EVERYTHING from 2012 MOU & undo changes you made since December that we didn't agree to. RETWEETS 252 LIKES 75 6:27 PM - 12 Jul 2015 	Basically the Eurogroup wants Greece to become a modern, efficiently run country by Wednesday. Could be extremely tricky and near impossible. News Article Here

Table 3.11: Identified Frequent Patterns After Major Announcements

3.3.5 Evaluation

In this section, we will evaluate the events which were identified by our event detection framework (the D-FPM), and also those which were detected by an alternative framework (the SFPM [39]) by applying the metrics Precision and Recall [168, 169], and then comparing and analysing results.

Precision and Recall are the basic evaluation measures which are used in information retrieval [130]. Precision is the ratio of the number of relevant documents retrieved to the total number of irrelevant and relevant documents retrieved; in other words it is the percentage of correct retrieved items to all retrieved items (it measures how useful the retrieval results are). Recall on the other hand, is the ratio of the number of relevant documents retrieved to the total number of relevant documents (the latter includes all relevant documents even the ones not retrieved); in other words it is the percentage of correct items that are retrieved (it measures how complete the retrieval results are). Equations 3.10 and 3.11 show how Precision and Recall are calculated, where tp refers to true positives, fp refers to false positives, and fn refers to false negative.

A perfect precision with a score of 1.0, means that all the retrieved documents were relevant, however it says nothing about completeness, whether all the relevant documents were retrieved or not. In contrast, a perfect recall with a score of 1.0, means that all relevant documents were retrieved, but says nothing about how many irrelevant documents were retrieved as well.

$$Precision = \frac{tp}{tp + fp} \tag{3.10}$$

$$Recall = \frac{tp}{tp + fn} \tag{3.11}$$

True positives (tp), is the total number of items that are correctly identified as belonging to a particular category. False positives (fp) on the other hand, is the total number of items that are incorrectly identified as belonging to the particular category.

True negatives (tn), refers to the total number of items that are correctly identified as not belonging to the particular category. False negatives (fn), refers to the total number of items that are incorrectly identified as not belonging to the particular category.

For example, take the case where the particular category was the events/topics found in the gold-standard list. The gold-standard list, is a manually created list showing the events found from news outlets during a certain period of time. Refer to Appendix 2 for a table showing the gold-standard list for the GE 2015 period (in Table A2.1), where you can see the list of news headlines as announced each day by the BBC NEWS (http://www.bbc.co.uk/news).

Accordingly, tp refers to the total number of events that are retrieved and found in the gold-standard list, fp refer to the total number of events that are incorrectly retrieved as events (not found in the gold-standard), tn refers to the total number of events that are correctly classified as not relating to an event, and finally fn refers to the total number of missed events (being not classified as events) while in fact they are true events found in the gold-standard list.

The F-measure [170] is a combination of both precision and recall; it is calculated by finding the harmonic mean of both. The balanced F-measure balances recall and precision in such a way that each of them is given equal weight. It is calculated using the following equation (Equation 3.12).

$$F1 - measure = 2 * \frac{precision * recall}{precision + recall} \tag{3.12}$$

The SFPM [39] is an approach that uses the FPM to find and extract topics from a textual stream (for more details refer to sub-section 2.5.2). It was chosen as an alternative method to be implemented to find the topics which emerged during the GE 2015 period. Thus we were able to compare findings between the SFPM and our event detection framework (the D-FPM). Table 3.12 shows the topics detected during the GE 2015 period (topics are separated by commas) using the SFPM, where $b = 10$ and $c = 2$. In more detail, it shows the identified topics from each window-batch (each day), along with the number of top selected terms, K, for each window (the value of K is a fixed number provided in advance). For each window-batch, K was set to a value matching the number of distinct terms satisfying the dynamic support value in our event detection framework. This leads to there being the same number of terms selected by both approaches, which in consequence allows us to straightforwardly compare the detected topics yielded by the two approaches.

W#	K	Topics
1	193	[alternative austerity ukip want, cleggs alex going salmond]
2	231	[literally year nowcast sun attack faked , northern bring westminster unique irelands , camerons turning miliband interview , message parliament strong students politicians , activist millar linda concerns , deal scottish year]
3	204	[launches ulster calling cuts unionist , nurses gps hell promising , country numerous ukip receives grammatical]

4	177	[independence trending leader number , undermine oppose parliament scottish let , deadline registered trending registration]
5	199	[scotland year registered westminster , comments affront year democracy , arriving started year stronger , basically accuses left pushing making , launches role play , rattled obviously majors affront , murdoch sun tells miliband]
6	200	[murdoch thinks want rupert , portobello morning year trail , members girls year teenage , welfare poor year poverty , winning scottish year westminster , foodbanks food year record]
7	175	[underestimates scottish excited miliband , projection ldem ukip weeks , campaign quirky spotted pics bbc , attack literally sun confirms , debt leader voters]
8	176	[inherited property million worth george wealth , misses cycle voters jacket , camerons voters english , brief endorsement voters history , confidently world voters miliband]
9	226	[scotland year ukip dem breaking, main promised equality want , bus owners year business let letter , crisis year ukip dem]
10	212	[interview humanist brand , independence scottish humanist overview]
11	246	[elections let seat scottish strong , hunt building jeremy surrey momentum , including cockup missing hull]
12	241	[reminder bbc year tonights , snp scottish year sun, unfair state defende electoral , conservative ukip dem , compared bankers years , getting audience year miliband]
13	297	[chamber face shop going commons , exdefence sec left right , referendum westminster sco scottish let shortest]
14	271	[protestors shouted ukip chaos , condemns aggressive ukip violent , lib ukip scottish miliband endorses brand , tour warm reception continues , tells shouted dead shout democracy]
15	334	[scotlands westminster year pension , kingdom parliament year united , saltire standing year girl , broken promises year , crowd eddie izzard scottish , lying pension state letter westminster , crisis year ukip]
16	360	[donts dos legally , jews accurate fucking angry]

17	531	[sacrificed suffragettes voted protected , fucking angry idiot accurate , pensions protected overs voted tuition , adorable choosing eating weird sure , schizophrenia folly evasions hatred , losing bad voted protected]
18	721	[absolutely long unemployed disabled old , women parliament female , foodbanks using nearly message loud , student unless poor disabled unemployed , disappointed fucking voted , disability suffer family congrats better , david cameron year , feeling feel happier , takes responsibility poor miliband sources , majority conservatives confirm , leadership quits resignation resign miliband lib , kingdom voted united , meet buckingham palace success]
19	162	[representation proportional leader ukip , form electoral leader bit bar context , conservatives leader voted rid bbc want , breaking westminster leader ukip]
20	72	[conservatives held year meeting tory , labour by election hed voters , snp agree lords nicolasturgeon abolishing , bankers modern britain letter politicians , scotlands westminster voted rid bbc want against , widely website share , pollsters entire polls , unresigned byelection hed farage forward , stories pics writing send article , attack hardwon rights]
21	44	[gebot scotland ukip lost , mps elected labour , business party ukip , results result election general , underestimated admits stole votes , plans tory voted scottish , westminster reports south electoral thanet kent win]
22	32	[scotland highest snp , party labour voters scottish , buying spent tories election , results week amnesty running protect human , right hardwon rights fight human , voters cameron election result]
23	39	[lab labour number share increase constituencies highest , north england thousands , result week election evidence coverage newspapers , tory voters voted young , mps parliament things , scottish politics political labour lab , leadership chuka umunna]
24	32	[politics big david cameron precious speech nhs , electioni elect election sits houseofcommons commons house , north labour lost ukip]
25	18	[general election politics political tories , cameron majority won tories , political politics election general tories]

26	19	[politics hunt snp south edin dundee constituency west , cumbernauld result election]
27-29	42-39-26	No found topics

Table 3.12: The SFPM Identified Topics

Table 3.13 shows the calculated precision, recall, and F-measure for our approach —the Dynamic-FPM (D-FPM) and also for the SFPM approach; in addition, it shows the number of news topics from the gold-standard list each day. When there were no news topics published on a certain day, this was shown in the gold-standard column as ‘no topics’. The highest F-measure value achieved between both approaches is shown in bold for each window.

An overview of the evaluation results is presented in Table 3.14, where we show the averages of precision, recall and F-measure. In general, a high precision score means that a particular approach has returned more relevant results than irrelevant ones, while a high recall means that the approach has returned most of the relevant results. Our approach (the D-FPM) clearly outperforms the SFPM, especially in terms of precision values (73% for the D-FPM approach but just 23% for the SFPM approach), which means that the D-FPM approach identified topics included in the gold-standard list topics more than the SFPM approach did. In most cases, the SFPM detects more events than the D-FPM approach does, however most of the events it detects are false positives (not matching events in the gold-standard list) and so degraded the SFPM’s precision value. A reason for the low precision value yielded by the SFPM approach could be the term selection criteria, which depends on an independent text stream. On the other hand, in the D-FPM approach, the terms selection criteria depends on the current day’s window size and its term frequencies.

In contrast, the recall values for both approaches are close to each other: 52% for the D-FPM approach and 53% for the SFPM. As the recall is the number of correctly identified events over the total number of events found in the gold-standard list, these close results could possibly mean that the events which are missed are not frequently mentioned in the Twitter stream (most likely the value of fn is the same for both

		D-FPM			SFPM		
W#	Gold-standard	Precision	Recall	F-measure	Precision	Recall	F-measure
1	3	2/2=1	2/3=0.67	**0.8**	1/2=0.5	1/3=0.33	0.4
2	4	1/2=0.5	1/4=0.25	0.33	2/6=0.33	2/4=0.5	**0.4**
3	2	1/1=1	1/2=0.5	**0.67**	1/3=0.33	1/2=0.5	0.4
4	No topics	Nothing detected	Nothing detected	NA	0/3	0/0	0
5	2	1/1=1	1/2=0.5	**0.67**	1/7=0.14	1/2=0.5	0.22
6	No topics	Nothing detected	Nothing detected	NA	0/6	0/0	0
7	1	1/1=1	1/1=1	**1**	1/5=0.2	1/1=1	0.33
8	1	0/0	0/1=0	0	0/5=0	0/1=0	0
9	2	1/1=1	1/2=0.5	**0.67**	1/4=0.25	1/2=0.5	0.33
10	No topics	Nothing detected	Nothing detected	NA	0/2	0/0	0
11	3	3/3=1	3/3=1	**1**	2/3=0.67	2/3=0.67	0.67
12	2	1/1=1	1/2=0.5	**0.67**	2/6=0.33	2/2=1	0.5
13	1	1/1=1	1/1=1	**1**	0/3=0	0/1=0	0
14	1	1/1=1	1/1=1	**1**	1/5=0.2	1/1=1	0.67
15	No topics	Nothing detected	Nothing detected	NA	0/7	0/0	0
16	2	0/1=0	0/2=0	0	0/2=0	0/2=0	0
17	No topics	Nothing detected	Nothing detected	NA	0/6	0/0	0
18	2	0	0	0	1/13=0.077	1/2=0.5	**0.133**
19	2	1/1=1	1/2=0.5	**0.67**	1/4=0.25	1/2=0.5	0.33
20	2	1/1=1	1/2=0.5	**0.67**	2/10=0.2	2/2=1	0.33
21	1	1/1=1	1/1 =1	**1**	1/7=0.143	1/1=1	0.25
22	2	0/0=0	0/2=0	0	1/6=0.16	1/2=0.5	**0.25**
23	No topics	Nothing detected	Nothing detected	NA	0/7	0/0	0
24	No topics	Nothing detected	Nothing detected	NA	0/3	0/0	0
25	No topics	Nothing detected	Nothing detected	NA	0/3	0/0	0
26	No topics	Nothing detected	Nothing detected	NA	0/2	0/0	0

Table 3.13: Precision, Recall, and F-measure Using GE 2015 Stream

approaches). Finally, for the F-measure value, which is the harmonic mean of the precision and recall, and provides a single measurement (so it is ideal for examining precision and recall together), the results are shown in the last column of Table 3.14. For the D-FPM approach, the F-measure score is almost double that yielded by the SFPM approach, 60% for the D-FPM approach and 30% for the SFPM approach.

	Precision	Recall	F1-measure
D-FPM Approach	12.5/17=0.73	8.92/17=0.52	10.15/17=0.6
SFPM	3.82/17=0.23	9/17=0.53	4.96/17=0.3

Table 3.14: Precision, Recall, and F-measure Overview Using GE 2015 Stream

In general, the SFPM missed many events because in most cases the terms identified forming the topic are very general. A sample of the identified topics from the first window-batch in the GE 2015 period as yielded by the D-FPM approach, the SFPM approach, and those yielded by the news headlines (from the gold-standard list) for that day are shown in Table 3.15. There were 3 news events found in the news headlines, but both the D-FPM approach and the SFPM approach detected 2 events only (they both missed one event). However it is clear that the terms identified by the SFPM are very general, and are hard to be tied to an event. Whereas using the D-FPM approach, important terms matching the news events are identified and can be linked with the published news events. In more detail, Table 3.15 shows the first event, which is "The Liberal Democrat leader launches his party's manifesto"; the terms forming the event in the D-FPM approach were "rivals, outright, win, dem, nick, clegg, says, lib, launch". From the identified terms we can say that the topic is related to the Liberal party manifesto launch. On the other hand the terms the SFPM identified as forming the event were "cleggs alex going salmond". It is obvious that the event discusses something related to Nick Clegg (the Liberal Democrat leader) but it is not clear what. Furthermore, if you look at the second topic from the news in Table 3.15, which was "The UKIP party low-tax revolution" and compare the terms identified by both approaches, you can clearly see that the event was detected by the D-FPM approach but not by the SFPM approach.

D-FPM	SFPM	BBC News on 29/4/2015
rivals, outright, win, dem, nick, clegg, says, lib, launch	cleggs, alex, going, salmond	Leader Nick Clegg launched his party's manifesto "Every vote for the Liberal Democrats matters". `http://www.bbc.co.uk/news/in-pictures-32315512`
lowtax, revolution, want, farage, nigel, ukip, set, free, pledges	alternative, austerity, ukip, want	UKIP would make working people better off through a "low-tax revolution", Nigel Farage has said as he launched his party's election manifesto. `http://www.bbc.co.uk/news/election-2015-32312687`
NA	NA	In Northern Ireland SDLP leader Alasdair McDonnell unveiled his party's manifesto at the Holiday Inn Hotel, Belfast. `http://www.bbc.co.uk/news/in-pictures-32315512`

Table 3.15: Detected Topics Sample on 29/4/2015 by D-FPM and the SFPM approach Compared to News

Before we conclude this chapter, we want to summarize the targeted terms selection criteria, which is the main aspect for topics detection using the feature pivot approach (discussed in Chapter 2). Table 3.16 shows a summary for the terms selection criteria considered in the following studies [39–41] (again were discussed in Chapter 2) along with our own intended framework (the D-FPM) for detecting the occurring events from streams of data. The SFPM [39, 41] selects the top K-terms with the highest ratio of appearance in both corpora (the investigated and the reference corpus), K is a number which is given as an input. The HUPC [40] on the other hand used a very low fixed threshold value but only considers the top 3k frequent terms.

Accordingly, in our topic/event detection framework (the D-FPM), we wanted to overcome the limitations inherent in determining the number of selected terms in advance; this pre-determination is especially problematic with respect to changeable and dynamic environments (i.e. data streams) such as social networks (Twitter). Thus, the term selection threshold (the support value) was dynamically defined by considering the number of terms and their frequencies across each text-post or message received.

	SFPM	HUPC	D-FPM
Determine no. of terms in advance	Yes	Yes	No
How?	Select the top K-terms. K is a number given as input. The selection of top K-terms is based on the probability of appearing in the current corpus and an independent corpus [39, 41].	Select the top 3k frequent terms to represent a batch. Each batch contains a fixed number of terms [40].	Terms with frequency satisfying the defined support value are only selected. No. of terms in each window is not fixed.
FP formulation threshold	Partially calculated ($Q_{b,c}$).	Fixed threshold (very small value that is close to 0).	Dynamic calculated support value.
Specify no. of FPs in advance	No	Yes	No

Table 3.16: Terms selection comparison

3.4 Summary

In this chapter, we have had introduced the Dynamic-FPM (D-FPM), which extends the FPM method by using a dynamic support value in order to replace the fixed a-priori set support value. We have developed a dynamic support definition method to be used with a FPM approach to detect the daily occurring events from text data streams.

The D-FPM differs from other approaches using the FPM on text streams [4, 39–41] to identify the occurring topics/events, by dynamically defining the support value for every batch in the text stream, rather than having it as fixed value that dose not change.

Specifically, we employed the FP-Growth algorithm along with our dynamic support definition method in order to detect the occurring events/topics from a Twitter stream. The identified events depend on what is frequently mentioned and extensively discussed in the investigated stream. The support is defined separately for each day, and is then used for detecting the events which occur in that day (using that day's window-batch). The support definition method mainly depends on the current day's window-batch size and the frequency of occurrence of keywords.

A strict support definition method was proposed for small windows to avoid the detection of insignificant events, hence, two versions of the support definition method were presented. A logistic regression model is used in order to classify each incoming window-batch for being either large or small window. The size of each incoming window-batch must be specified prior to defining the support value. Essentially, the dynamic support value in a window-batch is set based on the average frequency of the terms and the median value.

Experiments were conducted on Twitter streams related to the UK General Election 2015, and the Greece Crisis 2015, to detect the occurrence of events and to evaluate the dynamic support definition method. The detected events were associated with what was published on the same day on the news wires. If the detected event was found on the same day a related news headline was published, then it was said to be a true event and it was said to be a false or insignificant event otherwise.

We evaluated our event detection framework against the SFPM approach in [39]; this is a topic detection approach which is based on the FPM method. We compared the detected topics from both approaches with a gold-standard list of topics. The gold-standard list was created based on news articles published by the BBC News, regarding the GE 2015 period, see Appendix 2. We applied the precision and recall metrics in order to evaluate the identified topics yielded by both approaches. The results showed that the precision of our approach (the D-FPM) was three times higher than that of the SFPM approach, the recall on the other hand was almost the same for both approaches.

Chapter 4

Time-Series Stream Analysis and Event Identification

4.1 Introduction

With respect to the financial markets, capturing significant price movements is crucial as this presents investment opportunities. Looking at a high frequency time-series data stream in relation only to physical time (e.g., by looking at daily closing prices) fails to capture the full activity of price movements [34,131]. In finance, High Frequency Data (HFD) refers to data observations taken daily or at even finer time scales [33].

Detecting events (i.e., significant price movements) from high frequency time-series data streams is challenging due to their (the streams') characteristics — data elements arrive in real-time or near real-time and at high velocity, and the streams are of unbounded size. A further challenge is that, it is not possible to backtrack over the data elements which have arrived in the past, or keep track of and review the entire history.

The Directional Change (DC) approach, observes price fluctuations depending on intrinsic time rather than on physical time [34, 35]. Intrinsic time is an event based timing system (irregularly spaced in time), as opposed to physical time — which is a point based timing system depending on fixed time intervals [34, 131]. The DC is an event-driven approach for summarizing price movements; these movements consist of two type of events: downturn and upturn [133]. A DC event is detected if the market price change exceeds a fixed a-priori threshold value (positively or negatively).

A real world event is defined as the occurrence of something associated with both a place (where the event happened), and a time (when the event happened) [86]. Event detection is the process of searching for indications of events. In DC, an event is said to have been found (either an upturn or a downturn), if the price change between two values exceeds a given threshold value [35] (positively or negatively). In addition, this threshold value is fixed and is used for the entire data stream (i.e. it does not change).

The reasons behind the occurrence of such events are various and not easy to predict in advance (except with respect to regular events whose timing and duration are known). Events may also be of varying magnitude and significance. Hence, a threshold which is considered to be adequate for the detection of events under certain circumstances may not be able to facilitate the detection of events under different circumstances. Thus, using a fixed threshold value to detect events across the board is not always effective, especially for changeable and dynamic environments. Therefore, a dynamic threshold may be more appropriate — in order to allow for the detection of events of different magnitudes.

Usually when using the DC approach, more events are detected when a relatively small threshold value is set; this is desirable if the trend being examined is likely to continue in the same way for some time as investors can take action before prices go up or down even more. On the other hand, a small threshold is not desirable when there are transient events which play out over a short time and have limited effects. In addition, when large threshold values are used only larger scale and "true" events (i.e., those which have an influence beyond the stock price movements) are detected, but by the time an event is detected, generally the peak price has almost been reached. Hence, we believe that replacing a a-priori fixed threshold with a dynamic one may deal with these issues, by having a daily defined threshold value that is used only on that day to detect DC events if any.

Several studies including, but not limited to, [171–175] have shown that stock prices may respond to different kinds of events. These studies looked at associating particular events with particular stock price changes. Different types of events — economic, political, disasters, etc. — can have different effects on the different markets world-wide. These effects may be of different magnitudes and therefore they may have different "impacts", i.e., fluctuations in the price. Consider for example the on-going sovereign debt crisis in Europe. In particular, on the 27th October 2011, European politicians announced a deal for cutting the Greek debt in half. As a result, the S&P 500 index rose by 3.4%, while French and German stocks gained 5%. But just over the following week, all these stock gains were wiped out as Greece's Prime Minister announced a referendum on that deal. In subsequent days, and when other Greek politicians voiced their opposition to the referendum, stocks rose sharply again [176].

In this present study, we are not explicitly interested in identifying the various types of events and their nature or even their exact magnitude; what we are interested in is developing methods for event detection from streams of structured data — i.e., for detecting significant price fluctuations in price time-series data streams, so that investors or artificial software agents can detect movements in the market that they can potentially react to and take advantage of. We aim to adapt the DC event approach so that it becomes capable of running on high frequency data streams even more effectively. This we hope to achieve by using a dynamic threshold instead of a fixed, a-priori one. Our motivation is to provide (i) decision support methods for traders/investors; and (ii) automatic event detection/identification methods for software trading agents.

The rest of this chapter is organized as follows. The next section (Section 4.2) discusses the high frequency time-series stream analysis and event identification framework; followed by a description of the dynamic threshold definition method, which is the core concern of this chapter. The subsequent section (Section 4.3) presents the experimental evaluations, analysing and discussing the experiments conducted and their findings. The chapter ends with a summary and conclusions in Section 4.4.

4.2 Developing Methods for Event Identification

In this section, we first formulate the problem of event identification from high frequency time-series data streams in general, and then we introduce our framework.

Significant price movements within high frequency time-series data streams tend to be unevenly spaced; hence using physical, fixed time interval data (the daily closing price for example) to capture the state of a financial market may result in the missing of significant intraday price fluctuations [34, 131]. Moreover, this could cause investors to miss a buying or selling opportunity. Thus, we will employ the DC approach to price time-series data streams for markets that operate over specific opening and closing times. Price fluctuations will be considered as an event occurrence indicator. In addition, we aim to introduce a method to dynamically define the threshold value, instead of setting it, a-priori, at a fixed value, as has been typically the case in the use of the DC approach [6, 36–38, 42–50, 52, 134]. We believe that a dynamic threshold will enable us to better identify events.

A DC event is said to have been found when the current price *p(t)* moves higher than the last lowest price (p_l) by a given threshold value, or the current price *p(t)* moves lower than the last highest price (p_h) by a given threshold. DC events can be one of two forms: an upturn event or a downturn event. An upward run is the period between the current upturn event and the next downturn event, it consists of an upturn event and an upward OS event. While a downward run is the period between the current downturn event and the next upturn event, it consists of a downturn event and a downward OS event.

Such a "DC event" may or may not be an actual event that has been or will be reported via news outlets. So, once a price change between *p(t)* and p_l/p_h exceeds the threshold value (positively or negatively), then a DC event is triggered — without taking into account or considering any other influence. Other aspects which might be taken into account could include, for instance, the time elapsed between the current price *p(t)* and that of the last p_l/p_h price. However, it could be said that the longer the elapsed time between these two, the lesser, in terms of significance, the event becomes, and thus the higher the threshold should be. The probability that an event will be triggered in such circumstances could thus be reduced. Furthermore the previous day's price change, especially where that price change was in the opposite direction to the current trend where p_l/p_h values remains with no change (i.e. prices does not decrease/increase even more in the current trend), could be another such influencing aspect. Consider the following scenario in an upward run with 2% fixed threshold. Suppose the share prices which are being examined drop on a particular day by 1.9%. However, as long as the threshold is not exceeded (negatively, in this case), no downturn event is detected although the price decrease is significant. However, if on the next day the prices drop by only 0.1%, an event (a downturn event) is now detected as the threshold has now been met, even though this particular day's price change is not significant.

Although events may affect the price movements on a specific day, typically events may be unfolding over several days and hence it would be reasonable to examine and take into account the price fluctuations of the previous day as this can be regarded as a direct indicator of the occurrence of an event. So, detecting DC events from price time-series data streams based only on the price change between the current *p(t)* and

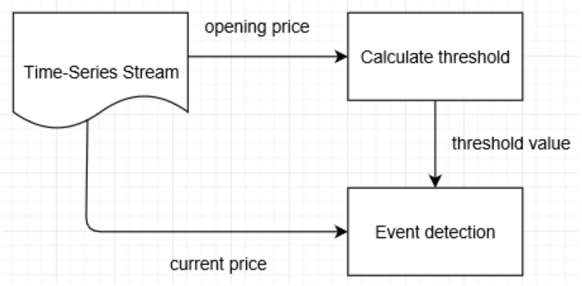


Figure 4.1: DC Abstract Model

the last p_h/p_l price is not enough, including the previous day's price transitions as well makes more sense. Hence, daily dynamic threshold values are defined in order to replace fixed a-priori thresholds, and to detect the occurring events if any.

The aim of our work is to build on the original Directional Change approach, which utilises a static threshold, and extend it to utilise a dynamic threshold. Our premise is that a dynamic threshold may be more useful and enable the more accurate detection of events — which can vary in magnitude and significance. To demonstrate and validate the developed methods, we will be using minute-by-minute stream data, i.e. stock prices, generated by a financial market.

The overall abstract model for the high frequency time-series data stream analysis and event identification is shown in Figure 4.1. Once the time-series data stream receives an opening price, then the dynamic threshold value is defined and used on that day to detect DC events if any. Otherwise if the received price is not an opening price, then it is just examined for event detection.

4.2.1 Introducing Dynamic Thresholds in the Directional Change

The threshold inherent to the DC approach is used to detect upturn and downturn events. Whenever the current price $p(t)$ is higher than the last low price (p_l) during a downward run, or is lower than the last high price (p_h) during an upward run, by a given threshold value, then it is said that an event has been found.

Accordingly, our dynamic threshold definition method considers the price change between the current price $p(t)$ and the last p_h/p_l, together with what of significance occurred during the previous day. In the following, we describe how the threshold is defined on a daily basis.

First, we start with the basic idea of the DC event detection aspect, which is calculating the percentage change between the current day's opening price (Cp_{open}) and the last high p_h or low p_l price reached in the course of the current upward/downward run. This was considered in order to capture significant price movements, Equation 4.1 shows the upward run price change and Equation 4.2 shows the downward run price change.

In the Upward run case:

$$upward\_PC = \frac{p_h - Cp_{open}}{p_h} \times 100 \tag{4.1}$$

While in the Downward run case:

$$downward\_PC = \frac{p_l - Cp_{open}}{p_l} \times 100 \tag{4.2}$$

After that, we go on to consider what happened on the previous day, specifically. Hence, we consider the percentage change between the previous day's opening price (Pp_{open}) and the previous day's closing price (Pp_{close}) to discover the overall effect of the price transitions that occurred on the previous day. If the difference between the opening and the closing price is large, this could be considered an indication of an event, see Equation 4.3.

$$previous\_PC = \frac{Pp_{open} - Pp_{close}}{Pp_{open}} \times 100 \tag{4.3}$$

Though by considering the price change that occurred the previous day may give an indication of whether an event has occurred or not, there can be extreme situations such as for instance, when the EU Referendum poll results was announced on Thursday 23 June 2016. The British Pound Sterling suffered its biggest one-day sell-off in recent history, as a reaction to the unexpected news that the UK had voted to leave the European Union. The Pound Sterling suffered a sharp drop in the early hours of Friday, in London, from \$1.50 against the US dollar to just \$1.33 — though it closed at \$1.3681[1]. Thus, to define the dynamic threshold, we also consider the percentage change between the previous day's closing price (Pp_{close}) and the current day's opening price (Cp_{open}) to uncover the price transition which has taken place overnight. The higher this percentage is, the higher is the likelihood that an event has occurred/is occurring, see Equation 4.4.

$$overnight\_PC = \frac{Pp_{close} - Cp_{open}}{Pp_{close}} \times 100 \tag{4.4}$$

The overall threshold is then defined to be the sum of the three aforementioned metrics $up/downward\_PC$, $previous\_PC$, and $overnight\_PC$ (as per Equation 4.5), unless something extreme is found on the previous day or overnight. If something has happened the previous day, then the threshold is set to be the sum of the $up/downward\_PC$ and $previous\_PC$ metrics only (Equation 4.6), otherwise if something significant has happened overnight which affected the difference between the previous day's closing price and the next day's opening price, then the threshold is the sum of the $up/downward\_PC$ and $overnight\_PC$ metrics only (Equation 4.7). This condition has been put in place (the reduction of the threshold value by using Equations 4.6, 4.7) to ensure the detection of an event — as we can be confident that such has indeed occurred. In addition, consider for example, something extreme (i.e. significant price change) happened the previous day and continued to happen overnight (between previous day's closing price and the reporting of the current day's opening price). If this provision were not made (the reduction of the threshold value by using either Equations 4.6 or 4.7 instead of Equation 4.5), it would results in a threshold value which was too high, and this would then suppress the detection of further events.

Furthermore, to avoid the situation where events would be detected based only on the price change between the current day opening price Cp_{open} and p_h/p_l, which is the $up/downward\_PC$ metric (such situations can take place in stable days when nothing noticed the previous day or overnight, i.e. metrics $previous\_PC$ and/or $overnight\_PC$ values are too small). We have considered adjusting the values of $previous\_PC$ and $overnight\_PC$ using small weighting values (w_1 and w_2); this would be to increase the values of these metrics in situations where they are too small and would have no effect. Finally, if the values of $previous\_PC$ or $overnight\_PC$ are already large, these weights are of small value and will still be satisfied within the day as prices are increasing or decreasing.

$$Threshold = up/downward\_PC + (previous\_PC \times w_1) + (overnight\_PC \times w_2) \tag{4.5}$$

$$Threshold = up/downward\_PC + (previous\_PC \times w_1) \tag{4.6}$$

$$Threshold = up/downeard\_PC + (overnight\_PC \times w_2) \tag{4.7}$$

[1] https://www.theguardian.com/business/live/2016/jun/24/global-markets-ftse-pound-uk-leave-eu-brexit-live-updates

These weights (w_1 and w_2) should help in preventing the spurious detection of events in circumstances where the values of *previous_PC* and *overnight_PC* can have no effect on the setting of the threshold value, and at the same time they should not reduce the chances of detecting an event when the values of *previous_PC* and *overnight_PC* are already large. Thus, we have considered the weights w_1 and w_2 for the various factors just explained above, which can influence the setting of the dynamic threshold. But these weights may differ from market to market and it may be more appropriate to identify the best approximate values experimentally. We have determined experimentally the best values for these weights in the specific market that we have used as part of our experimental work. This is illustrated in section 4.3.2.

Algorithm 3 describes the method for defining the daily dynamic threshold, which is then used to replace the fixed threshold when employing the DC in Algorithm 1. It takes as an input the price time-series data stream and returns the threshold as an output. Each incoming price from the time-series data stream is examined in relation to DC event detection. In line 2 the incoming price is checked to see whether it is an opening price, if so, then the current day's dynamic threshold is defined; if not, then the price is examined for DC event detection only. Lines 3-5 deal with the first day of the stream when an initial, fixed threshold is used for detecting DC events. Lines 6 to 27 are the core of the algorithm; they show how the daily dynamic threshold is defined as per the above.

Determining that something significant has happened the previous day or overnight which is performed by lines 18 and 21 of the dynamic threshold definition algorithm, Algorithm 3. In these circumstances, we can use either Equation 4.6 or Equation 4.7 for the dynamic threshold definition. A decision tree was built using a labelled training dataset to help make a decision of which dynamic threshold definition equation should be used. In other words, a decision tree was built to predict when Equation 4.6 and Equation 4.7 can be used (the training dataset and the decision tree details are shown later in the experimental work).

4.3 Experimental Work

In the experimental work section, we employ the DC event approach to high frequency time-series data streams both using our dynamic threshold definition method

Algorithm 3: Define a Dynamic Threshold Value for DC

```
Input  : $TSstream$: Time-series stream
Output: $Threshold$: the threshold defined value
for each incoming price $p_{(t)}$ in TSstream do
    if $p_{(t)}$ is opening price then
        if $p_{(t)}$ is firstday opening price then
            threshold=fixed threshold
        else
            $Pp_{open}$= previous day open price
            $Pp_{close}$= previous day close price
            $Cp_{open}$=$p_t$
            if Upturn Event then
                $P_h$=last high price in current upward run
                $upward\_PC$=percentage-change($P_h$, $Cp_{open}$)
            else
                $P_l$=last low price in current downward run
                $downward\_PC$=percentage-change($P_l$, $Cp_{open}$)
            end
            $previous\_PC$=percentage-change($Pp_{open}$, $Pp_{close}$)
            $overnight\_PC$=percentage-change($Pp_{close}$, $Cp_{open}$)
            if something noticed the overnight then
                Threshold=$up/downward\_PC$+($overnight\_PC \times w_2$)
            else
                if something noticed previous day then
                    Threshold=$up/downward\_PC$+($previous\_PC \times w_1$)
                else
                    Threshold=$up/downward\_PC$+($previous\_PC \times$
                     $w_1$)+($overnight\_PC \times w_2$)
                end
            end
        end
    end
end
```

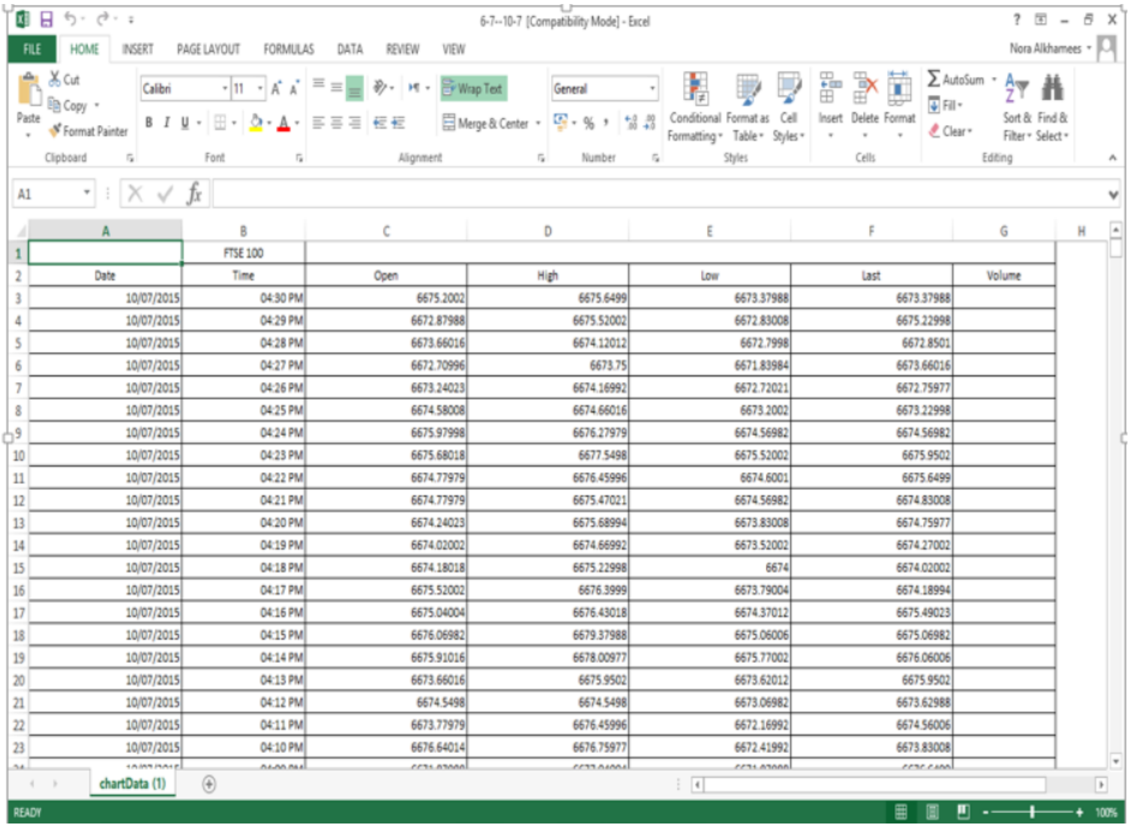
and using a range of different fixed threshold values —– to detect the occurrence of DC events (upturn and downturn events), and then compare and discuss results. We begin (in Sub-section 4.3.1) by describing the financial time-series stream collection and preparation process. Then, the dynamic threshold is set in Sub-section 4.3.2. We apply the DC approach in order to detect the occurring events in the following sub-section. Finally, we conclude this section by analysing and discussing the events detected using the daily defined dynamic threshold and different fixed thresholds.

4.3.1 Data Collection, Description and Preparation

In order to demonstrate the application of the DC approach and the dynamic threshold algorithm, we have collected data from the FTSE 100 index minute-by-minute prices from “Reuters Thomson One” [177], for the period from July 2015-May 2017; see Figure 4.2 for a snapshot of our collected data. The FTSE 100 index intraday prices have been collected on a week by week basis. The FTSE 100 is the index of the largest 100 companies in the London Stock Exchange (LSE). In the prices stream, if for any reason no price is given (a single minute price is missing), we take the previous minute price which was received. In other words, the last captured value is carried forward when missing data arises [178].

We chose the FTSE 100 index for our experiments because it captures the overall performance of the LSE market. Furthermore, afterwards when we want to put together and draw inferences from the text stream and the price time-series stream (later in Chapter 7), the FTSE 100 prices stream could be cross-referenced to the GE 2015 text stream and the Greece Crisis 2015 text stream, as it is directly linked to the UK General Elections (both are, of course, based in the UK) and also to the Greece Crisis as it can be affected by European level events.

On a minute-by-minute basis we obtained the opening, high, low, and closing prices. As there are generally no big differences between these four prices (opening, high, low, and closing), we take their mean average. Furthermore, in order to access and analyse any of the time-series data streams in JAVA, we had to download the Apache POI Java library (the Java API for Microsoft Documents) [179]. This was so that we could access the Excel spread-sheets and read the data in their columns and cells. Then we implemented the DC algorithm shown in Algorithm 1 to attempt to detect the DC events which may occur.



	A	B	C	D	E	F	G
1		FTSE 100					
2	Date	Time	Open	High	Low	Last	Volume
3	10/07/2015	04:30 PM	6675.2002	6675.6499	6673.37988	6673.37988	
4	10/07/2015	04:29 PM	6672.87988	6675.52002	6672.83008	6675.22998	
5	10/07/2015	04:28 PM	6673.66016	6674.12012	6672.7998	6672.8501	
6	10/07/2015	04:27 PM	6672.70996	6673.75	6671.83984	6673.66016	
7	10/07/2015	04:26 PM	6673.24023	6674.16992	6672.72021	6672.75977	
8	10/07/2015	04:25 PM	6674.58008	6674.66016	6673.2002	6673.22998	
9	10/07/2015	04:24 PM	6675.97998	6676.27979	6674.56982	6674.56982	
10	10/07/2015	04:23 PM	6675.68018	6677.5498	6675.52002	6675.9502	
11	10/07/2015	04:22 PM	6674.77979	6676.45996	6674.6001	6675.6499	
12	10/07/2015	04:21 PM	6674.77979	6675.47021	6674.56982	6674.83008	
13	10/07/2015	04:20 PM	6674.24023	6675.68994	6673.83008	6674.75977	
14	10/07/2015	04:19 PM	6674.02002	6674.66992	6673.52002	6674.27002	
15	10/07/2015	04:18 PM	6674.18018	6675.22998	6674	6674.02002	
16	10/07/2015	04:17 PM	6675.52002	6676.3999	6673.79004	6674.18994	
17	10/07/2015	04:16 PM	6675.04004	6676.43018	6674.37012	6675.49023	
18	10/07/2015	04:15 PM	6676.06982	6679.37988	6675.06006	6675.06982	
19	10/07/2015	04:14 PM	6675.91016	6678.00977	6675.77002	6676.06006	
20	10/07/2015	04:13 PM	6673.66016	6675.9502	6673.62012	6675.9502	
21	10/07/2015	04:12 PM	6674.5498	6674.5498	6673.06982	6673.62988	
22	10/07/2015	04:11 PM	6673.77979	6676.45996	6672.16992	6674.56006	
23	10/07/2015	04:10 PM	6676.64014	6676.75977	6672.41992	6673.83008	

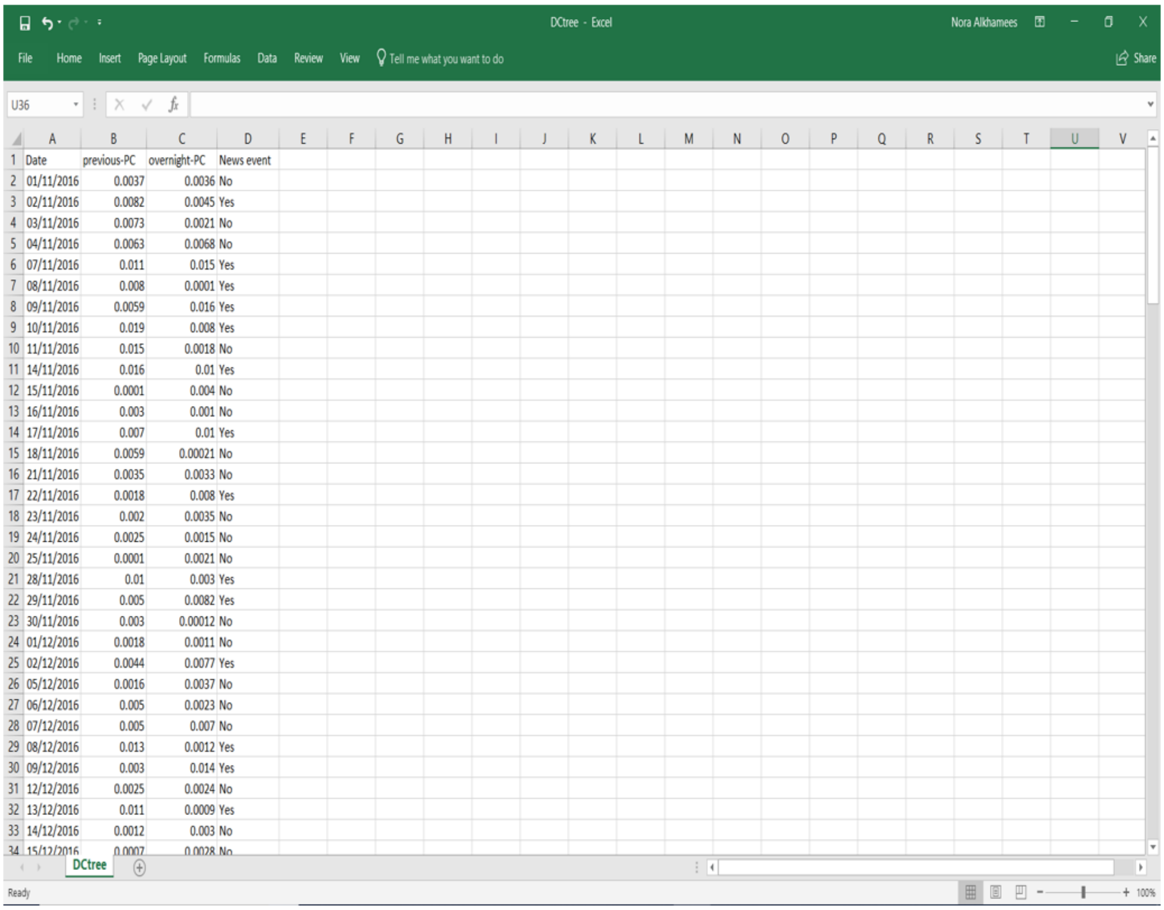
Figure 4.2: Excel Sheet Snap Shot for FTSE100 Data

4.3.2 Dynamic Threshold Definition

Before we can define the daily dynamic threshold, we have to decide when it is required to use either Equation 4.6 or Equation 4.7 for the dynamic threshold definition. The determination of when to apply which of these equations (Equation 4.6 and Equation 4.7) was performed by building a J84 decision tree using a labelled training dataset (of more than 150 days of the FTSE 100 prices from the "Reuters Thomson One" [177] from November 2016 till May 2017), using the WEKA data mining tool [180]. J48 decision tree is the JAVA implementation of C4.5 decision tree [181] in WEKA [180]. WEKA is a data mining toolbox consisting of a collection of machine learning algorithms, it is open source and is written in JAVA.

For each day in the training data set we had the following: date, previous day's percentage change, overnight percentage change, and whether something was noticed the previous day/overnight or not. The decision regarding whether something was noticed the previous day/overnight or not was manually made according to whether a news event (i.e news article) was released by the BBC News (www.bbc.co.uk) regarding the FTSE 100 on that day or not. There were 33 news articles relating to the FTSE 100 during the training period. So, in the "News Event" column of this training dataset, we had 33 rows with the value "Yes" and 112 rows with the value "No". A snapshot of the training dataset is shown in Figure 4.3. Appendix 3 shows the list of FTSE 100 News from the BBC during the training phase period.

The decision tree which was built is shown in Figure 4.4. If the percentage change between the previous day's closing price and current day's opening price (overnight) exceeds 0.7%, or if the percentage change between the previous day's opening and closing prices is greater than 0.76%, then it is deemed that something of significance has occurred the previous day or overnight. Thus Equations 4.6 or 4.7 are used to define the threshold value for that day. The overnight percentage (0.7%) was a little lower because it is considered a stronger indicator that something has happened, and is continuing to happen — more than the previous day's prices (0.76%).



1	Date	previous-PC	overnight-PC	News event
2	01/11/2016	0.0037	0.0036	No
3	02/11/2016	0.0082	0.0045	Yes
4	03/11/2016	0.0073	0.0021	No
5	04/11/2016	0.0063	0.0068	No
6	07/11/2016	0.011	0.015	Yes
7	08/11/2016	0.008	0.0001	Yes
8	09/11/2016	0.0059	0.016	Yes
9	10/11/2016	0.019	0.008	Yes
10	11/11/2016	0.015	0.0018	No
11	14/11/2016	0.016	0.01	Yes
12	15/11/2016	0.0001	0.004	No
13	16/11/2016	0.003	0.001	No
14	17/11/2016	0.007	0.01	Yes
15	18/11/2016	0.0059	0.00021	No
16	21/11/2016	0.0035	0.0033	No
17	22/11/2016	0.0018	0.008	Yes
18	23/11/2016	0.002	0.0035	No
19	24/11/2016	0.0025	0.0015	No
20	25/11/2016	0.0001	0.0021	No
21	28/11/2016	0.01	0.003	Yes
22	29/11/2016	0.005	0.0082	Yes
23	30/11/2016	0.003	0.00012	No
24	01/12/2016	0.0018	0.0011	No
25	02/12/2016	0.0044	0.0077	Yes
26	05/12/2016	0.0016	0.0037	No
27	06/12/2016	0.005	0.0023	No
28	07/12/2016	0.005	0.007	No
29	08/12/2016	0.013	0.0012	Yes
30	09/12/2016	0.003	0.014	Yes
31	12/12/2016	0.0025	0.0024	No
32	13/12/2016	0.011	0.0009	Yes
33	14/12/2016	0.0012	0.003	No
34	15/12/2016	0.0007	0.0028	No

Figure 4.3: "Something has Happened" Training Dataset

In addition, we had to set the weights w_1 and w_2 values, these weights are used to prevent the detection of DC events in cases where metrics $previos\_PC$ (which measures the previous day price change) and $overnight\_PC$ (which measures the overnight price change) have no or very limited effect. Refer to Table 4.1 which shows the values assigned to w_1 and w_2, the number of detected DC events, and the number of missed and falsely

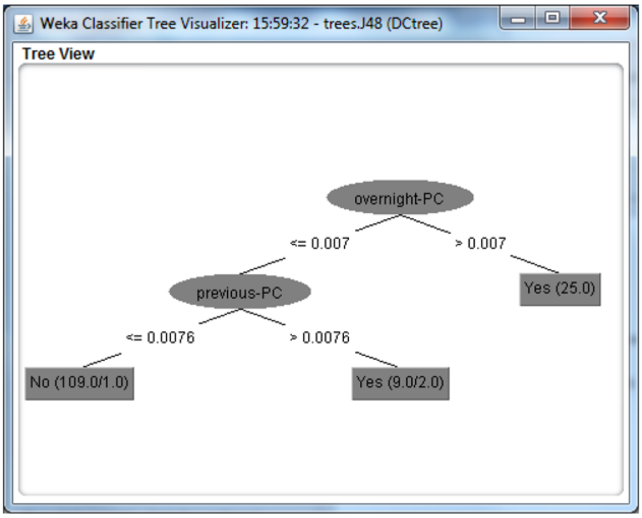


Figure 4.4: Something has Happened Decision Tree

identified events (if an event was detected on the same day that a FTSE 100 news headline was published, then the event was said to be a "true" event, otherwise it was said to be a false event). Initially we did not use any weights at all; we then gradually increased the value of both w_1 and w_2 to see how this would affect the threshold and so the events which could be detected.

w_1 (previous day weight)	w_2 (overnight weight)	Detected DC events	False/Missed events
No applied weight	No applied weight	24	5
No applied weight	10%	24	5
10%	No applied weight	22	3
10%	10%	22	3
10%	15%	22	3
15%	10%	20	2
15%	15%	20	2
15%	20%	20	1
20%	15%	18	4
20%	20%	18	4
20%	25%	16	5
25%	20%	18	3

Table 4.1: Weights w_1 and w_2 Assigning Attempts

We looked at the different values which could be used for these weights and what these values might represent, and we concluded that the metric *overnight_PC* can be given a higher weight than *previous_PC* as we are almost certain that the price change between the previous day's closing price and current day's opening price will even increase/decrease more throughout the day (that price change was identified overnight), thus a higher weight (and therefore threshold) will still be satisfied as prices will most probably continue to increase/decrease. The weighting values which facilitated the most effective detection of events (detecting events on the same day as news headlines were released) were assigned.

Hence the weighting values which were actually assigned are as follows:

$w_1 = 15\%$

$w_2 = 20\%$

As a data stream is, as the name suggests, a continuous stream of data, and the dynamic threshold definition requires the previous day's prices as an input, the threshold for the day on which processing starts, in the absence of the previous day's price information, will be set in a a-priori fashion based on the limited information available. The threshold will be set dynamically from the second day onwards. Each day, we aim to have a threshold value that is appropriate for detecting true or significant events. Threshold values that are too low, which can occur after the detection of a DC event where the values of $upward/downward\_PC$ are too low, as the values of p_h/p_l has been just set, may cause the process to trigger the reporting of spurious or false DC events. In order to avoid this, a minimum threshold was applied instead to ensure that this was avoided. In fact, none of the detected DC events were detected by that minimum threshold value.

The proposed dynamic threshold definition method can be used with real-time or near real-time data since the calculations depends only on the previous day's prices and the current day's opening price. Hence, as an API streams the prices, each streamed price is examined in turn, if it is an opening price then the dynamic threshold is defined and used for that day, otherwise the price is only examined for the purpose of DC event detection.

4.3.3 Applying the DC Event Approach

We ran an experiment on the FTSE 100 minute-by-minute data stream which was yielded from the period July 2015 until the end of February 2016. A dynamic threshold was defined daily and was used to detect the occurrence of DC events. Verifying the occurrence of events is not a straightforward matter, and for the purposes of this work we examined and cross-referenced the DC events with events which were mentioned in published news headlines from the BBC News (www.bbc.co.uk) regarding the FTSE 100 index. If a DC event was detected on the same day that a news headline was published,

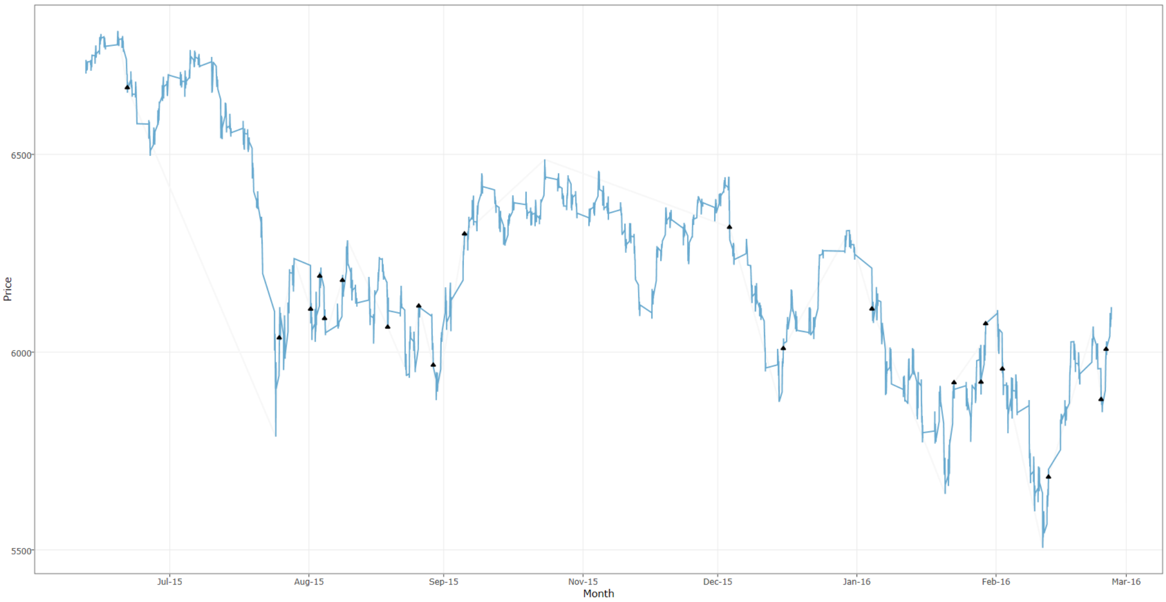


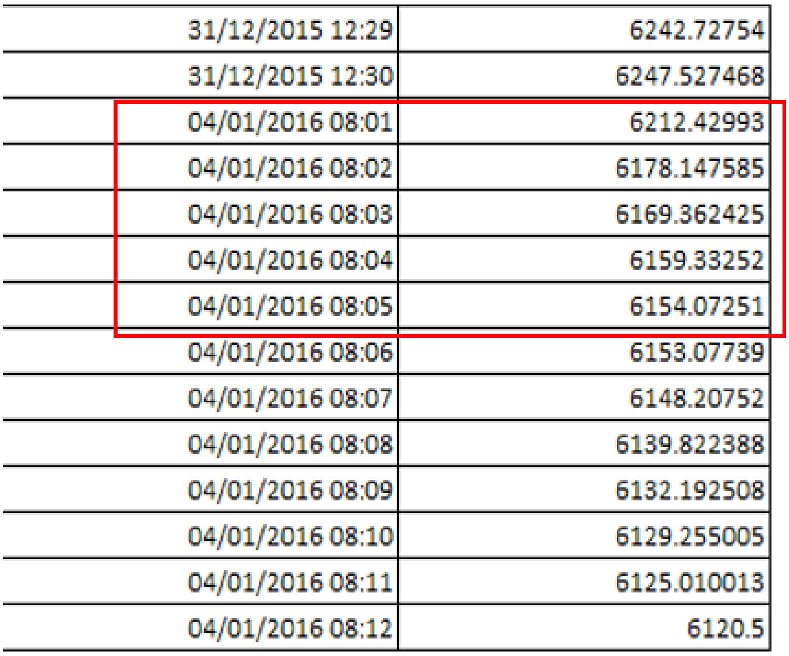
Figure 4.5: DC Detected Events from July 2015-Feb 2016

then the event was said to be a true event. In Appendix 4, we present the list of News Headlines (in Table A4.1) appearing in the BBC News for the FTSE 100 index in the period from July 2015 until February 2016.

During the investigation period we were able to detect 20 DC events from the minute-by-minute prices stream using a daily dynamically defined threshold values, ranging from 0.014- 0.071. Figure 4.5 shows the FTSE 100 index for the investigated period; here, each event is shown by a black triangle. Generally DC events alternate, the algorithm does not detect two of the same type of event sequentially (e.g. two downward events), but the latter will still be showing as an OS event. However, all the detected DC events, using the dynamic threshold, matched news items relating to the FTSE 100 which were published in major news outlets.

An event may be missed if the defined threshold value is too large and so is never exceeded by the data which is being received. However, if prices continue to increase or decrease an event may still be detected as the threshold value will change from the beginning of the next day. Thus, in order to avoid large threshold values if something significant has occurred the previous day and/or overnight, we consider either the percentage change generated by the previous day's or by the overnight prices to define the threshold value (Equations 4.6 and 4.7).

Furthermore, events may be missed if the price drastically changes immediately after the opening price has been reported (a sudden price change); this will result in a defined threshold value that is not able to capture or reflect the actual prices encountered, and so events may be missed. Consider Figure 4.6 for instance, where the prices are intensely falling after the opening price on 4th Jan 2016. Dealing with this issue (a sudden price change after opening price) by postponing the definition of the dynamic threshold value until the fifth minute-by-minute value after the opening price, and use the average of the first five (per minute) prices instead of the opening price by itself. We chose the first five minutes only, because we did not want to delay the definition of the threshold any longer than this — as it may cause other issues to come to the fore such as delaying the detection of an event. At the same time, if no sudden price change occurs, taking the average of the five minute-by-minute prices does not effect large changes to the threshold value, as the prices do not diverge significantly from the opening price. We were able to detect events in a more timely fashion when we considered the first five minute-by-minute prices in this way, instead of just the opening price alone. In addition and as a precautionary step in case a sudden price change does take place during the day, we kept track of the streamed prices (on a minute-by-minute basis). Thus the process can react if any such changes do indeed occur. Hence, we considered the percentage change between the current day's opening price and each minute-by-minute price. Once a price change is noticed, we use the defined threshold regardless of its value (no minimum threshold value is required).



31/12/2015 12:29	6242.72754
31/12/2015 12:30	6247.527468
04/01/2016 08:01	6212.42993
04/01/2016 08:02	6178.147585
04/01/2016 08:03	6169.362425
04/01/2016 08:04	6159.33252
04/01/2016 08:05	6154.07251
04/01/2016 08:06	6153.07739
04/01/2016 08:07	6148.20752
04/01/2016 08:08	6139.822388
04/01/2016 08:09	6132.192508
04/01/2016 08:10	6129.255005
04/01/2016 08:11	6125.010013
04/01/2016 08:12	6120.5

Figure 4.6: After Opening Intense Price Change

4.3.4 Discussion

In this sub-section, we show whether events are more effectively detected with a static threshold value or with a daily dynamic defined one. We looked at four different fixed threshold values (randomly chosen): the 3% threshold, which is considerably a low value, the 4% and 5% thresholds, which are medium sized, and lastly the 6% which can be considered a high threshold value. Fixed thresholds below 3% were not examined, because we want the number of detected DC events to be not far away from the number of published News events (i.e. News articles). As too low fixed threshold are easily satisfied, and so will detect more events.

Figure 4.7 demonstrates the DC events detected by the use of the 0.03 fixed threshold value; there were 22 DC events relating to the period from July 2015-Feb 2016. The use of the 0.04 fixed threshold resulted in 14 DC events for the same period, see Figure 4.8. Furthermore, Figure 4.9 shows the DC events detected by the use of the 0.05 fixed threshold; there were 12 DC events for the same period. With the use of the 0.06 fixed threshold, the number of DC events drops to just 10, see Figure to 4.10.

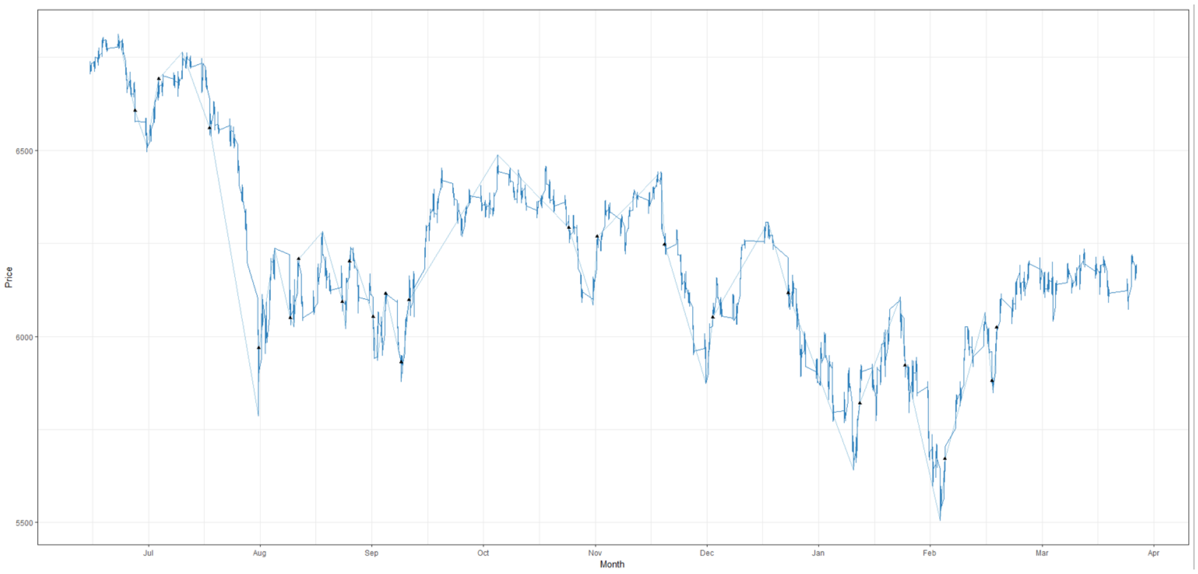


Figure 4.7: DC Detected Events from July 2015-Feb 2016 by the 3% Fixed Thresold

In the following sub-sections, we show some snapshots relating to the investigated period and discuss the detected DC events and findings. We will show different cases by looking at the DC events which were detected using our dynamic defined threshold, and also those which were detected using a number of different fixed threshold values. In addition, we show the news events which were published along with the dates they

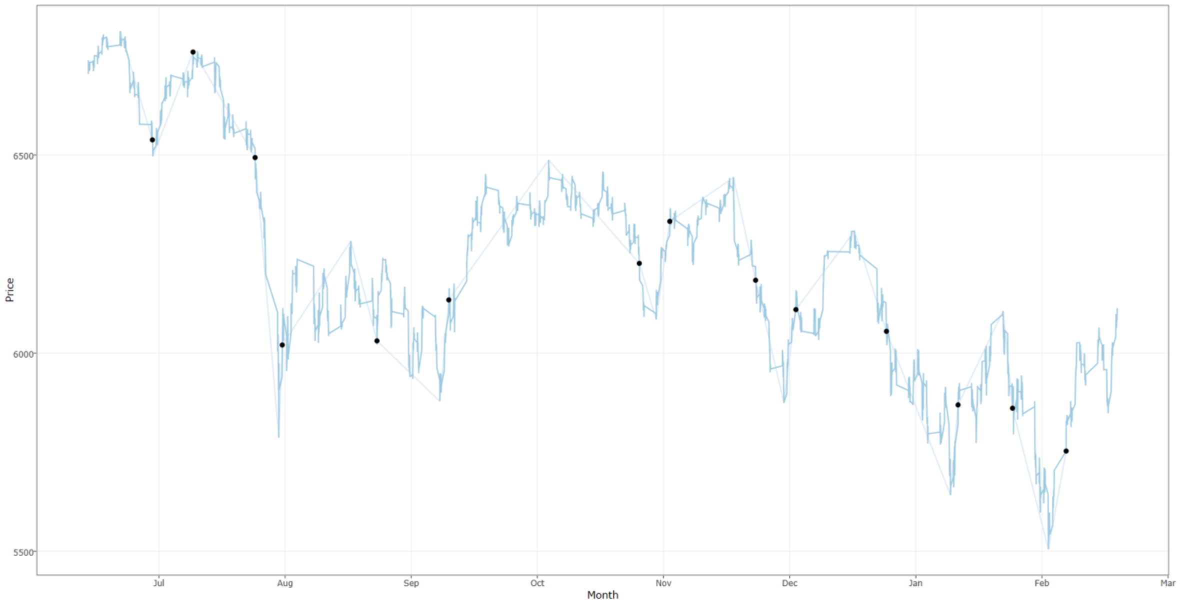


Figure 4.8: DC Detected Events from July 2015-Feb 2016 by the 4% Fixed Thresold

were published via the BBC News (www.bbc.co.uk) — regarding the FTSE 100 index.

4.3.4.1 Low Fixed Threshold Values May Detect Transit Events

A snapshot of the DC events detected from the period 30/9/2015 until 30/10/2015 is shown in Figure 4.11. A single upturn event was detected using our dynamic threshold; this was detected using the fixed thresholds as well.

Event 1 (an upturn event) was detected on 1st Oct using the 0.03 and the 0.04 fixed thresholds; also that day, an item of news was published regarding the FTSE 100 price decrease which occurred from noon until the end of the day's trading. The detected event was an upturn event (due to a price increase); however, the news item related to a FTSE 100 price decrease[2], refer to Figure 4.11 to see the price decrease which happened on the 1st Oct. Furthermore, using the 0.05 fixed threshold, an event was detected on the 2nd Oct. The 0.06 fixed threshold detected an upturn event on the 5th Oct, and later that day at 15:30 this upturn was detected using our dynamic threshold (which was at 0.071); at the release of the closing price that day, a FTSE 100 price increase was published in the news[3]. Using the 0.03 and 0.04 fixed thresholds an upturn event was detected even though the prices in general on the day in question were going down. Also, the 0.05 threshold detected an upturn event just before an attested, sharp FTSE 100 price increase took effect. Event 1 (an upturn event) was detected on the same day that the news event was published by only two of the experimental set-ups: the

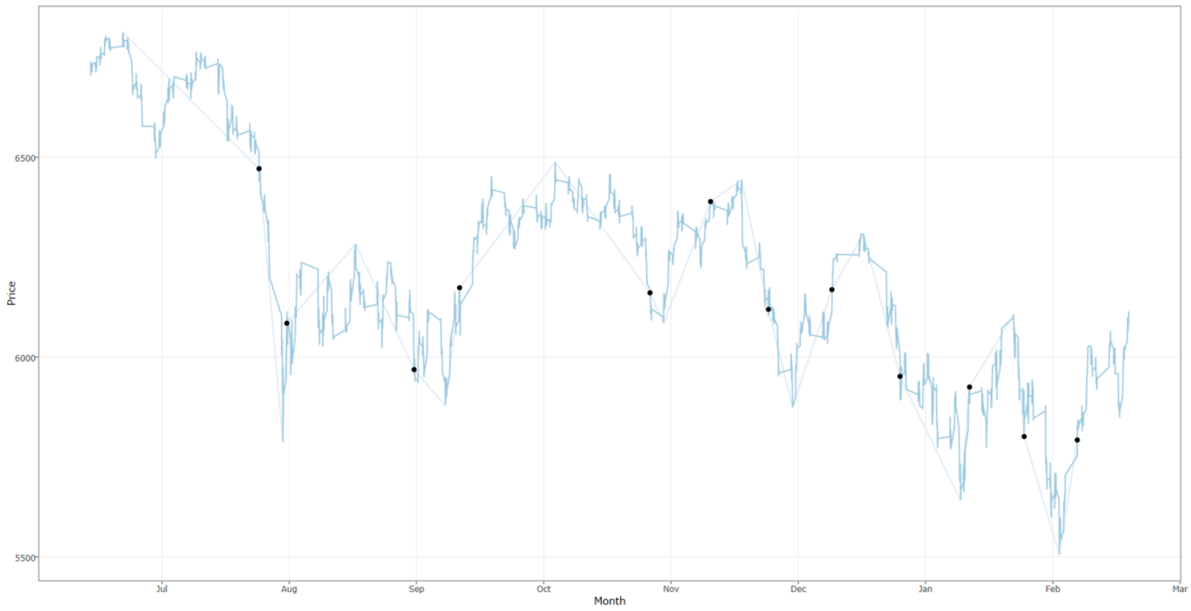


Figure 4.9: DC Detected Events from July 2015-Feb 2016 by the 5% Fixed Thresold

0.06 fixed threshold and our dynamic threshold. At the time of this event, our dynamic threshold was high as a result of the percentage change between the last lowest price ($P_l = 5878.71$), reached on 29th Sept at 8:09, and the current day's (5th Oct) opening price ($Cp_{open} = 6182.21$). Additionally the percentage change between previous day's (2nd Oct) closing price ($Pp_{close} = 6131.51$) and the current day's (5th Oct) opening price (the average of the first five minute-minute-by-minute prices was 6231.70) which had accumulated overnight was 0.016, hence the threshold on 5th Oct was 0.0712, and it was exceeded at 15:30 by a price of 6299.13.

4.3.4.2 Fixed Thresholds May Detect Events Before or After Prices Change

Another snapshot of the DC events detected from the period 1/12/2015 until 31/12/2015 is shown in Figure 4.12. Two events (a downturn and an upturn event) were detected using our dynamic threshold and also via most of the fixed thresholds (except for the 0.06 threshold which only detected a single DC event).

Event 2 (a downturn event) was detected using our dynamic threshold (which was set as 0.016) on 3rd Dec at 15:06 (the same day at closing price the first piece of news regarding the FTSE 100 price decrease was released[4]). Moreover an event was detected by the 0.03 fixed threshold on the 4th Dec, and was detected on the 8th Dec by the

[2] http://www.bbc.co.uk/news/business-34401041

[3] http://www.bbc.co.uk/news/business-34441540

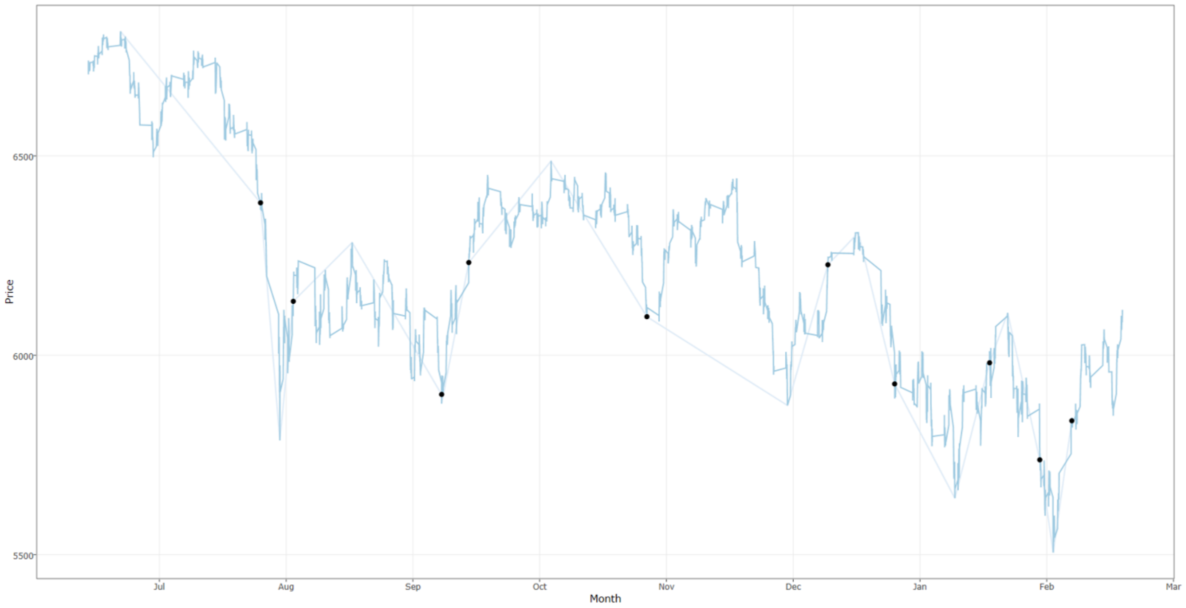


Figure 4.10: DC Detected Events from July 2015-Feb 2016 by the 6% Fixed Thresold

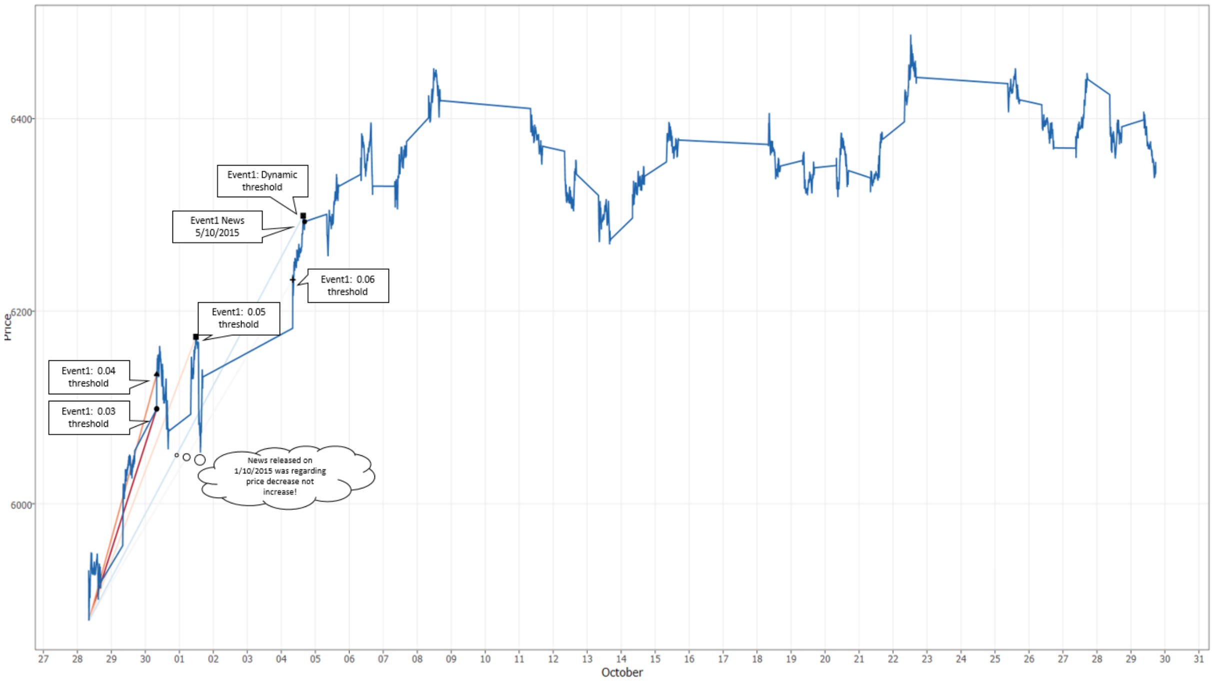


Figure 4.11: October 2015 DC Detected Events using our dynamic and fixed thresolds

0.04 fixed threshold, finally on 10th Dec an event was detected using the 0.05 fixed threshold (the second piece of news regarding the price decrease was released[5]). As, however, according to the use of the 0.06 threshold, the FTSE 100 stream was already on a downward run, no event was detected via this. The third item of news which was released in relation to Event 2 (a FTSE 100 price decrease) was published on the 11th Dec after the closing price was released - prices were again reducing sharply[6]. None of

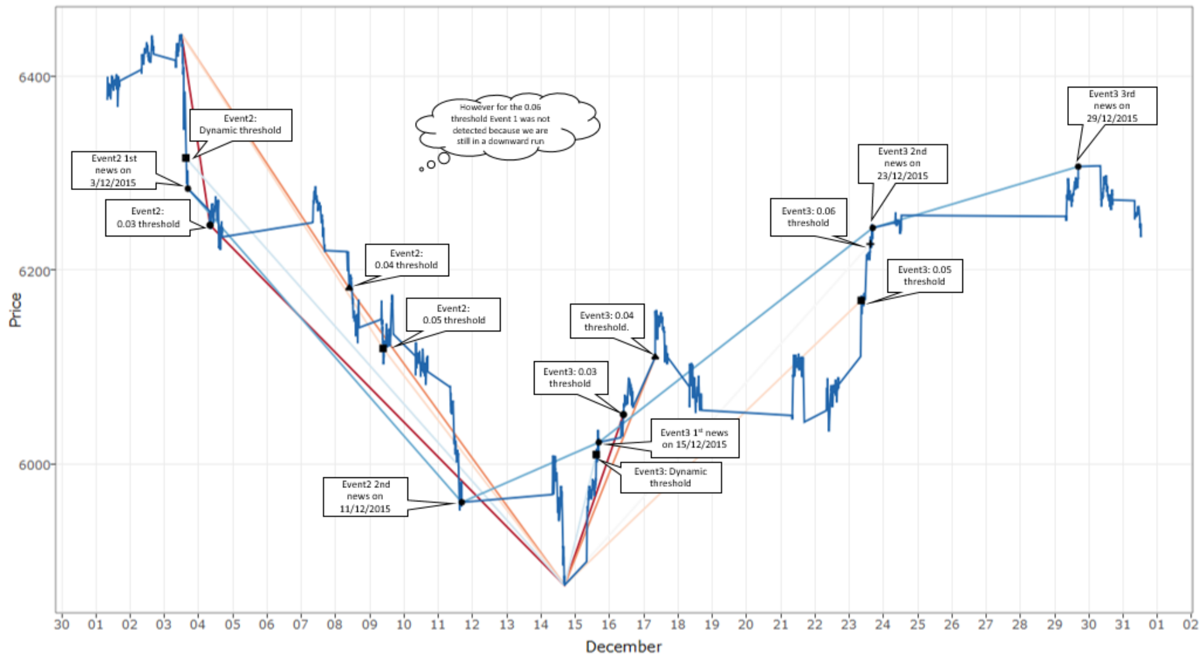


Figure 4.12: December 2015 DC Detected Events using our dynamic and fixed thresolds

the fixed thresholds we looked at were able to detect the downturn event in a timely manner and where a fixed threshold was used, the event was detected essentially between the release of the first and the second item of news.

Event 3 (an upturn event) was detected using the dynamic threshold (which was set as 0.023) on the 15th Dec at 14:42, and the first news reference to this event was released on the same day, after the closing price had been given[7]. Using the 0.03 fixed threshold, this upturn event was detected one day after the first piece of news had been published (the second piece of news regarding the price increase was released[8]), and using the 0.04 threshold it was detected two days after the first piece of news had been published. The 0.05 and 0.06 thresholds each detected an upturn event on the same day that the third related item of news was released (on the 23rd Dec), which was six days after the publication of the first item of news[9]. Furthermore, a fourth item of news was released concerning this FTSE 100 price increase on the 29th Dec[10]. The first item of news regarding the price increase was released on the same day that the dynamic

[4] http://www.bbc.co.uk/news/business-34992913
[5] http://www.bbc.co.uk/news/business-35059626
[6] http://www.bbc.co.uk/news/business-35070184

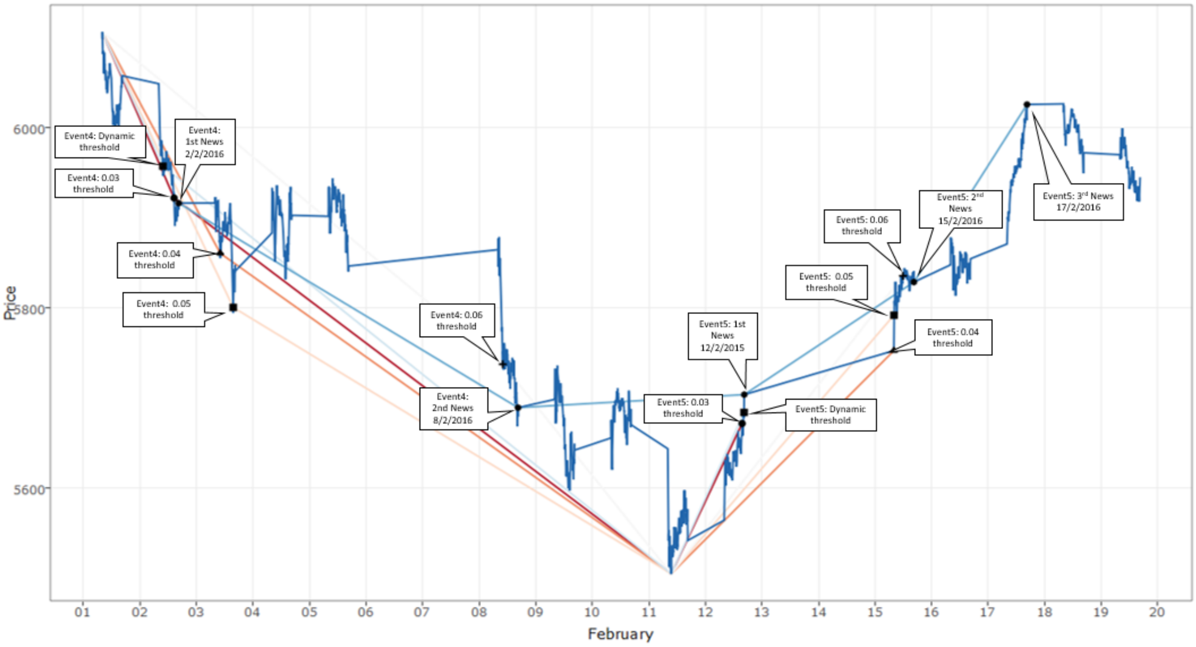


Figure 4.13: Feb 2016 DC Detected Events using our dynamic and fixed thresolds

threshold detected Event 3.

4.3.4.3 Dynamic Thresholds Can Effectively Spot Price Changes

A third and final snapshot of the DC detected events from the period 1/2/2016 till 19/2/2016 is shown in Figure 4.13. Once more, two events were detected using our dynamic threshold and using the fixed thresholds. Event 4 (a downturn event) was detected by the use of our dynamic threshold (which was set as 0.021) on 2nd February at 8:36, and later that day the event was also detected by the use of the 0.03 fixed threshold. The first related piece of news regarding this FTSE 100 price decrease[11] was published on the same day, once the closing price had been released. Moreover, on the 3rd Feb, this event 4 was detected by the use of the 0.04 and 0.05 fixed thresholds (one day after the first piece of news had been published). Lastly, on 8th Feb an event was detected by the use of the 0.06 fixed threshold, and on the same day and at the time of the release of the closing price the second news item regarding this FTSE 100 price decrease was published[12]. A third item of news regarding this price decrease was released on 9th Feb at closing price[13]. Event 4 (a downturn event) was detected on

[7] http://www.bbc.co.uk/news/business-35100085
[8] http://www.bbc.co.uk/news/business-35110494
[9] http://www.bbc.co.uk/news/business-35166667
[10] http://www.bbc.co.uk/news/business-35192384

the same day that the first news event was published only by the use of our dynamic threshold and the use of a 0.03 fixed threshold.

Event 5 (an upturn event) was detected by the use of the 0.03 fixed threshold on 12th Feb, and on the same date it was detected by the use of our dynamic threshold (which was set as 0.032). The first item of news regarding this FTSE 100 price increase was published on the same date once the closing price had been released[14]. Event 5 was detected by the use of the 0.04, 0.05, and 0.06 fixed thresholds on the 15th Feb, and the second item of news regarding this price increase was published on that same date once the closing price had been released[15]. In addition a third item of news regarding this increase was published on the 17th Feb once the closing price for this day had been released[16]. The Upturn event (Event 5) was detected with our dynamic threshold on the same day that the first news item was published regarding this FTSE 100 price increase.

All the DC events detected by the use of the dynamic threshold matched news headlines — which were published on the same day that the DC event was discovered at the closing price (22 News events in total in Appendix 4 reporting the first price increase/decrease). While with the fixed thresholds we looked at, only a few of the detected DC events matched news events, in most cases, the detected DC events identified either before or after a news event was published. Table 4.2 shows an analysis of the detected DC events for the different thresholds used (fixed and dynamic), specifically, it shows true and false detected events and sets the precision and recall values. In more details, all the identified DC events by the dynamic threshold were true events (matching news headlines reporting the first price increase/decrease), precision score was 1 and the recall score was 0.9 (only 2 events were missed[17][18]). The 0.03 fixed threshold resulted in the detection of even more events (22 events) than the dynamic threshold (20 events), but 10 out of those 22 DC events were detected either significantly before or significantly after the news event was published, precision and recall score was 54%. The use of the 0.04 fixed threshold, on the other hand, resulted in more DC events being false than "true" events (only 5 out of the 14 detected DC events were true events, precision score was 35%). Thus, the use of this threshold resulted in 9 out of the 14 DC events

[11] http://www.bbc.co.uk/news/business-35470062
[12] http://www.bbc.co.uk/news/business-35520828
[13] http://www.bbc.co.uk/news/business-35530328
[14] http://www.bbc.co.uk/news/business-35558163
[15] http://www.bbc.co.uk/news/business-35577308
[16] http://www.bbc.co.uk/news/business-35593629

Threshold Type	**Detected DC Events**	**True Events**	**False Events**	**Precision**	**Recall**
Dynamic	20	20	0	20/20=1	20/22=0.9
Fixed (0.03)	22	12	10	12/22=0.54	12/22=0.54
Fixed (0.04)	14	5	9	5/14=0.35	5/22=0.22
Fixed (0.05)	12	5	7	5/12=0.42	5/22=0.22
Fixed (0.06)	10	3	7	3/10=0.3	3/22=0.14

Table 4.2: Dynamic and Fixed Thresholds Detected DC Events Analysis

detected being unmatched by news reports. In addition, only 2 out of those 9 events were late detected, in that they matched news events but not items which corresponded to the first release of news regarding a price increase or decrease. With the use of the 0.05 fixed threshold, more than half of the known DC events were false — 7 out of the 12 attested DC events were incorrect (only 3 out of those 7 events were late detected, in that they matched news items but not items which corresponded to the first news published regarding a particular price increase or decrease). Both fixed thresholds 0.04 and 0.05 missed the same number of news events, recall score was 22%. Finally, the use of the 0.06 fixed threshold missed 70% of the detected DC events (precision score was 30%), 7 out of the 10 detected DC events were false events (6 out of those 7 events were late detected, in that they matched news events but not an item which represented the first published news regarding a particular price increase or decrease).

We have shown that in most cases the use of fixed thresholds fail to detect DC events which directly correspond to published news events. Fixed thresholds often result in the detection of a DC event either before or after its occurrence, because only the fixed threshold criteria is considered. On the contrary, the use of our dynamic threshold was able to detect DC events on the same day that news events were released. The dynamic threshold is defined daily and is flexible, it sometimes takes a low value and at other times it takes a large value. Furthermore, using dynamic threshold values events were generally detected on the same day as they were when small fixed threshold values were used (or even before), and at other times events were detected on the same day that the

[17] http://www.bbc.co.uk/news/business-34785068
[18] http://www.bbc.co.uk/news/business-35966896

use of big fixed threshold values caused them to be detected (or even after). However, the main determinants of the daily dynamic threshold values are the previous day's and the overnight price changes.

4.4 Summary

This chapter presented the dynamic threshold DC definition algorithm, which can replace the fixed a-priori given threshold, and is suitable for markets that are operating over specific opening and closing times. The defined dynamic threshold is flexible and is more suitable for dynamic and volatile environments such as financial time-series data streams. A fixed threshold has always been the case in the DC [6, 36–38, 42–50, 134].

In more detail, we have shown the applicability of the DC approach on price time-series data streams, especially, when employed with daily dynamic defined thresholds. An event is detected by DC when a price change between two relatively extreme values (the current price *p(t)* and either (p_h/p_l), depending on the type of DC event) exceeds the given threshold value. Our dynamic threshold is defined on a daily basis once the current day's opening price has been received, and is used on that day for detecting DC events. The dynamic threshold is basically defined on the basis of the previous day's price transitions and the current day's first 5 minute-by-minute prices. Depending on the previous day and/or overnight price change the suitable dynamic threshold definition method is used. A decision tree using a labelled training dataset is built to set the values for the previous day and overnight extreme price changes.

Experiments were conducted on a minute-by-minute price time-series stream to detect the occurrence of DC events using different fixed thresholds and also our daily dynamically defined one. The events which were detected were evaluated against what was published on the same day by a major news outlet regarding the particular stock or share in question. If the detected event was found on the same day that a relevant news headline was published, then it was said a true event, otherwise it was said to be a spurious or false event.

The results show that the use of a dynamic threshold leads to more accurately identified events than does the use of various fixed threshold values. Our dynamic threshold, used with the DC approach, facilitated the detection of DC events with different magnitudes, which is not applicable to the use of fixed thresholds, and so this is inherently an improvement. This work extends and further enhances the DC approach and is new with regards taking into account additional pertinent information to set up a dynamic threshold instead of a fixed one which facilitates more accurate event identification in a continuously changing market environment.

Chapter 5

DC Based Trading Strategy with Dynamic Thresholds

5.1 Introduction

Identifying significant price movements is crucial in financial markets as these price changes represent investment opportunities. Traditional methods for observing price fluctuations in financial markets are based on changes as related to physical time; such measures depend on a fixed time period of the trading session (e.g., daily closing price), however this fails to capture the full nature of the price movement activity, and hence important fluctuations may be missed which could present an opportunity to trade [34, 131]. This is so despite financial data (i.e. transactions) being recorded at higher level of detail —– described as HFD.

Thus, it is very difficult and challenging to observe and track price movements in high frequency time-series data streams consisting of intraday prices (HFD) when using a physical time scale. This is because such a time-scale depends on a fixed time period (a chosen time unit). A more flexible approach to time intervals should be taken. Furthermore, it is difficult to review the entire market's history, when streams of time-series data are arriving in almost real-time and at high velocity. Such analysis is made even more difficult by the fact that such data streams are essentially of an unbounded size.

As the experimental work has shown in the previous chapter, when using fixed threshold values with the DC approach, some events may be missed, or inaccurately identified (perhaps at the wrong time). This in essence means that the use of fixed thresholds could result in the missing of trading opportunities, and would therefore have an impact on the profits generated by a strategy based on them.

Hence in this chapter, we want to further evaluate the daily dynamically defined threshold value (presented in chapter 4). Using the dynamic threshold with the DC approach has led to the detection of events in a timely manner; this means that the DC events were detected on the same day that a news headline relating to them was published. The next natural step would then be to consider if the DC dynamic threshold algorithm can be used as part of a trading strategy. Thus, when a DC event is detected (a price change is spotted), then a trading opportunity (buying or selling) is opened up.

A successful trade is one which takes place at the right time, at an advantageous final price. The determination of the most profitable threshold value has always been an issue in relation to the DC approach when applied with trading strategies. Different studies have considered a number of different ways for specifying the best threshold value: for example, as in [37, 38, 52]. The authors of [38] proposed a trading strategy which combines the DC approach with a Genetic Programming (GP) algorithm. More specifically, the GP algorithm combines the use of a number of different threshold values to present a trading strategy. They found that a combination of multiple thresholds helps to focus buy or sell strategies at more favourable times.

Another effort was a trading strategy named the Directional-Change Trading (DCT1) which was proposed in [37]. The main idea was based on a learning process from historical data; where the most profitable trading strategy (Trend Following or Contrarian Trading) and threshold value are found before actually trading in the market.

Trend Following (TF) [182] is a rule-based trading technique where buying and selling orders are made according to the market trend; it neither forecasts nor predicts the market movement. The TF requires that the trading rules are set up prior to trading; once a trend is identified, the trading rules are activated until the next trend is identified. Basically a TF trader makes a buy order when the prices are increasing, and a sell order when the prices are falling. The Contrary Trading (CT) is a rule-based trading technique; however, it acts in the opposite way with regard to the trend direction. More specifically, a CT trader makes a sell order when the prices are increasing, and a buy order when the prices are falling.

More recent work in [36] presented a contrarian trading strategy based on the DC concept named the Backlash Agent (BA). Instead of finding the most profitable threshold value, they introduced two new threshold values ($Down-ind$ and $Up-ind$). In a downward run, if an Overshoot Value (OSV) is less than a threshold called $Down-ind$, then a buy order is made. The order is closed when the DC of the next upturn event is confirmed. The opposite happens when there is an upward run; if the OSV value exceeds a given threshold ($Up-ind$) then a sell order is made; while the order is closed when a new DC event (downturn event) is encountered. Equation 5.1 describes how the OSV is defined, where $theta$ is the static threshold value, and P_{DCC*} is the highest/lowest price during an upward/downward OS event.

$$OSV = \frac{(\frac{(P_t - P_{DCC*})}{P_{DCC*}})}{theta} \tag{5.1}$$

As it was difficult to find the most profitable value for the thresholds $Down-ind$ and $Up-ind$, the authors tried to automatically set it by applying 100 $Down-ind/Up-ind$ values on a training dataset prior to trading. The value whose adoption resulted in the highest profits was chosen as the most suitable value for the threshold $Down-ind/Up-ind$ and was then used when actually trading in the period under consideration.

In this chapter, we present a trading strategy that is based on the DC approach along with using a daily dynamic defined threshold value (based on what has happened the previous day) to replace the fixed given one. We consider a DC event as an opportunity for trading – as a price change is identified by such an event. Thus, once a DC event is detected and a price increase/decrease is noticed, then a trading action is activated. The trading decision, either buying or selling, depends on the previous day's price transitions.

The rest of this chapter is organized as follows. In the next section, (Section 5.2) we show how the DC approach and the previous day's price change (short term history) are used to define the trading strategy trading rules. Section 5.3 presents the experimental work, where we describe, discuss and analyse both the experiments conducted and the findings. The chapter ends with a summary and conclusions in Section 5.4.

5.2 The Dynamic Threshold-Trading Strategy Rules

We introduce a trading strategy that is based on the DC approach, our dynamically defined threshold, and a consideration of what happened the previous day (the short term history). We wanted to take advantage of what has occurred the previous day before trading; thus we consider the previous day's price change as an indication of the type of trading to be adopted (either CT or TF). For example, when we are experiencing an upward run and the previous day's price change (price increase) was large, then it is a selling opportunity rather than a buying opportunity, since the price has already increased significantly. Hence, we consider a large price change as a contrarian trading strategy opportunity.

If the price change that occurred the previous day or overnight was extreme, then we follow a CT strategy (as prices are becoming too high or too low depending on the trend, and we do not want to miss that opportunity), and if not, then we follow a TF trading strategy.

Whenever an OS event starts during either a downward run or an upward run, we keep track of every price change within that trend (every p_h/p_l price change). If the change in either p_h/p_l has been tracking an extreme event during the previous day, then the change of price is assumed to be large, and so we adopt a CT trading strategy. In particular, if the price change was in p_l (price decrease), then it is a buying opportunity as prices are becoming excessively low, otherwise if it is in p_h (representing a price increase) then it is a selling opportunity as price are becoming excessively high. In contrast, if a price change was only identified in p_h/p_l with nothing special identified during the previous day, then we follow a TF trading strategy. Finally, if the price change was in p_l (a price decrease), then it is a selling opportunity, otherwise if it is in p_h (a price increase), then it is a buying opportunity. Consider for instance in a downward run the previous day closing price was 5954 and the current day's opening price is 5885 (overnight price change was 1.14%). So, once the p_l value is changed (p_l is updated whenever the current price $p(t)$ is lower than the last p_l value), a buying trading action is triggered rather than selling, this is because the overnight price change was big and the prices are decreasing sharply.

In general, if all the money and shares available are used when trading, only a single trading action can take place for every trend (no sequential trading actions, i.e., 2 buy or 2 sell trades), since all the money or shares have already been used. However, a positive thing about our trading strategy is the ability to continue trading even if all the money or shares have already been used, as the next trading action might be the opposite way. Hence, if we have already traded and then suddenly the prices sharply decrease or increase, we may still be able to take an action and benefit from that price change. This is in contrast to other trading strategies which use the whole amount of money or shares, and cannot take further advantage of a big price change opportunity, as the trading rule that they are using has already been satisfied, and the money or shares available have already been consumed. In summary, when using our trading strategy, more trading actions can continue to be executed, even if the same trend is continuing and the previous action has used all the money or shares available.

The previous day's price change is said to be extreme if the percentage change in relation to what happened the previous day (between the previous day opening and closing price, *previous_PC*) or happened overnight (between the previous day closing price and current day opening price, *overnight_PC*) was greater than *previous_v*/*overnight_v*. The *previous_v* is a value that determines whether the price change encountered the previous day is significant or not, the *overnight_v*, on the other hand, determines the significant overnight price change. They are set by building a decision tree on a training dataset, more details regarding the training phase are shown later in the experimental work.

The DT-TS trading rules are as follows:

In a Downward run and during an OS event:

Rule 1: If (*overnight_PC* > *overnight_v*) or (*previous_PC* > *previous_v*) then generate a buy order.

If the percentage change between the previous day closing price and current day opening price is greater than *overnight_v* or the percentage change between the previous day opening and closing price is greater than *previous_v*, then generate a buy order.

Rule 2: If ($p(t) < p_l$), then generate a sell order.

If the current price is less than the lowest reached price, then generate a sell order.

In an Upward run and during an OS event:

Rule 1: If (*overnight_PC* > *previous_v*) or (*previous_PC* > *previous_v*) then generate a sell order.

If the percentage change between the previous day closing price and current day opening price is greater than *overnight_v* or the percentage change between the previous day opening and closing price is greater than *previous_v*, then generate a sell order.

Rule 2: If ($p(t) > p_h$), then generate a buy order.

If the current price is greater than the highest reached price, then generate a buy order.

Algorithm 4 shows the DT-TS trading rules, which are triggered with every price change in p_h/p_l during the course of the DC approach. Lines 2-11 relate to being in a downward OS, while lines 13-22 relate to being in a an upward OS. In summary, through the DT-TS, a trading action is opened up, if there is a price change during the OS period, if not then the next incoming price is examined. The type of trading to be applied (either CT or TF) depends on the previous day's circumstances. A trading rule is satisfied, if there is a sufficient number of monetary units or shares. The trading rule is closed when the confirmation point of the next DC event is encountered.

Algorithm 4: The DT-TS Trading Rules

```
   Input  : TSstream: Time-series stream
 1 for each incoming price $p_{(t)}$ in TSstream do
 2     if (Downward OS) then
 3         if $p_{(t)} < p_l$ then
 4             if (overnight_PC > overnight_v  or previous_PC > previous − v)
                 then
 5                 trading = CT
 6             else
 7                 trading = TF
 8             end
 9         else
10             examine Next $p_{(t)}$
11         end
12     else
13         if (Upward OS) then
14             if $p_{(t)} > p_h$ then
15                 if (overnight_PC > overnight_v  or previous_PC > previous_v)
                     then
16                     trading = CT
17                 else
18                     trading = TF
19                 end
20             else
21                 examine Next $p_{(t)}$
22             end
23         end
24     end
25 end
```

5.3 Experimental Work

In this section, we want to demonstrate the applicability of our trading strategy (the DT-TS) when used in conjunction with the DC approach on a minute-by-minute prices stream. We want to show that using dynamic thresholds when trading is more profitable than using fixed thresholds.

We have collected the FTSE 100 minute-by-minute prices from Reuters Thomson One [177] from July 2015 till end of October 2016. FTSE 100 is the index of the 100 largest companies in the LSE. The LSE operates on week days from 8 am until 4:30 pm. We obtained the opening, high, low, and closing prices on a minute-by-minute basis. As there are generally no large differences between these four prices (opening, high, low, and closing), we take their average.

In this research and for simplicity, the same money management approach in [36, 37, 51, 183] is adopted, which is the following: whenever a buy order is activated we use the entire amount of money available (use 100% of cash available when buying), and with sell orders we sell all the available shares (use 100% of shares available when selling). In the applied experiments, we initially start trading with 100k monetary units and zero shares. Additionally at this stage transaction costs (if any) are not taken into consideration. We name our trading strategy, when combined with our dynamic threshold, the DT-TS, and when it is combined with a fixed threshold, the Fixed Threshold Trading Strategy (FT-TS).

Next, we employ the DT-TS trading rules on a training phase and a testing phase. Using the training dataset, we set the values of when something extreme has been noticed the previous day and/or overnight. By the testing set, on the other hand, we test these values with our dynamic defined threshold (the DT-TS) and different fixed threshold values (the FT-TS) along with applying that time-series stream to other trading strategies (CT, TF, and Backlash Agent (BA) [36]) in order to further explore their behaviour. We conclude this section by evaluating the performance of the DT-TS against different fixed thresholds and other trading strategies.

5.3.1 Training phase

In order to set the $previous\_v$ and $overnight\_v$ values, we trained our trading model on a training dataset extracted from the FTSE 100 stream from July 2015 until end

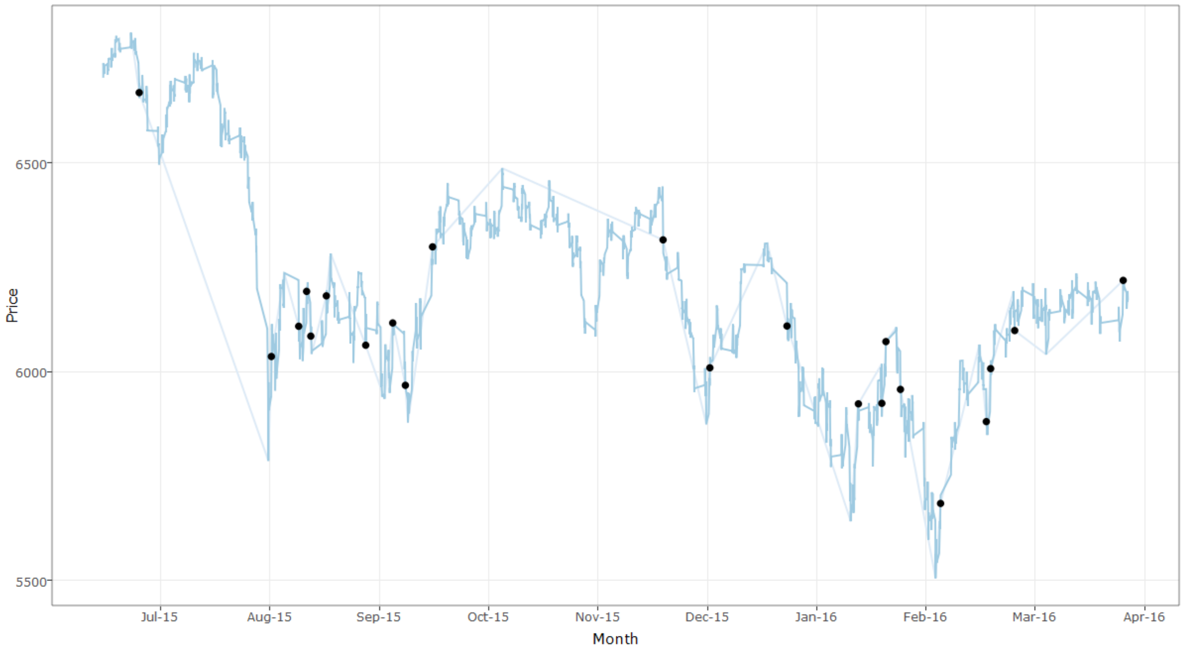
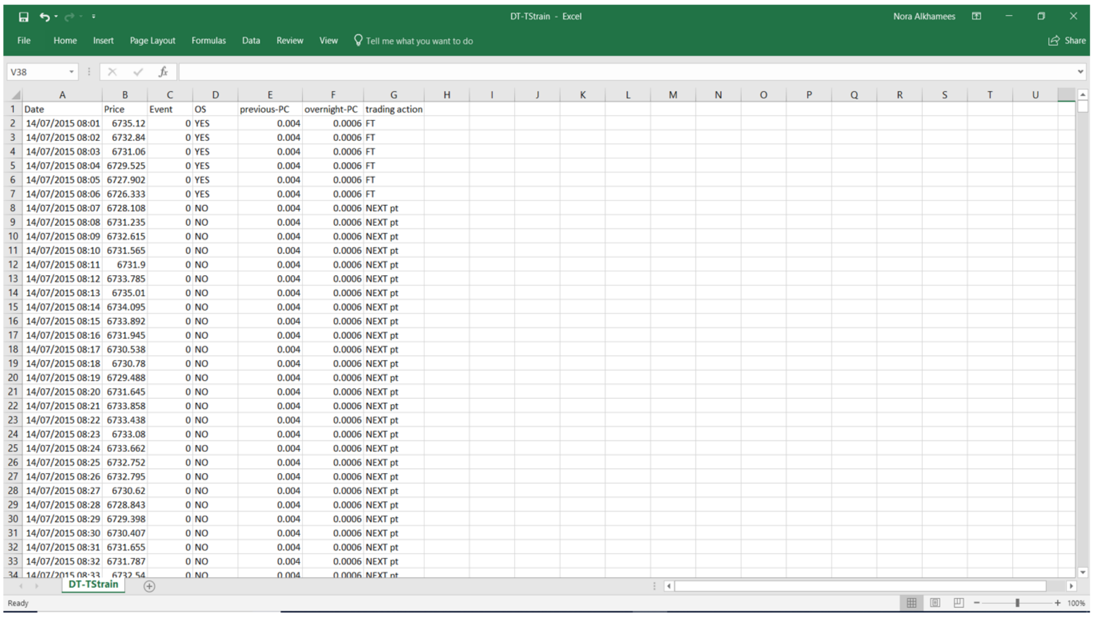


Figure 5.1: Detected DC Events from July 2015- March 2016

of March 2016. Figure 5.1 illustrates the training period and the events which were detected using the DC approach along with our daily dynamic defined threshold. We were able to detect 22 DC events using thresholds values ranging from (0.014-0.071). In the figure, each DC event is represented by a black dot.

Thus, the *previous_v* and *overnight_v* values were determined as a result of building a J84 Decision Tree to the training dataset. J48 is the Java implementation of C4.5 [181] in WEKA [180]. WEKA is an open source data mining toolbox written in JAVA and consisting of a collection of machine learning algorithms. We used more than 36 weeks of FTSE 100 minute-by-minute prices as our training dataset (from July 2015 until end of March 2016), refer to Figure 5.2 for a snapshot of the training dataset. Each minute of the training dataset has the following information: date and time, share price, percentage change between previous day's opening and closing prices (*previous_PC*), percentage change between previous day's closing and the current day's opening price (*overnight_PC*), whether this datum is an event confirmation point or not, whether this datum is an overshoot period and p_h/p_l has changed, and finally the trading decision. The trading decision can be either "next $p(t)$" (which means no trading takes place and wait for the next minute price), or TF, or CT. Figure 5.3 shows the decision tree which was built for extracting our trading strategy trading rules from the training dataset.

From the built decision tree in figure 5.3, we can conclude that the *overnight_v*



	A	B	C	D	E	F	G
1	Date	Price	Event	OS	previous-PC	overnight-PC	trading action
2	14/07/2015 08:01	6735.12	0	YES	0.004	0.0006	FT
3	14/07/2015 08:02	6732.84	0	YES	0.004	0.0006	FT
4	14/07/2015 08:03	6731.06	0	YES	0.004	0.0006	FT
5	14/07/2015 08:04	6729.525	0	YES	0.004	0.0006	FT
6	14/07/2015 08:05	6727.902	0	YES	0.004	0.0006	FT
7	14/07/2015 08:06	6726.333	0	YES	0.004	0.0006	FT
8	14/07/2015 08:07	6728.108	0	NO	0.004	0.0006	NEXT pt
9	14/07/2015 08:08	6731.235	0	NO	0.004	0.0006	NEXT pt
10	14/07/2015 08:09	6732.615	0	NO	0.004	0.0006	NEXT pt
11	14/07/2015 08:10	6731.565	0	NO	0.004	0.0006	NEXT pt
12	14/07/2015 08:11	6731.9	0	NO	0.004	0.0006	NEXT pt
13	14/07/2015 08:12	6733.785	0	NO	0.004	0.0006	NEXT pt
14	14/07/2015 08:13	6735.01	0	NO	0.004	0.0006	NEXT pt
15	14/07/2015 08:14	6734.095	0	NO	0.004	0.0006	NEXT pt
16	14/07/2015 08:15	6733.892	0	NO	0.004	0.0006	NEXT pt
17	14/07/2015 08:16	6731.945	0	NO	0.004	0.0006	NEXT pt
18	14/07/2015 08:17	6730.538	0	NO	0.004	0.0006	NEXT pt
19	14/07/2015 08:18	6730.78	0	NO	0.004	0.0006	NEXT pt
20	14/07/2015 08:19	6729.488	0	NO	0.004	0.0006	NEXT pt
21	14/07/2015 08:20	6731.645	0	NO	0.004	0.0006	NEXT pt
22	14/07/2015 08:21	6733.858	0	NO	0.004	0.0006	NEXT pt
23	14/07/2015 08:22	6733.438	0	NO	0.004	0.0006	NEXT pt
24	14/07/2015 08:23	6733.08	0	NO	0.004	0.0006	NEXT pt
25	14/07/2015 08:24	6733.662	0	NO	0.004	0.0006	NEXT pt
26	14/07/2015 08:25	6732.752	0	NO	0.004	0.0006	NEXT pt
27	14/07/2015 08:26	6732.795	0	NO	0.004	0.0006	NEXT pt
28	14/07/2015 08:27	6730.62	0	NO	0.004	0.0006	NEXT pt
29	14/07/2015 08:28	6728.843	0	NO	0.004	0.0006	NEXT pt
30	14/07/2015 08:29	6729.398	0	NO	0.004	0.0006	NEXT pt
31	14/07/2015 08:30	6730.407	0	NO	0.004	0.0006	NEXT pt
32	14/07/2015 08:31	6731.655	0	NO	0.004	0.0006	NEXT pt
33	14/07/2015 08:32	6731.787	0	NO	0.004	0.0006	NEXT pt
34	14/07/2015 08:33	6732.54	0	NO	0.004	0.0006	NEXT pt

Figure 5.2: The DT-TS Training Dataset for Building the Trading Strategy Decision Tree

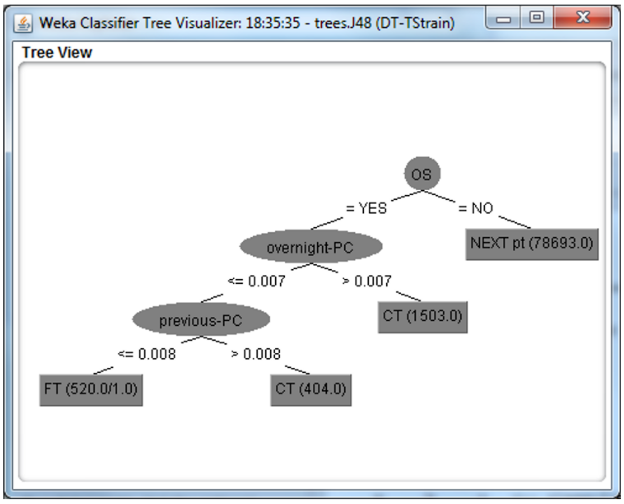


Figure 5.3: Trading Strategy Decision Tree

was set to 0.7% and the *previous_v* was set to 0.8%. Hence, the overnight and/or the previous day's price change is said to be extreme, if the *overnight_PC* was greater than 0.7%, or if the *previous_PC* was greater than 0.8%.

In addition, we ran the found trading strategy rules using different fixed thresholds (FT-TS). We tried a small fixed threshold (3%), some medium sized thresholds (4% and 5%), and a larger fixed threshold (6%), Figures 4.7, 4.8, 4.9, and 4.10 in the previous chapter (Chapter 4) show the DC events which were detected from July 2015-Feb 2016 using the static thresholds 3%, 4%, 5%, and 6% respectively. In addition, we tried different trading strategies (CT, TF, and BA [36]) to further compare the profitability and the number of trading actions taking place — between our proposed trading strategy and the other strategies. The profitability is measured at the end of a trading period by removing the gross loss ($Gloss$) of all losing trades from the gross profit ($Gprofit$) of all winning trades, see Equation 5.2. In the last section of the Experimental Work, we use other performance evaluation metrics to further measure the DT-TS performance.

$$TNprofit = Gprofit - Gloss \tag{5.2}$$

Table 5.1 reveals the profitability of each threshold, along with the number of detected DC events, the applied trading strategy, and lastly the number of trading actions taking place. It is clear that the dynamic threshold when used with our trading strategy (DT-TS) outperforms all the fixed thresholds which were examined and all other investigated trading strategies (FT-TS, CT, TF, and BA) —- with a profitability percentage of 65%. In general, using the DC approach, low threshold values lead to the detection of more events than larger thresholds, as the former are more easily exceeded than the latter. The more events which are detected, the more the trading actions which could take place, and so the higher the profits which could be reached.

Furthermore, Table 5.1 shows that the same number of detected DC events were yielded by the use of the dynamic threshold and by the use of the 0.03 fixed threshold (22 events), and when using our trading approach the number of trading actions facilitated by the use of the 0.03 threshold (FT-TS) were greater than the number facilitated by our dynamic threshold (DT-TS); 42 trading actions were facilitated by the DT-TS in contrast to 45 trading actions facilitated by the 0.03 FT-TS. However the dynamic threshold used with our trading strategy DT-TS was more profitable (65%) than the 0.03 fixed threshold FT-TS (62%) even though, using our dynamic threshold, the number of trading actions which took place were less. This was due to events being detected in a more timely fashion with our dynamic threshold, and so trading actions took place at better points in time and with better prices. Other fixed thresholds (0.04, 0.05, and 0.06) used with FT-TS, as shown in table 5.1 were less profitable because the number of detected events were less. However, the FT-TS still performs well compared to the other investigated trading strategies (CT, TF, and BA) when using fixed thresholds. The 0.03 threshold had the highest profitability percentage of 62% using the FT-TS, while the 0.04 threshold had the highest profitability percentage of 52% (with 14 detected DC events) using the FT-TS. The highest profitability percentage yielded

Threshold	No. events	Trading	No. Tradings	Profits
Dynamic	22	DT-TS	42	65%
		CT	23	34%
		TF	23	33%
		BA	23	38%
0.03 (Fixed)	22	FT-TS	45	62%
		CT	22	13%
		TF	23	42%
		BA	23	38%
0.04 (Fixed)	14	FT-TS	38	52%
		CT	14	12%
		TF	15	21%
		BA	12	44%
0.05 (Fixed)	12	FT-TS	30	41%
		CT	12	17%
		TF	13	7%
		BA	12	44%
0.06 (Fixed)	10	FT-TS	23	38%
		CT	10	21%
		TF	11	-2%
		BA	9	40%

Table 5.1: Training Period Analysis Using the Minute-by-Minute Prices Stream

by the use of the 0.05 threshold was 44% (with 12 detected DC events) using the BA (the FT-TS profits was 41%), and the highest profitability percentage yielded by the use of the 0.06 threshold was 40% (with 10 detected DC events) using the BA trading strategies (the FT-TS profits was 38%).

Table 5.1 demonstrates that the CT trading strategy performs better with high threshold values rather than with lower ones. While the TF trading strategy, on the other hand, performs better with low threshold values rather than with higher ones. This is because with the TF it is preferable to detect an event as early as possible (which is possible with small threshold values), as the trend is expected to continue in the same direction, thus the trading action taken tries to prevent losses when the prices are decreasing and increases the profits when the prices are rising. Using the CT in contrast, it is preferable to detect the event when the peak has almost been reached (which is possible using big threshold values), which means a sell action when prices are rising, and a buy action when prices are dropping. CT performed best with the 0.06 threshold and TF performed best with the 0.03 threshold value.

The BA trading strategy [36] was applied to our training set to find the most profitable down-ind and up-ind threshold values, which then were used to trade on the testing dataset.

Using our dynamic threshold it was not possible to find the most profitable value (as a single value) of down-ind and up-ind from the training dataset to be used when trading on the testing dataset. This is because we have a different threshold value every single day, which results in there being a different value for the down-ind and up-ind every single day as well. In order to deal with this issue, we first calculate the mean of the dynamic threshold values, which were in the range 0.014-0.071 (the dynamic thresholds' mean was 0.0281), then for that mean, we found the value of down-ind and up-ind. Accordingly, the values of down-ind and up-ind for the mean of the daily calculated threshold values were -0.16 and 0.28 respectively.

Using the 0.03 fixed threshold, the down-ind and up-ind values were -0.11 and 0.18 respectively. While using the 0.04 fixed threshold, down-ind and up-ind were -0.35 and 0.43 respectively. The 0.05 fixed threshold down-ind and up-ind values were -0.17 and 0.14 respectively. Lastly the 0.06 fixed threshold down-ind and up-ind values were -0.54 and 0.17 respectively.

5.3.2 Testing Phase

In order to test our trading strategy, we applied the trading rules which were discovered, alongside our dynamic threshold (the DT-TS) on a testing dataset of the FTSE 100 minute-by-minute prices for the period from April 2016 until the end of October 2016, and we also, once more, tested with four different fixed thresholds FT-TS (0.03, 0.04, 0.05, and 0.06) as well. Furthermore we evaluated our trading strategy's profitability against the CT, TF, and BA trading strategies. We initially started the trading strategy with 100k monetary units and zero shares. Figure 5.4 illustrates the DC events detected during the testing period using our dynamic threshold; DC events are shown as small black dots.

It is clear that the number of detected DC events across the testing period is small; this is because the market prices were almost going in the same direction (prices were gradually rising), refer to figure 5.4. However half of the detected events (8 out of 16

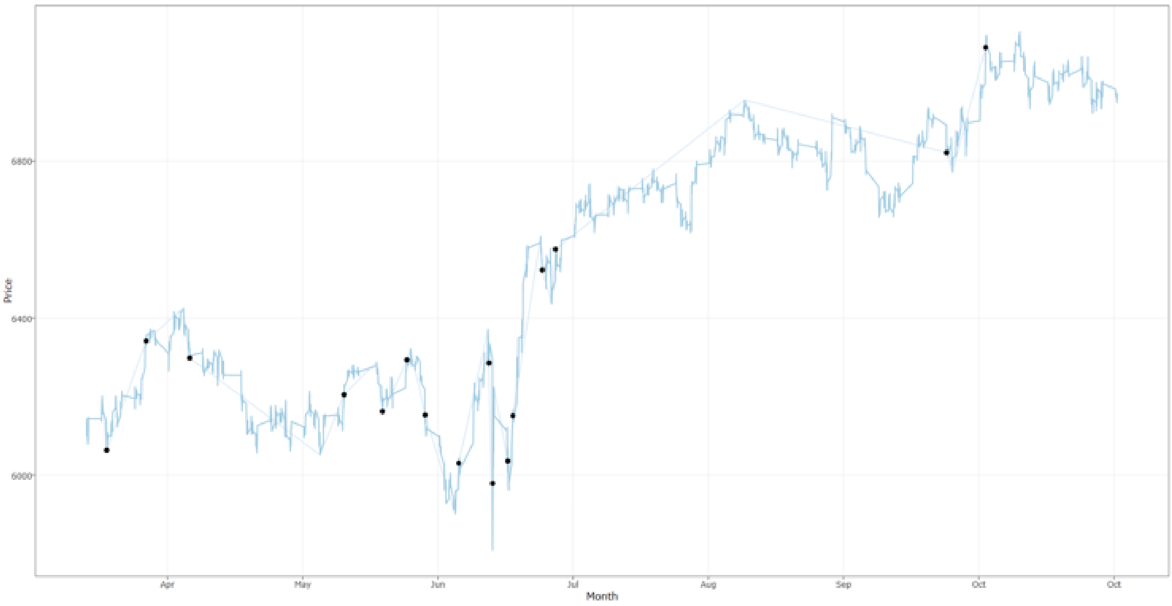


Figure 5.4: Detected DC Events from April-Oct 2016 (Testing Period)

events) took place in June 2016 ——the month of the EU Referendum. This explains why the profitability percentage is generally not high, as Brexit fears were growing. Figure 5.5 shows a bar chart comparing the profitability in the training set with the profitability in the testing sets and how our trading strategy, when used with a dynamic threshold (DT-TS), outperformed the other trading strategies which were investigated. In addition, with respect to the testing dataset, our trading strategy with our dynamic threshold (DT-TS) once more outperformed the use of the fixed thresholds which were trialled and the other investigated trading strategies (FT-TS, CT, TF, and BA) with 47% profits, refer to Table 5.2.

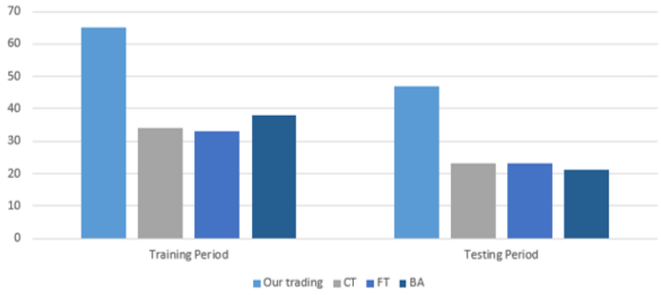


Figure 5.5: Profitability Comparision between Training and Testing sets

Figure 5.6 provides a bar chart showing the profitability in the testing period for all investigated thresholds and trading strategies. It is noticeable that our trading strategy with our dynamic threshold (DT-TS) gained the highest profits among all the thresh-

Threshold	No. events	Trading	No. Tradings	Profits
Dynamic	16	DT-TS	33	47%
		CT	16	23%
		TF	16	23%
		BA	16	21%
0.03 (Fixed)	10	FT-TS	16	35%
		CT	10	18%
		TF	10	20%
		BA	10	20%
0.04 (Fixed)	10	FT-TS	15	14%
		CT	10	29%
		TF	10	8%
		BA	10	20%
0.05 (Fixed)	4	FT-TS	9	17%
		CT	4	13%
		TF	4	10%
		BA	4	12%
0.06 (Fixed)	4	FT-TS	8	16%
		CT	4	17%
		TF	4	10%
		BA	0	0%

Table 5.2: Testing Set Analysis Using the Minute-by-Minute Prices Stream

olds and all the trading strategies. Our trading strategy with fixed thresholds (FT-TS) performed also well and was more profitable than most other trading strategies. This was so except in relation to the CT trading strategy when used with a 0.04 fixed threshold. The latter set-up was more profitable with 29% profits (an outlier). Using the 0.04 fixed threshold with our trading strategy (FT-TS), 3 out of the 10 detected DC events triggered the TF trading actions, and 7 DC events on the other hand triggered the CT trading action. Three sequential events which triggered CT trading actions (all influenced by the EU referendum and Brexit) were not actioned since there was insufficient cash for buying or shares for selling. After each CT trading action (the unsatisfied CT trading action), on the next day, prices were still rising or falling depending on the event but with nothing special happening the previous day, hence this triggered a TF trading action to take place, which in turns prevented the next CT trading action (belonging to the next DC event) from being satisfied.

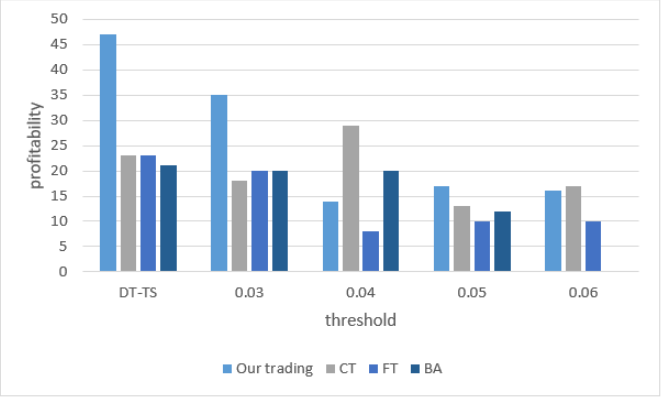


Figure 5.6: Profitability Comparision between Trading Strategies in the Testing set

In general, we do not want to miss a CT trading action because we are certain something has happened the previous day, and so it is an opportunity to trade. However if in the course of the same trend (a downward run or an upward run), a CT trading action is triggered but not satisfied (because of insufficient cash or shares), and this is followed by a single TF trading action (and the TF trading action is satisfied). Then in the next trend, another CT trading action is activated, this (the CT trading action) will also not be applied. In another words, in the same trend if there is an unsatisfied CT trading action followed by a single, satisfied, TF trading action, then if this is followed in turn by another CT trading action (belonging to the next trend), it will also not be satisfied as money or shares available have already been consumed.

5.3.3 Performance Evaluation Metrics

In this section, we measure the performance of the DT-TS trading strategy in compared fixed thresholds (the FT-TS) by applying the trading strategies performance metrics adopted from [184], to provide a more comprehensive view of our trading strategy's performance. We apply the total net profit, the profit factor, the percent profitable, the maximum drawdown, and the average trade net profit metrics.

Total net profit is the net value of the profits yielded by a trading strategy over a specific period of time. It is calculated by subtracting the gross loss of all trades which made a loss (i.e. losing trades) from the gross profit of all the trades which made a profit (i.e. winning trades). Gross profit is the sum of the returns from of all the trades which made a profit, while gross loss is the sum of the return from of all the trades which made a loss — in a specified period of time (this metric was used to measure the profitability when we applied the DT-TS), refer to Equation 5.2.

Profit factor is mainly for measuring the amount of profit per unit of risk, e.g. in the DT-TS (from Table 5.3) for every 1 monetary unit that is invested (unit of risk), 5.97 monetary units is returned (amount of profit). It is calculated by dividing the gross profit of all winning trades by the gross loss of all losing trades over a specific period of time, refer to Equation 5.3. Profit factor is always a positive value between zero (no wins) and infinity (no losses), a profit factor of value 1 means a break-even (wins are exactly equal to losses). In addition, the profit factor does not consider the number trades taking place, it only considers the amount of money gained or lost.

$$Profit - F = \frac{Gprofit}{Gloss} \tag{5.3}$$

The percent profitable measures the percentage of profitable trades in relation to all trades over a specific period of time; it provides no context to do with real net profits. It is calculated by dividing the number of winning trades by the total number of trades, refer to Equation 5.4. A high profitable percentage strategy can be found when more winning trades take place regardless to the amount of money gained. Thus a high profitable percentage is not always desirable, as it only considers the number of trades taking place, not the returns/profits gained.

$$Profit - P = \frac{Win_{trade}}{total_{trade}} \tag{5.4}$$

The maximum drawdown is the greatest price drop from the current maximum, in other words, it is the maximum losing streak that has been experienced. It represents the worst case scenario over a specific period of time. This metric is calculated by subtracting the peak value before the largest drop $peak_{value}$ from the lowest value before a new peak is achieved low_{value}, then divided by the peak value before the largest drop $peak_{value}$, refer to Equation 5.5. The maximum drawdown measures the size of the largest loss, but says nothing about the frequency of that loss.

$$MDD = \frac{peak_{value} - low_{value}}{peak_{value}} \tag{5.5}$$

Average trade net profit is the average amount of money gained or lost per trade. It is calculated by dividing the total net profit by the total number of trades, refer to Equation 5.6. This means, looking at DT-TS (from Table 5.3), each trade taking place is expected to average 3088.33.

$$Avg_{NetProfit} = \frac{TNprofit}{total_{trade}} \tag{5.6}$$

Trading strategy	Threshold	DC events	Total Net Profit	Profit Factor	Percent Profitable	Maximum Drawdown	Average Trade Net Profit
DT-TS	Dynamic	22	65%	5.97	76%	-5.18%	3088.33
FT-TS	0.03	22	62%	8.41	83%	-6.21%	2704.35
FT-TS	0.04	14	52%	15.46	84%	-2.03%	2721.37
FT-TS	0.05	12	41%	4.72	73%	-5.49%	2706.33
FT-TS	0.06	10	38%	6.37	67%	-1.83%	3112.75

Table 5.3: DT-TS Performance in Period 1 (from July 2015- March 2016), According to the Evaluation Metrics

Tables 5.3 and 5.4 illustrate the investigated periods: Table 5.3 refers to the period from July 2015 to March 2016 and Table 5.4 refers to the period from April 2016 to October 2016. Each table shows the threshold which was used as either a dynamic or a fixed threshold, and it also shows the number of detected DC events. It also presents the evaluation metrics we have adopted: the total net profit percentage, the profit factor, the percent profitable, the maximum drawdown, and lastly the average trade net profit.

It is noticeable that the DT-TS achieved the highest Total net profit in both periods. In period 1 (Table 5.3) it achieved 65%, and in period 2 (Table 5.4) it achieved 47%. For other evaluation metrics it is not possible to clearly say which trading strategy performed best, as none of the above metrics scored high with any of the trading strategies across both investigated periods. However DT-TS generally performed well, and always ranked in the highest three places with all evaluation metrics.

In addition, it is worth mentioning that the percent profitable metric records higher scores for lower threshold values: for example, the 0.03 fixed threshold in comparison to the 0.06 fixed threshold. This is because, with lower thresholds more DC events are detected and so more trading actions can take place, which in turns may lead to a high percent profitable score. On the other hand, the maximum drawdown evaluation metric may perform better with higher threshold values rather than lower ones, as with the 0.06 fixed threshold in comparison to the 0.03 fixed threshold; the latter has the worst maximum drawdown scores across both investigated periods. This is because with high threshold values fewer DC events are detected, which in turn can lead to fewer trading actions taking place, and so the amount of risk in trading is minimized.

Trading strategy	Threshold	DC events	Total Net Profit	Profit Factor	Percent Profitable	Maximum Drawdown	Average Trade Net Profit
DT-TS	Dynamic	16	47%	6.60	80%	-3.40%	3136.27
FT-TS	0.03	10	35%	6.00	85%	-3.48%	2522.36
FT-TS	0.04	10	14%	3.28	54%	-3.15%	1230.27
FT-TS	0.05	4	17%	5.15	71%	-2.20%	2383.14
FT-TS	0.06	4	16%	3.90	71%	-2.73%	2280.43

Table 5.4: DT-TS Performance in Period 2 (from April 2016- October 2016), According to the Evaluation Metrics

5.4 Summary

In this chapter, we introduced a new trading strategy based on the Directional Change approach but with using a daily dynamically defined threshold, named the Dynamic Threshold Trading Strategy (DT-TS). Apart from using a dynamic defined threshold, the DT-TS is a flexible strategy, in which it considers the previous day price transitions as an indicator for the type of trading (either CT or TF) to be followed, which also makes it different from other strategies that use the DC [36–38, 51, 52].

We applied the DC approach with a dynamic threshold that is defined daily based on the previous day's (between the previous day's opening and closing prices) and the overnight (between the previous day's closing price and the current day's opening price) price transitions. Such a dynamic threshold was used instead of a fixed threshold set a-priori. A DT-TS trading rule is opened up with every price increase/decrease during an upward/downward OS event. The trading decision to be taken depends on the previous day's and the overnight price transitions. The trading rule is closed when the next DC event is confirmed.

The DT-TS trading rules were constructed by building a decision tree based on a training dataset of the FTSE 100 minute-by-minute prices. If the previous day or overnight price change was greater than *previous_v*/*overnight_v* (*overnight_v* was set to 0.07 and *previous_v* was set to 0.08), then a CT trading strategy is followed (i.e. buying when it is a downward OS event and selling when it is an upturn OS event). Otherwise, if nothing extreme was noticed the previous day and overnight, then a TF trading is followed (i.e selling when it is a downward OS event and buying when it is an upturn OS event).

An experiment was conducted to evaluate the DT-TS trading rules using various fixed thresholds (0.03, 0.04, 0.05, and 0.06). Various trading strategies were also tested (CT, TF, and BA). The DT-TS trading strategy outperformed (gained more profits than) both the same strategy with fixed thresholds (FT-TS) and the other investigated trading strategies. Furthermore we evaluated the DT-TS trading strategy performance by applying the following key metrics: total net profit, profit factor, percent profitable, maximum drawdown, and average trade net profit. The DT-TS strategy scored highest in comparison to the use of fixed thresholds in terms of total net profit, and was always ranked in the top three places with respect to the other evaluation metrics.

Chapter 6

Exploring the Effectiveness of the DT-TS Approach on Financial Streams of Variable Frequencies

6.1 Introduction

Investors seek to identify price movements from price time-series data streams, as it may represent an investment opportunity. In the previous chapter (Chapter 5), we presented a trading strategy based on the DC approach which used a dynamic threshold which was set daily, we named this approach DT-TS. A DT-TS trading rule is activated with every price change in p_l/p_h during an OS event. The type of trading to be considered (either CT or TF) depends on the previous day's price transitions. If the previous day price transitions were extreme ($overnigh\_pc > overnight\_v$ or $previous\_PC > previous\_v$), then we follow a CT trading strategy. This is because the price change is assumed to be big and so we do not want to miss the opportunity (selling when prices are too high, and buying when prices are deeply down). Otherwise, if the previous day price changes were not extreme ($overnight\_pc \leq overnight\_v$ and $previous\_pc \leq previous\_v$), then we follow a TF trading strategy (buying when prices are increasing, and selling when prices decreasing).

One interesting question that arises in research on trading strategies is whether the same strategy can be equally effective on data streams of varying granularity: i.e., high frequency versus lower frequency streams. Hence, in this chapter, we aim to explore further the effectiveness of DT-TS as a trading strategy by applying it to two different time-series data streams, to see whether DT-TS performs better (achieves higher profitability) with higher frequency streams (minute-by-minute) or lower frequency streams (day-by-day).

The rest of this chapter is organized as follows. In the next section, we show how the dynamic threshold was defined for a low frequency time-series data stream (daily frequency). Section 6.3, shows how the dynamically defined threshold was applied with the DT-TS to such data streams. In Section 6.4, we discuss the performance of DT-TS from a profitability point of view in relation to both investigated streams: the higher frequency (minute-by-minute) and the lower frequency stream (day-by-day). The chapter ends with a summary and some conclusions in section 6.5.

6.2 Threshold Definition in a Daily Time-Series Stream

In the day-by-day prices stream, four prices are given at the end of each day, the opening, the high, the low and the closing price. The previous day's closing price is equal to the current day's opening price (in contrast to the situation with regard to the minute-by minute prices stream), which in consequence means that the percentage difference between the previous day's closing price and current day's opening price (the overnight price change) is always zero. As a result, it is clearly inappropriate for the daily dynamic threshold value to be set on the basis of the percentage difference between the previous day's opening and closing prices, and the percentage difference between the last high/low price and current day's high/low price, as this is how the threshold is defined if something is discovered the previous day. Therefore, we need to look at alternative ways of defining and setting the dynamic threshold.

As the daily stream provides four prices each day (opening, closing, high, and low), there are a number of indicators that we may want to take into account in setting the threshold value. For instance, we can consider the current day's opening and closing prices along with the high and low prices and/or the previous day's opening and closing prices along with high and low prices. Next, we will show some approaches in defining the dynamic threshold in the daily time-series stream.

The first approach (Approach 1) at defining a threshold value for the daily stream was undertaken by considering the percentage change between the current day's opening and closing prices (*current_PC*), and the percentage change between the current day's high and low prices ($currentHL - PC$), along with the percentage change between the last high/low price (p_h/p_l) and the current day's high/low price ($up/downward\_PC$), see Equation 6.1.

$$Threshold_1 = up/downward\_PC + current\_PC + currentHL\_PC \tag{6.1}$$

Another approach (Approach 2) was undertaken by considering the percentage change between the previous day's opening and closing prices $previous\_PC$, and the percentage change between the previous day's high and low prices $previousHL\_PC$, along with the percentage change between (p_h/p_l) and current day's high/low price $up/downward{-}PC$, refer to Equation 6.2.

$$Threshold_2 = up/downward\_PC + previous\_PC + previousHL\_PC \tag{6.2}$$

The third approach (Approach 3) at setting the dynamic threshold value for the day-by-day stream was undertaken by considering the percentage change between the previous day's opening and closing prices $previous\_PC$, and the percentage change between the current day's opening and closing prices $current\_PC$, along with the percentage difference between (p_h/p_l) and current day's high/low price $up/downward\_PC$, refer to Equation 6.3.

$$Threshold_3 = up/downward\_PC + previous\_PC + current\_PC \tag{6.3}$$

More approaches at setting the dynamic threshold value in lower frequency streams as follows. In Approach 4, the previous day high and low percentage change $previousHL\_PC$, and the current day high and low percentage change $currentHL\_PC$, along with the percentage change between (p_h/p_l) and current day's high/low price $up/downward\_PC$, refer to Equation 6.4. Approach 5 on the other hand, considers the previous day opening and closing price change $previous\_PC$, and the current day high and low percentage change $currentHL\_PC$, along with the percentage change between (p_h/p_l) and current day's high/low price $up/downward\_PC$, refer to Equation 6.5. Finally, in Approach 6, the dynamic threshold is set by considering the previous day high and low price change $previousHL\_PC$, and the current day opening and closing price change $current\_PC$, along with the percentage change between (p_h/p_l) and current day's high/low price

$up/downward\_PC$, refer to Equation 6.6.

$$Threshold_4 = up/downward - PC + previousHL\_PC + currentHL\_PC \tag{6.4}$$

$$Threshold_5 = up/downward - PC + previous\_PC + currentHL\_PC \tag{6.5}$$

$$Threshold_6 = up/downward\_PC + previousHL\_PC + current\_PC \tag{6.6}$$

6.3 Experimental Work

In the day-by-day prices stream, four prices are given at the end of each day, the opening, the high, the low and the closing price, we examine the daily closing price when trading. We have collected the FTSE 100 day-by-day prices from Reuters Thomson One [177] from July 2015 till end of October 2016.

6.3.1 Setting the Dynamic Threshold in Daily time-series stream

In this sub-section, we examine each of the approaches set for defining the dynamic threshold in the previous section (Approaches 1-6), and see how they will behave. Table 6.1 lists the dynamic threshold definition approaches in the daily stream.

Approach 1 (Equation 6.1) was able to detect 5 DC events, and 5 trading actions took place, and it yielded an 11% profit, refer to Table 6.1. For this approach, in most cases the daily defined threshold value was high, and so did not facilitate the detection of all the “true events” (the price change did not satisfy the threshold value criterion). Generally, a high threshold value may lead to the strategy missing events; furthermore, with a high threshold value, even when an event is detected, in most cases it is detected too late (i.e., not on the same day the corresponding news event is published). Some examples of events missed in this way or detected too late, from the daily stream and from the minute-by-minute stream, are as follows. An example of a missed event occurred on the 25th Aug 2015. Using the minute-by-minute stream an upturn event was detected on the same day 25th Aug (its threshold was set to 0.043), while using the day-by-day stream (on the same date) the threshold value was 0.052, and this was too high to allow the detection of any event (it still indicated a downward run until the 28th Oct, when an upturn event was finally detected). An example of late event detection on the other hand, took place on the 3rd Dec; a downturn event was detected using the minute-by-minute stream with a threshold of 0.015. However, using the daily stream on the same day (3rd Dec), the threshold was set at 0.029 (i.e., the threshold for the daily stream was higher than that set for the minute-by-minute stream), thus no event was detected with respect to the daily stream since the price change did not satisfy the relatively high threshold value. It was not until the 14th Dec that a downturn event was detected for the daily stream. As a result this form of the threshold calculation was rejected.

By Approach 2 (Equation 6.2) the DC events were detected more effectively (fewer events were missed in comparison to Approach 1); this was because the threshold was calculated based on the previous day's price transitions rather than the current day's. The number of DC events which were detected was 14 which was a significant improvement. Nevertheless, 8 events were still missed in comparison to those flagged as news events; moreover, 6 out of the 14 detected events were detected late. The number of trading actions which took place was 22, and a 24% profits was yielded, refer to Table 6.1.

The number of detected DC events, using Approach 3 (Equation 6.3), was 13; these generated 22 trading actions, and these in turn yielded a profitability of 37%, refer to Table 6.1. The threshold values which allowed the events to be detected were in the range 0.016-0.051. Even so, nine DC events were missed by this strategy in comparison to the events reported in the news. However, the DC events were detected more effectively (and detected in a more timely manner) than in approaches one and two. Consideration of both the previous day's and the current day's price transitions led to more effective calculated threshold values.

More approaches at defining effective threshold values are shown in Table 6.1. This table shows the approach number, a basic description of how the threshold was calculated, the number of detected DC events, the number of these that were detected late (Late events are events which are detected on the same day a news item regarding a price change was released, but was not the first news item reported that price change), the number of trading actions which took place, and finally the profitability (as a percentage). The highest profitability was achieved by Approach 3, while the lowest profitability was yielded by Approach 1. Figure 6.1 shows the DC events detected via all 6 approaches at defining an effective threshold value; each identified DC event is shown as a black dot on the prices graph. In addition, this figure demonstrates how

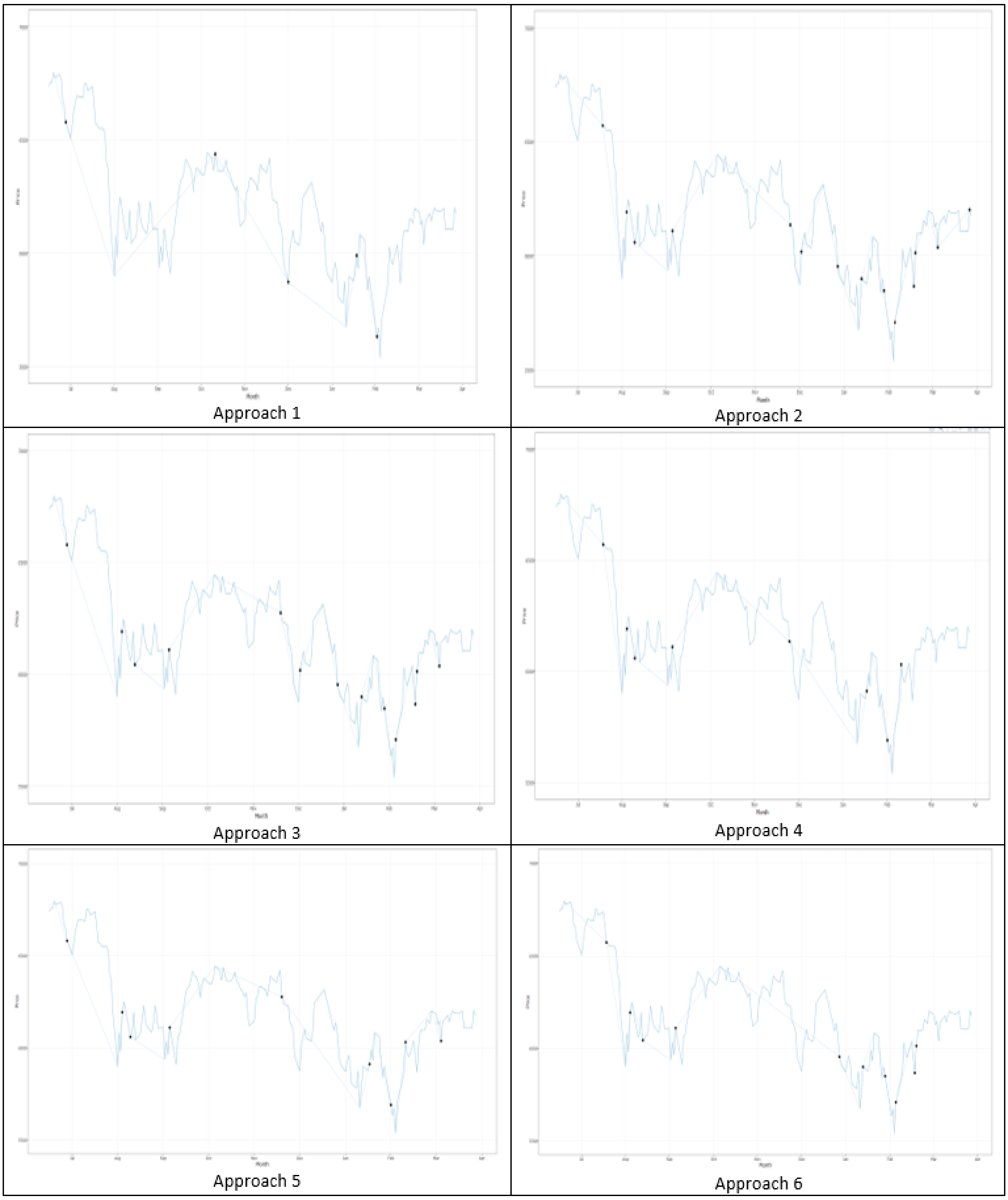
Approach#	Description	DC Events	Late Events	Trading Actions	Profits
Approach 1	Current day opening and close price, and high and low price.	5	5	5	11%
Approach 2	Previous day opening and closing price and high and low price.	14	6	22	24%
Approach 3	Previous day opening and closing price, and current day opening and close price.	13	5	22	37%
Approach 4	Previous day high and low price, and current day high and low price.	8	6	10	17%
Approach 5	Previous day opening and closing price, and current day high and low price.	10	5	12	25%
Approach 6	Previous day high and low price, and current day opening and close price.	9	5	12	21%

Table 6.1: The Daily Stream Threshold Calculation Approaches

each approach detected the DC events which occurred, and where events were missed or detected late. It is noticeable that Approach 3 was able to detect DC events in a more comprehensive and timely fashion than other approaches.

In general when using a daily prices stream, the calculated threshold values missed a large number of events; Approach 1 missed more than 75% of the events which occurred (which were the subject of published news items), while Approaches 2 and 3 missed more than 35% of events; Approaches 4, 5 and 6 were even worse in this respect, missing more than 50% of events. This was mainly because only a single price per day was compared against the threshold value, so if this price did not exceed (negatively or positively) the threshold, then no event was detected. In contrast, using the minute-by-minute stream, 510 prices (a price per minute) each day were compared to the threshold which had been calculated, and so there were far more opportunities whereby the current day's threshold value could be exceeded.

The dynamic threshold for the daily stream was defined more effectively when both the previous day's and the current day's price transitions were considered. In addition, the opening and closing prices were better calibrators than the high and low prices, as the percentage change between the high and the low prices were usually higher than the percentage change between the opening and the closing price — the higher the



Approach 1

Approach 2

Approach 3

Approach 4

Approach 5

Approach 6

Figure 6.1: The Detected DC Events in Different Threshold Calculation Approaches Using the Daily Prices Stream from July 2015- March 2016

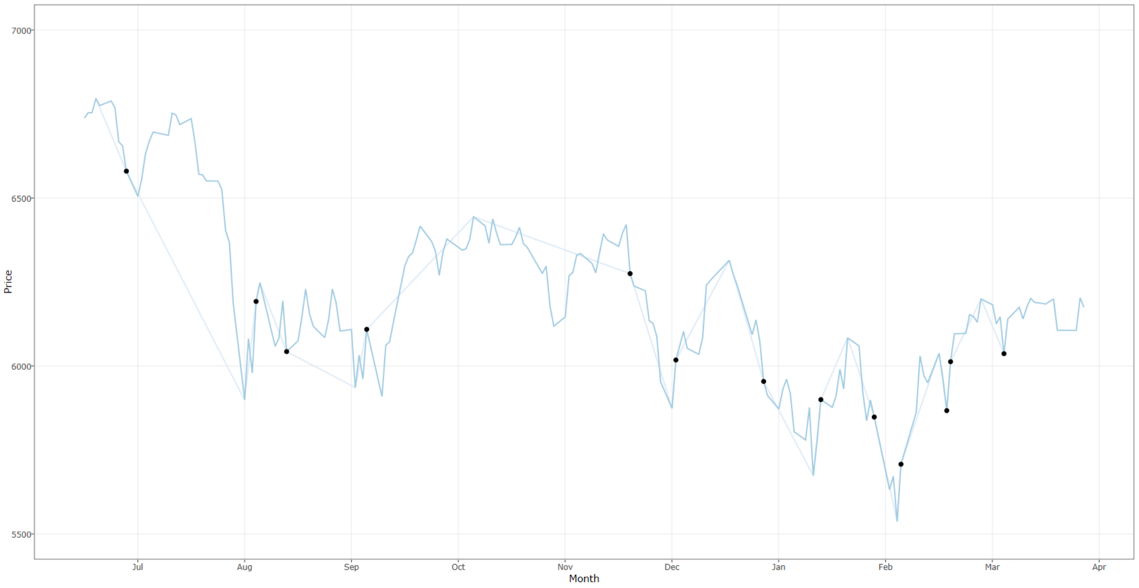


Figure 6.2: The Detected DC events using the daily prices stream in the period from July 2015 till end of March 2016

percentage change the higher was the threshold value, and the higher the threshold value the higher was the possibility of missing a "true" event (because the threshold criterion was less likely to be satisfied). As a result, the calculation tested in Approach 3 (in Equation 6.3) was adopted for calculating the threshold value for the daily stream.

6.3.2 Applying the DT-TS on Lower Frequency Streams

Figure 6.2 shows the events detected (represented as black dots) for the period from July 2015 until the end of March 2016, using both the daily stream and the daily defined threshold value. In this scenario, we were able to detect 13 DC events using threshold values ranging from (0.016-0.051). An analysis of the daily stream dynamic threshold (Approach 3) is shown in Table 6.2. We compare the results (i.e., the profitability) of our trading strategy using a dynamic threshold (DT-TS) and the results using a number of different fixed, a-priori thresholds (FT-TS for 0.03, 0.04, 0.05, and 0.06), and also those yielded by using a number of different trading strategies (CT, TF, and BA). The profitability is defined using the formula in Equation 5.2, where $Gprofit$ stands for the gross profit of all winning trades at the end of a trading period, and $Gloss$ is the gross loss of all losing trades in the same trading period.

Table 6.2 shows the type of threshold used (either dynamic or fixed), the trading strategy applied, the number of DC events detected, the number of trading actions which took place, and finally the profits made. The values of the calculated dynamic thresholds for the investigated period were in the range 0.016 to 0.051. The highest profits were yielded using DT-TS and the FT-TS with considerably low thresholds (0.03 and 0.04), regardless. The profits obtained via the BA strategy were not considered at this stage, since we were concerned, here (i.e. when using the BA), primarily with testing to find the most profitable values for the Down-ind and Up-ind values, so that these could then be used in the testing set.

The most profitable values for the Up-ind and Down-ind variables were found as follows. Using the dynamic threshold we first calculated the mean of the dynamic threshold values (which were in the range 0.016-0.051). The mean of the dynamic threshold values was 0.029. Then, for that mean, we found the value of Down-ind and Up-ind. Accordingly, the values of Down-ind and Up-ind calculated via the mean of the daily calculated threshold values were -0.01 and 0.25 respectively. Using the 0.03 fixed threshold, the Down-ind and Up-ind values were calculated as -0.71 and 0.93 respectively. Using the 0.04 fixed threshold, the Down-ind and Up-ind values were calculated as -0.16 and 0.129 respectively. The 0.05 fixed threshold yielded Down-ind and Up-ind values of -0.37 and 0.17 respectively. Lastly, for the 0.06 fixed threshold, the Down-ind and Up-ind values were -0.18 and 0.23 respectively.

In Table 6.2, it is noticeable that the number of DC events which were detected using the dynamic threshold was less than the number of DC events which were detected using the 0.03 threshold: there were 13 events detected using the dynamic threshold and 16 events detected using the 0.03 fixed threshold. In fact the more DC events which are detected, the more the chances are that trading actions can take place, and more trading actions can lead to more profit. However this is not the case with respect to the dynamic threshold and the 0.03 fixed threshold, as the profits yielded by using each of these are nearly equal (37% with the dynamic threshold and 38% with the 0.03 threshold); this is because, using the dynamic threshold, events are detected in a more timely fashion than they are when using a fixed threshold.

Figure 6.3 shows the detected events (represented as black dots) for the period from April 2016 until the end of Oct 2016, from the daily share-prices stream. We were able to detect six DC events using thresholds with values ranging from (0.015-0.058).

Table 6.3 illustrates the daily stream analysis for that period as well. Again, this shows the type of threshold used (either dynamic or fixed), the trading strategy applied, the number of DC events detected, the number of trading actions which took place, and

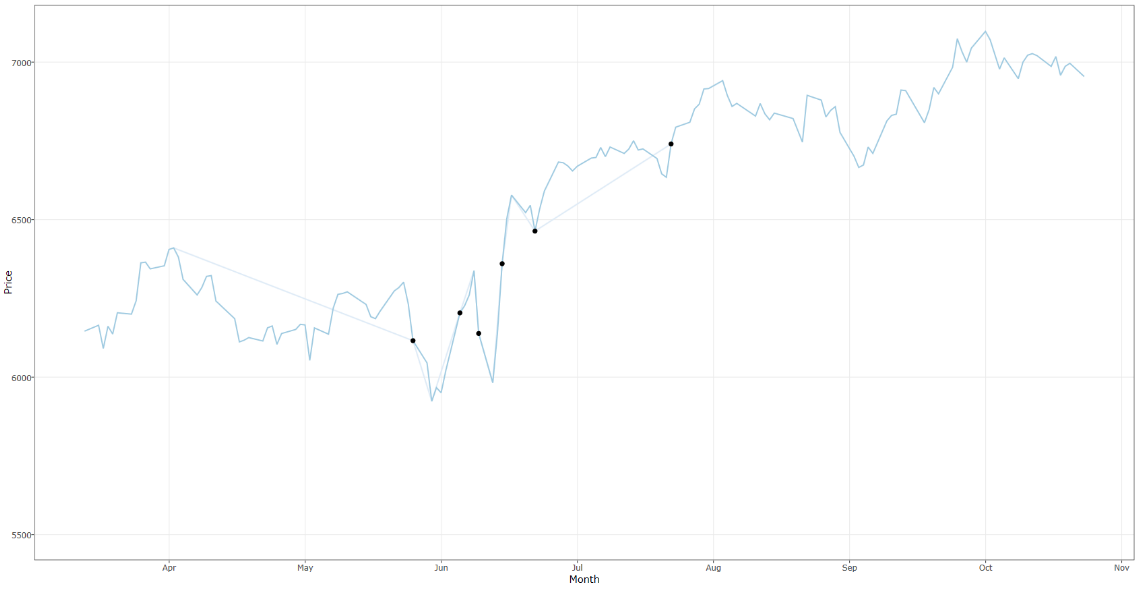


Figure 6.3: The Detected DC events using the daily prices stream in the period from April 2016 till end of Oct 2016

finally, the profits gained. Once more the use of the dynamic threshold with DT-TS, and the use of the 0.03 fixed threshold with FT-TS outperformed other threshold values/trading strategies. The dynamic threshold enabled the detection of 6 DC events and triggered 13 trading actions, while with the 0.03 fixed threshold, the numbers of DC events detected and the number of trading actions triggered were even higher (8 DC events and 19 trading actions). However, with both threshold schemes, the profits gained were equal to 33%. Furthermore, the use of the dynamic threshold and the 0.04 fixed threshold detected the same number of DC events (6 DC events) as the use of the dynamic threshold, but the profits yielded by the use of a fixed 0.04 threshold were only 24%. Thus the profits yielded by the use of a dynamic threshold were more than one-fourth higher than those yielded by using a 0.04 fixed threshold, even when the number of detected DC events were the same. One of the reasons for the higher profits when using a dynamic threshold, was that the use of dynamic thresholds result in events being detected in a more timely manner than does the use of fixed thresholds, which in consequence leads to a trading actions taking place at a more advantageous price situation.

The FT-TS, when used with a fixed threshold and when used for examining daily prices achieves better results than the other trading strategies, just as it does when looking at minute-by-minute prices. The use of the fixed 0.04 threshold (for FT-TS) achieved the highest profitability percentage, of 24%, and resulted in the detection

Threshold	No. events	Trading	No. Tradings	Profits
Dynamic	13	DT-TS	22	37%
		CT	10	12%
		TF	11	12%
		BA	10	43%
0.03 (Fixed)	16	FT-TS	23	38%
		CT	14	25%
		TF	15	3%
		BA	11	39%
0.04 (Fixed)	12	FT-TS	20	37%
		CT	12	36%
		TF	13	-4%
		BA	13	47%
0.05 (Fixed)	10	FT-TS	16	32%
		CT	8	19%
		TF	9	-4%
		BA	8	43%
0.06 (Fixed)	8	FT-TS	15	26%
		CT	6	18%
		TF	7	-5%
		BA	7	27%

Table 6.2: Daily Stream: July 2015 until End of March 2016 Stream Analysis

of 6 DC events, while the highest profitability percentage achieved by the use of a 0.05 threshold and FT-TS was 18% (with 4 DC events detected), and the highest profitability percentage using the 0.06 threshold (and also FT-TS) was 19% with 2 DC events detected.

In summary, when using a daily prices stream, both DT-TS and FT-TS with small fixed threshold values yield almost the same in terms of profits. Refer to Table 6.2 for the period from July 2015 until March 2016 and to Table 6.3 for the period from April 2016 until the end of October 2016; these show the results achieved by the use of a dynamic threshold for those periods, and by the use of a fixed, 0.03, threshold.

Threshold	No. events	Trading	No. Tradings	Profits
Dynamic	6	DT-TS	13	33%
		CT	4	20%
		TF	5	3%
		BA	4	13%
0.03 (Fixed)	8	FT-TS	19	33%
		CT	8	32%
		TF	9	3%
		BA	4	17%
0.04 (Fixed)	6	FT-TS	15	24%
		CT	2	4%
		TF	3	12%
		BA	2	4%
0.05 (Fixed)	4	FT-TS	11	18%
		CT	2	7%
		TF	3	10%
		BA	0	0%
0.06 (Fixed)	2	FT-TS	11	19%
		CT	0	0%
		TF	1	13%
		BA	0	0%

Table 6.3: Daily Stream: April 2016 until End of Oct 2016 Stream Analysis

Source	Events	Trading Actions	Profits
Daily prices stream	13 DC events	22	37%
Minute-by-minute prices stream	22 DC events	42	65%
BBC News	54 News events	NA	NA

Table 6.4: Event Detection Source Summary from July 2015-March 2016

6.4 Discussion

In this section, we look at the results (the profits gained) from both price time-series streams (the minute-by-minute and the daily prices streams) as a result of trading. Table 6.4 shows the sources considered for event detection (the day-by-day stream, the minute-by-minute stream, and the BBC News); the number of trading actions taking place (in the price streams only); and the total profits yielded (for the price streams only) for the period from July 2015-March 2016. The daily prices stream (the low frequency stream) yielded 13 detected events and a 37% profit; in comparison, the minute-by-minute stream (the higher frequency stream) yielded 22 detected DC events and a 65% profit. The BBC News on the other hand published more than 54 news articles regarding the FTSE 100, during the investigated period. In point of fact, it was possible that on some occasions more than one news article related to the same price increase or decrease, in the days following the first reported news event – as prices went even further lower or higher. An example of such repeated reporting regarding the same price decrease or increase, was that related to the price decrease which happened in early December. The first news article released was on the 3rd December[1], then the second article was published on the 10th December[2], and finally a third was produced on the 11th December[3]; thus three news articles were published which essentially related to the same price decrease event. This explains why, in general, the number of news articles exceeds the number of detected DC events.

Refer to Table 6.5 to see the average of the profits gained from the minute-by-minute stream for the whole period from July 2015-Oct 2016, and to Table 6.6 to see the average of the profits gained from the day-by-day stream for the same time period. Tables 6.5

[1] http://www.bbc.co.uk/news/business-34992913
[2] http://www.bbc.co.uk/news/business-35059626
[3] http://www.bbc.co.uk/news/business-35070184

Threshold	Trading	Profits
Dynamic	DT-TS	56%
	CT	29%
	TF	28%
	BA	30%
0.03 (Fixed)	FT-TS	49%
	CT	16%
	TF	31%
	BA	29%
0.04 (Fixed)	FT-TS	33%
	CT	21%
	TF	15%
	BA	32%
0.05 (Fixed)	FT-TS	29%
	CT	15%
	TF	9%
	BA	28%
0.06 (Fixed)	FT-TS	27%
	CT	19%
	TF	4%
	BA	20%

Table 6.5: Average profits in the minute-by-minute Stream Using Different Trading Strategies

and 6.6 show the threshold values which were adopted (either dynamic or fixed) with the different trading strategies (DT-TS, FT-TS, CT, TF, and BA).

Furthermore, the bar charts in Figures 6.4 and 6.5 summarize the profits gained across both price streams for the same investigated period. It is clear that the DT-TS out performs all the other trading strategies in relation to both the high and lower frequency streams. However, the use of the minute-by-minute stream (the high frequency stream) resulted in higher profits than those yielded from the daily stream (the low frequency stream) almost regardless of the trading strategy adopted. This is because events (price changes) can be detected in a more timely fashion when the minute-by-minute stream (showing intra-day prices) rather than the day-by-day stream (daily closing prices) is in use. The more timely an event detection is (since better timing leads to better opportunities in terms of price), the more profitable a trading action can

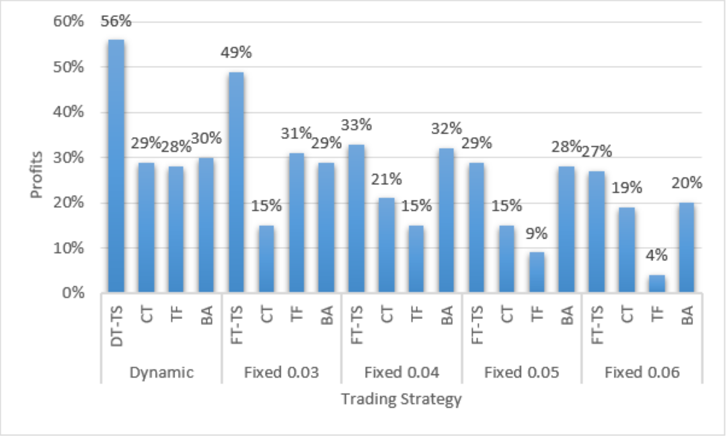


Figure 6.4: Profitability Comparision between Trading Strategies By the Minute-by-Minute Stream

be.

All the events detected from the minute-by-minute stream were found the same day as the news event was published, refer to Table 4.2. While for the daily stream in contrast more than 35% of the detected events (5 events of the 13 detected ones, refer to Table 6.1) were detected late. In addition, more events were missed in relation to the daily stream in comparison to the minute-by-minute stream; 38 events were detected in total for the minute-by-minute stream and 19 events for the daily stream, so almost half of the events which occurred were missed in the daily stream. An event is missed or detected late in the daily stream when the threshold value is not satisfied when examined against the daily streamed price. Each day a single price is examined for the daily stream (daily closing price), while for the minute-by-minute stream 510 prices are examined every day (a single price each minute); thus the chances of detecting an event in the minute-by-minute stream is a lot higher than it is for the daily stream. As a result, DT-TS works better (gains higher profits) with data streams of higher frequency levels than it does with lower frequency ones.

6.5 Summary

In this chapter, we explored the application of DT-TS in relation to financial time-series data streams with different data flow levels (different frequencies) to investigate whether DT-TS performs equally well on data streams of different frequencies and whether it achieves good results.

The DT-TS is a trading strategy based on the DC approach and a daily dynamically

Threshold	Trading	Profits
Dynamic	DT-TS	35%
	CT	16%
	TF	8%
	BA	28%
0.03 (Fixed)	FT-TS	36%
	CT	29%
	TF	3%
	BA	28%
0.04 (Fixed)	FT-TS	31%
	CT	20%
	TF	4%
	BA	26%
0.05 (Fixed)	FT-TS	25%
	CT	13%
	TF	3%
	BA	22%
0.06 (Fixed)	FT-TS	23%
	CT	9%
	TF	4%
	BA	14%

Table 6.6: Average profits in the Daily Stream Using Different Trading Strategies

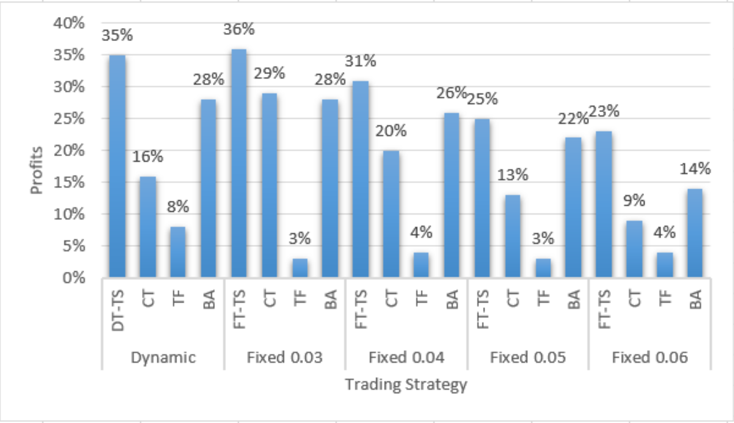


Figure 6.5: Profitability Comparision between Trading Strategies By the Daily Stream

defined threshold; the trading action to be taken depends on the previous day's price transitions. When a DC event is detected (a price change is identified), a trading action is triggered. The type of trading action taken (CT or TF) is determined by the behaviour of the prices on the previous day.

For the daily stream, before applying the DT-TS, we had to define the dynamic threshold to be used for spotting price transitions according to the DC approach. Different threshold definition methods were considered, and the one which was most effective at spotting the DC events was then chosen. Experiments were carried out on two FTSE 100 time-series streams (a day-by-day prices stream and a minute-by-minute prices stream) for more than 60 weeks; this was in order to further investigate the applicability of DT-TS on time-series streams of differing frequencies.

DT-TS, applied to a day-by-day stream, performed well in comparison to other trading strategies. However when looking at both streams, the daily stream and the minute-by-minute stream, we found that the DT-TS performs better (gains higher profits) with the higher frequency stream (the minute-by-minute stream) rather than lower ones.

Chapter 7

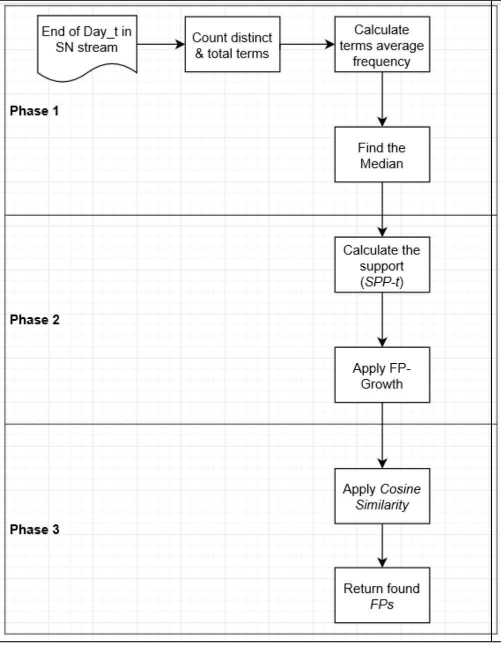
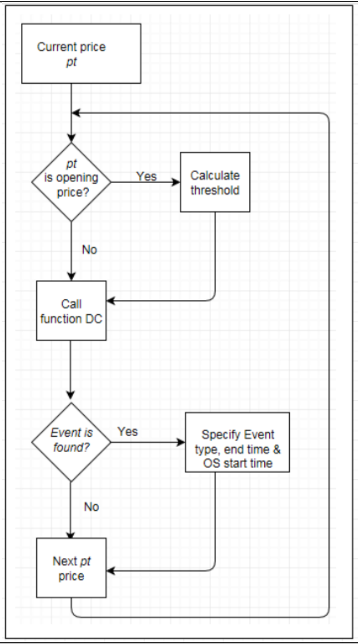
Investigating Interrelationships Across Data Streams

7.1 Introduction

In chapters 3 and 4 to 6, we developed methods for detecting events from social network streams and from financial market time-series data streams. The main thing we were aiming for was to develop and extend event detection methods so that they could function effectively on streams of data. As a result, we were able to detect the occurring events from text streams (Twitter) and price time-series data streams (stock prices). One of the questions that has arisen in particular in the domain of financial data is whether and to what extent events, as detected from social media, affect market prices. In this chapter, we bring the two strands of work together and we consider whether indeed we can cross-reference over both streams and see what relations could be found between them. Furthermore, we are interested in discovering whether the magnitude of the studied events — in terms of them being on a regional or on an international level, for instance — would have an effect on the relationships between streams, if any. The magnitude of the GE 2015 event in relation to the FTSE 100 is regional, the Greece Crisis on the other hand, is on an international or global level in relation to the FTSE 100.

As a preliminary step before going on to cross-referencing over both streams, a comparison summary is conducted in terms of the event identification process between the text stream (Twitter) and the time-series data stream (stock prices) starting from the stream data type and ending with when events can be found. In more details, Table

7.1 provides a comparison between the text stream and the time-series data stream: it shows both streams' data types; how and when the thresholds are set; the length of the threshold's validity; the method used to identify events; when events can be found; and finally, the overall structure of event detection in relation to each stream.

	Social network stream	**Time-Series stream**
Data Type	Text data	Financial (prices) data
Data used to calculate threshold	Current day data	Previous day data plus current day opening price
When threshold is set	End of current day	Beginning of current day
Duration of threshold usage	Threshold is used for a single day only	Threshold is used for a single day only
Method used	FP-Growth	DC approach
Events found if any	End of day	Once they occur
Overall Structure		

End of Day_t in SN stream
Count distinct & total terms
Calculate terms average frequency
Phase 1
Find the Median
Calculate the support (SPP-t)
Phase 2
Apply FP-Growth
Apply Cosine Similarity
Phase 3
Return found FPs

Table 7.1: Comparison between Proposed Event Detection Methods Applied to Social Network (Text) stream and Time-Series (financial) stream

In relation to the FTSE 100 prices stream and the GE 2015 tweets stream — during the UK 2015 General Election period, and before we can put together and draw inferences from the two streams (the social network stream and the time-series stream). The FTSE 100 daily prices, over the same period as was covered by the GE 2015 stream, were collected on a daily basis (the minute-by-minute prices were not available for that period). We wanted to further investigate the relation between the events detected from the GE 2015 tweets stream and those detected from the FTSE 100 prices stream. To do this, we downloaded the FTSE 100 prices for the period 1/3/2015-26/5/2015, and ran our DC event detection framework for (roughly) the same period to see if we would detect any DC events. We used 1/3/2015 as the beginning of our time-series stream instead of 15/4/2015, which was the beginning of the GE 2015 social network stream, in order to explore more price transitions, and so, set the values of p_l, p_h (lowest and highest prices reached) prior to the GE 2015 period to be able to detect the DC events during the GE 2015 period, if any. The period we explored is shown in Figure 7.1 (indicated within the vertical lines).

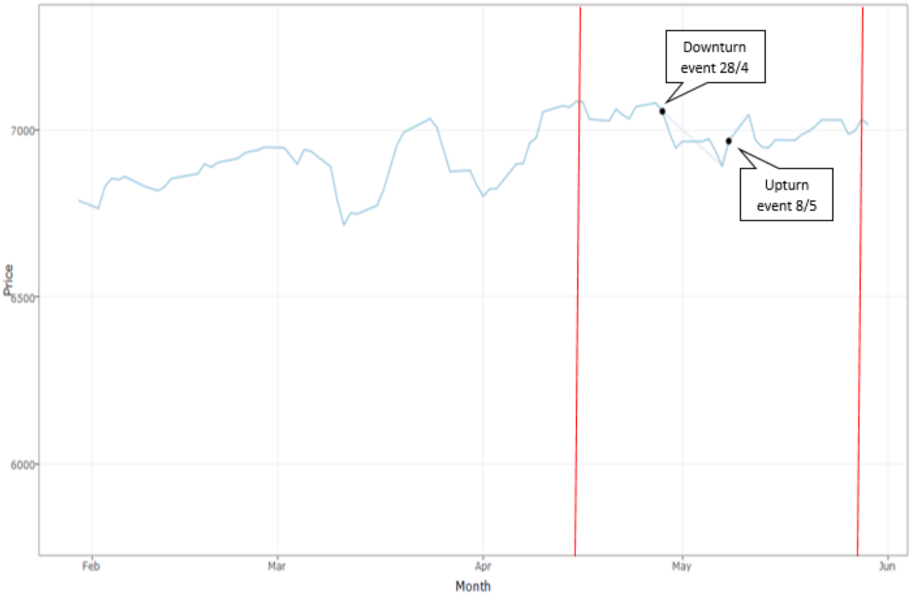
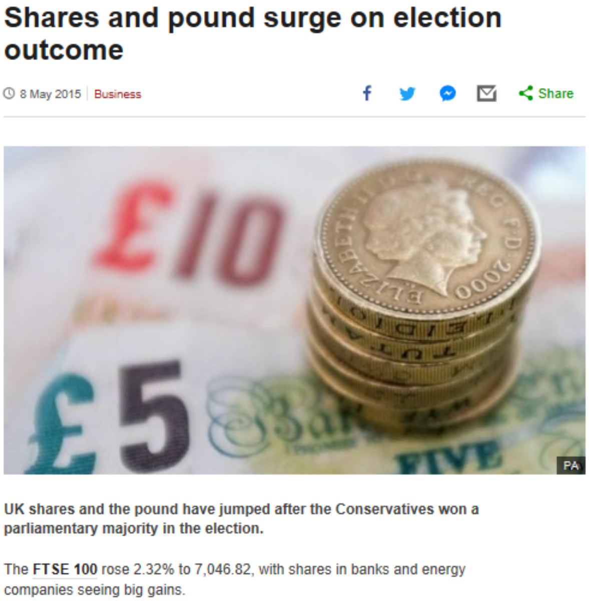


Figure 7.1: The FTSE 100 Daily Prices During the GE2015 Period (vertical red lines indicate the GE 2015 investigated period from 1/3/2015-26/5/2015)

On running the DC approach, using our daily dynamically defined threshold, we detected a downturn event on the 29th April (10 days prior to the elections day), where the FTSE 100 prices dropped by 1.1%[1]. Subsequently, an upturn event was detected on the 8th of May, after the announcement of the GE results; the FTSE 100 prices increased by 2% after the Conservatives won (refer to Figure 7.2 for an article from the BBC News[2], released on Friday 8th of May). Figure 7.3 shows the downturn event detected (using a daily dynamically defined threshold) on the 29th of April, while Figure 7.4 demonstrates the upturn event detected on the 8th of May. In fact, when investigating the behaviour of the FTSE 100 during the GE 2015 period via the news outlets, the only very large price transitions found were as a result of the Conservatives Party winning (the price decrease on 29th April was a result of some caution ahead of an interest rate policy meeting at the US Federal Reserve[1]).



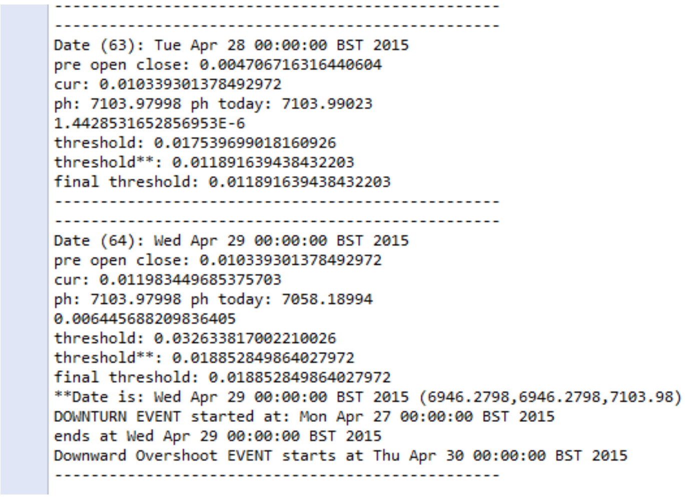
Shares and pound surge on election outcome

8 May 2015 | Business

UK shares and the pound have jumped after the Conservatives won a parliamentary majority in the election.

The **FTSE 100** rose 2.32% to 7,046.82, with shares in banks and energy companies seeing big gains.

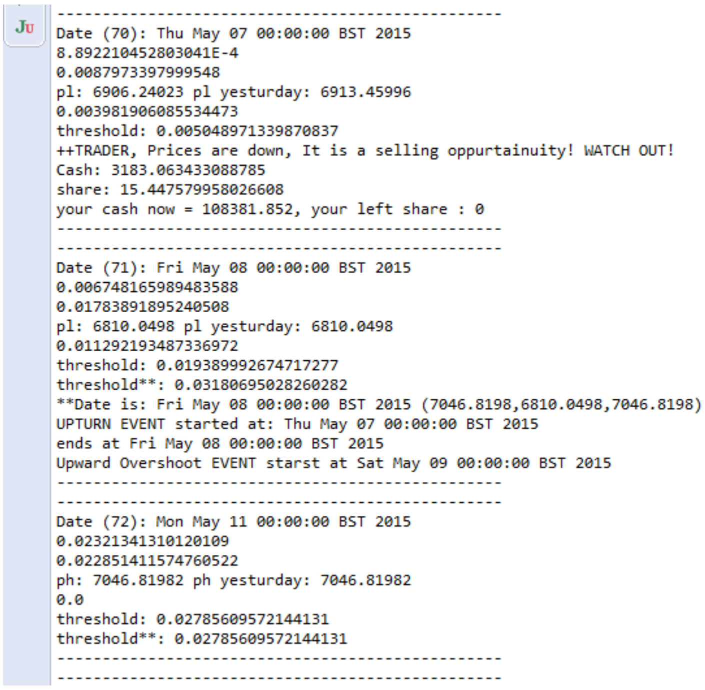
Figure 7.2: FTSE 100 After Conservitives winning GE 2015[2]



```
-------------------------------------------------
-------------------------------------------------
Date (63): Tue Apr 28 00:00:00 BST 2015
pre open close: 0.004706716316440604
cur: 0.010339301378492972
ph: 7103.97998 ph today: 7103.99023
1.4428531652856953E-6
threshold: 0.017539699018160926
threshold**: 0.011891639438432203
final threshold: 0.011891639438432203
-------------------------------------------------
-------------------------------------------------
Date (64): Wed Apr 29 00:00:00 BST 2015
pre open close: 0.010339301378492972
cur: 0.011983449685375703
ph: 7103.97998 ph today: 7058.18994
0.006445688209836405
threshold: 0.032633817002210026
threshold**: 0.018852849864027972
final threshold: 0.018852849864027972
**Date is: Wed Apr 29 00:00:00 BST 2015 (6946.2798,6946.2798,7103.98)
DOWNTURN EVENT started at: Mon Apr 27 00:00:00 BST 2015
ends at Wed Apr 29 00:00:00 BST 2015
Downward Overshoot EVENT starts at Thu Apr 30 00:00:00 BST 2015
-------------------------------------------------
```

Figure 7.3: The Downturn DC event on 29th of April

[1] http://www.bbc.co.uk/news/live/business-32452206
[2] http://www.bbc.co.uk/news/business-32644523



```
-------------------------------------------------
Date (70): Thu May 07 00:00:00 BST 2015
8.892210452803041E-4
0.0087973397999548
pl: 6906.24023 pl yesterday: 6913.45996
0.003981906085534473
threshold: 0.005048971339870837
++TRADER, Prices are down, It is a selling oppurtainuity! WATCH OUT!
Cash: 3183.063433088785
share: 15.447579958026608
your cash now = 108381.852, your left share : 0
-------------------------------------------------
-------------------------------------------------
Date (71): Fri May 08 00:00:00 BST 2015
0.006748165989483588
0.01783891895240508
pl: 6810.0498 pl yesturday: 6810.0498
0.011292193487336972
threshold: 0.019389992674717277
threshold**: 0.03180695028260282
**Date is: Fri May 08 00:00:00 BST 2015 (7046.8198,6810.0498,7046.8198)
UPTURN EVENT started at: Thu May 07 00:00:00 BST 2015
ends at Fri May 08 00:00:00 BST 2015
Upward Overshoot EVENT starst at Sat May 09 00:00:00 BST 2015
-------------------------------------------------
-------------------------------------------------
Date (72): Mon May 11 00:00:00 BST 2015
0.02321341310120109
0.022851411574760522
ph: 7046.81982 ph yesturday: 7046.81982
0.0
threshold: 0.02785609572144131
threshold**: 0.02785609572144131
-------------------------------------------------
-------------------------------------------------
```

Figure 7.4: The Upturn DC event on 8th of May

There have been various efforts attempting to find a relationship between the financial markets and the various different media sources provided by the World Wide Web: news articles, weblogs, social networks, etc. In fact, the majority of those efforts (including, but not limited to, [136, 139, 141, 185–189]) considering an additional source for studying the financial markets behaviours mainly depend on web-posts that mention the investigated market's data explicitly.

Tetlock in [186] measures the nature of the interaction between the media and daily stock market activity. This was one of the earliest attempts at finding evidence that news media content can predict stock market activity. More specifically, it was found that high media pessimism predicts downward pressure on stock market prices, and high or low pessimism predicts high trading volume in stock market.

The work in [190] found a positive correlation between the daily number of mentions of a company in the news (the Financial Times) and the daily transaction volume of that company's stock both one day before the news was released, and on the same day the news was released.

In [185], the authors wanted to assess whether emotions estimated from weblogs (the LiveJournal) can provide information about future prices of the S&P 500 index on the stock market. Using the Granger causality method they found that the anxiety emotion index contains information about future stock market prices that is not yet available in the market data itself. Alternatively, the study in [136] investigated the relationships between Twitter and the financial markets. They considered the sentiment and volume of tweets relating to the Dow Jones Industrial Average (DJIA) index along with the DJIA prices (market data). A relatively low Pearson correlation and Granger causality was found between the two investigated time series over the time period which was examined.

We are interested in putting together and drawing inferences from both streams; we aim to explore the relation between the text stream and the time-series data stream, if any. What we are specifically interested in is investigating whether events detected from the FTSE 100 prices stream, representing the share index of the 100 companies with the highest market capitalisation listed on the LSE, has an effect on events detected from a text stream – at either a regional level (the UK GE 2015) or a European level (the Greece crisis 2015). Unlike the previously mentioned studies, [136, 139, 141, 142, 144, 186–189], our text stream (the Twitter stream) is not directly related to the financial market data under investigation. In other words, the text stream does not generally explicitly discuss any of the particular assets or shares being looked at, instead it relates to a major event that is taking place, such as the UK general elections.

The rest of this chapter is organized as follows: the Correlation Coefficient is investigated in section 7.2; Section 7.3 explores the Granger causality method; and finally the chapter ends with a summary and conclusions in section 7.4.

7.2 Measuring the Relationship Through Pearson Correlation Coefficient

Correlation, in general, is a technique for investigating the relationship between two or more quantitative [191] values. The Pearson correlation coefficient (R) [192] measures the strength and direction of a linear relationship between two variables, X and Y. The Pearson equation is shown in Equation 7.1, where X and Y are the examined variables for the correlation and n is the number of values.

$$R = \frac{n(\sum XY) - (\sum X)(\sum Y)}{\sqrt{[n\sum X^2 - (\sum X)^2][n\sum Y^2 - (\sum Y)^2]}} \tag{7.1}$$

The output, R, is a value between [1,-1]: a value of 1 or close to 1 indicates a positive linear relationship; a zero value means that no linear relationship was found; and a value of -1 or close to -1 indicates a negative linear relationship.

The Pearson correlation method was applied to the data from both the GE 2015 period and the Greece crisis 2015 time period. As a starting point, we wanted to see if there was correlation between the number of posted tweets and the daily FTSE 100 index prices. Thus the variable, X, was the number of tweets posted daily, and Y was the daily FTSE 100 prices. Table 7.2 shows the values of X and Y throughout the GE 2015 period, where we had 29 windows, and Table 7.3 shows the values of X and Y throughout the Greece crisis period, where we had 17 windows.

R in the GE period=0.0973

R in the Greece crisis= 0.0553

These values of R, both for the GE period and for the Greece crisis period, indicate that no linear correlation was found between the number of tweets posted daily and the daily FTSE 100 prices. A high number of posted tweets (a high tweet' volume) on a certain day could coincide with either a FTSE 100 price increase or a FTSE decrease. This could be the reason for there being a poor relationship or no relationship found between them.

X	Y	X*Y	X*X	Y*Y
23836	7096.8	169159320.2	568154896	50364567.47
24294	7060.5	171527787	590198436	49850660.25
27210	6994.6	190323068.7	740384100	48924430.53
33622	7052.1	237105709.5	1130438884	49732115.79
27759	7062.9	196059038.4	770562081	49884555.03
23427	7028.2	164649646	548824329	49395597.98
22511	7053.7	158785845.1	506745121	49754686.44
22032	7070.7	155781666.7	485409024	49994801.25
29957	7104	212814528	897421849	50466816
27023	7030.5	189985201.5	730242529	49427930.25
35546	6946.3	246913172.9	1263518116	48251080.98
34281	6960.6	238616332	1175186961	48449953.72
34410	6986	240388260	1184048100	48804196
27052	6927.6	187405437.9	731810704	47991643.12
57347	6933.7	397626905.1	3288678409	48076198.39
73571	6887	506683477	5412692041	47430769
171829	7046.8	1210844564	29525205241	49657387.49
388966	7029.9	2734392045	1.51295E+11	49419492.63
13648	6933.8	94632499.74	186267904	48077579.74
7942	6949.6	55193723.98	63075364	48296941.52
5545	6973	38665285	30747025	48622729
4081	6960.5	28405800.5	16654561	48448560.25
3744	6968.9	26091561.23	14017536	48565565.84
2429	6995.1	16991098.14	5900041	48931425.38
2262	7007.3	15850512.16	5116644	49102250.56
1607	7013.5	11270694.5	2582449	49189182.25
2046	7031.7	14386858.6	4186116	49444807.63
2215	6949	15392035	4906225	48288601
1699	7033	11949067	2886601	49463089

Table 7.2: The values of X and Y in the GE 2015 period, where X is the number of posted tweets in a day and Y is the FTSE 100 daily prices

X	Y	X*Y	X*X	Y*Y
9294	6520.97998	60605987.93	86378436	42523179.9
19989	6608.58984	132099102.3	399560121	43673459.67
6888	6630.47021	45670678.81	47444544	43963135.21
3229	6630.47021	21409788.31	10426441	43963135.21
9287	6630.47021	61577176.84	86248369	43963135.21
1903	6585.77979	12532738.94	3621409	43372495.44
11241	6535.68018	73467580.9	126360081	42715115.42
6258	6432.20996	40252769.93	39162564	41373324.97
2219	6673.37988	14808229.95	4923961	44533999.02
6458	6673.37988	43096687.27	41705764	44533999.02
6058	6673.37988	40427335.31	36699364	44533999.02
17065	6737.9502	114983120.2	291214225	45399972.9
20083	6753.75	135635561.3	403326889	45613139.06
5357	6753.75	36179838.75	28697449	45613139.06
4695	6796.4502	31909333.69	22043025	46191735.32
3306	6775.08008	22398414.74	10929636	45901710.09
3143	6775.08008	21294076.69	9878449	45901710.09

Table 7.3: The values of X and Y in the Greece Crisis period, where X is the number of posted tweets in a day and Y is the FTSE 100 daily prices

We then considered the relationship between the two data streams, the text stream and the time-series data stream, by undertaking the number of tweets posted per day against the number of DC events detected per day. We wanted to see if a high number of posted tweets (a peak value), can increase the chances of identifying events in the financial stream, or not. Thus X was set to the number of tweets posted each day and Y was set to the number of DC events detected each day.

During the GE 2015 period we had 29 windows; refer to Table 7.4 to see the values of X and Y for this period. There were two DC events in the GE 2015 period detected from the daily FTSE 100 price stream: a downturn event on 29/4/2015 (Y is set to 1 in window 11), and an upturn event on 8/5/2015 (Y is set to 1 in window 18). As a result, after applying the Pearson Equation in 7.1, the value of R=0.64; this means that the number of detected DC events and the number of posted tweets in a day has a moderate positive relationship.

X	Y	X*Y	X*X	Y*Y
23836	0	0	568154896	0
24294	0	0	590198436	0
27210	0	0	740384100	0
33622	0	0	1130438884	0
27759	0	0	770562081	0
23427	0	0	548824329	0
22511	0	0	506745121	0
22032	0	0	485409024	0
29957	0	0	897421849	0
27023	0	0	730242529	0
35546	1	35546	1263518116	1
34281	0	0	1175186961	0
34410	0	0	1184048100	0
27052	0	0	731810704	0
57347	0	0	3288678409	0
73571	0	0	5412692041	0
171829	0	0	29525205241	0
388966	1	388966	1.51295E+11	1
13648	0	0	186267904	0
7942	0	0	63075364	0
5545	0	0	30747025	0
4081	0	0	16654561	0
3744	0	0	14017536	0
2429	0	0	5900041	0
2262	0	0	5116644	0
1607	0	0	2582449	0
2046	0	0	4186116	0
2215	0	0	4906225	0
1699	0	0	2886601	0

Table 7.4: The values of X and Y in the GE 2015 period, Where X is the Number of Tweets and Y is the Number of Detected DC Events

Table 7.5 demonstrates the values of X and Y throughout the Greece crisis 2015 period, where we had 17 windows. During that period the FTSE 100 prices were falling, on average; refer to Figure 7.5 to see an overall picture of the investigated period (indicated within the red vertical lines). In addition, the FTSE 100 released news items related to the Greece crisis 2015, came out on 29/6/2015, on which day the FTSE 100 prices went down 1.97%, a dramatic drop[3], and on 7/7/2015, when the FTSE 100 prices slipped again, by 0.8%[4]. However when applying the DC approach with our daily dynamically defined threshold on the FTSE 100 price stream, a downturn event was detected earlier (on 4/6/2015), so during the period relating to the Greece crisis which was investigated it was continuing on a downward run (downturn OS). Refer to Figure 7.6 to see a snapshot of the results of the DT-TS run on 29/6/2015, as the FTSE 100 prices went down to a level at which a buying trading action was triggered. As no DC event was detected, the value of Y for all the 17 windows was zero, and so the value of R was undefined.

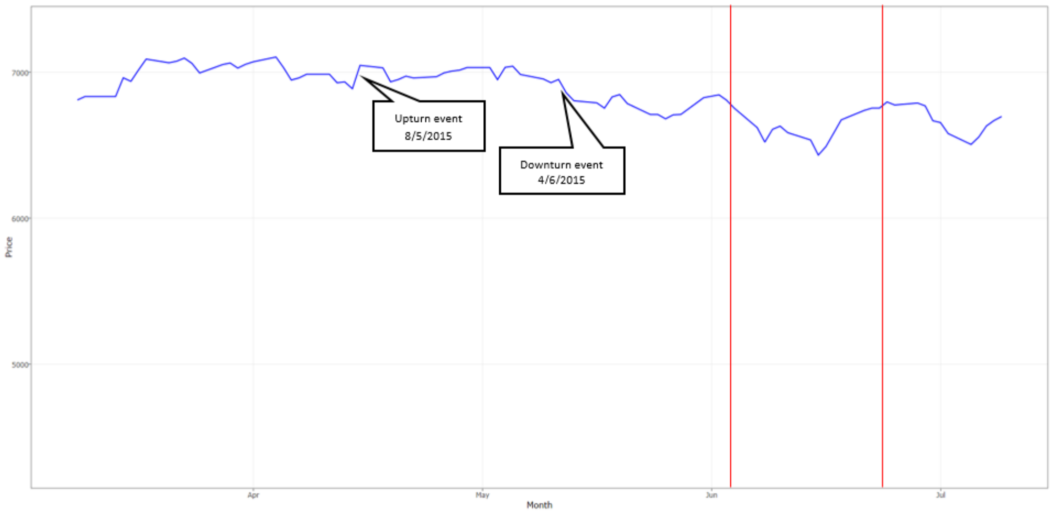


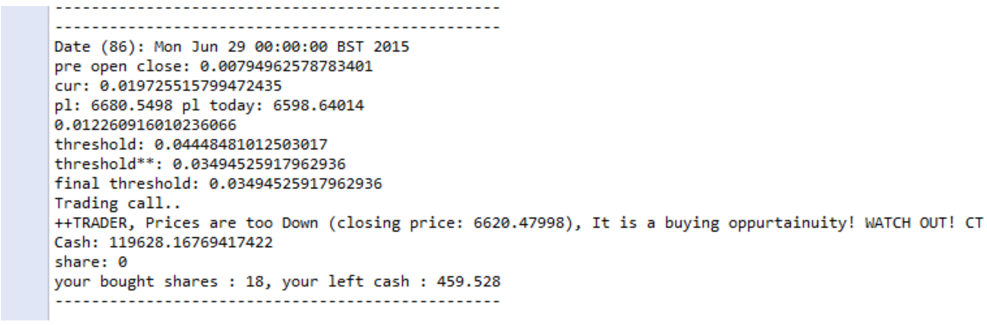
Figure 7.5: FTSE 100 During the Greece crisis 2015 (vertical red lines indicate the investigated period)

X	Y	X*Y	X*X	Y*Y
9294	0	0	86378436	0
19989	0	0	399560121	0
6888	0	0	47444544	0
3229	0	0	10426441	0
9287	0	0	86248369	0
1903	0	0	3621409	0
11241	0	0	126360081	0
6258	0	0	39162564	0
2219	0	0	4923961	0
6458	0	0	41705764	0
6058	0	0	36699364	0
17065	0	0	291214225	0
20083	0	0	403326889	0
5357	0	0	28697449	0
4695	0	0	22043025	0
3306	0	0	10929636	0
3143	0	0	9878449	0

Table 7.5: The values of X and Y in the Greece crisis 2015 period, where X is the Number of Tweets and Y is the Number of Detected DC Events

[3] http://www.bbc.co.uk/news/business-33307810

[4] http://www.bbc.co.uk/news/business-33405583



```
------------------------------------------------
------------------------------------------------
Date (86): Mon Jun 29 00:00:00 BST 2015
pre open close: 0.00794962578783401
cur: 0.019725515799472435
pl: 6680.5498 pl today: 6598.64014
0.012260916010236066
threshold: 0.04448481012503017
threshold**: 0.03494525917962936
final threshold: 0.03494525917962936
Trading call..
++TRADER, Prices are too Down (closing price: 6620.47998), It is a buying oppurtainuity! WATCH OUT! CT
Cash: 119628.16769417422
share: 0
your bought shares : 18, your left cash : 459.528
------------------------------------------------
```

Figure 7.6: Trading action taken place on 29/6/2015 as prices go sharply down

The final correlation calculation, applied to the FTSE 100 price stream and the Twitter stream, was one for which X was set to be the number of events detected from the text stream (Twitter) each day, and Y was set to be the number of DC events detected from the time-series data stream (FTSE 100 prices stream) each day — see Table 7.6. To this end, we mainly wanted to find out whether the detection of an event from one stream may be associated with the detection of an event from another stream. The Pearson Correlation (R) for the GE 2015 period was 0.62, which means a moderate positive relationship between the number of detected events in Twitter stream and the number of detected events in the FTSE 100 stream exists. Again it was not possible to calculate the Pearson correlation for the Greece crisis period as no DC events were detected for that.

Mainly we used the correlation coefficient to assess whether there is a positive linear correlation between events detected from the time-series data stream and events detected from the text stream. We wanted to see if the detection of an event from one stream was associated with an increased probability of detecting an event on the other stream. We found a moderate positive linear relationship ($R = 0.62$) between events detected from a Twitter stream (the GE 2015 tweets stream) and those detected from the FTSE 100 price stream in the GE 2015 period; this means that events from these streams may have a moderate positive correlation with each other and that they generally move in the same direction. However for the period of the Greece crisis, we could not apply the Pearson correlation test as no DC events were detected. Thus this means the correlation test can not be applied if no events are detected from any of the investigated data streams.

Overall, no significant correlation was found between the events detected from the financial time-series stream and those detected from the Twitter stream. Hence, the conclusion that we can draw from this investigation is there may be no direct correlation,

X	Y	X*Y	X*X	Y*Y
2	0	0	4	0
2	0	0	4	0
1	0	0	1	0
0	0	0	0	0
1	0	0	4	0
0	0	0	0	0
1	0	0	1	0
0	0	0	0	0
1	0	0	1	0
0	0	0	0	0
3	1	3	9	1
1	0	0	1	0
1	0	0	1	0
1	0	0	1	0
0	0	0	0	0
1	0	0	1	0
0	0	0	0	0
2	1	2	4	1
1	0	0	1	0
1	0	0	1	0
1	0	0	1	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0

Table 7.6: The values of X and Y in the GE2015 period, Where X is the Number of Detected Events from Twitter Stream and Y is the Number of Detected DC Events

or an easy to discern direct relationship between the two streams. Although we cannot draw this as a definitive conclusion, our examination suggests that in general it is not easy to identify such correlations.

7.3 Finding Possible Inter-links Using Granger Causality

Granger causality [148] is a statistical method for studying causal links between random variables. Specifically, it is applied in order to assess whether there is any potential predictability power in terms of one indicator for the other. Hence it is a method for investigating the flow of information between time-series; it said that X Granger causes Y, if time-lagged values of X are significant in helping to predict values of Y.

We applied the Granger causality test, not for testing true causality and prediction, but for measuring the precedence and information content of one variable in relation to another, if any such relation exists. Essentially, we wanted to see if events detected from the time-series data streams, using the DC approach with our dynamic threshold, precede events detected from the text streams, using the FPM approach and a dynamic support value, or vice versa.

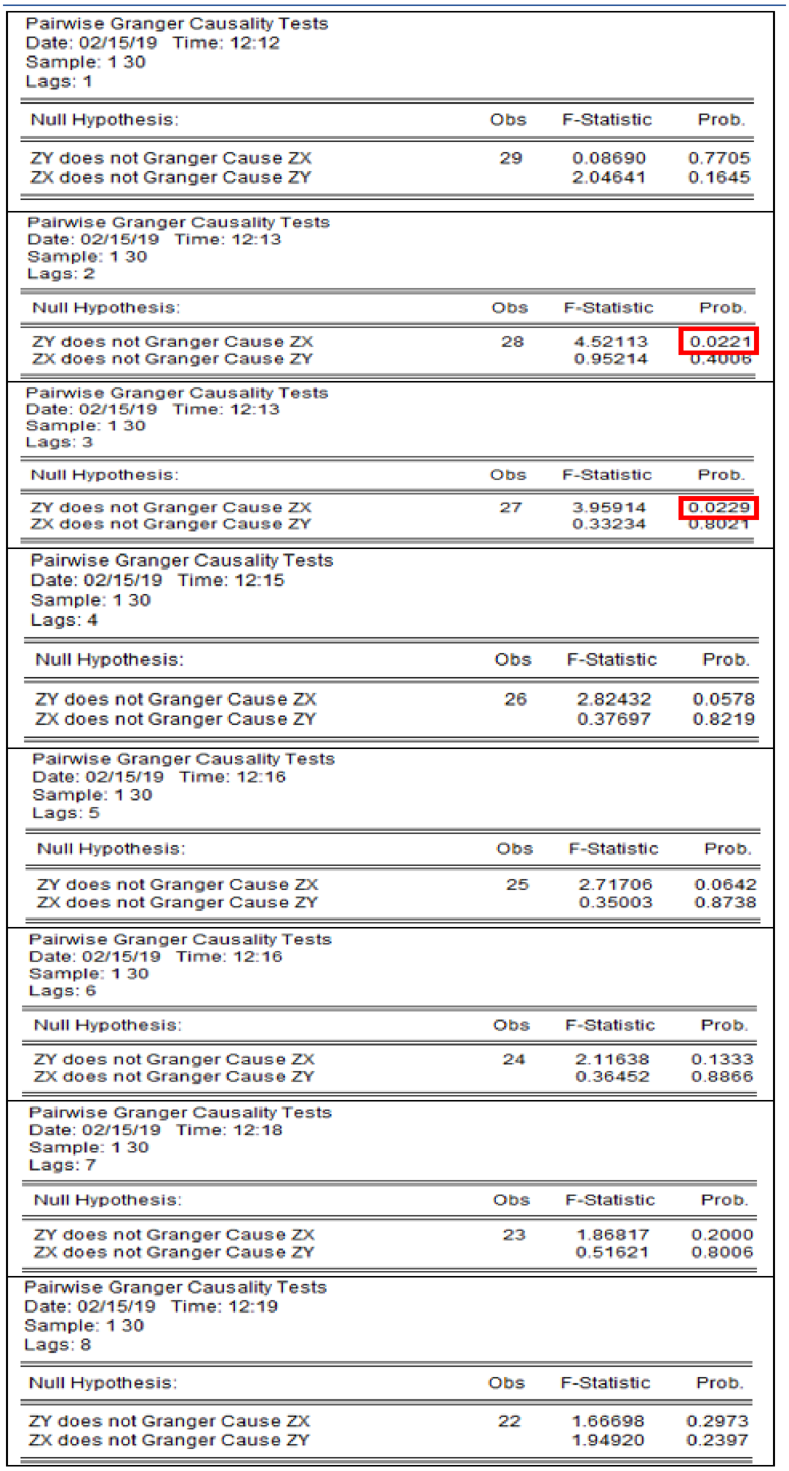
So, we wanted to apply the Granger causality test and see whether we could accept or reject the null hypothesis. The null hypothesis was that X does not Granger cause Y, and that Y does not Granger cause X. The standard significant *p value* level used to accept or reject the null hypothesis is Fisher's proposed level of 5% [193].

The following Granger causality analysis (named GC_1) was undertaken in relation to events detected from the text stream (Twitter stream) and events detected from the time-series data stream (FTSE 100 prices stream) in the GE 2015 period. Refer to Equation 7.2, where X_t represents the detected Twitter stream events on day$_t$, Y_t represents the detected time-series data stream events on day$_t$, and n represents the maximum time-lagged values. The ε_t is the random error rate for day$_t$, α, β_i, and λ_i are the regression parameters. Table 7.6 shows the values of X and Y for all 29 windows in the GE 2015 period.

$$GC = \alpha + \sum_{i=1}^{n} \beta_i X_{t-i} + \sum_{i=1}^{n} \lambda_i Y_{t-i} + \varepsilon_t \tag{7.2}$$

The Econometrics Views (Eviews 10) [194], which is a statistical package, was used for applying the Granger causality test. As a preliminary step, non-stationary time-series must be converted to stationary time series, as the Granger causality test is sensitive to non-stationary time series [195]. In a stationary time-series, all statistical properties remain constant over time.

The GC_1 Granger causality test was applied using a number of different time-lag values. Refer to Figure 7.7 to see the Granger causality results relating to time-lags from 1 to 8 days. With a time-lag of 2 and 3, the first hypothesis was rejected, as the p value equals 0.022, which is less than %5 (framed in red), and so the financial stream events do, relatively, Granger cause the text stream events —but the opposite is not the case.

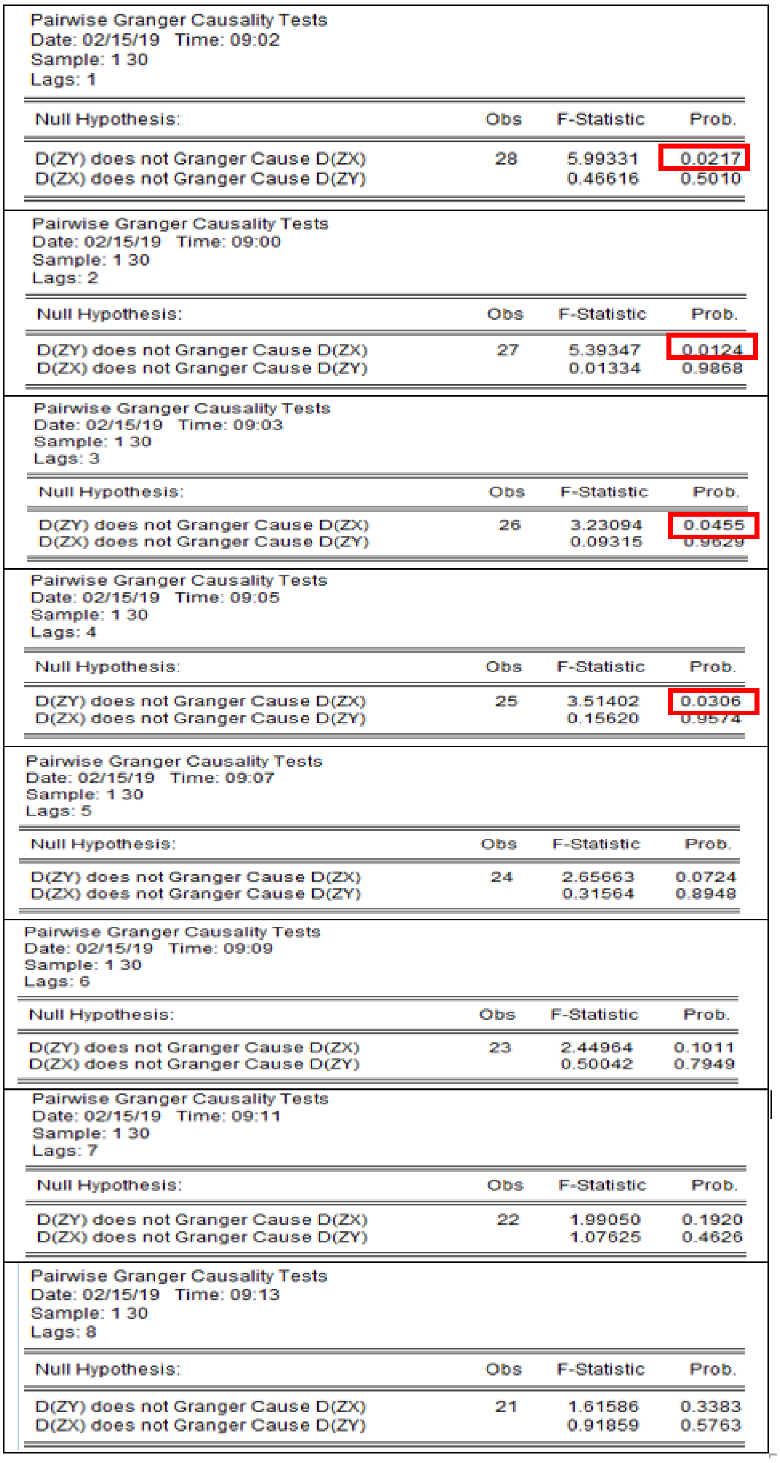


```
Pairwise Granger Causality Tests
Date: 02/15/19   Time: 12:12
Sample: 1 30
Lags: 1

 Null Hypothesis:                        Obs    F-Statistic    Prob.

 ZY does not Granger Cause ZX             29     0.08690      0.7705
 ZX does not Granger Cause ZY                    2.04641      0.1645

Pairwise Granger Causality Tests
Date: 02/15/19   Time: 12:13
Sample: 1 30
Lags: 2

 Null Hypothesis:                        Obs    F-Statistic    Prob.

 ZY does not Granger Cause ZX             28     4.52113      0.0221
 ZX does not Granger Cause ZY                    0.95214      0.4006

Pairwise Granger Causality Tests
Date: 02/15/19   Time: 12:13
Sample: 1 30
Lags: 3

 Null Hypothesis:                        Obs    F-Statistic    Prob.

 ZY does not Granger Cause ZX             27     3.95914      0.0229
 ZX does not Granger Cause ZY                    0.33234      0.8021

Pairwise Granger Causality Tests
Date: 02/15/19   Time: 12:15
Sample: 1 30
Lags: 4

 Null Hypothesis:                        Obs    F-Statistic    Prob.

 ZY does not Granger Cause ZX             26     2.82432      0.0578
 ZX does not Granger Cause ZY                    0.37697      0.8219

Pairwise Granger Causality Tests
Date: 02/15/19   Time: 12:16
Sample: 1 30
Lags: 5

 Null Hypothesis:                        Obs    F-Statistic    Prob.

 ZY does not Granger Cause ZX             25     2.71706      0.0642
 ZX does not Granger Cause ZY                    0.35003      0.8738

Pairwise Granger Causality Tests
Date: 02/15/19   Time: 12:16
Sample: 1 30
Lags: 6

 Null Hypothesis:                        Obs    F-Statistic    Prob.

 ZY does not Granger Cause ZX             24     2.11638      0.1333
 ZX does not Granger Cause ZY                    0.36452      0.8866

Pairwise Granger Causality Tests
Date: 02/15/19   Time: 12:18
Sample: 1 30
Lags: 7

 Null Hypothesis:                        Obs    F-Statistic    Prob.

 ZY does not Granger Cause ZX             23     1.86817      0.2000
 ZX does not Granger Cause ZY                    0.51621      0.8006

Pairwise Granger Causality Tests
Date: 02/15/19   Time: 12:19
Sample: 1 30
Lags: 8

 Null Hypothesis:                        Obs    F-Statistic    Prob.

 ZY does not Granger Cause ZX             22     1.66698      0.2973
 ZX does not Granger Cause ZY                    1.94920      0.2397
```

Figure 7.7: The Granger causality GC_1 test results for 1-8 lags, where only lags 2 and 3 rejects the null hypothesis in the first regression (framed in red)

Another Granger causality analysis (GC_2) was undertaken between the number of tweets posted daily (the daily tweets' volume) on the text stream and the daily FTSE 100 prices from the time-series data stream. Refer to Equation 7.2 where X_t represents the number of tweets (the tweets' volume) on day$_t$, Y_t represents the FTSE 100 price as streamed on day$_t$ (Table 7.2 shows the values of X and Y for all 29 windows), and n represents the maximum time-lagged values. We wanted to see whether the FTSE 100 price changes Granger cause the tweets' volume or the other way around, (that is, the daily posted tweets' volume Granger causes the FTSE 100 price changes).

The GC_2 Granger causality test was applied using a number of different time-lag values (from 1-8 days). Refer to Figure 7.8 to see a summary of the GC_2 tests related to time-lags of from 1 to 8 days. The first hypothesis was rejected with the first four time-lag values (from 1-4 days), as the p value was less than %5 (framed in red), and so the FTSE 100 daily price does Granger cause the Twitter stream daily tweet volume with 1, 2, 3, and 4 days lag — but the opposite is not the case.

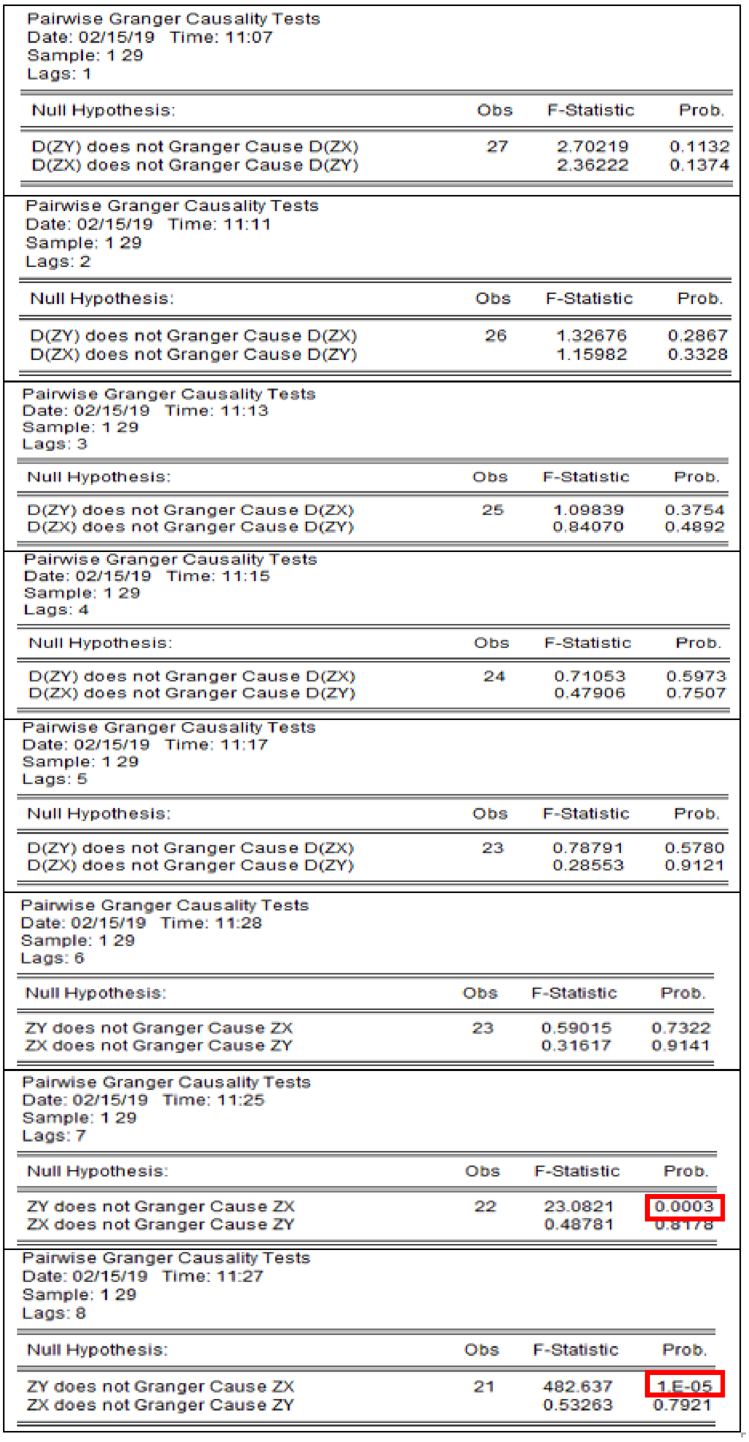


```
Pairwise Granger Causality Tests
Date: 02/15/19   Time: 09:02
Sample: 1 30
Lags: 1

 Null Hypothesis:                          Obs    F-Statistic    Prob.

 D(ZY) does not Granger Cause D(ZX)         28     5.99331      0.0217
 D(ZX) does not Granger Cause D(ZY)                0.46616      0.5010

Pairwise Granger Causality Tests
Date: 02/15/19   Time: 09:00
Sample: 1 30
Lags: 2

 Null Hypothesis:                          Obs    F-Statistic    Prob.

 D(ZY) does not Granger Cause D(ZX)         27     5.39347      0.0124
 D(ZX) does not Granger Cause D(ZY)                0.01334      0.9868

Pairwise Granger Causality Tests
Date: 02/15/19   Time: 09:03
Sample: 1 30
Lags: 3

 Null Hypothesis:                          Obs    F-Statistic    Prob.

 D(ZY) does not Granger Cause D(ZX)         26     3.23094      0.0455
 D(ZX) does not Granger Cause D(ZY)                0.09315      0.9629

Pairwise Granger Causality Tests
Date: 02/15/19   Time: 09:05
Sample: 1 30
Lags: 4

 Null Hypothesis:                          Obs    F-Statistic    Prob.

 D(ZY) does not Granger Cause D(ZX)         25     3.51402      0.0306
 D(ZX) does not Granger Cause D(ZY)                0.15620      0.9574

Pairwise Granger Causality Tests
Date: 02/15/19   Time: 09:07
Sample: 1 30
Lags: 5

 Null Hypothesis:                          Obs    F-Statistic    Prob.

 D(ZY) does not Granger Cause D(ZX)         24     2.65663      0.0724
 D(ZX) does not Granger Cause D(ZY)                0.31564      0.8948

Pairwise Granger Causality Tests
Date: 02/15/19   Time: 09:09
Sample: 1 30
Lags: 6

 Null Hypothesis:                          Obs    F-Statistic    Prob.

 D(ZY) does not Granger Cause D(ZX)         23     2.44964      0.1011
 D(ZX) does not Granger Cause D(ZY)                0.50042      0.7949

Pairwise Granger Causality Tests
Date: 02/15/19   Time: 09:11
Sample: 1 30
Lags: 7

 Null Hypothesis:                          Obs    F-Statistic    Prob.

 D(ZY) does not Granger Cause D(ZX)         22     1.99050      0.1920
 D(ZX) does not Granger Cause D(ZY)                1.07625      0.4626

Pairwise Granger Causality Tests
Date: 02/15/19   Time: 09:13
Sample: 1 30
Lags: 8

 Null Hypothesis:                          Obs    F-Statistic    Prob.

 D(ZY) does not Granger Cause D(ZX)         21     1.61586      0.3383
 D(ZX) does not Granger Cause D(ZY)                0.91859      0.5763
```

Figure 7.8: The Granger causality GC_2 results for 1-8 lags, where lags 1,2,3 and 4 rejects the null hypothesis in the first regression (framed in red)

A third Granger causality analysis (GC_3) was undertaken in relation to the number of tweets posted daily (daily tweets' volume) on the text stream and the DC events detected from the time-series data stream. Refer to Equation 7.2,where X_t represents the number of tweets posted on day$_t$ (tweets' volume),Y_t represents the number of DC detected events on day$_t$ (Table 7.4 shows the values of X and Y for all 29 windows), and n represents the maximum time-lagged values. We wanted to see if the detection of DC events Granger causes the tweets' volume, or the other way around — the number of tweets posted daily Granger causes the detection of DC events.

The GC_3 Granger causality test was applied using a number of different time-lag values (from 1-8 days): Figure 7.9 shows a summary of the results. For time-lags of 7 and 8 days the first hypothesis is rejected since the p value is less than 5% (framed in red in Figure 7.9). Hence, the occurrence of DC events may Granger cause tweets volume, with 7 and 8 days lags but not the other way around.

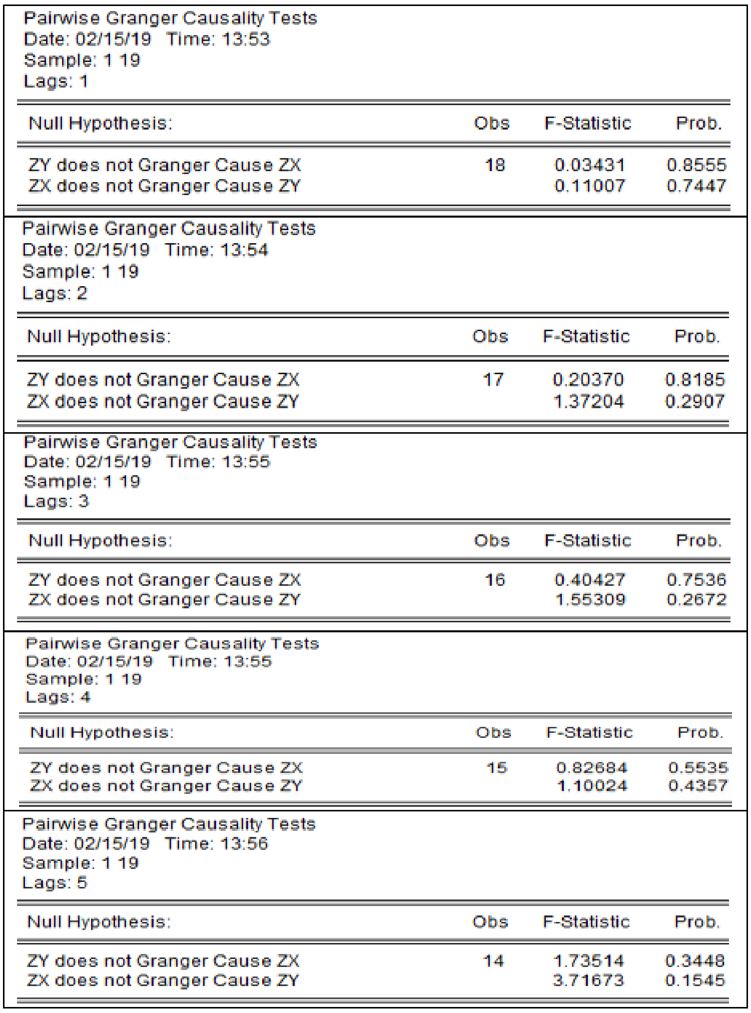


```
Pairwise Granger Causality Tests
Date: 02/15/19   Time: 11:07
Sample: 1 29
Lags: 1

 Null Hypothesis:                              Obs    F-Statistic    Prob.

 D(ZY) does not Granger Cause D(ZX)             27      2.70219     0.1132
 D(ZX) does not Granger Cause D(ZY)                     2.36222     0.1374

Pairwise Granger Causality Tests
Date: 02/15/19   Time: 11:11
Sample: 1 29
Lags: 2

 Null Hypothesis:                              Obs    F-Statistic    Prob.

 D(ZY) does not Granger Cause D(ZX)             26      1.32676     0.2867
 D(ZX) does not Granger Cause D(ZY)                     1.15982     0.3328

Pairwise Granger Causality Tests
Date: 02/15/19   Time: 11:13
Sample: 1 29
Lags: 3

 Null Hypothesis:                              Obs    F-Statistic    Prob.

 D(ZY) does not Granger Cause D(ZX)             25      1.09839     0.3754
 D(ZX) does not Granger Cause D(ZY)                     0.84070     0.4892

Pairwise Granger Causality Tests
Date: 02/15/19   Time: 11:15
Sample: 1 29
Lags: 4

 Null Hypothesis:                              Obs    F-Statistic    Prob.

 D(ZY) does not Granger Cause D(ZX)             24      0.71053     0.5973
 D(ZX) does not Granger Cause D(ZY)                     0.47906     0.7507

Pairwise Granger Causality Tests
Date: 02/15/19   Time: 11:17
Sample: 1 29
Lags: 5

 Null Hypothesis:                              Obs    F-Statistic    Prob.

 D(ZY) does not Granger Cause D(ZX)             23      0.78791     0.5780
 D(ZX) does not Granger Cause D(ZY)                     0.28553     0.9121

Pairwise Granger Causality Tests
Date: 02/15/19   Time: 11:28
Sample: 1 29
Lags: 6

 Null Hypothesis:                         Obs    F-Statistic    Prob.

 ZY does not Granger Cause ZX              23      0.59015     0.7322
 ZX does not Granger Cause ZY                      0.31617     0.9141

Pairwise Granger Causality Tests
Date: 02/15/19   Time: 11:25
Sample: 1 29
Lags: 7

 Null Hypothesis:                         Obs    F-Statistic    Prob.

 ZY does not Granger Cause ZX              22      23.0821     0.0003
 ZX does not Granger Cause ZY                      0.48781     0.8178

Pairwise Granger Causality Tests
Date: 02/15/19   Time: 11:27
Sample: 1 29
Lags: 8

 Null Hypothesis:                         Obs    F-Statistic    Prob.

 ZY does not Granger Cause ZX              21      482.637     1.E-05
 ZX does not Granger Cause ZY                      0.53263     0.7921
```

Figure 7.9: The Granger causality GC_3 results for 1-8 lags, where lags 7 and 8 rejects the null hypothesis in the first regression (framed in red)

The Granger causality test was not applied to the period of the Greece crisis since no DC events were detected for that period, neither GC_1 nor GC_3 were applicable. However we attempted to see if the FTSE 100 prices stream Granger caused the number of tweets posted daily (the daily tweets' volume), which is the GC_2, as it did for the GE 2015 period. Refer to Figure 7.10 to see the Granger causality test results for time-lags from 1-

5 days (the maximum time-lag was 5 only since the number of observations (the number of windows) for the Greece crisis period were less than the number of observations for the GE period). Here, we could not reject the null hypothesis, which means that the FTSE 100 daily prices do not Granger cause the number of tweets posted daily (in contrast to the results for the GE period). An interpretation of this result could be that this is because the UK GE was a regional event and so is more closely tied to the FTSE 100 index, while the Greece crisis, on the other hand, was a European matter and was not much connected or tied to the FTSE 100.



```
Pairwise Granger Causality Tests
Date: 02/15/19   Time: 13:53
Sample: 1 19
Lags: 1

Null Hypothesis:                    Obs    F-Statistic    Prob.

ZY does not Granger Cause ZX         18      0.03431     0.8555
ZX does not Granger Cause ZY                 0.11007     0.7447

Pairwise Granger Causality Tests
Date: 02/15/19   Time: 13:54
Sample: 1 19
Lags: 2

Null Hypothesis:                    Obs    F-Statistic    Prob.

ZY does not Granger Cause ZX         17      0.20370     0.8185
ZX does not Granger Cause ZY                 1.37204     0.2907

Pairwise Granger Causality Tests
Date: 02/15/19   Time: 13:55
Sample: 1 19
Lags: 3

Null Hypothesis:                    Obs    F-Statistic    Prob.

ZY does not Granger Cause ZX         16      0.40427     0.7536
ZX does not Granger Cause ZY                 1.55309     0.2672

Pairwise Granger Causality Tests
Date: 02/15/19   Time: 13:55
Sample: 1 19
Lags: 4

Null Hypothesis:                    Obs    F-Statistic    Prob.

ZY does not Granger Cause ZX         15      0.82684     0.5535
ZX does not Granger Cause ZY                 1.10024     0.4357

Pairwise Granger Causality Tests
Date: 02/15/19   Time: 13:56
Sample: 1 19
Lags: 5

Null Hypothesis:                    Obs    F-Statistic    Prob.

ZY does not Granger Cause ZX         14      1.73514     0.3448
ZX does not Granger Cause ZY                 3.71673     0.1545
```

Figure 7.10: The Granger causality GC_2 results for 1-5 lags in the Greece crisis 2015 period

In summary, a low Granger causality was found between the daily time-series data stream events and the text stream events (and only where there was a 2-3 day time-lag). Thus no statistically significant evidence was found in terms of the Granger causality test for there being a causal link between events detected from the financial stream and events detected from the text stream.

The Granger causality tests (GC_2 and GC_3) were carried out in order to gain more insights into both streams, the time-series data stream and the text stream, and in order to see which stream might be an indicator for the other. Similarly, a relatively low Granger causality was found in, GC_3, between the daily financial stream events (DC events) and the daily tweets' volume (with 7 and 8 day time-lags only). This means there is no clear indication of precedence or derivation from one time-series to the other. However in GC_2, a significant dependence was found between the FTSE 100 daily price change and the number of tweets posted daily (tweets' volume), with time-lags of 1, 2, 3, and 4 days.

7.4 Summary

In this chapter, we tried to bring together and draw inferences from the two data streams: the text (social network) stream and the price time-series data stream. We applied the Pearson correlation and the Granger causality tests to investigate the relationships between the events detected across both data streams. In the text stream, we were looking for text-posts relating to a particular major event (such as the GE 2015 or the Greece crisis 2015), not for text-posts relating to the financial market data under examination. Basically, we were curious about the relation between the identified events from the text streams and the price transitions taking place in the time-series data stream.

A relatively low Pearson correlation and Granger causality was found between events detected from the price time-series data stream and events detected from the text stream. One of the main reasons for this low percentage was that the text stream is not primarily focused on the market data under examination; instead it is mostly focused on a major event that is happening.

Another reason was a general issue which we found in relation to price time-series data stream events: such events come in only two forms, a price increase or a price decrease, and these kinds of events alternate (i.e., no downturn event follows a downturn event). This is because, if a price decrease is spotted, and then after some time the prices goes down even further, no additional price decrease event is identified (such was the case with the FTSE 100 index price during the Greece crisis period). While in the text stream, on the other hand, events are detected without any consideration of previous events.

In addition, another issue with respect to price time-series data streams is that during a period of investigation, a number of different events may take place that are not necessarily related to the same major (financial or other) event. In other words, different shares or assets may be affected by different influences. This is unlike the text stream from which we retrieve only text-posts related to the investigated event, and so only related events (topics) are identified. In the GE 2015 time period for example, more precisely on the 29th April (just 8 days prior to the election day), the FTSE 100 index went down by 1.1% because of cautiousness ahead of an interest rate policy meeting at the US Federal Reserve [1], which was not related to the GE 2015. And so linking over two different streams (a text stream related to a particular major event and a price time-series data stream reporting on particular stocks or assets), and finding relations between the events detected from both streams is not always possible.

In summary, we could not clearly identify a relationship between the investigated data streams. There was no concrete relation or link found between the streams considered (the text stream and the price time-series data stream) in terms of the events detected from each of them. This was in-line with the work in [136], where an event from Twitter is said to be found if tweets volume peak is identified. The identified events were related to stock prices using Correlation coefficient and Granger causality. They found a low Pearson correlation and Granger causality across both time series for the investigated time period. In addition, a recent work in [196], found that a correlation between the public sentiment of local elections in Twitter and the FTSE 100 movements does exist but is not determined as statistically significant, the same was also found for evidence of causation.

Chapter 8

Conclusions and Future Work

A data stream is a continuous flow of data, produced at high and irregular rates. The size of a data stream is unbounded; streams are of potentially infinite length [8]. This thesis has mainly focused on developing event detection methods so that they can effectively operate on streams of data. The principal goal was to develop and extend methods for detecting events for structured and unstructured data streams. The novelty of the work presented lies in the way that the dynamic and changing nature of data streams is adopted in order to develop event identification methods.

In the first section of this chapter, we summarize the work that we have undertaken and the contributions we believe we have made. Then, the second section will examine some potential areas for future research – which can be extensions of this work.

8.1 Summary

In this thesis, we investigated the challenge of event identification from different streams of data. In particular, and motivated by the need to identify the occurrence of events in the domain of finance, which is a dynamic and volatile domain that is typically affected by a range of events, we investigated and developed event identification methods for text and financial time-series data streams. Research main objectives listed in Chapter 1 are revisited in Sub-section 8.1.1. In addition, the contributions of this work have been described in the previous chapters are summarised in Sub-section 8.1.2.

8.1.1 Objectives

Three main research objectives were addressed as following.

- Event detection from unstructured data streams was achieved by proposing the Dynamic-FPM (D-FPM) framework which is based on employing a Frequent Pattern Mining (FPM) method with a dynamic defined support value instead of having a fixed given one. The proposed dynamic support measures the co-occurrence and frequency of terms and better fits with the changing nature of text data streams.

- Event detection from structured data streams was addressed by employing the Directional Change (DC) approach on a structured time-series data stream using a daily dynamic defined threshold value in order replace the use of a fixed, a-priori threshold. The dynamic defined threshold has led to more effective detecting of events (i.e. significant price changes) than a fixed threshold value that is used throughout the whole stream (i.e. does not change).

- Cross-reference across various data streams in order to explore and further investigate the relationship between events identified from different data streams (the structured and unstructured data streams). In general, a relatively low relationship was found between events identified from price time-series stream and text (Twitter) stream.

8.1.2 Contributions

We started our investigation thorough a literature review, in Chapter 2, of the domain of event identification and we identified a number of drawbacks and some problems with existing techniques that have informed this work.

In Chapter 3, we developed and extended a method for detecting events from the data yielded by unstructured text streams, using a FPM method. Applying FPM methods on text streams requires setting the term selection criteria, in other words, it requires setting of the support value. In [4], the authors set the minimum support value to a fixed value ($\sigma = 0.03$). In [39, 41], the number of selected terms K was given in advance by the users, and the selection of K terms was based on a reference independent corpus. In [40] on the other hand, they used a very small fixed support value, but then they limited the number of identified topics to a number given as an input. The authors argue of that it was easier for users to indicate how many patterns they would like to see than specifying a minimum support threshold. However, we think that in dynamic and changeable environments, such as data streams, it is very hard to specify the number of selected terms nor the number of identified patterns in advance, hence we introduced a dynamic support definition method (the D-FPM).

We employed the FP-Growth algorithm [3] on a social network stream (Twitter) to identify events/topics which occurred from each window-batch, if any. In addition, we dynamically defined the support value, which is a metric for terms retrieval in a text stream. Generally, the support value for the FP-growth algorithm has been defined a-priori, as a fixed value. Our dynamic support value directly replaces this fixed one. This change allowed our event detection framework to cope better with the nature of data streams. A different support value is assigned to each window-batch; the support value is defined mainly based on the keyword co-occurrences and frequencies. In fact, we proposed and tested two dynamic support definition methods: which one should be selected for a particular batch depends on the window sizes (i.e., the numbers of tweets) which are likely to be encountered. A logistic regression model was built to classify each incoming window-batch for being either large or small sized window.

We collected tweets which were related to major events (the UK General Elections 2015, and the Greece Crisis 2015). We employed the FP-Growth algorithm, using our daily proposed dynamic support value, to each of these major-event streams. The topics we detected were evaluated with respect to what was being published by the major news outlets at the same time. If a topic was found to be mentioned on the same day that a news article was published, then the topic was said to represent a true event – and to be a false or insignificant one otherwise. More than 88% of the topics that were identified using our event detection framework were found on the same day that news articles concerning that topic were released. In addition, we evaluated our event detection framework against the SFPM framework [39], which is also a topic detection framework, using the GE 2015 stream. For this comparison, we applied the precision, recall, and F-measure [168–170] evaluation metrics. The results showed that our event detection framework (the D-FPM) achieved a precision approximately three times greater than that achieved by the SFPM framework. Recall was about the same for both frameworks. The F-measure, which is a combination of both precision and recall, was around two times higher for the D-FPM framework than for the SFPM framework.

In Chapter 4, we showed how we developed a method for detecting events from structured time-series data streams; in relation to this, we used a high frequency time-series data stream (stock prices). We introduced a dynamic threshold definition algorithm in order to replace the use of a fixed a-priori threshold in the DC approach. The DC approach [35] is an approach for summarizing market price movements. An event according to DC is detected if the price change between two points exceeds or is below (depending on the event being either a downturn or an upturn) the given threshold value. The DC approach has always been used with fixed thresholds [6, 36–38, 42–50, 134]. Hence, this work is a novel contribution as it shows how the DC approach can be used on streams of data with a dynamic threshold which makes it more flexible and suitable for dynamic and volatile environments that markets are.

The dynamic threshold is defined on a daily basis (after receiving the current day's opening prices) to be compatible with the operation of the stock market, which runs on week days from 8 am – 4:30 pm. The threshold, essentially, is set on the basis of the previous day's prices and the overnight price transitions. Depending on the previous day and/or overnight price change the suitable dynamic threshold definition method is used. A decision tree was built to set the value of the previous day and overnight price changes.

We performed experiments on the FTSE 100 minute-by-minute price stream, attempting to detect the occurrence of DC events using different fixed thresholds and also using our daily dynamically calculated one. The detected events were evaluated against what was published on the same day on the BBC News regarding the FTSE 100 index. If a detected event was found on the same day that a news headline was published, then it was said to be a "true" event, otherwise it was said to be a spurious or false event.

The results showed that our dynamic threshold leads to a more accurate identification of "true" events than the various different fixed threshold values that we tried. The daily dynamic thresholds used with the DC approach were able to facilitate the detection of DC events of various different magnitudes; such a criteria is not applicable to event detection via fixed thresholds, and so the use of a dynamic threshold is clearly an improvement.

From the litreture we noticed that finding the most profitable threshold value has always been an issue when using the DC approach when trading [37, 38, 52]. Thus, to further evaluate the use of our dynamic threshold, in Chapter 5, we proposed a trading strategy based on the DC approach and our dynamically defined threshold, named the Dynamic Threshold-Trading Strategy (DT-TS). We found that, using a dynamic threshold that is calculated daily, we were able to more effectively (i.e. in a more timely manner) spot price changes. Once a DC event is detected, and with every price increase/decrease, the DT-TS trading rule is opened up. The trading action to be taken (either buying or selling) depends on the previous day's price transitions. The DT-TS is flexible and can cope with price fluctuation, hence when price transitions are encountered, it considers the previous day price change as an indicator for the type of trading to be taken (either buying or selling). A trading rule is closed when the next DC event is confirmed.

An experiment was conducted on the FTSE 100 minute-by-minute prices stream to evaluate the DT-TS trading rules using different fixed thresholds as well as different trading strategies. The DT-TS outperformed (achieved better profitability than) the use of fixed thresholds and all the other trading strategies investigated.

Furthermore, in Chapter 6 we explored the functionality of the proposed trading strategy (the DT-TS) using another alternative data stream with a different level of data flow. Thus, instead of using a higher frequency data stream (minute-by-minute), we tried a lower frequency one (day-by-day).

An experiment was performed on the FTSE 100 day-by-day prices stream in order to evaluate the DT-TS trading rules which we had constructed. For comparison purposes, we tested various different fixed thresholds as well as various different trading strategies against the DT-TS strategy using dynamically calculated thresholds. We found that the DT-TS performs almost equally as well (from a profits point of view) as a strategy using a low fixed threshold. In addition, when comparing the streams we were using, the daily stream and the minute-by-minute stream, we concluded that the DT-TS performs better (gains higher profits) when the higher frequency stream, the minute-by-minute stream rather than the day-by-day stream, is used.

In Chapter 7, we were interested to know whether we can put together and draw inferences across both streams, the text stream and the price time-series data stream, to see what relationships could be found, if any. We wanted to see if the events detected from the FTSE 100 prices stream had an effect on the events which could be detected from a text stream. The text streams reflected events on both a domestic level (the UK GE 2015) and a European level (the Greece crisis 2015).

Therefore, we applied the Pearson correlation and the Granger causality tests to investigate the relationships between events detected across both data streams. In summary, we found relatively low Pearson correlations and Granger causalities between events detected from the time-series data stream and those detected from the text stream. Our findings were in-line with the findings in [136, 196]

Hence, in conclusion, through the work in this thesis we have shown that methods that are used for event identification on streams of data need to be flexible and be able to cope with the dynamic nature of such streams and the underlying domains.

8.2 Future Work

Although the methods that we have developed can be deployed for event identification on text and financial time-series data streams, our work and methods can be extended in several directions.

In the text stream analysis and event identification, we used the FP-Growth and a dynamic support value which was defined daily (i.e. for each window-batch) to identify the topics/events which occurred. In the future, we would want to rank the identified topics — where more than one topic is identified from a single window-batch. The higher the rank of an event, the more the important the terms it include. Hence, along with the general list of keywords used to retrieve relevant text posts (i.e. tweets) from Twitter API, and in order to adapt to the dynamic nature of social network streams, where tweets are retrieved in almost real-time. An additional list of keywords can be created by including new emerging terms from the high rank identified events from previous batches and can last for some time only, until a new additional list is created. Creating an additional keyword list could help in identifying more true events and so could increase the recall value.

We employed the DC approach in the high frequency time-series stream analysis and event identification phase. This approach summarizes price movements based on a fixed, a-priori threshold value which is used to identify and spot price transitions.

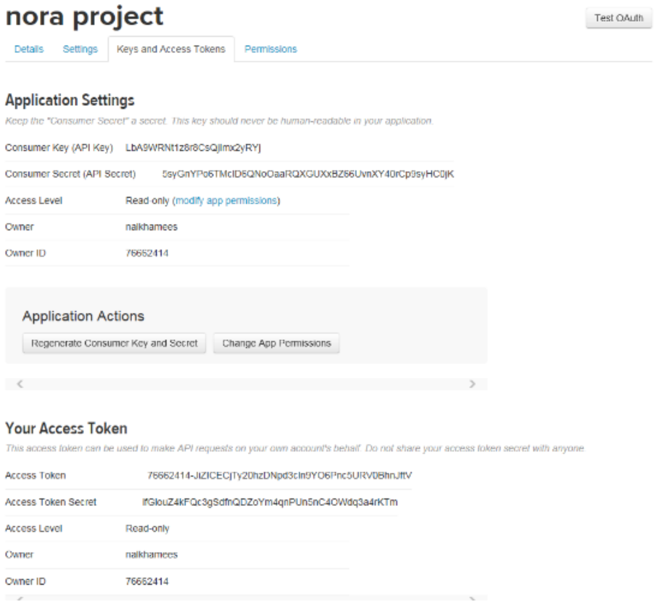
We defined a dynamic threshold to replace the fixed one — in order to make the method more consistent with the nature of data streams. The definition of the dynamic threshold requires a percentage value to be set, during the training phase, which is related to something which has happened the previous day or overnight ($previous - v/overnight - v$). The main source for setting that value was based on the release of news articles on the same day, or not. However, we may want to detect DC events from a high frequency time-series data stream for any individual stock or share, with regard to which major news outlets may not report events because of other more important news items. In addition, some companies may wish to hide their performance from the news outlets. As a result, in order to define the threshold value, we think that we should consider an additional source for setting the values which are taken to indicate that something significant has occurred the previous day or overnight. Looking at another aspect such as the trading volume or the difference between the asking price and bid price (in the case of individual shares) may help in defining the threshold value. In fact, we did not face this issue in our experiments since we considered only the FTSE 100 index as whole, which is the share index of the 100 companies listed in the London Stock Exchange with the highest market capitalisation. Hence the news outlets in the UK (such as the BBC) will always give exposure to the FTSE 100 price fluctuations and associated news.

Also, in terms of the high frequency time-series stream used, the dynamic threshold definition method and our proposed trading strategy, the DT-TS, both essentially relay on the previous day given data. In particular, they depend on the previous day closing and opening prices (previous day and overnight price changes). Thus, in the future, we plan to develop a strategy that can be applied to other financial markets whose nature is different (i.e. does not have opening and closing prices), as the Foreign Exchange Market, which operates 24 hours a day (continuous streaming).

Appendix 1: Tweets Collection Using Twitter API

As 7th of May 2015 was selected to be the British Parliament Election Day, and since this event happens once every five years, as well as being much discussed by all classes of Britain citizens. We have chosen the 2015 British Parliament elections to be the source of our data set. We started collecting tweets posted regarding that event from 15-4-2015 using Twitter API stream.

First, we created a Twitter developer account, then created an application and generate the API keys to allow us connect to Twitter, as shown in the figure A1.1.



Application Settings

Keep the "Consumer Secret" a secret. This key should never be human-readable in your application.

Consumer Key (API Key) LbA9WRNt1z6r6CsQjlmx2yRYj

Consumer Secret (API Secret) 5syGnYPo6TMcID6QNoOaaRQXGUXxBZ66UvnXY40rCp9syHC0jK

Access Level Read-only (modify app permissions)

Owner nalkhamees

Owner ID 76652414

Application Actions

Your Access Token

This access token can be used to make API requests on your own account's behalf. Do not share your access token secret with anyone.

Access Token 76652414-JiZICECjTy20hzDNpd3cln9YO6Pnc5URV0BhnJftV

Access Token Secret lfGlouZ4kFQc3gSdfnQDZoYm4qnPUn5nC4OWdq3a4rKTm

Access Level Read-only

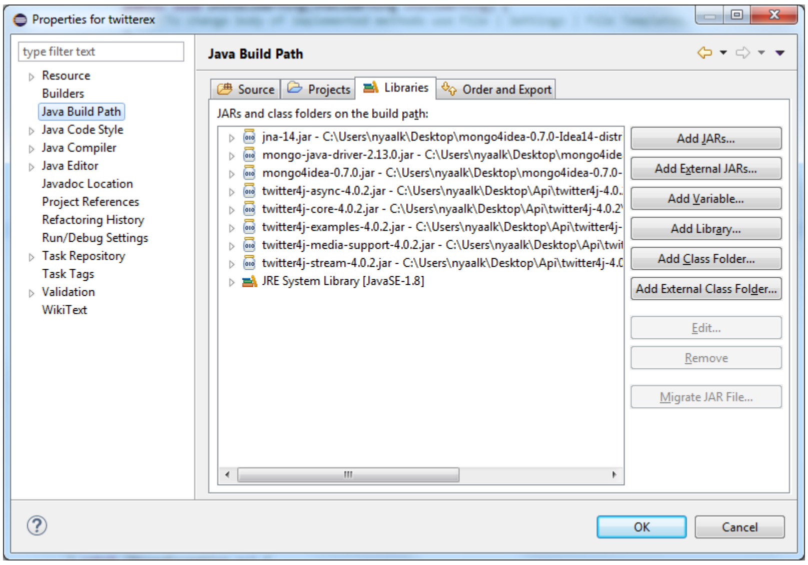
Owner nalkhamees

Owner ID 76652414

Figure A1.1: Twitter Application

Next, we downloaded the Eclipse IDE environment, and added the *Twitter4j* library along with *Mongodb* java driver library to our project, as shown in figure A1.2.

In the same time we installed *Mongodb* for Windows and connect to it using Command Line Prompt. We have to connect to *mongod* server, and specify a path for our

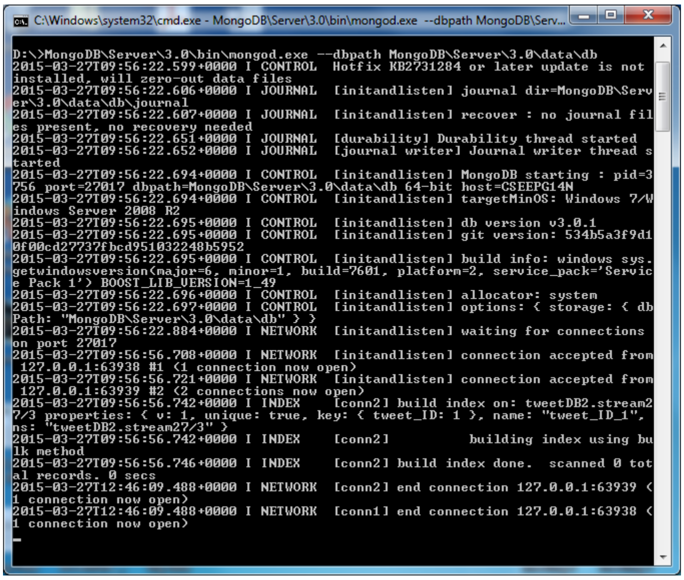


JARs and class folders on the build path:

jna-14.jar - C:\Users\nyaalk\Desktop\mongo4idea-0.7.0-Idea14-distr
mongo-java-driver-2.13.0.jar - C:\Users\nyaalk\Desktop\mongo4ide
mongo4idea-0.7.0.jar - C:\Users\nyaalk\Desktop\mongo4idea-0.7.0-
twitter4j-async-4.0.2.jar - C:\Users\nyaalk\Desktop\Api\twitter4j-4.0.
twitter4j-core-4.0.2.jar - C:\Users\nyaalk\Desktop\Api\twitter4j-4.0.2
twitter4j-examples-4.0.2.jar - C:\Users\nyaalk\Desktop\Api\twitter4j-
twitter4j-media-support-4.0.2.jar - C:\Users\nyaalk\Desktop\Api\twit
twitter4j-stream-4.0.2.jar - C:\Users\nyaalk\Desktop\Api\twitter4j-4.0
JRE System Library [JavaSE-1.8]

Figure A1.2: Add Java Libraries

database as shown in figure A1.3.


C:\Windows\system32\cmd.exe - MongoDB\Server\3.0\bin\mongod.exe --dbpath MongoDB\Serv...

```
D:\>MongoDB\Server\3.0\bin\mongod.exe --dbpath MongoDB\Server\3.0\data\db
2015-03-27T09:56:22.599+0000 I CONTROL  Hotfix KB2731284 or later update is not
installed, will zero-out data files
2015-03-27T09:56:22.606+0000 I JOURNAL  [initandlisten] journal dir=MongoDB\Serv
er\3.0\data\db\journal
2015-03-27T09:56:22.607+0000 I JOURNAL  [initandlisten] recover : no journal fil
es present, no recovery needed
2015-03-27T09:56:22.651+0000 I JOURNAL  [durability] Durability thread started
2015-03-27T09:56:22.652+0000 I JOURNAL  [journal writer] Journal writer thread s
tarted
2015-03-27T09:56:22.694+0000 I CONTROL  [initandlisten] MongoDB starting : pid=3
756 port=27017 dbpath=MongoDB\Server\3.0\data\db 64-bit host=CSEEPG14N
2015-03-27T09:56:22.694+0000 I CONTROL  [initandlisten] targetMinOS: Windows 7/W
indows Server 2008 R2
2015-03-27T09:56:22.695+0000 I CONTROL  [initandlisten] db version v3.0.1
2015-03-27T09:56:22.695+0000 I CONTROL  [initandlisten] git version: 534b5a3f9d1
0f00cd27737fbcd951032248b5952
2015-03-27T09:56:22.695+0000 I CONTROL  [initandlisten] build info: windows sys.
getwindowsversion(major=6, minor=1, build=7601, platform=2, service_pack='Servic
e Pack 1') BOOST_LIB_VERSION=1_49
2015-03-27T09:56:22.696+0000 I CONTROL  [initandlisten] allocator: system
2015-03-27T09:56:22.697+0000 I CONTROL  [initandlisten] options: { storage: { db
Path: "MongoDB\Server\3.0\data\db" } }
2015-03-27T09:56:22.884+0000 I NETWORK  [initandlisten] waiting for connections
on port 27017
2015-03-27T09:56:56.708+0000 I NETWORK  [initandlisten] connection accepted from
 127.0.0.1:63938 #1 (1 connection now open)
2015-03-27T09:56:56.721+0000 I NETWORK  [initandlisten] connection accepted from
 127.0.0.1:63939 #2 (2 connections now open)
2015-03-27T09:56:56.742+0000 I INDEX    [conn2] build index on: tweetDB2.stream2
7/3 properties: { v: 1, unique: true, key: { tweet_ID: 1 }, name: "tweet_ID_1",
ns: "tweetDB2.stream27/3" }
2015-03-27T09:56:56.742+0000 I INDEX    [conn2]          building index using bu
lk method
2015-03-27T09:56:56.746+0000 I INDEX    [conn2] build index done.  scanned 0 tot
al records. 0 secs
2015-03-27T12:46:09.488+0000 I NETWORK  [conn2] end connection 127.0.0.1:63939 (
1 connection now open)
2015-03-27T12:46:09.488+0000 I NETWORK  [conn1] end connection 127.0.0.1:63938 (
1 connection now open)
_
```

Figure A1.3: Connect to MongoDB

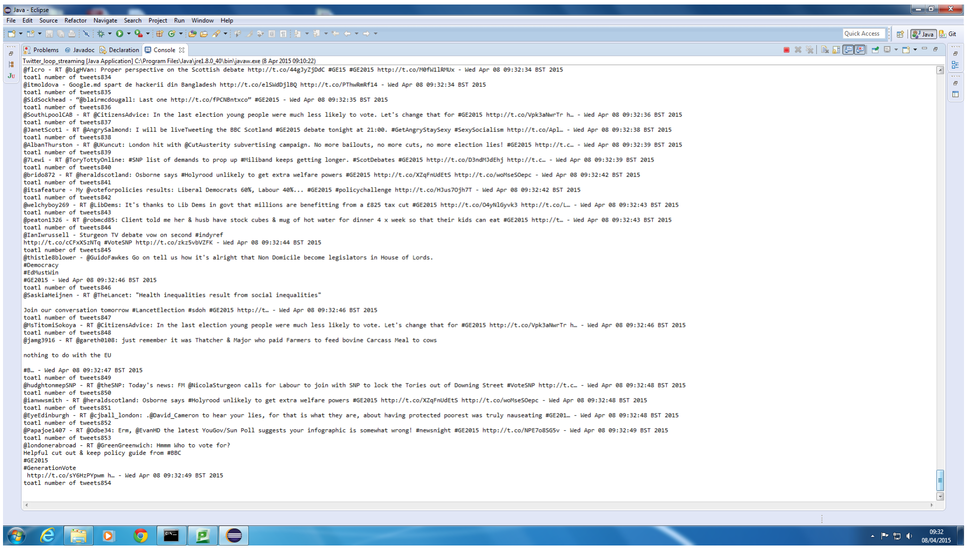
Once we are done from all the preparations, we created a new java project and started by connecting to twitter API stream by setting up the Oath parameters from the created twitter application.

We looked for tweets containing the following keywords "British Elections, GE2015, VoteUkIP2015, GE15", retrieved them and then store them in Mongodb collections.

For every tweet we saved the following parameters:

User name, followers count, retweet count, mention count, time tweet was posted, geographic location of the posted tweet, tweet id, and tweet text. The following figure

(figure A1.4) shows us the collected tweets on the run console.



```
Twitter_loop_streaming [Java Application] C:\Program Files\Java\jre1.8.0_40\bin\javaw.exe (8 Apr 2015 09:10:22)
@flcro - RT @bigHVan: Proper perspective on the Scottish debate http://t.co/44gJyZjDdC #GE15 #GE2015 http://t.co/M0fW1lRMUx - Wed Apr 08 09:32:34 BST 2015
toatl number of tweets834
@itmoldova - Google.md spart de hackerii din Bangladesh http://t.co/elSWdDjlBQ http://t.co/PThwRmRf14 - Wed Apr 08 09:32:34 BST 2015
toatl number of tweets835
@SidSockhead - "@blairmcdougall: Last one http://t.co/fPCNBntxco" #GE2015 - Wed Apr 08 09:32:35 BST 2015
toatl number of tweets836
@SouthLpoolCAB - RT @CitizensAdvice: In the last election young people were much less likely to vote. Let's change that for #GE2015 http://t.co/Vpk3aNwrTr h… - Wed Apr 08 09:32:36 BST 2015
toatl number of tweets837
@JanetScot1 - RT @AngrySalmond: I will be liveTweeting the BBC Scotland #GE2015 debate tonight at 21:00. #GetAngryStaySexy #SexySocialism http://t.co/Apl… - Wed Apr 08 09:32:38 BST 2015
toatl number of tweets838
@AlbanThurston - RT @UKuncut: London hit with @CutAusterity subvertising campaign. No more bailouts, no more cuts, no more election lies! #GE2015 http://t.c… - Wed Apr 08 09:32:39 BST 2015
toatl number of tweets839
@7Lewi - RT @ToryTottyOnline: #SNP list of demands to prop up #Miliband keeps getting longer. #ScotDebates #GE2015 http://t.co/D3ndMJdEhj http://t.c… - Wed Apr 08 09:32:39 BST 2015
toatl number of tweets840
@brido872 - RT @heraldscotland: Osborne says #Holyrood unlikely to get extra welfare powers #GE2015 http://t.co/XZqFnUdEtS http://t.co/woMseSOepc - Wed Apr 08 09:32:42 BST 2015
toatl number of tweets841
@itsafeature - My @voteforpolicies results: Liberal Democrats 60%, Labour 40%... #GE2015 #policychallenge http://t.co/HJus7Ojh7T - Wed Apr 08 09:32:42 BST 2015
toatl number of tweets842
@welchyboy269 - RT @LibDems: It's thanks to Lib Dems in govt that millions are benefitting from a £825 tax cut #GE2015 http://t.co/O4yNlGyvk3 http://t.co/L… - Wed Apr 08 09:32:43 BST 2015
toatl number of tweets843
@peaton1326 - RT @robmcd85: Client told me her & husb have stock cubes & mug of hot water for dinner 4 x week so that their kids can eat #GE2015 http://t… - Wed Apr 08 09:32:43 BST 2015
toatl number of tweets844
@IanIwrussell - Sturgeon TV debate vow on second #indyref
http://t.co/cCFxXSzNTq #VoteSNP http://t.co/zkz5vbVZFK - Wed Apr 08 09:32:44 BST 2015
toatl number of tweets845
@thistle8blower - @GuidoFawkes Go on tell us how it's alright that Non Domicile become legislators in House of Lords.
#Democracy
#EdMustWin
#GE2015 - Wed Apr 08 09:32:46 BST 2015
toatl number of tweets846
@SaskiaHeijnen - RT @TheLancet: "Health inequalities result from social inequalities"

Join our conversation tomorrow #LancetElection #sdoh #GE2015 http://t… - Wed Apr 08 09:32:46 BST 2015
toatl number of tweets847
@MsTitomiSokoya - RT @CitizensAdvice: In the last election young people were much less likely to vote. Let's change that for #GE2015 http://t.co/Vpk3aNwrTr h… - Wed Apr 08 09:32:46 BST 2015
toatl number of tweets848
@jamg3916 - RT @gareth0108: just remember it was Thatcher & Major who paid Farmers to feed bovine Carcass Meal to cows

nothing to do with the EU

#B… - Wed Apr 08 09:32:47 BST 2015
toatl number of tweets849
@hudghtonmepSNP - RT @theSNP: Today's news: FM @NicolaSturgeon calls for Labour to join with SNP to lock the Tories out of Downing Street #VoteSNP http://t.c… - Wed Apr 08 09:32:48 BST 2015
toatl number of tweets850
@ianwwsmith - RT @heraldscotland: Osborne says #Holyrood unlikely to get extra welfare powers #GE2015 http://t.co/XZqFnUdEtS http://t.co/woMseSOepc - Wed Apr 08 09:32:48 BST 2015
toatl number of tweets851
@EyeEdinburgh - RT @cjball_london: .@David_Cameron to hear your lies, for that is what they are, about having protected poorest was truly nauseating #GE201… - Wed Apr 08 09:32:48 BST 2015
toatl number of tweets852
@Papajoe1407 - RT @Odbe34: Erm, @EvanHD the latest YouGov/Sun Poll suggests your infographic is somewhat wrong! #newsnight #GE2015 http://t.co/NPE7o8SG5v - Wed Apr 08 09:32:49 BST 2015
toatl number of tweets853
@londonerabroad - RT @GreenGreenwich: Hmmm Who to vote for?
Helpful cut out & keep policy guide from #BBC
#GE2015
#GenerationVote
 http://t.co/sY6HzPYpwm h… - Wed Apr 08 09:32:49 BST 2015
toatl number of tweets854
```

Figure A1.4: Run Console

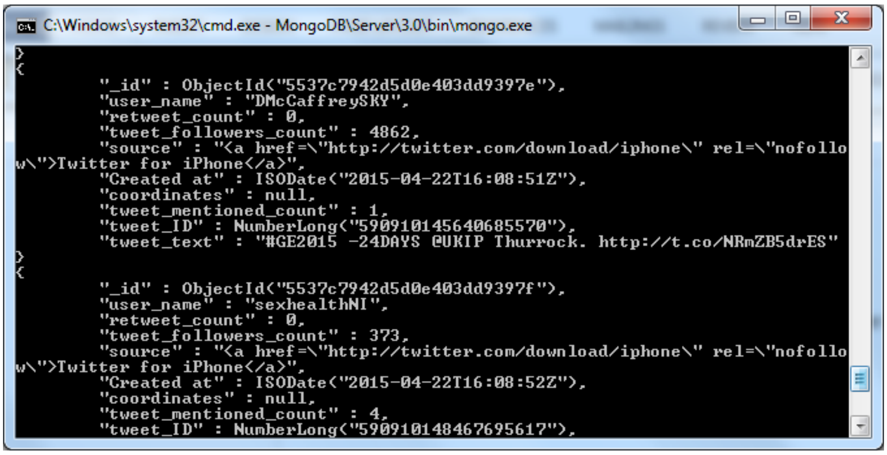
Here in figure A1.5, we show the saved tweets in *mongodb* stream collection, this done using Command Prompt Line. So, we first run the *mongo.exe* then ask for showing the available databases by using the command *show dbs*, after choosing the desired database by the following command *use db name*, then by using the *show collections* command we can view all the available collections in the specified database. To view the saved tweets in the collection of interest we use the following command *db.collection name.find()*. As we should iterate over all the tweets we used the while command as following:

var myCursor = db.collection name.find( );

while (myCursor.hasNext())

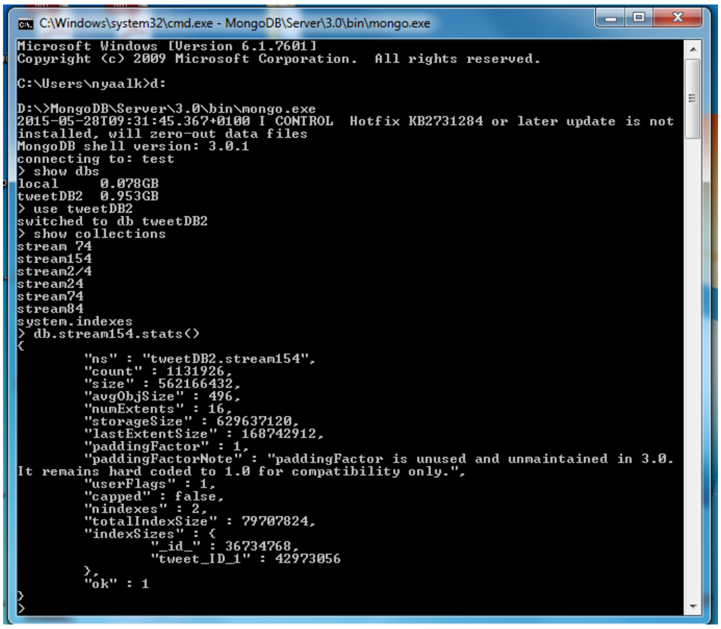
print(tojson(myCursor.next()));

Finally, we will show statistics related to our *stream154* collection using the following command *db.collection name.stats()*, as showing the number of tweets saved since 15/4-26/5-2015 (exceeds 1 million tweet), see figure A1.6.


C:\Windows\system32\cmd.exe - MongoDB\Server\3.0\bin\mongo.exe

```
}
{
        "_id" : ObjectId("5537c7942d5d0e403dd9397e"),
        "user_name" : "DMcCaffreySKY",
        "retweet_count" : 0,
        "tweet_followers_count" : 4862,
        "source" : "<a href=\"http://twitter.com/download/iphone\" rel=\"nofollo
w\">Twitter for iPhone</a>",
        "Created at" : ISODate("2015-04-22T16:08:51Z"),
        "coordinates" : null,
        "tweet_mentioned_count" : 1,
        "tweet_ID" : NumberLong("590910145640685570"),
        "tweet_text" : "#GE2015 -24DAYS @UKIP Thurrock. http://t.co/NRmZB5drES"
}
{
        "_id" : ObjectId("5537c7942d5d0e403dd9397f"),
        "user_name" : "sexhealthNI",
        "retweet_count" : 0,
        "tweet_followers_count" : 373,
        "source" : "<a href=\"http://twitter.com/download/iphone\" rel=\"nofollo
w\">Twitter for iPhone</a>",
        "Created at" : ISODate("2015-04-22T16:08:52Z"),
        "coordinates" : null,
        "tweet_mentioned_count" : 4,
        "tweet_ID" : NumberLong("590910148467695617"),
```

Figure A1.5: Tweets in MongoDB


C:\Windows\system32\cmd.exe - MongoDB\Server\3.0\bin\mongo.exe

```
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Users\nyaalk>d:

D:\>MongoDB\Server\3.0\bin\mongo.exe
2015-05-28T09:31:45.367+0100 I CONTROL  Hotfix KB2731284 or later update is not
installed, will zero-out data files
MongoDB shell version: 3.0.1
connecting to: test
> show dbs
local     0.078GB
tweetDB2  0.953GB
> use tweetDB2
switched to db tweetDB2
> show collections
stream 74
stream154
stream2/4
stream24
stream74
stream84
system.indexes
> db.stream154.stats()
{
        "ns" : "tweetDB2.stream154",
        "count" : 1131926,
        "size" : 562166432,
        "avgObjSize" : 496,
        "numExtents" : 16,
        "storageSize" : 629637120,
        "lastExtentSize" : 168742912,
        "paddingFactor" : 1,
        "paddingFactorNote" : "paddingFactor is unused and unmaintained in 3.0.
It remains hard coded to 1.0 for compatibility only.",
        "userFlags" : 1,
        "capped" : false,
        "nindexes" : 2,
        "totalIndexSize" : 79707824,
        "indexSizes" : {
                "_id_" : 36734768,
                "tweet_ID_1" : 42973056
        },
        "ok" : 1
}
>
```

Figure A1.6: Collection Statistics

Appendix 2: Gold Standard List During the UK General Elections 2015

In Appendix 2, we demonstrate a gold standard list (in the following table) showing a list of news articles published in the BBC News during the UK General Elections 2015. For each day we show the date, the event, and the linked news headline.

Date	Event	News Headline
15/4/2015	Leader Nick Clegg launched his party's manifesto.	Leader Nick Clegg launched his party's manifesto "Every vote for the Liberal Democrats matters".
	SDLP leader unveiled his party's manifesto.	In Northern Ireland SDLP leader Alasdair McDonnell unveiled his party's manifesto at the Holiday Inn Hotel, Belfast.
	UKIP leader Nigel Farage launches his party manifesto.	UKIP leader Nigel Farage and deputy chairwoman Suzanne Evans posed with a copy of their party's manifesto at its launch in Aveley. http://www.bbc.co.uk/news/in-pictures-32315512
16/4/2015	David Cameron has launched the Scottish Conservative manifesto.	David Cameron launches the Scottish Conservative manifesto calling the Labour and SNP a 'coalition of chaos'.http://www.bbc.co.uk/news/election-2015-scotland-32320466

	Daily Express owner Richard Desmond gives UKIP a £1m donation.	Richard Desmond, whose publishing company owns the Daily and Sunday Express, has donated £1m to UKIP.http://www.bbc.co.uk/news/election-2015-32340976
	TV election debate by the 5 leaders of Westminster's opposition.	The leaders of five of Westminster's opposition parties take part in the latest live TV election debate http://www.bbc.co.uk/news/election-2015-32331542
	Launch of the Alliance Party manifesto.	The Alliance Party will bring a "unique" voice to Westminster, ensuring the people of Northern Ireland are heard, the deputy leader has said at the launch of the party's manifesto. http://www.bbc.co.uk/news/election-2015-northern-ireland-32334446
17/4/2015	Labour is unveiling its Scottish manifesto in Glasgow.	Labour is unveiling its Scottish manifesto in Glasgow. The party also launched its youth manifesto in Lincoln and Ed Miliband called for an end to unpaid internships. http://www.bbc.co.uk/news/election-2015-32348029
	The UUP launched its manifesto and outlined tax reductions and extra money for mental health as part of its price for joining a coalition.	In Northern Ireland the Ulster Unionist Party (UUP) launched its manifesto and outlined tax reductions and extra money for mental health as part of its price for joining a coalition. http://www.bbc.co.uk/news/election-2015-32348029

20/4/2015	Nothing	http://www.telegraph.co.uk/news/general-election-2015/11549094/General-Election-2015-in-pictures-20-April-2015.html
21/4/2015	John Major warns form a labour-SNP government. He claims the SNP would deliver ”a daily dose of blackmail” to a Labour government.	Former Conservative PM Sir John Major warned that a Labour-SNP government would be ”a recipe for mayhem”. He added that he supporter of David Cameron as prime minister”. https://www.bbc.com/news/election-2015-32394684
	Liberal Democrat launched its Scottish manifesto.	Scottish Liberal Democrat leader Willie Rennie, Jo Swinson and candidate Mike Crockart launched the Scottish Liberal Democrat manifesto in front of the Forth Bridge in South Queensferry. http://www.bbc.co.uk/news/in-pictures-32396621
	More people register to vote ’than ever before’ for the GE 2015.	A record-breaking 469,000 people registered to vote online in one day for the 2015 general election - as the deadline closed on 20 April. https://www.bbc.co.uk/news/election-2015-32401218
22/4/2015	Nothing	http://www.telegraph.co.uk/news/general-election-2015/11554084/General-Election-2015-in-pictures-22-April-2015.html

23/4/2015	The IFS report comes after it analysed each of the party manifestos.	Four of the major parties have not provided ”anything like full details” on plans to cut the deficit, the ISF (Institute for Fiscal Studies) has said. http://www.bbc.co.uk/news/election-2015-32424739
24/5/2015	Ed Miliband took on foreign policy speech, and added David Cameron has presided over the biggest loss of UK influence in a generation.	Ed Miliband the Labour leader took on foreign policy speech, and added David Cameron has presided over the biggest loss of UK influence in a generation. http://www.bbc.co.uk/news/in-pictures-32446027
27/4/2015	Labour would exempt first-time buyers in England, Wales and Northern Ireland from paying stamp duty when buying homes.	Ed Miliband says Labour would exempt first-time buyers in England, Wales and Northern Ireland from paying stamp duty when buying homes below £300,000, for three years. http://www.bbc.co.uk/news/election-2015-32481973
	TNS survey suggests the SNP is on course to win 57 out of 59 Scottish seats and Labour to 1 seat.	The TNS Scottish poll was conducted face to face over a two week period. It gave the SNP 54% - the party’s highest rating since the 2014 referendum, with Labour on 22% - its lowest http://www.bbc.co.uk/news/live/election-2015-32476999
28/4/2015	Nothing	http://www.bbc.co.uk/news/live/election-2015-32487471

29/4/2015	SNP leader Nicola Sturgeon speech in Glasgow.	SNP leader Nicola Sturgeon delivered a speech in Glasgow on the day a new poll suggested that her party could win all 59 Scottish seats in the GE. http://www.bbc.co.uk/news/election-2015-scotland-32523804
	Jim Murphy took the Scottish Labour message to Glasgow.	Jim Murphy took the Scottish Labour Party message to Glasgow. http://www.bbc.co.uk/news/in-pictures-32515926
	Labour and Green candidates left off postal ballots.	More than 480 postal ballot papers have been sent out without the names of the Green and Labour Party candidates. https://www.bbc.co.uk/news/election-2015-32514019
30/4/2015	The Scottish Sun has endorsed the SNP while the London-based edition backed the Conservatives.	The Scottish Sun has endorsed the SNP - while the London-based edition backed the Conservatives.
	David Cameron, Ed Miliband and Nick Clegg are preparing for the BBC's Question Time Election Leaders Special.	David Cameron, Ed Miliband and Nick Clegg are preparing for the BBC's Question Time Election Leaders' Special. http://www.bbc.co.uk/news/election-2015-32530833
1/5/2015	Ed Miliband said, he would rather lose than doing a SNP deal!	Miliband will struggle to form a government if Labour is not biggest party, warns McConnell http://www.bbc.co.uk/news/live/election-2015-32543196

4/5/2015	Jim Murphy and Eddie Izzard were confronted by protesters in Glasgow	Scottish Labour leader Jim Murphy and comedian Eddie Izzard were heckled by opponents during general election campaigning in Glasgow. http://www.bbc.co.uk/news/election-2015-scotland-32581803
5/5/2015	No news, they are saying leaders are relaxed, laughing and taking selfies with crowds.	http://www.bbc.co.uk/news/in-pictures-32593176
6/5/2015	Final day of campaigning before GE polls are open.	Final day of campaigning before GE polls open. http://www.bbc.co.uk/news/live/election-2015-32603709
	Polls suggest no party will win the majority of seats.	Polls suggest no party will win enough seats for majority. http://www.bbc.co.uk/news/live/election-2015-32603709
7/5/2015	Nothing	
8/5/2015	2 news: Conservatives won, resigning of Ed Miliband, Nick Clegg and Nigel Farage	Conservative Party won the GE Resigning of leaders Ed Miliband, Nick Clegg and Nigel Farage http://www.bbc.co.uk/news/election-2015-32633008
11/5/2015	Nigel Farage resignation rejected	Farage stays as UKIP leader after resignation rejected http://www.bbc.co.uk/news/uk-politics-32696505
	BBC licence fee in doubt as John Whittingdale is named culture secretary.	John Whittingdale becomes UK culture secretary http://www.bbc.co.uk/news/entertainment-arts-32690777

12/5/2015	The BBC will be forced to slash its drama output if the licence fee is cut.	BBC drama at 'a tipping point' with licence fee cuts http://www.bbc.co.uk/news/entertainment-arts-32702746
	Cameron's new cabinet meeting.	Cameron's new cabinet meets http://www.bbc.co.uk/news/election/2015/england
13/5/2015	'No electoral fraud' in Nigel Farage-contested Thanet South seat.	"No evidence" of electoral fraud has been found in the Thanet South seat contested in the general election by Nigel Farage, police have said. https://www.bbc.co.uk/news/uk-politics-32725167
14/5/2015	Nigel Farage vows to remain UKIP leader.	Nigel Farage vows to remain UKIP leader, despite a row over his leadership of the party.
	Mary Creagh and Yvette Cooper announce bids to become Labour leader.	Mary Creagh and Yvette Cooper announce bids to become Labour leader. http://www.bbc.co.uk/news/live/uk-politics-32726181
15/5/2015	Nothing	
18-22/5/2015	Nothing	

Table A2.1: Golden Standard List

Appendix 3: FTSE 100 News Articles from November 2016- May 2017

In Appendix 3, we demonstrate a list of FTSE 100 news articles in the period from November 2016 until May 2017.

Date	News
2/11/2016	Next jumps 4% despite sales slipping. The FTSE 100 falls 0.6% https://www.bbc.com/news/live/business-37801333
7/11/2016	A strong day for stocks in Europe has seen London's benchmark FTSE 100 close the day up by 1.7%, or 113.64 points, at 6806.90. http://www.bbc.co.uk/news/live/business-37869316
8/11/2016	FTSE 100 closes 0.5% higher as investors got over their nerves about the US presidential election result. https://www.bbc.com/news/live/business-37875270
9/11/2016	US Election 2016: Markets meltdown fails to materialise. In London the FTSE 100 index dropped 2% at the start of trading before recovering to end the day 1% up. https://www.bbc.co.uk/news/business-37921036
10/11/2016	FTSE 100 dragged down by stronger pound. A new burst of volatility has driven the pound up against the US dollar, triggering a late selloff in London. The FTSE 100 index finished 1.2% lower at 6,827.98 points. https://www.bbc.co.uk/news/business-37934624
14/11/2016	The FTSE 100 has closed higher despite paring earlier gains, the index closed 0.34%, or 22.75 points, higher at 6,753.18 points. https://www.bbc.co.uk/news/live/business-37949837

17/11/2016	The London market ended higher on Thursday, but shares in Royal Mail fell 7% after its latest results. The FTSE 100 share index finished 45 points higher at 6,794.7 points. http://www.bbc.co.uk/news/business-38010843
22/11/2016	FTSE 100 has closed up 0.57% or 38.86 points at 6,816.82. It was boosted by the dip in the pound and higher commodity prices. https://www.bbc.com/news/live/business-38024672/page/2
28/11/2016	The FTSE 100 has closed down by 41.28 points or 0.60% at 6,799.47. https://www.bbc.co.uk/news/live/business-38103550
29/11/2016	At close, the benchmark FTSE 100 index was down 27.47 points, or 0.4%, at 6,772. http://www.bbc.co.uk/news/business-38141953
2/12/2016	FTSE 100 closes down ahead of Italian referendum. The benchmark share index was down 28.21 points or 0.43% at 6,730.72, with investors cautious ahead of Italy's referendum at the weekend. http://www.bbc.co.uk/news/business-38180179
8/12/2016	The FTSE 100 closed up 0.42% to 6,931.55 points, led by a 4.6% rise for WPP after shares in the advertising giant were upgraded to a 'buy' by broker Jefferies. https://www.bbc.co.uk/news/live/business-38183110
9/12/2016	The FTSE 100 index closed up 52.66 points at 6,954.21, notching up five consecutive days of gains. http://www.bbc.co.uk/news/business-38261236
13/12/2016	The FTSE 100 has held on to this morning's gains. It's trading 0.55% higher at 6,928. http://www.bbc.co.uk/news/live/business-38261853
28/12/2016	Strong gains for mining companies helped power the FTSE 100 to close at a new record high. The FTSE rose 0.5%, or 37.9 points, to 7,106.08 points http://www.bbc.co.uk/news/business-38449901
30/12/2016	London's benchmark FTSE 100 index reached its highest level to date on the last day of trading for 2016. The move upwards was - 32.5 points or 0.42% - it left the FTSE at an unprecedented 7,142.83. http://www.bbc.co.uk/news/business-38467258

9/1/2017	Share prices in London rose to a new record high point, with the FTSE 100 index closing 28 points up at 7,238. http://www.bbc.co.uk/news/business-38553048
10/1/2017	The FTSE 100 index has closed at a record high for the ninth day in a row, the longest such streak in history. The UK's benchmark share index ended Tuesday's trading session at 7,275.47, up 37.7 points on the day. http://www.bbc.co.uk/news/business-38567233
11/1/2017	The UK market opened lower as analysts digested a feast of Christmas trading updates from retailers. After closing at its 10th record high in a row on Wednesday, the FTSE 100 opened 22.03 points lower at 7,268.46. http://www.bbc.co.uk/news/business-38581030
13/1/2017	London's FTSE 100 has chalked up its 12th consecutive record high as shares in housebuilders led the way. The blue-chip index closed 0.6%, or 45.4 points, higher at 7,337.8 - the 14th time it has ended higher. http://www.bbc.co.uk/news/business-38608250
23/1/2017	FTSE 100 closes lower as pound hits one-month dollar high. Meanwhile the benchmark FTSE 100 index shed 47.26 points to 7,151.18 https://www.bbc.co.uk/news/business-38715793
3/2/2017	By midday, the benchmark FTSE 100 share index was 37.05 points higher at 7,177.80. https://www.bbc.co.uk/news/business-38852510
1/3/2017	The benchmark FTSE 100 index jumped 1.59% to 7,382.9 points https://www.bbc.co.uk/news/business-39125987
6/3/2017	By the close, the FTSE 100 was down 24 points at 7,350.12. https://www.bbc.co.uk/news/business-39177346
16/3/2017	The benchmark FTSE 100 rose 47.31 points to a record closing high of 7,415.57 https://www.bbc.co.uk/news/business-39289651
21/3/2017	At the close, the FTSE 100 was down 0.69% at 7,378.34 points, with mining companies the top five fallers. http://www.bbc.co.uk/news/business-39337885

18/4/2017	The pound rose strongly and share prices in London fell sharply after Theresa May announced plans to call a general election on 8 June. The benchmark FTSE 100 share index fell 180 points, or 2.5%, to 7,148. https://www.bbc.co.uk/news/business-39627859
19/4/2017	London's leading shares fell again on Wednesday but sterling was steady as investors continued to digest the consequences of the UK's snap election call. The FTSE 100 share index ended the day down 33 points, or almost 0.5%, at 7,114.3 points. https://www.bbc.co.uk/news/business-39639744
24/4/2017	The FTSE 100 index clawed back some of the losses triggered by last week's announcement of a general election, closing up 2% at 7,264.68. https://www.bbc.co.uk/news/business-39691366
28/4/2017	Barclays pulls FTSE 100 lower. The FTSE 100 index closed almost 0.5% lower at 7,203 https://www.bbc.com/news/business-39743647
16/5/2017	The FTSE 100 has closed above 7,500 for the first time despite inflation being at the highest rate since September 2013. Closing at 67.66 points, or 0.91%, higher at 7,522.03 https://www.bbc.co.uk/news/live/business-39895846
19/5/2017	At the close on Friday, the FTSE 100 was up 34.29 points at 7,470.71. https://www.bbc.co.uk/news/business-39972658
24/5/2017	The benchmark FTSE 100 gained 29.6 points or 0.4% to reach 7,514.90. FTSE 100 closes up led by Easyjet. https://www.bbc.co.uk/news/business-40026733

Table A3.1: FTSE 100 News from November 2016 untill May 2017

Appendix 4: FTSE 100 News Articles from July 2015- February 2016

In Appendix 4, we present the list of News Headlines (in table A4.1) appearing in the BBC News for the FTSE 100 index in the period from July 2015 until February 2016. There are 22 News events in total, news events with price change less than 1% are not counted. We realised that news with price change less than 1% are transient and published if the change in FTSE 100 was related to oil or mining sectors only.

Date	Price Change	News Headline
22/7/2015	Decrease	The FTSE 100 falls, with shares in chip-designer Arm Holdings down after Apple's latest earnings forecast disappointed the market. It closed down1.5% brought by miners and ARM Holdings. http://www.bbc.co.uk/news/business-33619756
27/7/2015	Decrease	The FTSE 100 index closed down 74.68 points at 6,505.13. Merlin Entertainments was the biggest faller on the FTSE 100 after the firm warned profits would be hit by the effects of the rollercoaster crash at its Alton Towers theme park. http://www.bbc.co.uk/news/business-33672315
5/8/2015	Increase	The London market gained nearly 1% on Wednesday, with mining shares among the top risers. FTSE 100's risers board, with the index up 65.84 points at 6,752.41. (less than 1%) http://www.bbc.co.uk/news/business-33784771

19/8/2015	Decrease	The FTSE 100 fell to its lowest level since January amid concerns about the outlook for commodity prices and Chinese growth. The FTSE 100 index closed down 122.84 points at 6,403.45. http://www.bbc.co.uk/news/business-33984773
20/8/2015	Decrease	FTSE 100 falls for eighth day due to global concerns. The FTSE 100 index closed down 35.56 points at 6,367.89. http://www.bbc.co.uk/news/business-34000349
21/8/2015	Decrease	The FTSE has fallen 5.2%, or 363 points, since Monday. On Friday the index closed 2.8% lower, while markets in Paris and Frankfurt saw falls of about 3%. http://www.bbc.co.uk/news/business-34015798
24/8/2015	Decrease	FTSE 100 loses more than £60bn after China's 'Black Monday'. London's FTSE 100 index closed down 4.6% at 5,898.87. https://www.bbc.co.uk/news/business-34038147
25/8/2015	Increase	China has reduced its main interest rate to boost growth in its economy. The move has boosted global share prices further, with London's FTSE 100 jumping 3%. http://www.bbc.co.uk/news/uk-34052618
27/8/2015	Increase	The FTSE 100 gained £60bn in value today, the biggest one-day rise in the index since October 2011. European markets have jumped after stronger than expected US GDP figures and a boost in US stocks, the FTSE 100 closed up 3.56% at 6,192.03. http://www.bbc.co.uk/news/live/business-34006003
28/8/2015	Increase	FTSE 100 ends the week on a high note after China-driven carnage. The index of London's biggest listed companies rose 55.91 points or 0.9pc on Friday to end at 6,247.94. http://www.telegraph.co.uk/finance/markets/marketreport/11831650/F 100-ends-the-week-on-a-high-note-after-China-driven-carnage.html

1/9/2015	Decrease	FTSE 100 hit by weak Chinese manufacturing data. The UK's FTSE 100 index closed down 189.40 points, or 3%, to 6,058.54. http://www.bbc.co.uk/news/business-34113452
3/9/2015	Increase	London's FTSE 100 maintained its opening gains in mid-morning trade, 1.64% higher at 6,183.34 points, with Easyjet being the biggest winner. http://www.bbc.co.uk/news/live/business-34087604
4/9/2015	Decrease	The FTSE 100 extended its losses after the release of the latest US jobs figures. The FTSE 100 index - which had been down about 1.7% before the figures were released - dropped further, closing down 151 points, or 2.4%, at 6,042.92. http://www.bbc.co.uk/news/business-34150161
9/9/2015	Increase	FTSE ends day higher as market sentiment builds. London's leading shares ended the day in positive territory as global investors regained confidence. The FTSE 100 index to close 82.91 points or 1.35% up at 6,229.01. http://www.bbc.co.uk/news/business-34195413
15/9/2015	Increase	FTSE 100 stages afternoon rally to close higher. The FTSE 100 closed up 0.87%, or 53 points, to 6,137.6, led by engineering firm Weir Group. http://www.bbc.co.uk/news/business-34255127
18/9/2015	Decrease	Not a great end to the week for the markets. The FTSE 100 is off 1%, while Paris and Frankfurt have sunk about 2.5%. http://www.bbc.co.uk/news/live/business-34221507
22/9/2015	Decrease	FTSE 100 slides as mining shares suffer. The FTSE 100 closed down 172.87 points, or 2.8%, to 5,935.84. http://www.bbc.co.uk/news/business-34322790
25/9/2015	Increase	FTSE 100 rebounds on Janet Yellen comments. At closing, the FTSE 100 index was up 147.52 points, or 2.47%, at 6,100.01. http://www.bbc.co.uk/news/business-34356486

28/9/2015	Decrease	The FTSE 100 closed 2.46% lower at 5958.86, completely wiping out the gains it made on Friday. http://www.bbc.co.uk/news/live/business-34358976
29/9/2015	Decrease	The London market closed lower but recovered initial sharp falls helped by mining stocks. The FTSE 100 of 100 leading shares fell 49.62 points to 5,909.24, a fall of 0.83%. http://www.bbc.co.uk/news/business-34388563
5/10/2015	Increase	The main UK market rose by 2.8%, with Glencore leading the way on reports it was looking to sell some of its agricultural assets to cut debts. http://www.bbc.co.uk/news/business-34441540
10/11/2015	Decrease	Miners led the FTSE 100 lower on Tuesday as commodity prices slid, with the FTSE 100 index fell 0.3% to 6,275.28 points. (less than 1%) http://www.bbc.co.uk/news/business-34774577
12/11/2015	Decrease	FTSE 100 lower as Rolls-Royce shares plunge. At close, the FTSE 100 index was down 118.52 points, or 1.88%, at 6,178.68. http://www.bbc.co.uk/news/business-34785068
17/11/2015	Increase	FTSE 100 rises as market recovery continues. At the end of the day, the FTSE 100 index was up 122.38 points, or 1.99%, at 6,268.76. http://www.bbc.co.uk/news/business-34841724
18/11/2015	Increase	Mining companies helped push London's leading shares higher, with the FTSE 100 closing up 10.21 points, or 0.2%, at 6,278.97. (less than 1%) http://www.bbc.co.uk/news/business-34855200
26/11/2015	Increase	FTSE 100 bolstered by mining firms. The benchmark FTSE 100 index closed up 55.5 points at 6,393.13. http://www.bbc.co.uk/news/business-34930384
3/12/2015	Decrease	At close, the FTSE 100 was 145.93 points lower at 6,274.00. http://www.bbc.co.uk/news/business-34992913

10/12/2015	Decrease	FTSE 100 falls as Sports Direct shares slump. The index closed down by 38.63 points at 6,088.05. (less than 1%) http://www.bbc.co.uk/news/business-35059626
11/12/2015	Decrease	The FTSE 100 index sunk 2.2% to close under 6,000 points - its seventh consecutive daily decline. The FTSE 100 index fell 135.2 points to 5,952 - the lowest level since late September. http://www.bbc.co.uk/news/business-35070184
15/12/2015	Increase	The FTSE 100 broke back through the 6,000 level as a rise in supermarket shares helped to lift the UK's benchmark index. The FTSE 100 closed up 143.73 points, or 2.5%, at 6,017.79. http://www.bbc.co.uk/news/business-35100085
16/12/2015	Increase	London's top shares continued their renewed rise in Wednesday trading, the day after the benchmark index rose back above 6,000. The FTSE 100 rose 43.4 points to close at 6061.19. http://www.bbc.co.uk/news/business-35110494
23/12/2015	Increase	The market climbed strongly, with mining and energy-related shares leading the way after prices of metals rose and oil prices stabilised. The benchmark FTSE 100 index closed up 157.88 points, or 2.6%, at 6,240.98. http://www.bbc.co.uk/news/business-35166667
29/12/2015	Increase	London's leading shares ended the day higher as a modest recovery in oil prices helped to boost European and US markets. The benchmark FTSE 100 index rose 59.93 points, or 0.96%, to 6,314.57. http://www.bbc.co.uk/news/business-35192384
4/1/2016	Decrease	FTSE 100 falls 2.6% amid global sell-off. The FTSE 100 index was down 161.83 points, or 2.6%, at 6,080.49 http://www.bbc.co.uk/news/business-35219846
5/1/2016	Increase	FTSE 100 ends higher after erratic day. It closed up 0.72% at 6,137.24. (less than 1%) http://www.bbc.co.uk/news/business-35230677

6/1/2016	Decrease	FTSE 100 dragged down by mining stocks. The 100-share index lost 66.86 to 6,073.38 http://www.bbc.co.uk/news/business-35241049
7/1/2016	Decrease	FTSE 100 down 2% on Chinese woes. The benchmark FTSE 100 index fell 119.30 points to 5,954.08. http://www.bbc.co.uk/news/business-35250182
14/1/2016	Decrease	FTSE 100 falls as global sell-off resumes. The FTSE 100 closed 0.7% lower, it bounced off lows to close down 42 points at 5,918. http://www.bbc.co.uk/news/business-35310093
18/1/2016	Decrease	FTSE 100 closes at three-year low ended the day at its lowest level for more than three years on Monday. It was down 24.18 points at 5,779.92, amid continuing concerns about the slowdown in China's growth. (less than 1%) http://www.bbc.co.uk/news/business-35341826
20/1/2016	Decrease	The FTSE 100 slumped 3.5% as investors fretted over global growth prospects and falling oil prices. The UK's benchmark index closed down 203.2 points at 5673.58. http://www.bbc.co.uk/news/business-35359796
22/1/2016	Increase	Shares up as global stock market have rallied for a second day, boosted by rising oil prices. London's FTSE 100 index closed more than 2% higher. http://www.bbc.co.uk/news/business-35379549
25/1/2016	Decrease	FTSE 100 closes lower as oil prices fall. Following a late sell-off, the FTSE 100 index closed 23 points, or 0.4%, lower at 5,877. (less than 1%) http://www.bbc.co.uk/news/business-35398845
27/1/2016	Increase	The London market closed at its highest level since January 6 as oil prices ended above $32 a barrel. The FTSE 100 index closed up 1.3%, or 78.9 points, at 5,990.3. http://www.bbc.co.uk/news/business-35417225

28/1/2016	Decrease	The London market closed lower but oil and mining companies were showing good gains, as the price of oil rose. The FTSE 100 index fell 58.59 points to 5,931.78 points - a fall of 0.98%. http://www.bbc.co.uk/news/business-35426719
29/1/2016	Increase	Strong FTSE gains after Japan rate cut. London shares rose sharply on Friday, boosted by the Bank of Japan, which cut interest rates to negative, and a strong start on Wall Street. The FTSE 100 index rose 127.7 points, or 2.15%, to close at 6,059.5. http://www.bbc.co.uk/news/business-35437636
2/2/2016	Decrease	Oil giants' woes drag down FTSE 100. It lost 2.3% or 138.09 points to close at 5,922.01. http://www.bbc.co.uk/news/business-35470062
8/2/2016	Decrease	FTSE 100 slumps on weak bank shares. The FTSE 100 finished 2.7% lower at 5,689 points as analysts said investors were turning against financial shares because of shaky global growth. http://www.bbc.co.uk/news/business-35520828
9/2/2016	Decrease	London's leading shares closed at a three year low, giving up early gains after being dragged down by mining stocks. By the end of the trading day, the benchmark FTSE 100 index was down 57.17 points or 1.0% to 5,632.19 http://www.bbc.co.uk/news/business-35530328
10/2/2016	Increase	FTSE 100 rises as banks recover. The FTSE 100 closed up 36.34 points, or 0.7%. (less than 1%) http://www.bbc.co.uk/news/business-35539475
11/2/2016	Decrease	The index of London's leading shares has fallen 2.4%, The FTSE 100 closed down 124 points at 5,549 points. http://www.bbc.co.uk/news/business-35548117
12/2/2016	Increase	FTSE rebounds as Rolls-Royce surges. By the close, the benchmark FTSE 100 was up 170.6 points, or 3%, at 5,707.6. http://www.bbc.co.uk/news/business-35558163

15/2/2016	Increase	London's leading share index ended the day 2.04% higher on Monday after Tokyo posted its biggest one-day rise since late 2008. http://www.bbc.co.uk/news/business-35577308
17/2/2016	Increase	The FTSE 100 was 2.87% higher - a gain of 168.15 points - at 6,030.32. http://www.bbc.co.uk/news/business-35593629
18/2/2016	Increase	London's main share market closed lower on Thursday, dragged down by oil and mining companies. The FTSE 100 ended down 55.78 at 5974.54 points. http://www.bbc.co.uk/news/business-35602312
24/2/2016	Decrease	Shares and the pound both saw falls on Wednesday amid uncertainty over the global economy and the UK's position in Europe. The FTSE 100 fell 101.04 points to close at 5,861.27 - a loss of 1.69%. http://www.bbc.co.uk/news/business-35649100
25/2/2016	Increase	The FTSE 100 ended the day with gains of 145.63 points, or 2.48pc, at 6,012.81. https://www.telegraph.co.uk/business/2016/02/25/ftse-100-rebounds-but-china-stocks-plunge-6pc-ahead-of-g20-meeti/

Table A4.1: FTSE 100 News From July 2015 - Feb 2016

Glossary

fn False negatives. 65, 66, 71

fp False positives. 65, 66

tn True negatives. 65, 66

tp True positives. 65, 66

BA Backlash Agent. 108–112, 117, 126, 131

CEP Complex Event Processing. 10, 11, 14

CQL Continuous Query Language. 10

CT Contrary Trading. 101, 103, 105, 107–114, 117, 119, 126, 131, 135

D-FPM Dynamic-FPM. xii, 43, 60, 69, 71–74

DBMS Data Base Management System. 10

DC Directional Change Approach. v, vii, viii, xii, 5, 29–32, 75–79, 82, 83, 85, 86, 88, 89, 91, 97–102, 105–107, 109, 111–113, 115–117, 119, 123, 126–128, 131, 135, 137, 138, 144, 147, 149, 153, 156, 160, 161, 163

DSMS Data Stream Management System. 10

DT-TS Dynamic Threshold Trading Strategy. v, vi, x–xii, 105, 106, 108, 109, 111, 112, 114–117, 119, 120, 126–128, 131, 133, 135, 161, 163

EPL Event Processing Language. 11

FPM Frequent Pattern Mining. iv, v, 23, 26, 27, 35, 37, 39, 40, 45, 56, 66, 73, 74, 149, 159

FPs Frequent Patterns. 39, 59–61, 64

FSD First Story Detection. 16

FT-TS Fixed Threshold Trading Strategy. 106, 107, 109, 111, 112, 114, 117, 126–128, 131

GE 2015 UK General Elections 2015. x, xi, 34, 44, 45, 48–51, 53–55, 61, 64, 66, 71, 74, 84, 137, 138, 142–145, 147, 149, 155–157, 160

GP Genetic Programming. 101

HFD High Frequency Data. 3, 10, 75, 100

HUP High Utility Pattern. 29

HUPC High Utility Pattern Clustering. 29, 72, 73

JMA Japan Metrological Agency. 20

LDA Latent Dirichlet Allocation. 17

LR logistic regression. x, 52–55, 61

LSE London Stock Exchange. 84, 106, 141

LSH Locality Sensitive Hashing. 16

LTT Location-Time Constrained Topic. 16

MDW Milano Design Week. 11, 12

NLP Natural Language Processing. 20

OS Overshoot Event. 31, 78, 89, 102–105, 117, 119, 146

OWL Web Ontology Language. 9

POI Point Of Interest. 12, 13

POS Part Of Speech Tagging. 20

RDF Resource Description Framework. 8, 9, 11–13

SFPM Soft Frequent Pattern Mining. 27, 28, 61, 65, 66, 69, 71–74, 160

SIOC Semantically-Interlinked Online Communities. vii, 9, 11, 12

SN Social Network. 36–39, 43–45, 57, 60

SSA Social Set Analysis. 14

SVM Support Vector Machine. 19

TF Trend Following. 101, 103, 105, 107–114, 117, 119, 126, 131, 135

TF-IDF Term Frequency-Inverse Document Frequency. 15, 16, 22

WWW World Wide Web. 6, 8, 50, 61

XML Extensible Markup Language. 9

Bibliography

[1] S. F. Irleand, “Sioc core ontology specification.” `http://rdfs.org/sioc/spec/`, 2010. [Online accessed 22-May-2015].

[2] M. Balduini, I. Celino, D. Dell’Aglio, E. Della Valle, Y. Huang, T. Lee, S.-H. Kim, and V. Tresp, “Bottari: An augmented reality mobile application to deliver personalized and location-based recommendations by continuous analysis of social media streams,” *Web Semantics: Science, Services and Agents on the World Wide Web*, vol. 16, pp. 33–41, 2012.

[3] J. Han, J. Pei, Y. Yin, and R. Mao, “Mining frequent patterns without candidate generation: A frequent-pattern tree approach,” *Data mining and knowledge discovery*, vol. 8, no. 1, pp. 53–87, 2004.

[4] J. Guo, P. Zhang, and L. Guo, “Mining hot topics from twitter streams,” *Procedia Computer Science*, vol. 9, pp. 2008–2011, 2012.

[5] J. Bollen, H. Mao, and X. Zeng, “Twitter mood predicts the stock market,” *Journal of Computational Science*, vol. 2, no. 1, pp. 1–8, 2011.

[6] M. Aloud, E. Tsang, R. Olsen, and A. Dupuis, “A directional-change event approach for studying financial time series,” *Economics: The Open-Access, Open-Assessment E-Journal*, vol. 6, no. 2012-36, 2012.

[7] E. Della Valle, S. Ceri, D. F. Barbieri, D. Braga, and A. Campi, *A first step towards stream reasoning.* Springer, 2009.

[8] B. Babcock, S. Babu, M. Datar, R. Motwani, and J. Widom, “Models and issues in data stream systems,” in *Proceedings of the twenty-first ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pp. 1–16, ACM, 2002.

[9] G. Krempl, I. Žliobaite, D. Brzeziński, E. Hüllermeier, M. Last, V. Lemaire, T. Noack, A. Shaker, S. Sievi, M. Spiliopoulou, *et al.*, "Open challenges for data stream mining research," *ACM SIGKDD explorations newsletter*, vol. 16, no. 1, pp. 1–10, 2014.

[10] Gartner, "Gartner says solving 'big data' challenge involves more than just managing volumes of data," *Stamford*, 2011.

[11] M. A. Beyer and D. Laney, "The importance of'big data': a definition," *Stamford, CT: Gartner*, 2012.

[12] T. Das and P. M. Kumar, "Big data analytics: A framework for unstructured data analysis," *International Journal of Engineering Science & Technology*, vol. 5, no. 1, p. 153, 2013.

[13] N. Veeranjaneyulu, M. N. Bhat, and A. Raghunath, "Approaches for managing and analyzing unstructured data," *International Journal on Computer Science and Engineering*, vol. 6, no. 01, 2014.

[14] S. Ananiadouo, *National centre for text mining: Introduction to tools for Researches*, 2008. http://sitecore.jisc.ac.uk/publications/briefingpapers/2008/bpnationalcentrefortextminingv1.aspx/.

[15] K. K. Khedo, R. Perseedoss, A. Mungur, *et al.*, "A wireless sensor network air pollution monitoring system," *arXiv preprint arXiv:1005.1737*, 2010.

[16] B. Fang, Q. Xu, T. Park, and M. Zhang, "Airsense: an intelligent home-based sensing system for indoor air quality analytics," in *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, pp. 109–119, ACM, 2016.

[17] J. Gabrys, H. Pritchard, and B. Barratt, "Just good enough data: Figuring data citizenships through air pollution sensing and data stories," *Big Data & Society*, vol. 3, no. 2, p. 2053951716679677, 2016.

[18] A. Signorini, A. M. Segre, and P. M. Polgreen, "The use of twitter to track levels of disease activity and public concern in the u.s. during the influenza a h1n1 pandemic," *PLOS ONE*, vol. 6, pp. 1–10, 05 2011.

[19] F. Gesualdo, G. Stilo, M. V. Gonfiantini, E. Pandolfi, P. Velardi, A. E. Tozzi, *et al.*, “Influenza-like illness surveillance on twitter through automated learning of naïve language,” *PLoS One*, vol. 8, no. 12, p. e82489, 2013.

[20] B. Pan, Y. Zheng, D. Wilkie, and C. Shahabi, “Crowd sensing of traffic anomalies based on human mobility and social media,” in *Proceedings of the 21st ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pp. 344–353, ACM, 2013.

[21] P. Anantharam, P. Barnaghi, K. Thirunarayan, and A. Sheth, “Extracting city traffic events from social streams,” *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 6, no. 4, p. 43, 2015.

[22] R. Tönjes, P. Barnaghi, M. Ali, A. Mileo, M. Hauswirth, F. Ganz, S. Ganea, B. Kjærgaard, D. Kuemper, S. Nechifor, *et al.*, “Real time iot stream processing and large-scale data analytics for smart city applications,” in *poster session, European Conference on Networks and Communications*, sn, 2014.

[23] M. M. Rathore, A. Ahmad, A. Paul, and S. Rho, “Urban planning and building smart cities based on the internet of things using big data analytics,” *Computer Networks*, vol. 101, pp. 63–80, 2016.

[24] A. Acar and Y. Muraki, “Twitter for crisis communication: lessons learned from japan’s tsunami disaster,” *International Journal of Web Based Communities*, vol. 7, no. 3, pp. 392–402, 2011.

[25] T. Sakaki, M. Okazaki, and Y. Matsuo, “Earthquake shakes twitter users: real-time event detection by social sensors,” in *Proceedings of the 19th international conference on World wide web*, pp. 851–860, ACM, 2010.

[26] D. Pohl, A. Bouchachia, and H. Hellwagner, “Automatic sub-event detection in emergency management using social media,” in *Proceedings of the 21st International Conference on World Wide Web*, pp. 683–686, ACM, 2012.

[27] Q. Li, T. Wang, P. Li, L. Liu, Q. Gong, and Y. Chen, “The effect of news and public mood on stock movements,” *Information Sciences*, vol. 278, pp. 826–840, 2014.

[28] E. Ferrara and Z. Yang, "Measuring emotional contagion in social media," *PloS one*, vol. 10, no. 11, p. e0142390, 2015.

[29] W. Dou, X. Wang, D. Skau, W. Ribarsky, and M. X. Zhou, "Leadline: Interactive visual analysis of text data through event identification and exploration," in *Visual Analytics Science and Technology (VAST), 2012 IEEE Conference on*, pp. 93–102, IEEE, 2012.

[30] L. D'Monte, "Swine flu's tweet tweet causes online flutter," *Business Standard*, vol. 29, 2013.

[31] M. Isaac and S. Ember, "For election day influence, twitter ruled social media," *The New York Times. Retrieved from https://www. nytimes. com/2016/11/09/technology/for-election-day-chatter-twitterruled-social-media. html*, 2016.

[32] J. Han, H. Cheng, D. Xin, and X. Yan, "Frequent pattern mining: Current status and future directions," *Data Mining and Knowledge Discovery*, vol. 15, no. 1, pp. 55–86, 2007.

[33] R. S. Tsay, *Analysis of financial time series*, vol. 543. John Wiley & Sons, 2005.

[34] D. M. Guillaume, M. M. Dacorogna, R. R. Davé, U. A. Müller, R. B. Olsen, and O. V. Pictet, "From the bird's eye to the microscope: A survey of new stylized facts of the intra-daily foreign exchange markets," *Finance and stochastics*, vol. 1, no. 2, pp. 95–129, 1997.

[35] J. B. Glattfelder, A. Dupuis, and R. B. Olsen, "Patterns in high-frequency fx data: discovery of 12 empirical scaling laws," *Quantitative Finance*, vol. 11, no. 4, pp. 599–614, 2011.

[36] A. Bakhach, E. Tsang, W. L. Ng, and V. L. R. Chinthalapati, "Backlash agent: A trading strategy based on directional change," in *2016 IEEE Symposium Series on Computational Intelligence (SSCI)*, pp. 1–9, Dec 2016.

[37] M. Aloud, "Directional-change event trading strategy: Profit-maximizing learning strategy," in *COGNITIVE 2015, The Seventh International Conference on Advanced Cognitive Technologies and Applications*, pp. 123–129, IARIA, 2015.

[38] M. Kampouridis and F. E. Otero, "Evolving trading strategies using directional changes," *Expert Systems with Applications*, vol. 73, pp. 145–160, 2017.

[39] G. Petkos, S. Papadopoulos, L. Aiello, R. Skraba, and Y. Kompatsiaris, "A soft frequent pattern mining approach for textual topic detection," in *Proceedings of the 4th International Conference on Web Intelligence, Mining and Semantics (WIMS14)*, p. 25, ACM, 2014.

[40] J. Huang, M. Peng, and H. Wang, "Topic detection from large scale of microblog stream with high utility pattern clustering," in *Proceedings of the 8th Workshop on Ph.D. Workshop in Information and Knowledge Management*, PIKM '15, (New York, NY, USA), pp. 3–10, ACM, 2015.

[41] S. Gaglio, G. Lo Re, and M. Morana, "Real-time detection of twitter social events from the user's perspective," in *Communications (ICC), 2015 IEEE International Conference on*, pp. 1207–1212, June 2015.

[42] T. Bisig, A. Dupuis, V. Impagliazzo, and R. Olsen, "The scale of market quakes," *Quantitative Finance*, vol. 12, no. 4, pp. 501–508, 2012.

[43] M. Aloud, M. Fasli, E. Tsang, A. Dupuis, and R. Olsen, "Stylized facts of trading activity in the high frequency fx market: An empirical study," *Journal of Finance and Investment Analysis*, vol. 2, no. 4, pp. 145–183, 2013.

[44] A. Bakhach, E. Tsang, and W. L. Ng, "Forecasting directional changes in financial markets," tech. rep., Working Paper WP075-15 Centre for Computational Finance and Economic Agents (CCFEA), University of Essex, 2015.

[45] E. Tsang, R. Tao, and S. Ma, "Profiling financial market dynamics under directional changes," tech. rep., Working Paper WP074-15, Centre for Computational Finance and Economic Agents (CCFEA), University of Essex, 2015.

[46] M. Aloud and M. Fasli, "Exploring trading strategies and their effects in the foreign exchange market," *Computational Intelligence*, 2016.

[47] Y. Shynkevich, T. M. McGinnity, S. Coleman, Y. Li, and A. Belatreche, "Forecasting stock price directional movements using technical indicators: investigating window size effects on one-step-ahead forecasting," in *2014 IEEE Conference*

on Computational Intelligence for Financial Engineering & Economics (CIFEr), pp. 341–348, IEEE, 2014.

[48] M. E. Aloud, "Profitability of directional change based trading strategies: The case of saudi stock market," *International Journal of Economics and Financial Issues*, vol. 6, no. 1, 2016.

[49] E. Tsang, R. Tao, A. Serguieva, and S. Ma, "Profiling high-frequency equity price movements in directional changes," *Quantitative Finance*, pp. 1–9, 2016.

[50] A. Bakhach, E. Tsang, and H. Jalalian, "Forecasting directional changes in the fx markets," in *2016 IEEE Symposium Series on Computational Intelligence (SSCI)*, pp. 1–8, Dec 2016.

[51] A. M. Bakhach, E. P. Tsang, and V. Raju Chinthalapati, "Tsfdc: A trading strategy based on forecasting directional change," *Intelligent Systems in Accounting, Finance and Management*, 2018.

[52] J. Gypteau, F. E. Otero, and M. Kampouridis, "Generating directional change based trading strategies with genetic programming," in *European Conference on the Applications of Evolutionary Computation*, pp. 267–278, Springer, 2015.

[53] T. Berners-Lee, J. Hendler, and O. Lassila, "The semantic web," *Scientific American*, pp. 29–37, May 2001.

[54] A. Maedche and S. Staab, "Ontology learning for the semantic web," *IEEE Intelligent systems*, vol. 16, no. 2, pp. 72–79, 2001.

[55] W3C, "Rdf concepts." `http://www.w3.org/TR/2014/REC-rdf11-concepts-20140225/`, 2004. [Online accessed 22-May-2015].

[56] W3C, "Owl ontology guide." `http://www.w3.org/TR/owl-guide/`, 2004. [Online accessed 22-May-2015].

[57] I. Horrocks and U. Sattler, "Ontology reasoning in the shoq (d) description logic," in *IJCAI*, vol. 1, pp. 199–204, 2001.

[58] X. H. Wang, D. Q. Zhang, T. Gu, and H. K. Pung, "Ontology based context modeling and reasoning using owl," in *Pervasive Computing and Communications*

Workshops, 2004. Proceedings of the Second IEEE Annual Conference on, pp. 18–22, Ieee, 2004.

[59] J. Z. Pan, "A flexible ontology reasoning architecture for the semantic web," *IEEE Transactions on Knowledge & Data Engineering*, no. 2, pp. 246–260, 2007.

[60] T. Gu, H. K. Pung, and D. Q. Zhang, "A service-oriented middleware for building context-aware services," *Journal of Network and computer applications*, vol. 28, no. 1, pp. 1–18, 2005.

[61] E. Thomas, J. Z. Pan, and Y. Ren, "Trowl: Tractable owl 2 reasoning infrastructure," in *Extended Semantic Web Conference*, pp. 431–435, Springer, 2010.

[62] E. Sirin, B. Parsia, B. C. Grau, A. Kalyanpur, and Y. Katz, "Pellet: A practical owl-dl reasoner," *Web Semantics: science, services and agents on the World Wide Web*, vol. 5, no. 2, pp. 51–53, 2007.

[63] L. Golab and M. T. Özsu, "Issues in data stream management," *ACM Sigmod Record*, vol. 32, no. 2, pp. 5–14, 2003.

[64] F. Lécué, S. Tallevi-Diotallevi, J. Hayes, R. Tucker, V. Bicer, M. L. Sbodio, and P. Tommasi, "Star-city: semantic traffic analytics and reasoning for city," in *Proceedings of the 19th international conference on Intelligent User Interfaces*, pp. 179–188, ACM, 2014.

[65] F. Lécué, S. Tallevi-Diotallevi, J. Hayes, R. Tucker, V. Bicer, M. Sbodio, and P. Tommasi, "Smart traffic analytics in the semantic web with star-city: Scenarios, system and lessons learned in dublin city," *Web Semantics: Science, Services and Agents on the World Wide Web*, vol. 27, pp. 26–33, 2014.

[66] A. Yoshihara, K. Fujikawa, K. Seki, and K. Uehara, "Predicting stock market trends by recurrent deep neural networks," in *Pacific rim international conference on artificial intelligence*, pp. 759–769, Springer, 2014.

[67] D. Wang, X. Liu, and M. Wang, "A dt-svm strategy for stock futures prediction with big data," in *Computational Science and Engineering (CSE), 2013 IEEE 16th International Conference on*, pp. 1005–1012, IEEE, 2013.

[68] A. Navon and Y. Keller, “Financial time series prediction using deep learning,” *arXiv preprint arXiv:1711.04174*, 2017.

[69] A. Margara, J. Urbani, F. van Harmelen, and H. Bal, “Streaming the web: Reasoning over dynamic data,” *Web Semantics: Science, Services and Agents on the World Wide Web*, vol. 25, pp. 24–44, 2014.

[70] B. Babcock, S. Babu, M. Datar, R. Motwani, and J. Widom, “Models and issues in data stream systems,” in *Proceedings of the twenty-first ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pp. 1–16, ACM, 2002.

[71] D. F. Barbieri, D. Braga, S. Ceri, E. Della Valle, and M. Grossniklaus, “C-sparql: Sparql for continuous querying,” in *Proceedings of the 18th international conference on World wide web*, pp. 1061–1062, ACM, 2009.

[72] E. FP7, “Large knowledge collider larkc.” `http://www.larkc.org/`, 2008. [Online accessed 9-May-2015].

[73] D. C. Luckham, *The Power of Events: An Introduction to Complex Event Processing in Distributed Enterprise Systems.* Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 2001.

[74] G. Cugola and A. Margara, “Tesla: a formally defined event specification language,” in *Proceedings of the Fourth ACM International Conference on Distributed Event-Based Systems*, pp. 50–61, ACM, 2010.

[75] EsperTech Inc, “Esper tech event series intelligence.” `http://www.espertech.com/index.php`, 2006. [Online accessed 9-May-2015].

[76] M. Balduini, E. Della Valle, D. Dell’Aglio, M. Tsytsarau, T. Palpanas, and C. Confalonieri, “Social listening of city scale events using the streaming linked data framework,” in *The Semantic Web–ISWC 2013*, pp. 1–16, Springer, 2013.

[77] C. Railean and A. Moraru, “Discovering popular events from tweets,” in *Proceedings of the 16th International Multiconference Information Society, Ljubljana, Slovenia*, 2013.

[78] B. Flesch, A. Hussain, and R. Vatrapu, “Social set visualizer: A set theoretical approach to big social data analytics of real-world events,” in *2015 IEEE International Conference on Big Data (Big Data)*, Oct 2015.

[79] A. Hussain and R. Vatrapu, "Social data analytics tool (sodato)," in *Advancing the Impact of Design Science: Moving from Theory to Practice*, pp. 368–372, Springer, 2014.

[80] M. Knobel and C. Lankshear, "Online memes, affinities, and cultural production," *A new literacies sampler*, vol. 29, pp. 199–227, 2007.

[81] L. M. Aiello, G. Petkos, C. Martin, D. Corney, S. Papadopoulos, R. Skraba, A. Göker, I. Kompatsiaris, and A. Jaimes, "Sensing trending topics in twitter," *IEEE Transactions on Multimedia*, vol. 15, no. 6, pp. 1268–1282, 2013.

[82] S. Phuvipadawat and T. Murata, "Breaking news detection and tracking in twitter," in *Web Intelligence and Intelligent Agent Technology (WI-IAT), 2010 IEEE/WIC/ACM International Conference on*, vol. 3, pp. 120–123, IEEE, 2010.

[83] G. Salton and M. J. McGill, "Introduction to modern information retrieval," 1986.

[84] A. Berger, R. Caruana, D. Cohn, D. Freitag, and V. Mittal, "Bridging the lexical chasm: statistical approaches to answer-finding," in *Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 192–199, ACM, 2000.

[85] S. Petrović, M. Osborne, and V. Lavrenko, "Streaming first story detection with application to twitter," in *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pp. 181–189, Association for Computational Linguistics, 2010.

[86] J. Allan, *Topic detection and tracking: event-based information organization*, vol. 12. Springer Science & Business Media, 2002.

[87] X. Zhou and L. Chen, "Event detection over twitter social media streams," *The VLDB journal*, vol. 23, no. 3, pp. 381–400, 2014.

[88] N. Panagiotou, I. Katakis, and D. Gunopulos, "Detecting events in online social networks: Definitions, trends and challenges," in *Solving Large Scale Learning Tasks. Challenges and Algorithms*, pp. 42–84, Springer, 2016.

[89] M. Cataldi, L. Di Caro, and C. Schifanella, "Emerging topic detection on twitter based on temporal and social terms evaluation," in *Proceedings of the Tenth International Workshop on Multimedia Data Mining*, p. 4, ACM, 2010.

[90] M. Mathioudakis and N. Koudas, “Twittermonitor: trend detection over the twitter stream,” in *Proceedings of the 2010 ACM SIGMOD International Conference on Management of data*, pp. 1155–1158, ACM, 2010.

[91] C. Li, A. Sun, and A. Datta, “Twevent: segment-based event detection from tweets,” in *Proceedings of the 21st ACM international conference on Information and knowledge management*, pp. 155–164, ACM, 2012.

[92] J. Weng and B.-S. Lee, “Event detection in twitter.,” *ICWSM*, vol. 11, pp. 401–408, 2011.

[93] B. Goethals, “Frequent set mining,” in *Data mining and knowledge discovery handbook*, pp. 377–397, Springer, 2005.

[94] D. M. Blei, “Probabilistic topic models,” *Communications of the ACM*, vol. 55, no. 4, pp. 77–84, 2012.

[95] D. M. Blei, A. Y. Ng, and M. I. Jordan, “Latent dirichlet allocation,” *Journal of machine Learning research*, vol. 3, no. Jan, pp. 993–1022, 2003.

[96] C. W. Fox and S. J. Roberts, “A tutorial on variational bayesian inference,” *Artificial intelligence review*, vol. 38, no. 2, pp. 85–95, 2012.

[97] X. Han and T. Stibor, “Efficient collapsed gibbs sampling for latent dirichlet allocation [j],” *Journal of Machine Learning Research*, vol. 13, pp. 63–78, 2010.

[98] D. Ramage, D. Hall, R. Nallapati, and C. D. Manning, “Labeled lda: A supervised topic model for credit attribution in multi-labeled corpora,” in *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1-Volume 1*, pp. 248–256, Association for Computational Linguistics, 2009.

[99] D. Quercia, H. Askham, and J. Crowcroft, “Tweetlda: supervised topic classification and link prediction in twitter,” in *Proceedings of the 4th Annual ACM Web Science Conference*, pp. 247–250, ACM, 2012.

[100] T. Hofmann, “Probabilistic latent semantic indexing,” in *ACM SIGIR Forum*, vol. 51, pp. 211–218, ACM, 2017.

[101] J. Manyika, M. Chui, B. Brown, J. Bughin, R. Dobbs, C. Roxburgh, and A. H. Byers, "Big data: The next frontier for innovation, competition, and productivity," *The McKinsey Global Institute*, 2011.

[102] N. B. Ellison and D. M. Boyd, "Social network sites: Definition, history, and scholarship," *Journal of Computer-Mediated Communication*, vol. 13, no. 1, pp. 210–230, 2007.

[103] M. Hasan, M. A. Orgun, and R. Schwitter, "A survey on real-time event detection from the twitter data stream," *Journal of Information Science*, pp. 1–21, 2017.

[104] S. Petrovic, M. Osborne, R. McCreadie, C. Macdonald, I. Ounis, and L. Shrimpton, "Can twitter replace newswire for breaking news?," in *Proceedings of the Seventh International AAAI Conference on Weblogs and Social Media (ICWSM13)*, 2013.

[105] M. Osborne and M. Dredze, "Facebook, twitter and google plus for breaking news: Is there a winner?," in *ICWSM*, 2014.

[106] A. Ritter, O. Etzioni, S. Clark, *et al.*, "Open domain event extraction from twitter," in *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 1104–1112, ACM, 2012.

[107] A. Ritter, S. Clark, O. Etzioni, *et al.*, "Named entity recognition in tweets: an experimental study," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pp. 1524–1534, Association for Computational Linguistics, 2011.

[108] R. Saurí, R. Knippen, M. Verhagen, and J. Pustejovsky, "Evita: A robust event recognizer for qa systems," in *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing*, HLT '05, (Stroudsburg, PA, USA), pp. 700–707, Association for Computational Linguistics, 2005.

[109] I. Mani and G. Wilson, "Robust temporal processing of news," in *Proceedings of the 38th Annual Meeting on Association for Computational Linguistics*, pp. 69–76, Association for Computational Linguistics, 2000.

[110] D. Zhou, L. Chen, and Y. He, "A simple bayesian modelling approach to event extraction from twitter," in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 700–705, ACL, 2014.

[111] A. X. Chang and C. D. Manning, "Sutime: A library for recognizing and normalizing time expressions.," in *LREC*, pp. 3735–3740, 2012.

[112] J. R. Finkel, T. Grenager, and C. Manning, "Incorporating non-local information into information extraction systems by gibbs sampling," in *Proceedings of the 43rd annual meeting on association for computational linguistics*, pp. 363–370, Association for Computational Linguistics, 2005.

[113] K. Gimpel, N. Schneider, B. O'Connor, D. Das, D. Mills, J. Eisenstein, M. Heilman, D. Yogatama, J. Flanigan, and N. A. Smith, "Part-of-speech tagging for twitter: Annotation, features, and experiments," in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers-Volume 2*, pp. 42–47, Association for Computational Linguistics, 2011.

[114] X. Wang, L. Tokarchuk, and S. Poslad, "Identifying relevant event content for real-time event detection," in *Advances in Social Networks Analysis and Mining (ASONAM), 2014 IEEE/ACM International Conference on*, pp. 395–398, Aug 2014.

[115] W. Xie, F. Zhu, J. Jiang, E.-P. Lim, and K. Wang, "Topicsketch: Real-time bursty topic detection from twitter," *IEEE Transactions on Knowledge and Data Engineering*, vol. 28, no. 8, pp. 2216–2229, 2016.

[116] R. Agrawal, T. Imieliński, and A. Swami, "Mining association rules between sets of items in large databases," in *ACM SIGMOD Record*, vol. 22, pp. 207–216, ACM, 1993.

[117] R. Agrawal, R. Srikant, *et al.*, "Fast algorithms for mining association rules," in *Proc. 20th int. conf. very large data bases, VLDB*, vol. 1215, pp. 487–499, 1994.

[118] J. Han, J. Pei, and Y. Yin, "Mining frequent patterns without candidate generation," in *ACM sigmod record*, vol. 29, pp. 1–12, ACM, 2000.

[119] M. J. Zaki, "Scalable algorithms for association mining," *IEEE Transactions on Knowledge and Data Engineering*, vol. 12, no. 3, pp. 372–390, 2000.

[120] Y. Zhu and D. Shasha, "Statstream: Statistical monitoring of thousands of data streams in real time," in *VLDB'02: Proceedings of the 28th International Conference on Very Large Databases*, pp. 358–369, Elsevier, 2002.

[121] P. S. Tsai, "Mining frequent itemsets in data streams using the weighted sliding window model," *Expert Systems with Applications*, vol. 36, no. 9, pp. 11617–11625, 2009.

[122] G. S. Manku and R. Motwani, "Approximate frequency counts over data streams," in *VLDB'02: Proceedings of the 28th International Conference on Very Large Databases*, pp. 346–357, Elsevier, 2002.

[123] C. Giannella, J. Han, J. Pei, X. Yan, and P. S. Yu, "Mining frequent patterns in data streams at multiple time granularities," *Next generation data mining*, vol. 212, pp. 191–212, 2003.

[124] Y. Chi, H. Wang, S. Y. Philip, and R. R. Muntz, "Catch the moment: maintaining closed frequent itemsets over a data stream sliding window," *Knowledge and Information Systems*, vol. 10, no. 3, pp. 265–294, 2006.

[125] J.-d. Ren and K. Li, "Online data stream mining of recent frequent itemsets based on sliding window model," in *Machine Learning and Cybernetics, 2008 International Conference on*, vol. 1, pp. 293–298, IEEE, 2008.

[126] K. Li, Y.-y. Wang, M. Ellahi, and H.-a. Wang, "Mining recent frequent itemsets in data streams," in *Fuzzy Systems and Knowledge Discovery, 2008. FSKD'08. Fifth International Conference on*, vol. 4, pp. 353–358, IEEE, 2008.

[127] H.-F. Li and S.-Y. Lee, "Mining frequent itemsets over data streams using efficient window sliding techniques," *Expert Systems with Applications*, vol. 36, no. 2, pp. 1466–1477, 2009.

[128] J. Cheng, Y. Ke, and W. Ng, "Maintaining frequent itemsets over high-speed data streams," in *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pp. 462–467, Springer, 2006.

[129] H. M. Nabil, A. S. Eldin, and M. A. E.-F. Belal, "Mining frequent itemsets from online data streams: Comparative study," *International Journal of Advanced Computer Science and Applications (IJACSA)*, vol. 4, 2013.

[130] J. W. Perry, A. Kent, and M. M. Berry, "Machine literature searching x. machine language; factors underlying its design and development," *Journal of the Association for Information Science and Technology*, vol. 6, no. 4, pp. 242–254, 1955.

[131] R. Gençay, M. Dacorogna, U. A. Muller, O. Pictet, and R. Olsen, *An introduction to high-frequency finance*. Elsevier, 2001.

[132] B. Mandelbrot and H. M. Taylor, "On the distribution of stock price differences," *Operations research*, vol. 15, no. 6, pp. 1057–1062, 1967.

[133] E. Tsang, "Directional changes, definitions," tech. rep., Working Paper WP050-10 Centre for Computational Finance and Economic Agents (CCFEA), University of Essex Revised 1, 2010.

[134] A. Dupuis and R. Olsen, "High frequency finance: using scaling laws to build trading models," *Handbook of exchange rates*, pp. 563–584, 2012.

[135] A. Mittal and A. Goel, "Stock prediction using twitter sentiment analysis," *Standford University, CS229 (2011 http://cs229. stanford. edu/proj2011/GoelMittal-StockMarketPredictionUsingTwitterSentimentAnalysis. pdf)*, vol. 15, 2012.

[136] G. Ranco, D. Aleksovski, G. Caldarelli, M. Grčar, and I. Mozetič, "The effects of twitter sentiment on stock price returns," *PloS one*, vol. 10, no. 9, p. e0138441, 2015.

[137] J. Si, A. Mukherjee, B. Liu, Q. Li, H. Li, and X. Deng, "Exploiting topic based twitter sentiment for stock prediction.," *ACL (2)*, vol. 2013, pp. 24–29, 2013.

[138] J. Si, A. Mukherjee, B. Liu, S. J. Pan, Q. Li, and H. Li, "Exploiting social relations and sentiment for stock prediction.," in *EMNLP*, vol. 14, pp. 1139–1145, 2014.

[139] T. T. P. Souza, O. Kolchyna, P. C. Treleaven, and T. Aste, "Twitter sentiment analysis applied to finance: A case study in the retail industry," *arXiv preprint arXiv:1507.00784*, 2015.

[140] M. Nofer and O. Hinz, “Using twitter to predict the stock market,” *Business & Information Systems Engineering*, vol. 57, no. 4, pp. 229–242, 2015.

[141] H. Mao, S. Counts, and J. Bollen, “Predicting financial markets: Comparing survey, news, twitter and search engine data,” *arXiv preprint arXiv:1112.1051*, 2011.

[142] Y. Mao, W. Wei, B. Wang, and B. Liu, “Correlating s&p 500 stocks with twitter data,” in *Proceedings of the first ACM international workshop on hot topics on interdisciplinary social networks research*, pp. 69–72, ACM, 2012.

[143] Y. Mao, W. Wei, and B. Wang, “Twitter volume spikes: analysis and application in stock trading,” in *Proceedings of the 7th Workshop on Social Network Mining and Analysis*, p. 4, ACM, 2013.

[144] W. Wei, Y. Mao, and B. Wang, “Twitter volume spikes and stock options pricing,” *Computer Communications*, vol. 73, pp. 271–281, 2016.

[145] E. J. Ruiz, V. Hristidis, C. Castillo, A. Gionis, and A. Jaimes, “Correlating financial time series with micro-blogging activity,” in *Proceedings of the fifth ACM international conference on Web search and data mining*, pp. 513–522, ACM, 2012.

[146] X. Zhang, H. Fuehres, and P. A. Gloor, “Predicting stock market indicators through twitter “i hope it is not as bad as i fear”,” *Procedia-Social and Behavioral Sciences*, vol. 26, pp. 55–62, 2011.

[147] T. Brants and A. Franz, “Web 1t 5-gram version 1,” 2006.

[148] C. W. Granger, “Investigating causal relations by econometric models and cross-spectral methods,” *Econometrica: Journal of the Econometric Society*, pp. 424–438, 1969.

[149] A. Perrin., “Social networking usage: 2005-2015,” Report 1296, Pew Research Center, October 2015.

[150] T. Joachims, “Text categorization with support vector machines: Learning with many relevant features,” in *European conference on machine learning*, pp. 137–142, Springer, 1998.

[151] A. M. Mood, *Introduction to the Theory of Statistics.* NY, USA: McGraw-hill, 1950.

[152] A. Zubiaga, M. Liakata, R. Procter, K. Bontcheva, and P. Tolmie, "Towards detecting rumours in social media," *arXiv preprint arXiv:1504.04712*, 2015.

[153] A. Gupta, H. Lamba, P. Kumaraguru, and A. Joshi, "Faking sandy: characterizing and identifying fake images on twitter during hurricane sandy," in *Proceedings of the 22nd international conference on World Wide Web*, pp. 729–736, ACM, 2013.

[154] T. Owoputi and B. O'Connor, "Twokenize." `https://github.com/brendano/ark-tweet-nlp/blob/master/src/cmu/arktweetnlp/Twokenize.java`, 2014.

[155] M. F. Porter, "An algorithm for suffix stripping," *Program*, vol. 14, no. 3, pp. 130–137, 1980.

[156] S. H. Walker and D. B. Duncan, "Estimation of the probability of an event as a function of several independent variables," *Biometrika*, vol. 54, no. 1-2, pp. 167–179, 1967.

[157] D. W. Hosmer Jr and S. Lemeshow, *Applied logistic regression.* John Wiley & Sons, 2004.

[158] N. Jindal and B. Liu, "Opinion spam and analysis," in *Proceedings of the 2008 International Conference on Web Search and Data Mining*, pp. 219–230, ACM, 2008.

[159] Y.-I. Lou and M.-L. Wang, "Fraud risk factor of the fraud triangle assessing the likelihood of fraudulent financial reporting," *Journal of Business & Economics Research (JBER)*, vol. 7, no. 2, 2011.

[160] S. Viaene, R. A. Derrig, B. Baesens, and G. Dedene, "A comparison of state-of-the-art classification techniques for expert automobile insurance claim fraud detection," *Journal of Risk and Insurance*, vol. 69, no. 3, pp. 373–421, 2002.

[161] R. Maranzato, A. Pereira, A. P. do Lago, and M. Neubert, "Fraud detection in reputation systems in e-markets using logistic regression," in *Proceedings of the 2010 ACM Symposium on Applied Computing*, pp. 1454–1455, ACM, 2010.

[162] C. Phua, V. Lee, K. Smith, and R. Gayler, "A comprehensive survey of data mining-based fraud detection research," *arXiv preprint arXiv:1009.6119*, 2010.

[163] I. Kurt, M. Ture, and A. T. Kurum, "Comparing performances of logistic regression, classification and regression tree, and neural networks for predicting coronary artery disease," *Expert Systems with Applications*, vol. 34, no. 1, pp. 366–374, 2008.

[164] J. Zhu and T. Hastie, "Classification of gene microarrays by penalized logistic regression," *Biostatistics*, vol. 5, no. 3, pp. 427–443, 2004.

[165] P. Fournier-Viger, A. Gomariz, T. Gueniche, A. Soltani, C.-W. Wu, and V. S. Tseng, "Spmf: a java open-source pattern mining library," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 3389–3393, 2014.

[166] G. Salton, *Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computer*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1989.

[167] P. Jaccard, "Étude comparative de la distribution florale dans une portion des alpes et des jura," *Bulletin del la Société Vaudoise des Sciences Naturelles*, vol. 37, pp. 547–579, 1901.

[168] D. D. Lewis, "Representation quality in text classification: An introduction and experiment," in *Speech and Natural Language: Proceedings of a Workshop Held at Hidden Valley, Pennsylvania, June 24-27, 1990*, 1990.

[169] D. D. Lewis, "Evaluating text categorization," in *Proceedings of the Workshop on Speech and Natural Language*, HLT '91, pp. 312–318, Association for Computational Linguistics, 1991.

[170] J. Makhoul, F. Kubala, R. Schwartz, R. Weischedel, *et al.*, "Performance measures for information extraction," in *Proceedings of DARPA broadcast news workshop*, pp. 249–252, Herndon, VA, 1999.

[171] R. C. Fair, "Events that shook the market," *The Journal of Business*, vol. 75, no. 4, pp. 713–731, 2002.

[172] V. Niederhoffer, "The analysis of world events and stock prices," *The Journal of Business*, vol. 44, no. 2, pp. 193–219, 1971.

[173] D. K. Pearce and V. V. Roley, “Stock prices and economic news,” Working Paper 1296, National Bureau of Economic Research, March 1984.

[174] D. M. Cutler, J. M. Poterba, and L. H. Summers, “What moves stock prices?,” *Journal of Portfolio Management*, vol. 15, no. 2, 1989.

[175] R. Forsythe, F. Nelson, G. R. Neumann, and J. Wright, “Anatomy of an experimental political stock market,” *The American Economic Review*, pp. 1142–1161, 1992.

[176] L. Pástor and P. Veronesi, “Political uncertainty and risk premia,” *Journal of Financial Economics*, vol. 110, no. 3, pp. 520–545, 2013.

[177] Thomson Reuters, “Thomson one.com..” `http://www.thomsonone.com/`. [Online accessed 27-November-2015].

[178] P. Kokic, “Standard methods for imputing missing values in financial panel/time series data,” tech. rep., working paper 2, mu ANTARIS Gmb à, Frankfurt am Main, 2001.

[179] Apache Software Foundation, “Apache poi - the java api for microsoft documents.” `http://Poi.apache.org`, 2001. [Online accessed 10-July-2015].

[180] I. H. Witten, E. Frank, M. A. Hall, and C. J. Pal, *The WEKA Workbench. Data Mining: Practical machine learning tools and techniques.* Morgan Kaufmann, fourth edition ed., 2016.

[181] J. R. Quinlan, *C4. 5: programs for machine learning.* Elsevier, 2014.

[182] S. Fong and J. Tai, “The application of trend following strategies in stock market trading,” in *INC, IMS and IDC, 2009. NCM’09. Fifth International Joint Conference on*, pp. 1971–1976, IEEE, 2009.

[183] M. E. Aloud, *Modelling the High-Frequency FX Market: an Agent-Based Approach.* PhD thesis, University of Essex, United Kingdom, 2013.

[184] R. Pardo, *The evaluation and optimization of trading strategies*, vol. 314. John Wiley & Sons, 2011.

[185] E. Gilbert and K. Karahalios, "Widespread worry and the stock market.," in *ICWSM*, pp. 59–65, 2010.

[186] P. C. Tetlock, "Giving content to investor sentiment: The role of media in the stock market," *The Journal of finance*, vol. 62, no. 3, pp. 1139–1168, 2007.

[187] P. C. Tetlock, M. Saar-Tsechansky, and S. Macskassy, "More than words: Quantifying language to measure firms' fundamentals," *The Journal of Finance*, vol. 63, no. 3, pp. 1437–1467, 2008.

[188] F. Lillo, S. Miccichè, M. Tumminello, J. Piilo, and R. N. Mantegna, "How news affects the trading behaviour of different categories of investors in a financial market," *Quantitative Finance*, vol. 15, no. 2, pp. 213–229, 2015.

[189] M. Siering, ""boom" or "ruin"–does it make a difference? using text mining and sentiment analysis to support intraday investment decisions," in *System Science (HICSS), 2012 45th Hawaii International Conference on*, pp. 1050–1059, Jan 2012.

[190] M. Alanyali, H. S. Moat, and T. Preis, "Quantifying the relationship between financial news and the stock market," *Scientific reports*, vol. 3, p. 3578, 2013.

[191] S. M. Stigler, "Francis galton's account of the invention of correlation," *Statistical Science*, pp. 73–79, 1989.

[192] K. Pearson, "Note on regression and inheritance in the case of two parents," *Proceedings of the Royal Society of London*, vol. 58, pp. 240–242, 1895.

[193] R. A. Fisher, "Statistical methods for research workers," in *Breakthroughs in Statistics*, pp. 66–70, Springer, 1992.

[194] HIS Inc, "Eviews 10." `https://store.eviews.com/`, 2017.

[195] C. W. Granger and P. Newbold, "Spurious regressions in econometrics," *Journal of econometrics*, vol. 2, no. 2, pp. 111–120, 1974.

[196] T. M. Nisar and M. Yeung, "Twitter as a tool for forecasting stock market movements: A short-window event study," *The Journal of Finance and Data Science*, vol. 4, no. 2, pp. 101–119, 2018.