University College London

Department of Computer Science

A thesis submitted in partial fulfilment of the requirements for the degree of Master of Science in Computational Finance/Financial Risk Management, University College London

Twitter Event Detection and Topic Modeling

Author
Yiheng Chen

Academic Supervisor
Dr Paolo Barucca
Department of Computer Science
University College London

Industrial Supervisor
Mr Kumar Dixit
Department
Financial Conduct Authority

April 23, 2022

This dissertation is submitted as part requirement for the MSc Computational Finance degree at UCL. It is substantially the result of my own work except where explicitly indicated in the text.

Abstract

In this paper, we propose an abnormal event detection framework. The framework can process tweet data, perform time series analysis based on the number of tweets, analyze anomalies, and extract the topics of tweets within an abnormal time interval. In addition, a classifier is constructed to recognize specific tweets by using the text vectorization methods and four machine learning methods. The classifiers can be used to quickly identify tweets with specific topics and sentiments. The research and verification of this paper are based on the tweet data of 9 banks including Barclays Bank, Santander Bank, and HSBC in the past two months. Tweet collection is based on the Twitter Streaming API, which can collect tweets related to a specific account in real-time. This framework can be used to help financial supervision and has the efficiency advantage of processing large amounts of text data.

Acknowledgments

I would like to thank my academic supervisor Dr Paolo Barucca at Department of Computer Science for his continuous guidance and advice on my thesis work. Further, I would like to thank my industrial supervisor Mr Kumar Dixit at Financial Conduct Authority for providing me with the opportunity of doing my thesis at FCA, as well as for the invaluable feedback and guidance along the way. Finally, I would like to thank my family and friends for their continuous support.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

In recent years, social media such as Twitter has rapidly risen and become an important platform for social groups to share their opinions and insights. Besides, social media has also become one of the major channels for businesses to reach their customers. For example, many banks are serving the needs of their consumers around the clock via their own official Twitter accounts by answering questions. The posts not only provide information to the customers but also serve as an important source of data for tracking abnormal events without geographical restrictions. As a result of this virtual evolution, data analysis on social media has become an important research topic with many applications of the business world and public governance.

It is very difficult to analyze massive tweets and identify abnormal events only by relying on human resources. Therefore, it is necessary to build a computer-assisted system to assist human beings in processing tweets. Nowadays, Natural Language Processing(NLP) technology is developing rapidly, in which the topic modeling methods can be used to extract topics from massive text data. So we propose a framework for processing tweets and anomaly detection, which is based on time series analysis, sentiment analysis, topic modeling methods, and machine learning methods.

This thesis is in cooperation with the Financial Conduct Authority (FCA), which is the official regulator of financial institutions and markets in the UK. In response to regulatory requirements, we collect and analyse tweets that are relevant to the specific bank's official account, such as HSBC and Barclays. We are interested in bank-related abnormal events and have detected payment system failures, fraudulent transfers, virtual currency transactions and other topics from the tweets. Although there have been many papers on topic modeling related to tweets, the contribution of this article is to focus on abnormal event monitoring in the banking field.

1.2 Thesis Goal

There are two main goals in this thesis. The first goal (part A) is to perform time-series anomaly detection methods and then using topic modeling methods to analyze the topic distribution of tweets in the abnormal time interval. The second target (part B) is to build supervised learning classifiers by using text-vectorization methods and machine learning methods. The prerequisite for training such a classifier is to transfer the tweets into a labeled data set. For data labeling, sentiment analysis and topic modeling are applied.

1.3 Thesis Structure

In this thesis, chapter 2 explains the main body of mathematical principles which are applied in this thesis, including the time series decomposition model in Section 2.1 and the LDA model in Section 2.2. In Chapter 3, we propose the frameworks of this project. Firstly we introduced how data is collected and processed in Section 3.1. Then we introduce two frameworks in Section 3.2 and Section 3.3 for achieving the two goals. In Chapter 4, we first explore the whole data set in section 4.1. Then we represent the result of part A and part B in Section 4.2 and Section 4.3 respectively. In Chapter 5, we summarize the main results of this project and discuss the direction for future research.

Chapter 2

Theory

2.1 Decomposable Time Series Model

In Harvey and Peters (1990), a decomposable time series model is proposed for describing time series data. There are three main components: trend, seasonality, and events, as seen in Equations 2.1:

$$y(t) = g(t) + s(t) + h(t) + \epsilon_t \tag{2.1}$$

where $g(t)$ represents nonperiodic changes in time series, $s(t)$ is the changes in seasonal fashion, for example, weekly or monthly, and $h(t)$ models the effect of holidays on an irregular basis. ϵ_t is an error term for describing unpredictable changes or noise.

2.1.1 The Trend Model

The Trend is a key factor in the growth model. This component describes the main direction of change in the entire time series. For example, the main trend predicted by Facebook's users is approximated as the population growth in nature (Taylor and Letham (2018)). In this thesis, the number of tweets sent to bank customer service accounts is related to the number of bank customers, so the growth of bank customers is the main trend that affects the number of tweets. The basic form of growth in logistic growth model is:

$$g(t) = \frac{C}{1 + \exp(-k(t - m))} \tag{2.2}$$

where C shows maximum carrying ability, for example, in this thesis this term can be explained as the maximal customer number of banks that can be achieved; k represents growth speed, i.e. the number of increasing customer numbers of a bank; m is an offset parameter for no actual meaning.

The above formula represents a naive situation. In real life, the capacity C and the growth rate k at different times are changing, rather than constant. In the optimized

model, the capacity C becomes a time-varying variable $C(t)$, meaning the capacity is changing at different times. There are S change-points at which time s_j the growth rate k is allowed to change, and $\delta \in \mathbb{R}^S$, where δ_j represents the change of growth rate at time point s_j. The calculation of adjustment in j is:

$$\gamma_j = (s_j - m - \sum_{l<j} \gamma_l)(1 - \frac{k + \sum_{l<j} \delta_l}{k + \sum_{l \leq j} \delta_l}) \tag{2.3}$$

More complex, the piecewise logistic growth model, which changes the growth rate over time, can expressed as:

$$g(t) = \frac{C(t)}{1 + \exp(-(k + a(t)^T \delta)(t - (m + a(t)^T \gamma)))} \tag{2.4}$$

where:

$$a_j(t) = \begin{cases} 1, & \text{if } t > s_j. \\ 0, & \text{otherwise.} \end{cases} \tag{2.5}$$

2.1.2 The Seasonality Model

By using Fourier series, arbitrary smooth seasonal effects can be approximated by:

$$s(t) = \sum_{n=1}^{N} (a_n \cos(\frac{2\pi nt}{P}) + b_n \sin(\frac{2\pi nt}{P})) \tag{2.6}$$

where P stand for a period of time, for example, $P = 7$ means predict for weekly data. The β is defined as a vector with 2N size $[a_1, b_1, ..., a_N, b_N]^T$, which is needed for fitting seasonality. This vector can be estimated by building a matrix with previous and future data. If the seasonality is weekly, and we set $N = 7$, then we have:

$$X(t) = [\cos(\frac{2\pi(1)t}{7}), ..., \sin(\frac{2\pi(10)t}{7})] \tag{2.7}$$

Finally the seasonality can be computed as:

$$s(t) = X(t)\beta \tag{2.8}$$

with $\beta \sim Normal(0, \sigma^2)$.

2.1.3 The Event Factor

Holidays or major events, such as plague outbreaks or parades, are not periodic, and these have a greater impact on the time series. It is also true for people tweeting online. For example, festivals may cause people to have more free time to communicate online instead of spending on work, or under the current COVID-19 impact, serious The outbreak may

result in the temporary closure of local bank branches, leading to an increase in online complaints.

In this model, each event is annotated as i, and D_i represents a set of continuous-time series for this event. Then an indicator function can be built for determining whether a specific time t is in the event i:

$$Z(t) = [1(t \in D_1), ..., 1(t \in D_L)] \tag{2.9}$$

$$h(t) = Z(t)k \tag{2.10}$$

where k is parameter for each event i, representing changes in the future with $k \sim Normal(0, v^2)$.

2.2 Latent Dirichlet Allocation

Latent Dirichlet Allocation (LDA) is a classic generative probalistic model introduced Blei et al. (2003). This method is usually applied for topic modeling. Topic modeling is using statistical methods or machine learning methods to identify semantic topics from a collection of documents, and address mapping relationships of documents to topics and topics to words. There are wide applications of this technique, for example social network monitoring, information retrieval, or recommendation systems.

2.2.1 Terminology

In this subsection, we define the topic modeling problem in a mathematical way and explain the notation for that in this subsection. In the bag-of-words representation, the order of words in the sentence is ignored, and the document is represented by one-hot-encoding. The document d_m is defined as a collection of words, where $d_m = \{w_{m,1}, w_{m,2}, ..., w_{m,N}\}$. In this thesis, each tweets can be defined as a document. The collection of documents is *corpus*, where $D = \{d_1, d_2, ..., d_M\}$. The collection of all unique words in *corpus* is *dictionary (V)*, where $V = \{w_1, w_2, ..., w_N\}$. Topic Z is defined as latent concept, consist of a collection of words which belongs to this topic, where $Z = \{z_1, z_2, ..., z_k\}$.

2.2.2 Structure of Probability Graph

Figure 2.1 shows the process of generating topics and words by LDA method. The following steps are assumed to generate a document:

1 For each document d_m in corpus:

1.1 draw a document-topic distribution $\vec{\Theta}_m \sim Dirichlet(\alpha)$

1.2 draw a topic assignment $z_{m,n} \sim Multinomial(\vec{\Theta}_m)$

2 For each words in document d_m:

1.1 draw a topic-word distribution $\vec{\varphi}_k \sim Dirichlet(\beta)$

1.2 draw a word $w_{m,n} \sim Multinomial(\varphi_{z_{m,n}})$

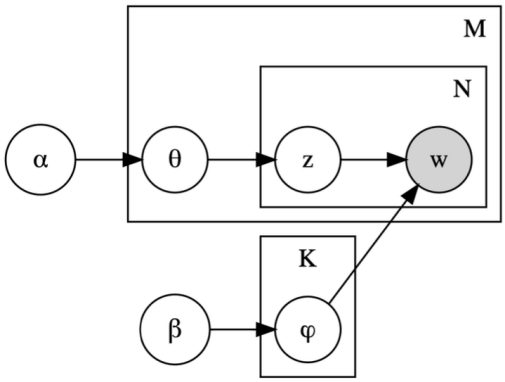


Figure 2.1: Plate Model presentation for LDA (Blei et al. (2003))

In this algorithm, the parameter α and β parameterize two Dirichlet distributions. In topic modeling, an intuitive explanation for these two parameters is: α controls the document-topic density, and the α controls the topic-word density. The LDA process has the following distribution:

$$p(D, z, \varphi, \Theta \mid \alpha, \beta) = \prod_{m=1}^{M} p(\theta_m \mid \alpha) \prod_{k=1}^{K} p(\varphi_k \mid \beta)(\prod_{n=1}^{N} P(w_{m,n} \mid z_{m,n}, \varphi_k) P(z_{m,n} \mid \theta_m) \tag{2.11}$$

where $D = \{d_1, d_2, ..., d_M\}$ is the whole documents, $w_{m,n}$ is the occurrence of word w_n in document d_m, and in the representation of one-hot-encoding, $z_{m,n}$ is the topic of word w_n in document d_m,$\Theta_m \in \mathbb{R}^K$ is the proportion of topics in d_m, $\Theta_{m,k}$ is the proportion of topics z_k in d_m, $\varphi_k \in \mathbb{R}^N$ is the distribution of word occurrence, and $\varphi_{k,n}$ is the word occurrence of word w_n in topic z_k.

$$p(\Theta_m \mid \alpha) = \frac{\Gamma(\sum_k \alpha_k)}{\prod_k \Gamma(\alpha_k)} \prod_k \Theta_{t,k}^{\alpha_k - 1} \tag{2.12}$$

For a given $D = \{d_1, d_2, ..., d_M\}$, the hyperparameters of LDA can be calculated using *Maximum Likelihood Estimation*:

$$LL(\alpha, \beta) = \sum_{d=1}^{M} \log p(w_d \mid \alpha, \beta) \tag{2.13}$$

The key problem in topic modeling is computing the posterior distributions of the latent variables in the model given the observed data by reversing the generation process(Darling (2011)). By assuming to all hyperparameters are given, topic and word distribution can be calculated based on the word frequency $w_{m,n}$ which is easily calculated for a given corpus:

$$p(z, \varphi, \Theta \mid D, \alpha, \beta) = \frac{p(D, z, \varphi, \Theta \mid \alpha, \beta)}{p(D \mid \alpha, \beta)} \tag{2.14}$$

In the practice, Gibbs Sampling is used for approximate calculation because $p(D \mid \alpha, \beta)$ is in general intractable to compute (Blei et al. (2003)). In the next section, we will discuss the application of approximate inference algorithms for solving this problem. Gibbs sampling methods are chosen for this calculation in this thesis.

2.2.3 Gibbs Sampling

Gibbs sampling is one kind of Markov Chain Monte Carlo (MCMC) algorithms (Gilks et al. (1995)). The aim of MCMC methods is to construct a Markov chain, by which a stationary distribution can be achieved. After convergency, the stationary distribution is approximately equal to the target posterior distribution. In the application in LDA, this method is used to compute document-topic distribution Θ, topic-word distribution φ, and topic assignment for words z. In Darling (2011), *Collapsed Gibbs Sampler* is introduced for computing the desired topic assignment, which is defined as:

$$p(z_i \mid z_{-i}, \alpha, \beta, w) \tag{2.15}$$

where z^{-i} is the notation to represent the allocation of all topics except for z_i. Using chain rules, the Gibbs sampling equation for LDA is given by Darling (2011) as:

$$p(z_i \mid z_{-i}, \alpha, \beta, w) \propto (n_{d,k}^{-i} + \alpha_k) \frac{n_{k,w}^{-i} + \beta_w}{\sum_w n_{k,w}^{-i} + \beta_w} \tag{2.16}$$

Chapter 3

Proposed Frameworks

In the introduction, we have mentioned that there are two tasks in this thesis: one is to use time series anomaly detection and topic modeling for abnormal event monitoring. The other is to build an event classifier using machine learning algorithms. In this chapter, we introduce the two frameworks in Section 3.2 and Section 3.3 separately. In Section 3.1, we first show the process of data collection pre-processing applied in this work and get familiar with the tweets data.

3.1 Data Collection

In this section, we introduce the whole process of data collection and cleaning. We first introduce the function of Twitter API and its application in this project, as well as technical details. Then we introduce the Twitter account information of the research target bank. After this, we introduce how to preprocess the collected tweets for model learning, including the process of aggregating tweet paragraphs into long text, and cleaning irrelevant vocabulary of tweets.

3.1.1 Twitter Streaming API

Although there are many ways to collect tweets, such as web crawling, there are many latent risks, for example, illegal authorization or missing important data. Fortunately, Twitter provides developers with official, legal, and efficient tweets collection tools for academic usage purposes. In this thesis, we use the *Twitter Streaming API*[1]. This API can collect tweets in real-time by searching keywords or tracking users. It is based on the HTTP protocol and uses JSON format to transmit data, which includes detailed tweets and user information. In this thesis, for maintaining real-time characteristics of data, we choose this API for the data collection process. The *Tweepy*[2] python library is used to

[1] https://developer.twitter.com/en/docs/twitter-api/v1/tweets/filter-realtime/guides/basic-stream-parameters

[2] https://www.tweepy.org/

connect the API.

Authentication, Connection, Listening, and Disconnection are the four stages of using the Streaming API. Developers must apply for official authorisation from the Twitter developer page in order to utilize the API. To connect to the API, the developer will receive the four strings: Consumer Key, Consumer Secret, Access Token, and Access Token Secret after a successful application. These four strings need to be provided when connecting to the API for determining the user's identity.

For each tweet, we collected the following information: tweet content, tweet release time, tweet id, publisher username, user description, user area, tweet-related bank, tweet forwarding number, tweet response number, The number of likes of tweets. The sample information of collected tweets can be seen in Figure 3.1. The "text" feature is used for topic modeling. In the text, the symbol is used to know the information of which accounts are mentioned by these tweets. The "created" feature is used to record the time when the tweets are sent and convert the tweets data set into time series for anomaly detection. The "tweets_id" can be used as a unique ID to identify tweets.

	id	text	bank_relevant	created	tweet_id	retweets	reply	favorite	screenname	description	loc
0	1	@BarclaysUKHelp Hi Billy, noted thank you so m...	Barclays,	2021-07-09T10:39:03	1413447603774103554	0	0	0	bumpercarshardy	she/her	NaN
1	2	@NatWest_Help Oh that's a shame it was a great...	NatWest,	2021-07-09T10:39:49	1413447795093106690	0	0	0	LouLou_No_72	Premier League Music Television Literature Com...	London
2	3	@Jamie_ORourke @benncoxx @TwittyBen @Tommylost...	Santander,	2021-07-09T10:40:04	1413447856074018818	0	0	0	gadget78	I am philosophically an atheist humanist,polit...	UK
3	4	@Tetrisdroid @AndrewDriver @santanderuk @santa...	Santander,	2021-07-09T10:40:50	1413448052577239040	0	0	0	KateWilliams_10	NaN	NaN
4	5	@gadget78 @benncoxx @TwittyBen @Tommylostaccou...	Santander,	2021-07-09T10:41:14	1413448152896610307	0	0	0	Jamie_ORourke	Jamie O'Rourke Actor with a ginger beard http...	Hereford/Wales/UK
...	...	...	...	...	...	...	...	...	...	...	...
17813	17814	@TSB I keep trying to log into my internet ban...	TSB Bank,	2021-08-20T01:23:22	1428528049004457985	0	0	0	Natalie28773855	NaN	NaN
17814	17815	@BarclaysUKHelp @Crypto906932337 Can you also ...	Barclays,	2021-08-20T01:29:19	1428529545729044488	0	0	0	RufusMuMu	Crypto\n\nBitcoin\n\nEthereum\n\nAltcoins\n\n🙌👌	NaN
17815	17816	@_TheRealAlex_ Hi, Alex. Apologies for any fru...	Barclays	2021-08-20T01:31:34	1428530115458314244	0	0	0	BarclaysUKHelp	Need to ask us something? We're here to help y...	NaN
17816	17817	@BarclaysUKHelp @MrNobody707 Please can you ex...	Barclays,	2021-08-20T01:32:03	1428530235805290498	0	0	0	RufusMuMu	Crypto\n\nBitcoin\n\nEthereum\n\nAltcoins\n\n🙌👌	NaN

Figure 3.1: Tweets data sample.

3.1.2 Banks of Collection

There are two modes for collecting tweets by this API, either by following the User ID of our interest or tracking the keywords containing in tweets. In this thesis, we use the mode *Follow* because we are interested in all tweets relevant to specific bank accounts. All tweets sent by related accounts and tweets sent by other users mentioning these banks will be collected. The bank account lists are shown in Table 3.1.

3.1.3 Data Preprocessing

Data Preprocessing is a crucial step before applying models on the data because the raw data in the real world contains noise, outlier, or redundance which have a negative influence on the result. The tweet data collected from the API needs to be preprocessed before being processed by the topic model. In addition to the topic-related vocabulary, these tweet data include URLs, emojis, short words, misspellings, and meaningless words. In the preprocessing process, they need to be appropriately filtered and deleted. The data

Bank of Interest		
Name of Banks	Official Twitter Accounts	Account IDs
Barclay UK	@$BarclaysUKHelp$	3046525515
Co-op	@$CooperativeBank$	23970506
HSBC	@$HSBC\_UK$	2922732233
Lloyds	@$LloydsBank$	147932302
NatWest	@$NatWest\_Help$	284540385
Santander	@$santanderukhelp$	962692878
TSB Bank	@TSB	746954528
Virgin Money	@$VirginMoney$	22484810
Virgin Money	@$AskVirginMoney$	2463268225
Clydesdale Bank	@$askclydesdale$	2496100651
Clydesdale Bank	@$clydesdalebank$	2327070487

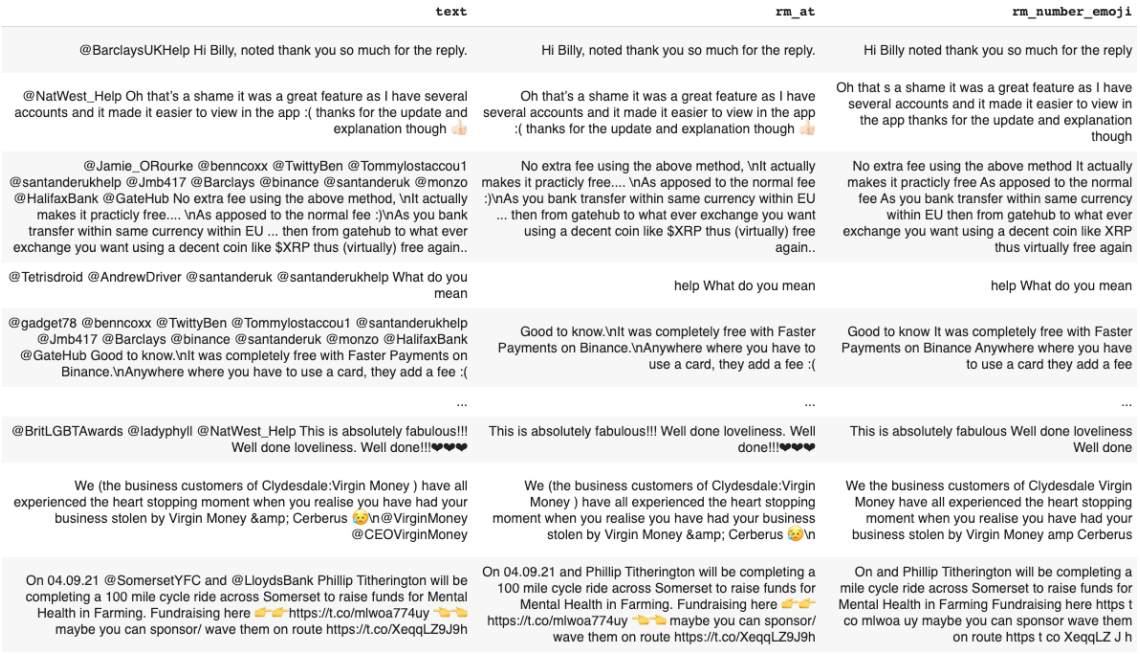
Table 3.1: This table lists Twitter account names and IDs of banks used for data collection in this thesis.

cleaning process is shown Figure 3.2 and Figure 3.3.

Remove URLs and special characters Tweets often contain URL links to other websites. These URLs are long and have no meaning, so they need to be deleted. Besides, punctuations, numbers, emojis, and other special characters are not in the member of English words, so they need to be cleared out. We use the regular expressions library in python to identify strings containing 'https', 'www', and special characters and delete them from the tweets.

Remove Short or Infrequent Words In order to avoid over-learning, we delete words that appear less than 30 times in the entire tweet document. In addition, words that are too short are also deleted. In this project, words with a character length of less than three are also deleted.

Remove Stopwords Stop-words are defined as the words that are frequently used without special meaning, for example, 'is', 'in', 'the'. These words in the English language do not connect to any specific topics, therefore they need to be deleted when they occur in collected tweets. In this thesis, we use the python library *NLTK* to perform the stopwords removing, which contains a library of stopwords. During the text processing, words in tweets that are also included in the stopwords dictionary would be cleared out. Because the tweets we collected are mainly composed of bank customers' complaints and bank service staff's responses, some common vocabulary words used frequently in the conversation have also been deleted, such as 'help', 'thank', 'please', 'sorry', 'yeah' etc. These words are not included in the *NLTK* library, so we create a customized list to perform the delete.



text	rm_at	rm_number_emoji
@BarclaysUKHelp Hi Billy, noted thank you so much for the reply.	Hi Billy, noted thank you so much for the reply.	Hi Billy noted thank you so much for the reply
@NatWest_Help Oh that's a shame it was a great feature as I have several accounts and it made it easier to view in the app :(thanks for the update and explanation though 👍🏻	Oh that's a shame it was a great feature as I have several accounts and it made it easier to view in the app :(thanks for the update and explanation though 👍🏻	Oh that s a shame it was a great feature as I have several accounts and it made it easier to view in the app thanks for the update and explanation though
@Jamie_ORourke @benncoxx @TwittyBen @Tommylostaccou1 @santanderukhelp @Jmb417 @Barclays @binance @santanderuk @monzo @HalifaxBank @GateHub No extra fee using the above method, \nIt actually makes it practicly free.... \nAs apposed to the normal fee :)\nAs you bank transfer within same currency within EU ... then from gatehub to what ever exchange you want using a decent coin like $XRP thus (virtually) free again..	No extra fee using the above method, \nIt actually makes it practicly free.... \nAs apposed to the normal fee :)\nAs you bank transfer within same currency within EU ... then from gatehub to what ever exchange you want using a decent coin like $XRP thus (virtually) free again..	No extra fee using the above method It actually makes it practicly free As apposed to the normal fee As you bank transfer within same currency within EU then from gatehub to what ever exchange you want using a decent coin like XRP thus virtually free again
@Tetrisdroid @AndrewDriver @santanderuk @santanderukhelp What do you mean	help What do you mean	help What do you mean
@gadget78 @benncoxx @TwittyBen @Tommylostaccou1 @santanderukhelp @Jmb417 @Barclays @binance @santanderuk @monzo @HalifaxBank @GateHub Good to know.\nIt was completely free with Faster Payments on Binance.\nAnywhere where you have to use a card, they add a fee :(	Good to know.\nIt was completely free with Faster Payments on Binance.\nAnywhere where you have to use a card, they add a fee :(	Good to know It was completely free with Faster Payments on Binance Anywhere where you have to use a card they add a fee
...	...	...
@BritLGBTAwards @ladyphyll @NatWest_Help This is absolutely fabulous!!! Well done loveliness. Well done!!!❤❤❤	This is absolutely fabulous!!! Well done loveliness. Well done!!!❤❤❤	This is absolutely fabulous Well done loveliness Well done
We (the business customers of Clydesdale:Virgin Money) have all experienced the heart stopping moment when you realise you have had your business stolen by Virgin Money & Cerberus 😠\n@VirginMoney @CEOVirginMoney	We (the business customers of Clydesdale:Virgin Money) have all experienced the heart stopping moment when you realise you have had your business stolen by Virgin Money & Cerberus 😠\n	We the business customers of Clydesdale Virgin Money have all experienced the heart stopping moment when you realise you have had your business stolen by Virgin Money amp Cerberus
On 04.09.21 @SomersetYFC and @LloydsBank Phillip Titherington will be completing a 100 mile cycle ride across Somerset to raise funds for Mental Health in Farming. Fundraising here 👉👉https://t.co/mlwoa774uy 👈👈 maybe you can sponsor/ wave them on route https://t.co/XeqqLZ9J9h	On 04.09.21 and Phillip Titherington will be completing a 100 mile cycle ride across Somerset to raise funds for Mental Health in Farming. Fundraising here 👉👉 https://t.co/mlwoa774uy 👈👈 maybe you can sponsor/ wave them on route https://t.co/XeqqLZ9J9h	On and Phillip Titherington will be completing a mile cycle ride across Somerset to raise funds for Mental Health in Farming Fundraising here https t co mlwoa uy maybe you can sponsor wave them on route https t co XeqqLZ J h

Figure 3.2: Twitter Data Cleaning Part A: This picture shows the process of tweet cleaning. The *text* column shows the collected raw data. *rm_at* column represents the text with the @ symbol removed. *rm_number_emoji* removes punctuation, special characters and numbers.

Word Stemming This step is used to convert English vocabulary into roots. The meaning of this is to make data more informatively concentrated, reduce word redundancy, and make it easier for models to process. For example, words 'achieving', 'achieved', 'achieves' are stemmed into 'achieve'. In this project, stemming is implemented by using *NLTK.PorterStemmer*[3] software.

Tokenization Tokenization is used for splitting sentences into small units, for example, words. This is a necessary procedure for use the data as input for topic modeling approaches.

3.1.4 Data Aggregation

For the LDA-U model, tweets that are sent from the same users should be aggregated into one long tweet. This process is implemented by grouping tweets by the feature "screenname". Thus the new data set can be used by the LDA-U.

[3]https://www.nltk.org/howto/stem.html

rm_shortword	rm_url	rm_stopword_stemming	tokenize
Billy noted thank you much for the reply	Billy noted thank you much for the reply	billi note thank much repli	[billi, note, thank, much, repli]
that shame was great feature have several accounts and made easier view the app thanks for the update and explanation though	that shame was great feature have several accounts and made easier view the app thanks for the update and explanation though	shame great featur sever account made easier view app thank updat explan though	[shame, great, featur, sever, account, made, easier, view, app, thank, updat, explan, though]
extra fee using the above method actually makes practicly free apposed the normal fee you bank transfer within same currency within then from gatehub what ever exchange you want using decent coin like XRP thus virtually free again	extra fee using the above method actually makes practicly free apposed the normal fee you bank transfer within same currency within then from gatehub what ever exchange you want using decent coin like XRP thus virtually free again	extra fee use method actual make practicli free appos normal fee bank transfer within currenc within gatehub ever exchang want use decent coin like xrp thu virtual free	[extra, fee, use, method, actual, make, practicli, free, appos, normal, fee, bank, transfer, within, currenc, within, gatehub, ever, exchang, want, use, decent, coin, like, xrp, thu, virtual, free]
help What you mean	help What you mean	help mean	[help, mean]
Good know was completely free with Faster Payments Binance Anywhere where you have use card they add fee	Good know was completely free with Faster Payments Binance Anywhere where you have use card they add fee	good know complet free faster payment binanc anywher use card add fee	[good, know, complet, free, faster, payment, binanc, anywher, use, card, add, fee]
...	...	...	...
This absolutely fabulous Well done loveliness Well done	This absolutely fabulous Well done loveliness Well done	absolut fabul well done loveli well done	[absolut, fabul, well, done, loveli, well, done]
the business customers Clydesdale Virgin Money have all experienced the heart stopping moment when you realise you have had your business stolen Virgin Money amp Cerberus	the business customers Clydesdale Virgin Money have all experienced the heart stopping moment when you realise you have had your business stolen Virgin Money amp Cerberus	busi custom clydesdal virgin money experienc heart stop moment realis busi stolen virgin money amp cerberu	[busi, custom, clydesdal, virgin, money, experienc, heart, stop, moment, realis, busi, stolen, virgin, money, amp, cerberu]
and Phillip Titherington will completing mile cycle ride across Somerset raise funds for Mental Health Farming Fundraising here https mlwoa maybe you can sponsor wave them route https XeqqLZ	and Phillip Titherington will completing mile cycle ride across Somerset raise funds for Mental Health Farming Fundraising here mlwoa maybe you can sponsor wave them route XeqqLZ	phillip titherington complet mile cycl ride across somerset rais fund mental health farm fundrais mlwoa mayb sponsor wave rout xeqqlz	[phillip, titherington, complet, mile, cycl, ride, across, somerset, rais, fund, mental, health, farm, fundrais, mlwoa, mayb, sponsor, wave, rout, xeqqlz]

Figure 3.3: The *rm_shortword* column shows the text in the last column in the above figure after removing the words with length of characters less than 3. The *rm_url* column shows text with url removed, for example links start with 'www' or 'https'. The *rm_stopword_stemming* column shows the text with stopwords removed, for example 'the', 'for', and all other remaining words are stemmed. The *tokenize* column shows the process of tokenizing the text, so that the word is converted into tokens.

3.2 Part A: Anomaly Detection and Topic Modeling

In this chapter, we use time-series detection techniques and topic models for abnormal event detection and extraction of information. The assumption for time series anomaly detection in this thesis is: There are correlations between tweets number and events of banks. For example, the bank mobile application offline may raise the increasing complaints and reports from clients, and part of this feedback will be in the form of tweets. Therefore, the number of tweets is a significant feature for event detection. When it comes to topic modeling, it is noteworthy that it is impossible to rely on manual methods to understand the subject of the tweet due to the huge amount of data in this internet age. On the one hand, topic determining depending on humans is expensive and inflexible: when the number of tweets increases exponentially, more manpower than usual is required to read the tweets, and this labor is hard to be employed in a short time. On the other hand, this approach is time-consuming. Compared with human approaches, the algorithm-based topic modeling method has the characteristics of fast, cost-saving, and flexible, therefore has huge advantages. The goal of Part A is to use time series analysis to find the abnormal time interval based on the tweets number, and use topic modeling methods to extract valuable topics of the tweets in the unusual time span. The overall project framework is shown in Figure 3.4.

3.2.1 Time Series Formation

When collecting the tweets data, the created time as feature of tweet is also included. Therefore, we are able to index tweets by time. The first step in part A is convert the collected data into time series, which is defined in this thesis as **a sequence of number of tweets, counted every hours by its created time**. In this time series, every data points is indexed by time, and its value represents the number of tweets collected in this time interval. There is various parameters can be chosen, for example tweets can be counted every three hours or by days, even by weeks or months. The smaller the time interval for calculating the number of tweets, the more time series data points will be obtained, and the model will be more sensitive to abnormal points. This increases the sensitivity of the model, and on the other hand causes the model more unstable. We choose hourly basis for time series formation is because the limitation of the data size. The collection of data is time-consuming. In an trade-off between the data size and data quality, we choose this parameter for constructing the time series.

3.2.2 Outlier Detection

Generally, outlier is defined as timestamp-value pair $\langle t, x_t \rangle$ in a time series x, where the observed value x_t is *significantly* different from the expected value $\mathbb{E}(x_t)$ (Laptev et al.

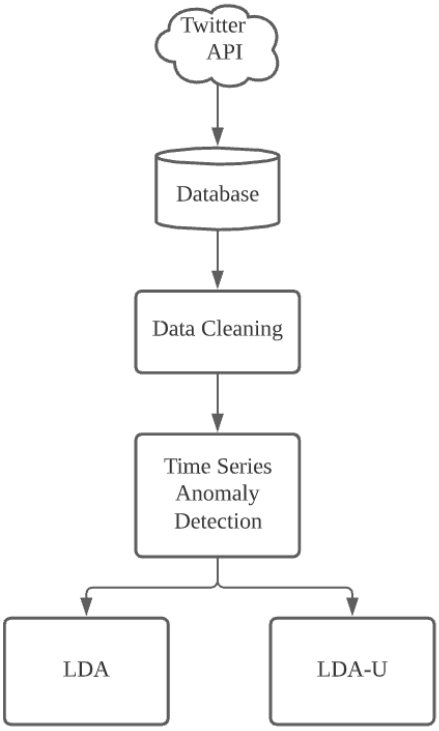


Figure 3.4: Framework A: Anomaly Detection and Topic Modeling

(2015)). In this thesis, the outlier represents potential event that occurs unusual, and it results in complaint from customer of banks. The goal of outlier detection is to output the anomaly time span in which the tweets number is usual.

There are two approaches for outlier detection: plug-in methods and decomposition-based methods (Laptev et al. (2015)). The first method models normal behavior of time series and detect outlier based on the deviation of observed data from predicted data. The second method is based on noise component of time series decomposition model. Data points whose noise value exceeds the threshold will be marked as abnormal. In this thesis, we use the second approach. The implementation is by using Facebook software *Prophet*.

Prophet[4] is an open-source software developed by the Facebook team, which is based on *an additive model* (Harvey and Peters (1990)). This method has advantages over the generative model in flexibility, less work in data pre-processing, fast-fitting speed and interpretable parameters (Taylor and Letham (2018)).

The *Prophet* and *Time Series Decimposition Model* are chosen for this work because the tweets data have the following characters that can be well process by the model, which are:

- Seasonality: Intuitively, the number of tweets follows a seasonal trend. In a working

[4]https://facebook.github.io/prophet/

day, tweets during non-working hours should be more than during working hours; tweets during nights is supposed to be less than during the day. In a scope of week, the number of tweets from working days should be more than during the weekend. These assumptions are validated by the time series analysis result in the Chapter 4.2.

- Missing data or large outlier: Tweet data is collected in real time. Due to API and network quality reasons, data collection may be incomplete and there are missing points. After an abnormal event breaks out, the number of tweets will increase exponentially, so there will be extremely large outliers.

- Irregular events: Some irregular large-scale events will also affect the number of tweets. For example, legal demonstrations approved by the government in advance will cause some roads to be temporarily closed. Customers who did not know the news in advance will found that financial services cna be be provided when they went to the closed branches. They may complain this to the bank customer service on Twitter, which causes the number of tweets increase significantly.

- Non-linear growth trends: Theoretically, there is a positive correlation between the number of bank-related tweets and the growth of bank customers. The more customers there are, the more feedbacks and inquires will be sent to bank official twitter account. The growth of customers is non-linear and will be limited, because banks cannot be unlimited expand his business.

As can be seen on the *website of prophet*[5], the above characteristics enable the data to be well fitted and predicted by the model. In Chapter 4.2, we show the results of time series analysis of tweets and describe in detail the characteristics of data related to banks.

[5]https://research.fb.com/prophet-forecasting-at-scale/

	Document_No	Dominant_Topic	Topic_Perc_Contrib	Keywords	Text
4	4	2.0	0.7675	money, payment, card, binanc, cryptocurr, safe, crypto, natwest, block, stop	Good to know. It was completely free with Faster Payments on Binance. Anywhere where you have to use a card, they add a fee :(Good to know. It was completely free with Faster Payments on Binance. Anywhere where you have to use a card, they add a fee :(Whichever bank advertises the ability to freely transfer the customers own funds and use binance are going to make a killing as everyone will go to them No, as I said, if you did a faster payments transfer of fiat from the bank it was completely free. Any card transaction anywhere in the world has a fee, it depends who picks up the tab, for most crypto purchases its the customer. https://t.co/NaauUlju3B But currently the option to do this transfer is suspended due to FUD, attacks on Binance and banks blocking transfers Cheers :)
5	5	0.0	0.7884	call, app, team, messag, work, servic, number, abl, onlin, account	Ok, thanks, so my online banking has to be working? Unfortunately mine stopped working a while ago. Is there a way to fix it without going into a branch? This is what I was told before. great, thanks We would recommend calling us on this number: https://t.co/pseyiCITFd. The team can check your access then assist you in getting back online. ^PM Youre welcome Alex. ^PM
6	6	3.0	0.5485	account, branch, today, open, close, last, name, amp, post, code	I Prefer to go to one of your branches but I'm on holidays so can my friend fix that for me , I mean he can go any of your branches I get that thanks Sir We wouldnt be able to discuss this matter with anyone thats not named on the account. If there is an account in your name, wed need to speak to you to resolve this matter. ^PM Youre welcome. ^PM

Figure 3.5: This figure represents how the dominant topic dataframe looks like in our data set. In this figure, each row shows an aggregated tweet instance. The second column is the document number. The third column is the assigned dominant topic number by LDA-U. the fourth column is the proportion of the dominant topic to all topics. The fifth column shows the keywords for the respective dominant topic. The last column is the aggregated tweet text.

3.2.3 Topic Modeling

In this thesis, we use two topic modeling methods to model the topics from tweets, one is the traditional LDA model (LDA), and the other is the LDA-U model (Jónsson and Stolee (2015)) (LDA-U). The difference between them is not the model itself, but the training data set. LDA uses the original tweets data set, and LDA-U uses the long text data set which is aggregated from several short texts. LDA-U is proposed to alleviate the inefficiency in the short text (Hong and Davison (2010)). In this thesis, we perform a user-based aggregation for obtaining datasets from LDA-U. Tweets with the same "screenname" attribute would be combined together. We use the *Gensim*[6] software to implement the LDA and LDA models.

LDA generates document-topic distribution and topic-word distribution. Therefore, each document (tweet) is assigned with a topic distribution. We set the topic with the highest probability in each tweet as the dominant topic of the tweet, and use the topic number to label this tweet. This process is shown in Figure 3.5. In the LDA-U, the aggregated tweets instead of the original tweets are labeled.

[6]https://radimrehurek.com/gensim/

3.2.4 Evaluation Metrics

In hyperparameter tuning, appropriate evaluation metrics are needed to indicate the performance of the model. The mainstream evaluation methods for topic modeling methods can be roughly divided into:

1. Perplexity
2. Coherence Scores
3. Human Evaluation

Perplexity Perplexity is proposed by Brown et al. (1992) as a metric to evaluate the similarity between actual text and text predicted by models. The lower this metric, the better is the prediction performance. Log-likelihood and cross-entropy are calculated in this evaluation method. However, this metric negatively affects human evaluation and understandability (Chang et al. (2009)). Thus in our model, other metrics will be used for achieving better interpretability of our topics. The formula of perplexity is defined as:

$$L(w) = \log p(w \mid \Phi, \alpha) = \sum_{d} \log p(w_d \mid \Phi, \alpha) \tag{3.1}$$

$$Perplexity(w) = \exp\{\frac{L(w)}{countoftokens}\} \tag{3.2}$$

Where w is test set, Φ represents the given topic matrix, the α parameter determines the topic distribution, and w refers to the document that need to predict.

Coherence Values Coherence scores are proposed to be a class of promising metrics for evaluating topic models, and it shows a positive correlation to understandability by humans(Chang et al. (2009)). Newman et al. (2010) proposed UCI, based on *pointwise mutual information*(PMI). This metric measures the statistical independence of observing two words in close proximity (Newman et al. (2010)). The formula of this method is defined as:

$$PMI(w_i, w_j) = \log \frac{P(w_i, w_j) + \epsilon}{P(w_j)} \tag{3.3}$$

UCI calculates point-wise mutual information for word pairs based on sliding windows. The UCI coherence can be defined as the following formula:

$$C_{UCI} = \frac{2}{N(N-1)} \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} PMI(w_i, w_j) \tag{3.4}$$

Then Mimno et al. (2011) proposed $UMass$. $UMass$ is an intrinsic measurement, and it uses document concurrency counts and logarithmic conditional probability for calculating the score. Besides, each word only forms a word pair with the word in front of it,

which is also called one-preceding segmentation. This metric is defined mathematically as:

$$C_{UMass} = \frac{2}{N \cdot (N-1)} \sum_{i=2}^{N} \sum_{i-1}^{j=1} \log \frac{P(w_i, w_j) + \epsilon}{P(w_j)} \tag{3.5}$$

Normalized PMI (NPMI) is based on sliding windows to count the Word co-occurrence of the topic top word, which is called context vector. The mathematical expression of $NPMI$ is defined as:

$$v_{ij} = NPMI(w_i, w_j) = \frac{\log \frac{P(w_i, w_j) + \epsilon}{P(w_i)P(w_j)}}{-\log(P(w_i, w_j) + \epsilon)} \tag{3.6}$$

where w_i, w_j represents words,v_{ij} is j-th element in the $\vec{v_i}$, which is a context vector.

The C_V is a metric with employment of a sliding window, and one-set segmentation of the top words, based on NPMI and cosines similarity (Röder et al. (2015)). In this paper, C_V shows the best performance of the measurement.

Human Evaluation In Chang et al. (2009), human evaluation can be applied in two forms: word intrusion and topic intrusion. Word intruder is the word that is semantically different from other words in one topic or collection, for example 'rock' may be the word intruder of the word list ['apple', 'orange', 'banana','rock'], because the majority of words in this list belong to fruit, and rock does not belong to it. One word is injected into one list of words generated by topic modeling, which belong to the same topic. The task of human evaluation is to find the word intruder. The topic intrusion is similar to word intrusion: topic is presented as top n words. One topic intruder is injected into the collection of topics generated by the model, and the task is to determine which topic is the topic intruder. The test can be in the form of a questionnaire survey, allowing participants to find the intruder. The greater the probability of determining the word and topic intruder, the better the interpretability of the model. In this thesis, we evaluate the model manually based on the knowledge from data exploration.

In this paper, perplexity is used as a metric to tune the number of iterations. The combination of coherence score and manual evaluation is used to tune the number of model topics. In the actual tuning process, we found that only relying on the coherence value to tune the model can not assure to the output of a human-understandable model. In principle, we use manual evaluation as the most important indicator of model tuning, which is also mentioned in Röder et al. (2015).

3.3 Part B: Event Classifier

Part B of this paper is to build a classifier for tweets. The task of the classifier is to detect whether the input tweet belongs to a specific topic with a specific sentiment. For example, a classifier can be designed to detect whether a tweet belongs to a "cryptocurrency" topic with negative sentiment. To build a supervised learning classifier, we need a data set with feature X and target value y. In this article, the training feature X is the feature vector transformed from the tweet text, and the target value y consists of two parts: sentiment scores marked with positive, negative, and neutral; and topics, which can be marked with topic numbers. The whole process of part B is shown in Figure 3.6

It is worth mentioning the process and motivation of data labeling in this article. What is worth mentioning is the process and motivation of data labeling in this article. Sentiment analysis and sentiment score labeling are carried out in a lexicon-based method (Taboada et al. (2011)), and topic labeling is done through LDA Blei et al. (2003). So the labeling is completely automatic and there is no manual labeling.

Readers may have two doubts when reading this. First, why use automatic labeling instead of manual labeling. The reason is that we did this not because the automatic labeling approaches over-perform manual labeling, but because of time and manpower constraints. Manually data labeling for emotions and topic classification performs more accurately than the machine learning approaches in this thesis. Therefore, we prefer to use manually labeled data for classifier training in future work for practical usage. The second doubt is since lexicon-based sentiment analysis and LDA topic modeling can meet the requirement of the classification task, why do we need to use machine learning again to build the classifier. The first reason is that as mentioned above, the labeling quality of these two methods is not high, so we hope to train the classifier with manually labeled data in the future to obtain better classification performance. The second reason is that once the classifier is trained, it can be faster applied without hyperparameter tuning (in LDA hyperparameter tuning is complex), which is very useful in real-time processing for practical application scenarios.

3.3.1 Text Vectorization Methods

The text vectorization methods are applied for transforming text data into vectors that can be processed by computer. The generated vectors will be used as features for training the supervised classification algorithms. The vector can also be used for calculating the similarity between the two documents using the cosine distance. In this project, we use four text vector representations:

- Bag-of-Words (BOW)
- TF-IDF

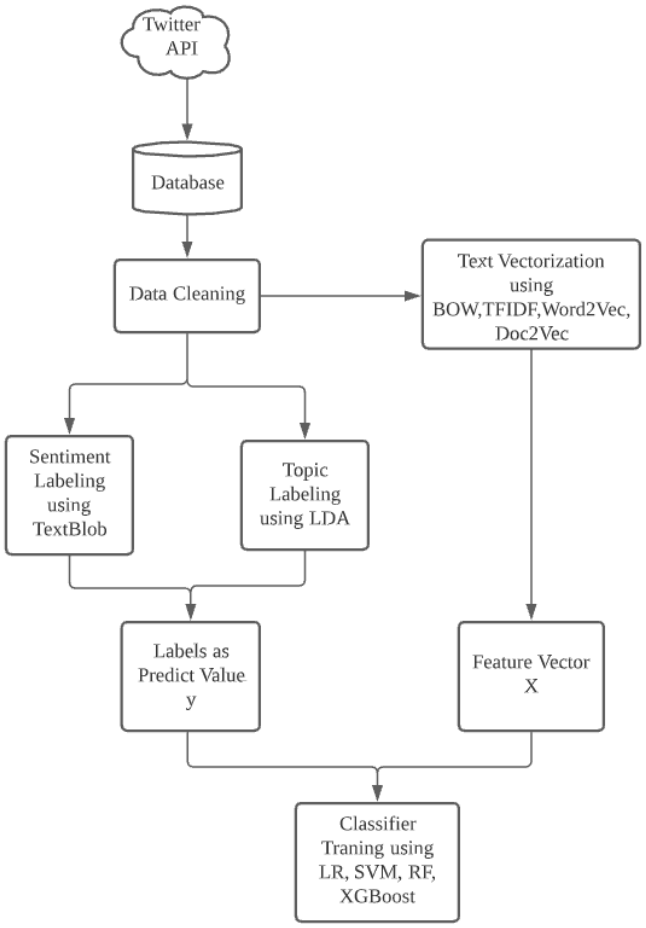


Figure 3.6: Framework B: Event Classifier

- Word2Vec
- Doc2Vec

Bag-of-Words The BOW model (Brownlee (2020)) treats documents as a pure collection of words. The word order in the document is ignored. In this model, a dictionary is generated based on all the words in a given corpus, each word has a unique index. The documents in the corpus are therefore represented as vectors, where the i-th element represents the number of times the i-th word in the dictionary appears in the document. Assuming that there are M documents and N unique words in the corpus, then the corpus can be expressed as an $M \times N$ matrix.

The advantages of this method are: the computation is not heavy, and easy to understand. The disadvantages of this method are: 1. The word order and context information in the text are lost 2. It causes spatial redundancy, that is, words that do not appear in the document are also required space for storage. This problem can be solved by using sparse vectors, i.e. only non-zero elements are stored in the vector. 3. All words in the document

are treated the same, without considering the weight configuration problem. For example: In a document, the word 'cryptocurrency' can describe the topic of the document clearly, and the word 'bank' cannot. In the BOW method, the importance weights of the words are the same, so the information extraction ability is limited.

TF-IDF TF-IDF is the abbreviation of the term frequency-inverse document frequency (Sammut and Webb (2010)). This method takes into account the uniqueness and importance of different words to different documents, where term frequency calculates the frequency of words appearing in the current document, and inverse document frequency calculates the frequency of words appearing in the entire corpus. The higher the frequency of a word in the current document and the lower the frequency of occurrence in the entire corpus, the higher the importance of the current word to the current document. There are many variants of TF-IDF calculation, a common formula is:

$$tfidf(t, d, D) = tf(t, d) \times idf(t, D) \tag{3.7}$$

Where:

$$tf(t, d) = \log(1 + freq(t, d)) \tag{3.8}$$

$$idf(t, D) = \log(\frac{N}{count(d \in D : t \in d)}) \tag{3.9}$$

Where the parameter t is term or word,d is document, N is the whole corpus. Compared with BOW, TF-IDF use the word frequency to measure the importance of a word in a document. However, this method cannot reflect the context information.

Word2Vec Word2Vec is a neural network-based method for converting text data into low-dimensional text vectors. It was proposed by the Google team Mikolov et al. (2013a). This method maps words into a fixed low-dimensional vector, and the word vectors contain semantic relationships and context information. Word2Vec trains word vectors according to the relationship between contexts. Two examples of Word2Vec generated by the data set in this thesis are shown in Figure A.1 and Figure A.2. There are two training modes, Skip Gram and CBOW(Continuous bag of words), in which Skip Gram predicts the context based on the target word, while CBOW predicts the target word according to the context. The structure of these two models is shown in Figure 3.7.

The input and output of the neural network are both $[1 \times N]$ dimensional vectors represented by One-Hot Encoding. The trained weight parameters in the model are output as word vectors. Because the skip-gram model only inputs one word at a time, it performs better against the over-fitting problems for frequently occurring words than the CBOW model. The same words will have diversified meanings in different contexts, for example, 'Amazon' in context 'forest' and 'shopping' are explained differently.

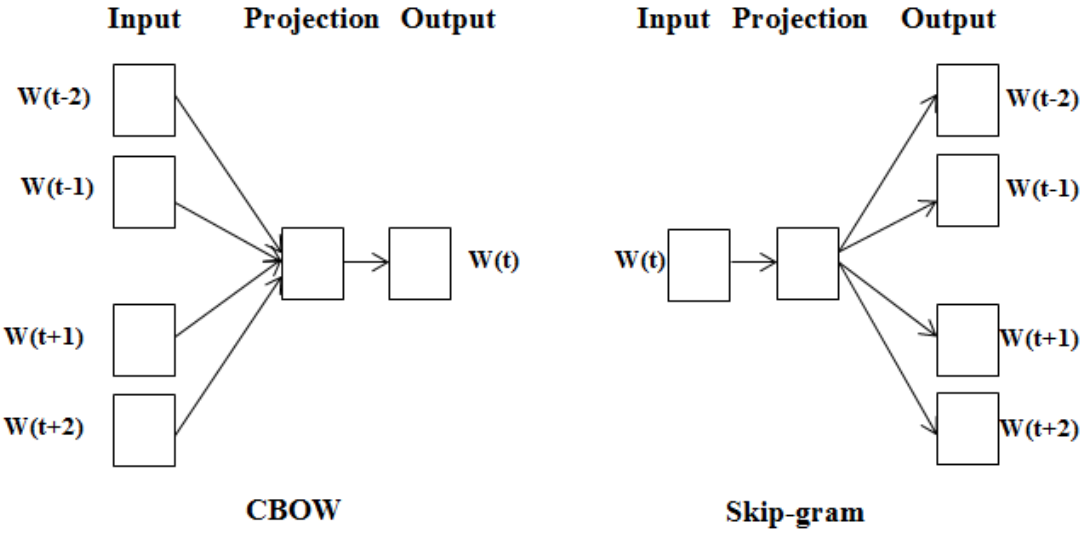


Figure 3.7: CBOW and Skip-gram Model (Mikolov et al. (2013b))

Doc2Vec Doc2vec is an unsupervised learning algorithm, which is used to represent documents with vectors. This model is proposed by Tomas Mikolov based on the word2vec model (Mikolov et al. (2013c)). The difference between Doc2vec and word2vec is that it adds a new sentence vector Paragraph vector, which is regarded as another word vector in the input layer of the neural network. After training, the model will output all the word vectors in the training sample and the sentence vector corresponding to each sentence

3.3.2 Dataset Labeling

Labeling is the process to assign labels to each tweet. There are two labels for each tweet: sentiment label and topic label.

Sentiment Analysis Labeling In order to convert our tweets data set into a labeled data set, we use sentiment analysis to perform the task. The goal is to label each tweet with a tag, which can be either positive, negative, or neutral. In this project, we use a Lexicon-based approach (Taboada et al. (2011)) for text sentiment labeling, because the document is the only input that is required. This approach is based on a sentiment dictionary, in which each word is assigned a sentiment score in the range between -1 to 1. The positive value represents positive sentiment, and the negative value shows negative sentiment, and 0 shows neutral. When performing sentiment analysis on a document, one of the ways is to calculate the average of the sentiment scores of all words in the document. The result is the sentiment score for the whole document. In this project, we apply the *TextBlob*[7] software to score the sentiment of each tweet.

[7]https://textblob.readthedocs.io/en/dev/

Topic Number Labeling The LDA model is applied to label the tweets with the topic class. Every tweet is assigned with a topic distribution by the LDA model, and the topic with the maximum probability is the dominant topic for a tweet. Each tweet is then labeled with its dominant topic.

3.3.3 Machine Learning Methods

We use four machine learning algorithms to build the classifier, which are:

Logistic Regression (LR) Logistic regression (McCullagh and Nelder (1989)) is a regression model used for classification problems. The most common is the binary classification logistic regression algorithm. Logistic regression is a variant of the linear regression algorithm using the sigmoid function, which can compress the output from the set of real numbers to between 0-1. The formula of LR is:

$$F(x) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x)}} \tag{3.10}$$

Where β_0 and β_1 are coefficients similar to linear regression.

Support Vector Machine (SVM) SVM is a robust supervised learning algorithm that can be used to deal with classification or regression problems (Hsu et al. (2003)). The working mechanism of SVM is to generate a hyperplane and divide the data points into their respective correct classes. The formula of SVM is:

$$\frac{1}{2} w^T w + C \sum_{i=1}^{l} \xi_i \tag{3.11}$$

subject to:

$$y_i(w^T \varnothing(x_i) + b) \geq 1 - \xi_i, \xi_i \geq 0 \tag{3.12}$$

Where x_i represents data points, y_i is the tag or class of the data instances, and l stands for the number of data points. $\varnothing$ is the kernel function for mapping data points to higher dimension.

Random Forest Random forest (Breiman (2001))is composed of multiple decision trees. It is a machine learning algorithm that can be used for classification and regression problems. Random forest is an ensemble learning method, based on the bagging method. It combines several poor performance classifiers into a stronger performance classifier. The workflow of random forest is as follows:

1. The original data set is sampled with replacement to generate a training set for each decision tree.

2. When the decision tree node needs to be split, randomly choose m variables from M variables (input variables), with $m << M$. Then splitting is based on these m variables.

3. No pruning during the growing process.

4. Repeating the above processes to generate random forest.

5. Generate output by aggregating results from all decision trees.

XGBoost XGBoost is the abbreviation of Extreme Gradient Boosting, which is an implementation based on the gradient boosting framework (Chen and Guestrin (2016)).XGBoost's supports both decision trees and linear models as its base model. In the tree base model, XGBoost supports two methods of splitting nodes: greedy algorithm and approximation algorithms. XGBoost has the following advantages:

1. Supports customized loss functions

2. Regulation term to the objective function is available to prevent over-fitting

3. Supports parallel operations

3.3.4 Evaluation

In the binary classification problem, the confusion matrix is used to evaluate the performance of the classifier or predictor. The confusion matrix(Ting (2017)) consists of four elements, namely:

1. True Positives (TP): Both the actual value and predict value are positive, the prediction is true.

2. True Negatives (TN): Because both the truth and the projected value are positive, the prediction is true.

3. False Positives (FP): Because the actual value is negative and the anticipated value is positive, the prediction is incorrect.

4. False Negatives (FN): Because the actual value is positive and the anticipated value is negative, the prediction is incorrect.

Based on the above four metrics, we further have the following three metrics: Precision, Recall and F1-score (Goutte and Gaussier (2005)). The formulas of these three metrics are as follows:

$$Precision = \frac{TP}{TP + FP} \tag{3.13}$$

$$Recall = \frac{TP}{TP + FN} \tag{3.14}$$

$$F1 - score = \frac{2 \times Recall \times Precision}{Recall + Precision} \tag{3.15}$$

The ability of a classification model to recognize positive samples is referred to as recall. The stronger the model's capacity to recognize positive samples, the higher the recall. Precision demonstrates the model's ability to distinguish between positive and negative samples. The stronger the model's capacity to distinguish negative samples, the higher the precision. The F1-score (Goutte and Gaussier (2005)) is a hybrid of the two. The higher this score, the more robust the classification model. We use F1-score to evaluate the classifier model in this thesis. The classifier is designed for classifying tweets with the specific sentiment of the specific topic, so the positive instances are less than the negative instances in the tweet data set. Therefore this score is chosen because it is suitable for unbalanced data sets.

Chapter 4

Result and Discussion

In this chapter, the whole picture of the collected data set, results of modeling, and discussion are included. In Section 4.1, we first explore the data set to get an intuitive understanding of the collected tweets data set. Data size, user distribution, hashtags, and sentiments in tweets are analyzed. In Section 4.2, we show the result of project A, which includes two parts: time-series anomaly detection and topic modeling results. Section 4.3 covers project B, in which the performance of classifiers is shown and discussed.

4.1 Data Discussion

The collected Twitter data set are the fundamental research object in this thesis. Therefore, data exploration and visualization can help us get insight into the data set and make a reasonable decision on model selection.

4.1.1 General Data Description

The data set of tweets were collected from 2021-07-09 to 2021-08-28. Totally, the data set consists of 17893 tweets, in which 7749 tweets are sent from the 11 official bank accounts(two accounts for VirginMoney and two accounts for Clydesdale), and 10090 tweets are sent from 5987 bank clients or third parties. In the following pages, we call the latter data source the "client or customer" for conciseness. If a tweets is sent from a bank, or a bank is mentioned by the symbol "@" in the tweets sent from a client, we call this tweet is relevant to this bank. The distribution of tweets by bank relevance is shown in Figure 4.1. As shown in the figure, when it comes to the total number of tweets, the top three banks are Barclays Bank, Santander Bank and HSBC, which accounted for 19.4%, 17.8% and 17.0% of the total tweets. In contrast, the three banks with the least number of tweets are Clydesdale Bank, the Co-operative Bank, and the TSB Bank, which account for 0.3%, 5.3%, and 6.1% of the total tweets respectively. It can be assumed that there is a positive correlation between the number of tweets and the size of the bank. Focusing on

the tweets sent from clients, the distribution of tweets relating to banks is shown as the yellow part of the bar in Figure 4.1. HSBC, Lloyds, and Santander occupy the top three positions with 2166, 1622, and 1572 tweets respectively. In contrast, Clydesdale Bank, TSB Bank, and Co-operative Bank occupy the last three positions with 42, 601, and 632 tweets.

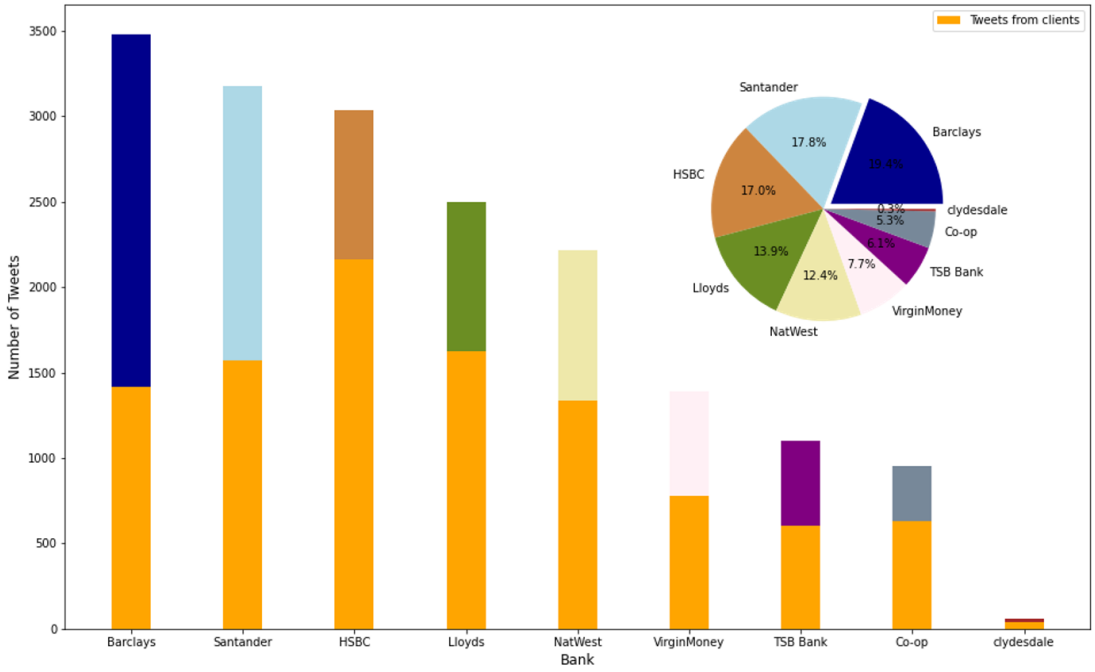


Figure 4.1: The Number of tweets grouped by banks. Bars represent the number of tweets relating to the respective banks. The below orange part of bars are the number of tweets that sent from clients, and the above part shows the tweets sent from official bank accounts themselves. The pie chart shows the proportional distribution of all tweets (clients and banks) for each bank.

We are also interested in the distribution of the number of tweets sent by customers. In other words, we are interested in how many tweets were sent the most clients on average. In Figure 4.2, the distribution of the number of tweets sent by clients are shown in the form of *violin plot*[1]. As shown in the figure, most of clients send 1 to 4 tweets to banks, which indicates that most customers do not communicate in detail with banks via Twitter, but only report some events or complaints.

Since we collect tweets from bank accounts for UK customer service, so the majority of locations should be in the UK. In Figure 4.3, we can see the location where tweets were sent from. The locations shown in the figure are all meets our expectation. Through the data analysis, we found that most tweets do not have location information. So in practical usage, this type of data may hard to be collected.

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

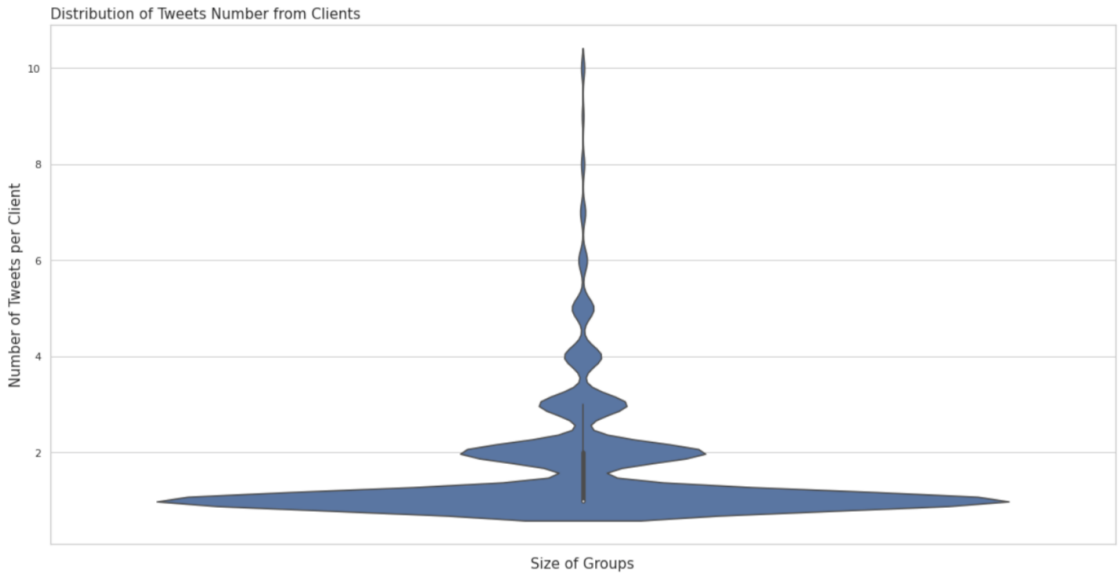


Figure 4.2: Distribution of a number of tweets sent by the client each person. This is presented in the violin plot. The ordinate axis represents the number of tweets, and the area in the blue area represents the size of the group of customers who send the corresponding number of tweets.

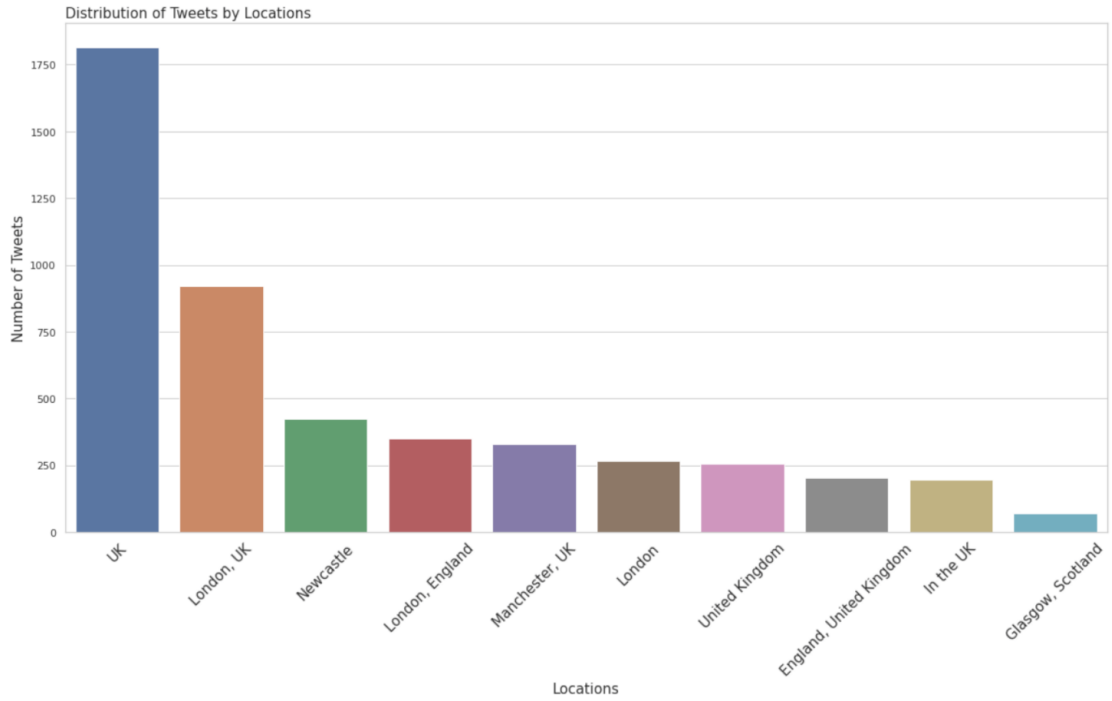


Figure 4.3: Locations where tweets are sent from.

4.1.2 Correlation of Hashtags

A hashtag is a metadata tag that begins with the # sign, which is popular in social media platform such as Twitter. By using Hashtags, tweets can be cross-referenced and clustered on the platform. For example, #BlackLivesMatter or #COVID19 are very popular, and hundreds of millions of tweets are linked with these hashtags. Hashtag is very useful to allow us make a preliminary data exploration on the tweets in this thesis. The results obtained by hashtag analysis can also be used for mutual verification and comparison with the topics obtained by the topic model in section 4.2.

The hashtag correlation matrix is shown in Figure 4.4. It is obtained based on calculating the hashtags co-occurrence frequency (Wang et al. (2016)) in the tweets data set. Hashtag pairs that appear frequently in the same tweet are assigned to high correlation scores. The relevant hashtags that noteworthy are: *#hsbc* has a strong correlation with *#fraud* and *#money* ; *#boycottbarclay* is strongly related to *#cryptocurrency* and *#binance*; *#natwest* is strongly related to *#defi* and *#MyMoneyMyChoise*.

Based on this information, we can have a basic understanding of the main complaints from customers. HSBC is strongly correlated to complaints of fraud transactions. Clients of Barclays mainly discuss the bank's policy of cryptocurrency and Binance, which make clients uncomfortable. The clients of Natwest complain about they are not allowed to use their own money to make investment choices.

We already have the basic knowledge of the content in this data set. However, hashtags alone cannot give us a profound and comprehensive understanding of latent topics. There are the following drawbacks of analyzing only use hashtag: On the one hand, tweets must not contain hashtag-related topics, and some topics in tweets are unique and no popular hashtags available. On the other hand, some users don't like to use hashtags. In our data set, only 928 tweets are labeled with hashtags. Therefore, this data size is not enough for us to extract all useful information. In the next section, we will show the result of topic modeling for more detailed exploration.

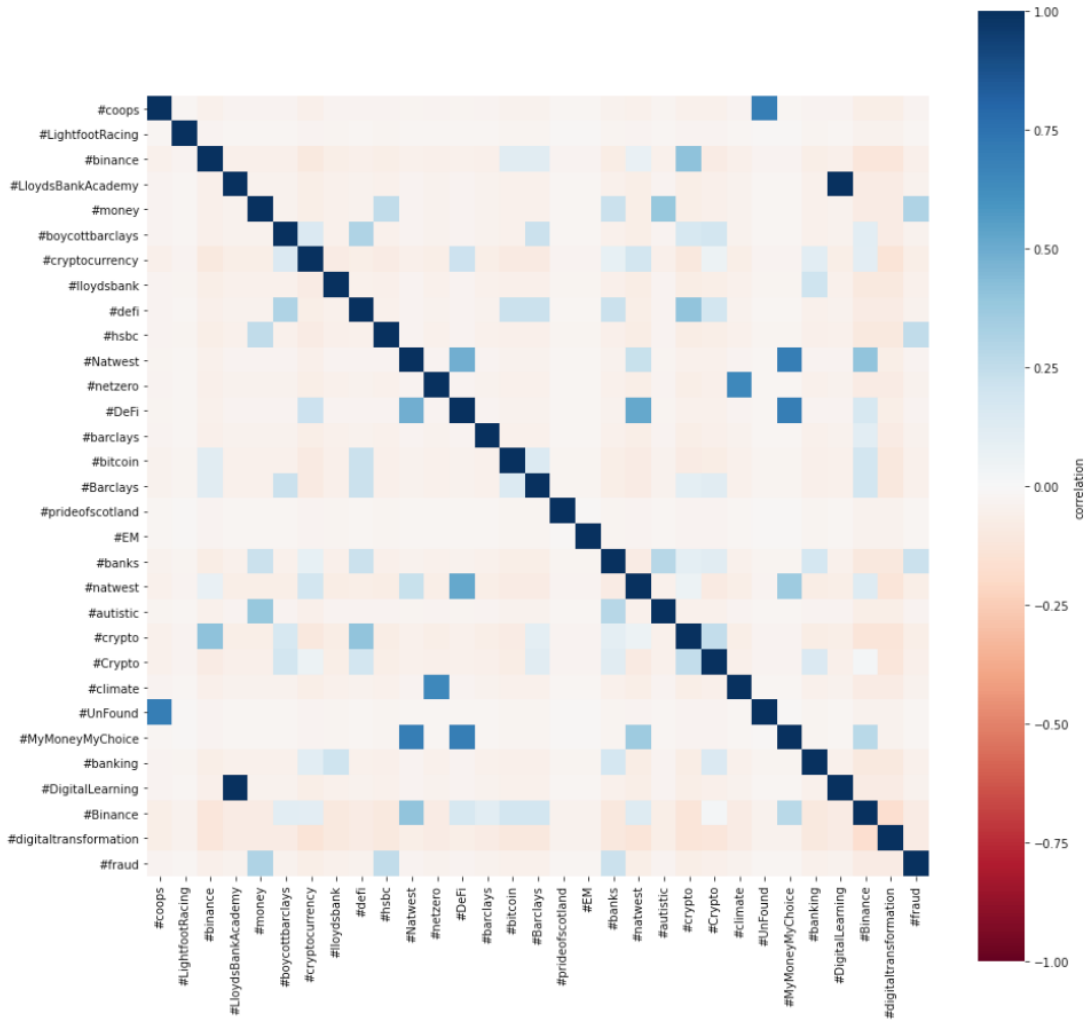


Figure 4.4: The figure shows the correlation between hashtags. Blue represents positive correlation and red represents negative correlation. The deepness of color represents the degree of correlation.

4.1.3 Sentiment Analysis

Having a basic knowledge of the sentiment distribution in the data set can help us get a deeper insight into our tweets. In Table 4.1, we show the sentiment distribution grouped by the data source, i.e. from banks and from clients. As can be seen from the table, 24% of the tweets sent by clients are negative, 76% are neutral or positive. In comparison, only 2% of tweets from banks are negative. This result makes sense because the tweets sent by the bank are generally sent from professional customer service. There will be many polite words in the reply tweets to the customers. In contrast, customers tend to complain directly if they are unsatisfied with banks, so the proportion of negative sentiment tweets from customers is greater than that of the bank.

Overall Sentiment Distribution			
From	Positive	Neutral	Negative
Client	41.4%	34.2%	24.4%
Bank	62.4%	35.4%	2.3%

Table 4.1: Distribution of Tweets Sentiment from Bank and Client.

In addition, the sentiment distribution of tweets related to banks is also a informative topic. Therefore, the tweets sent by customers are grouped by the related banks. The result is shown in Table 4.2. In the table, each row represents the sentiment distribution of tweets relating to a banks. It can be seen that the proportion of positive tweets received by Lloyds Bank is the highest with 44.5%. In comparison, only 36.5% of tweets related to co-op are positive. The bank with the highest proportion of negative tweets is also Co-ops with 29.9%. The bank with the lowest proportion of negative tweets is Clydesdale Bank with 21.7%. Overall, the difference of sentiment distribution between the banks is small, which means the degree of customer satisfaction reflected on twitter among banks are not much different.

Sentiment Distribution to Banks			
Name of Banks	Positive	Neutral	Negative
Barclay UK	39.6%	32.3%	28.0%
Co-op	36.5%	33.5%	29.9%
HSBC	41.8%	33.5%	24.7%
Lloyds	44.5%	33.6%	21.9%
NatWest	44.4%	29.7%	25.8%
Santander	40.1%	38.7%	21.2%
TSB Bank	40.1%	35.3%	24.6%
Virgin Money	39.3%	39.0%	21.6%
Clydesdale Bank	43.5%	34.8%	21.7%

Table 4.2: Sentiment Distribution of Tweets relating to Banks.

4.2 Part A: Event Detection

In this section, we show the result of time series anomaly detection in subsection 4.2.1 and two topic models in subsection 4.2.2. Then we use the outcomes of these two subsections to represents a practical case study, which is shown in subsection 4.2.3.

4.2.1 Anomaly Detection Results

The First step for applying anomaly detection is to convert the collected raw data into time series. We group data by using the feature of created time of tweets and count the number of tweets on an hourly basis. From 2021-07-09 10:00:00 to 2021-08-28 06:00:00, we have 1197 hours of data, and they are converted to 1197 data points in time series data format. These data are represented as black points, and predict value by Prophet represented as a blue line, are shown in Figure 4.5. In the whole picture, those points are distributed discretely over time. At midday, the number of tweets is really high. The blue line shows that the model fits and predicts the trend of the number of tweets with daily ups and downs.

In Figure 4.6, the weekly seasonality of the time series data is shown. On the weekdays, the trends of changes are positive, and the highest value is reached on Tuesday with nearly 8. In comparison, the trend during weekends, i.e. from Saturday to Monday, are negative. This weekly seasonality shows that clients tend to send tweets to banks from Tuesday to Friday at most. We discuss only clients here is because tweets from bank are usually for replying to tweets from clients.

Figure 4.7 shows the daily trend of number of tweets. As can be seen in the figure, from 05:15 to 17:20, the trend are positive. In other time the number of tweets is decreasing. This shows that people prefer to send tweets during the day rather than at night. This is consistent with our subjective guess.

In Figure 4.8, we can see that the abnormal points are obtained according to the time series anomaly detection model. The outliers detected from July 27 to July 29 are very dense. We will explore the main topics of tweets sent in this time period in subsection 4.2.3 as a case study.

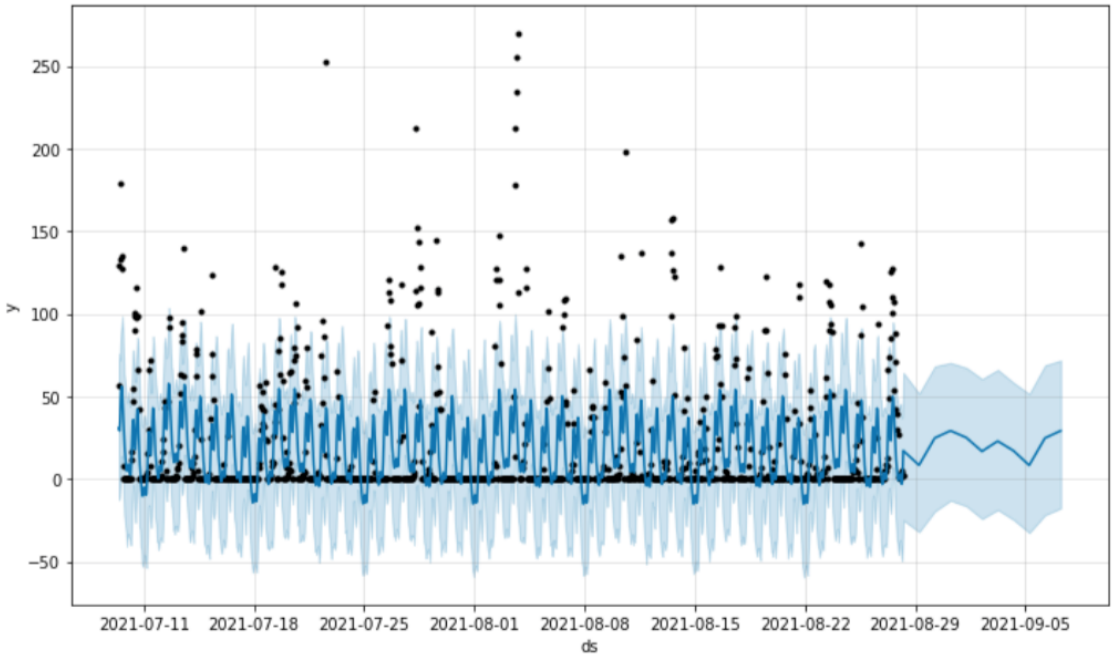


Figure 4.5: The number of tweets grouped on an hourly basis. Black points show the number of tweets collected in the corresponding time. The blue line shows the trend prediction by Prophet.

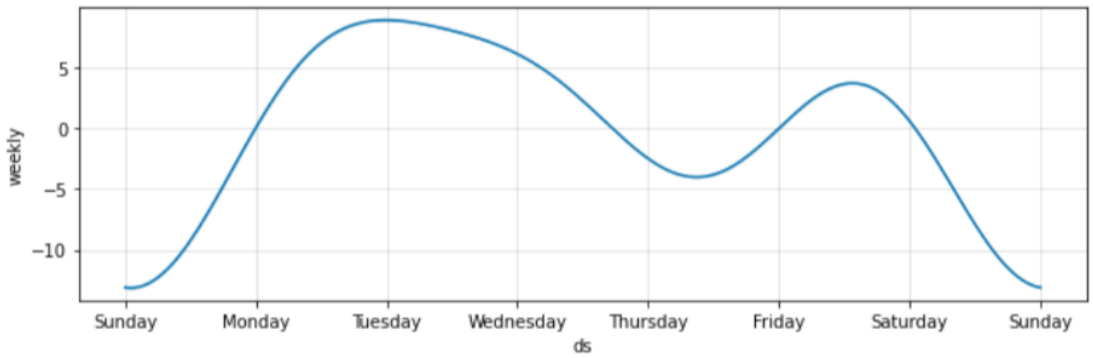


Figure 4.6: Seasonality weekly

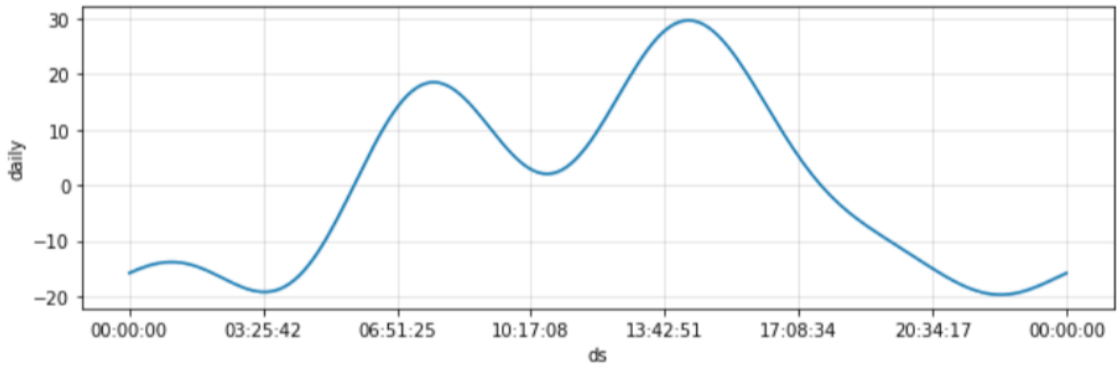


Figure 4.7: Seasonality daily

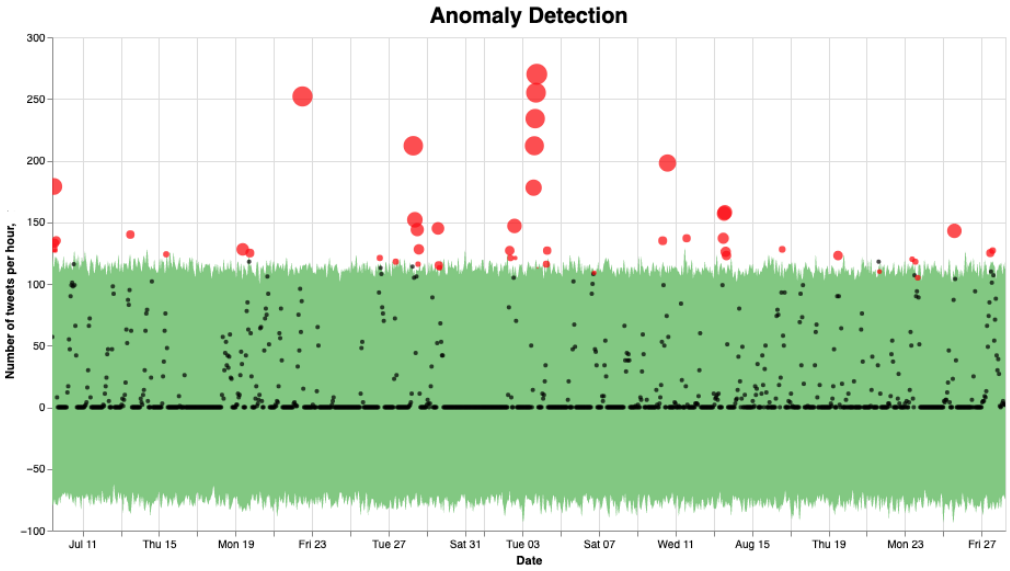


Figure 4.8: Time series anomaly detection result by using Prophet. The outlier points are marked as red color. The size of points indicates the degree of anomaly.

4.2.2 Hyperparameter Tuning of Topic Modeling

Before using the LDA and LDA-U models for topic modeling, we need to first tune the hyperparameters of the topic model. This step is to ensure that the model can produce the high-quality and human-understandable results of topics.

4.2.2.1 α and β

Two parameters α and β belong to the hyperparameter of the LDA and LDA-U model. They control the document-topic density and topic-word density in the process of topic generating. One of the effective ways for finding the optimal setting is applying grid search, which will exhaust all the combinations of α and β and find the optimal setting. However, due to the computational limitation, We choose to use the existing research results. There are many papers investigating the setting of LDA and LDA-U models for the task of tweets processing. We use the same hyperparameter setting proposed in Jónsson and Stolee (2015), see Table 4.3.

Model	Hyperparameters
LDA	$\alpha = 10/K$, $\beta = 0.01$
LDA-U	$\alpha = 10/K$, $\beta = 0.01$

Table 4.3: Hyperparameter Setting of α and β in LDA and LDA-U models. K is the number of topics which is set in the model.

4.2.2.2 Convergency of Iterations

Iterations refer to the number of times we perform a loop over each document by training the LDA and LDA-U model. Sufficient iterations make the model generate more meaningful topics. However, too many iterations may also cause unnecessary computation. Therefore, it is important to know the iteration number that allows the performance value of the model convergent by all other hyperparameters unchanged. We use the log-likelihood as an evaluation metric, and train the model with different iterations, which is in the range of 1 to 100 with step 2. In Figure 4.9, an experiment of iteration evaluation by an LDA model with a topic number of 15 is shown. As can be seen that, the log-likelihood increases sharply when the iteration number increases from 1 to 10. When the value of iteration reaches 20, the metric almost converges. From 20 to 100, there is almost no change in log-likelihood, which means more iteration can not improve the performance of the model. After sufficient experiments, we choose iteration number 20 as the reasonable iteration value for LDA and LDA-U models used in this thesis.

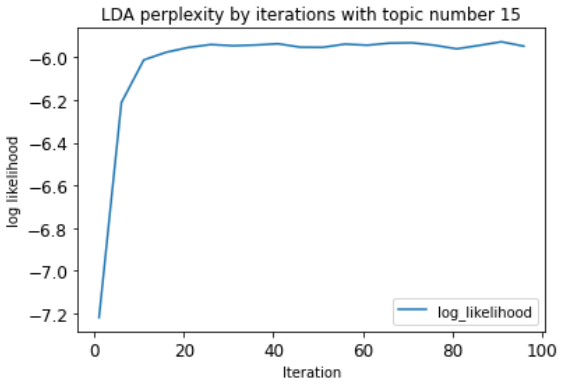


Figure 4.9: Log likelihood with different iteration in LDA model.

4.2.2.3 Topic Number

Topic number controls how many topics are generated from LDA and LDA-U models. While a model with an over-small topic value may cause missing important topics and information, setting an over-large topic number may cause redundances and hard-to-understand topics. In this project, we evaluate the topics based on numerical methods and human evaluation combined. For obtaining coherence values, We make the experiment by setting the topic number in the range of 1 to 50 and record the coherence values of each Model. We use the coherence values firstly as a reference to determine the interval of the candidate of topic numbers, and then train the models with those topic numbers for human evaluation. Figure 4.10 and Figure 4.11 show the average coherence values of topics with different topic numbers in LDA and LDA-U model.

In Figure 4.10, the coherence values of C_{UMass}, C_{NPMI} and C_{UCI} decrease as the topic number increase.C_{UMass}, C_{NPMI} and C_{UCI} are variants of PMI (Röder et al. (2015)), which is based on calculating the probability of word co-occurrence. Therefore this may explain why these three metrics show a similar trend. This result shows that as the topic number increases, the probability of co-occurrence of the top words in the same topics generated by the LDA model are lower. Therefore the topic number should be constrained in a relatively small number, best before the elbow point 20, shown in the C_{UMass}, C_{NPMI} and C_{UCI}. Conversely, C_V shows an increasing trend as the topic number increases. Before flatten out, there is an elbow space between 5 and 15. It indicates that the range of 5 to 15 would be a good space for selecting the topic number. Combining the observations of the above four indicators, We trained the LDA model with the topic numbers 5, 12, 15, 20, and printed the generated topics and top words of topics for human evaluation based on the understanding of the data set. The LDA model with topic number 5 covers limited topics, for example the topic of cryptocurrency is ignored. The LDA model with topic number 20 covers all topics that we found manuelly, but there are many overlaps between topics, which indicates over-fitting. Through comparison, We choose 12 is a appropriate

topic number for LDA model for further research and case study.

In Figure 4.11, the coherence values of C_{UMass}, C_{NPMI} and C_{UCI} show a similar trend as in Figure 4.10. Different from LDA, the C_V shows a trend of first increasing to the peak at topic number 5 and then dropping to a flat line. This indicator shows that we need to choose the number of topics between 5 to 10 for maximizing the coherence score. It is noteworthy that there is a point at 8 in C_V that the downward trend has been eased. By training and comparing the result of topics with those models, we choose 8 as a reasonable topic number for the LDA-U model.

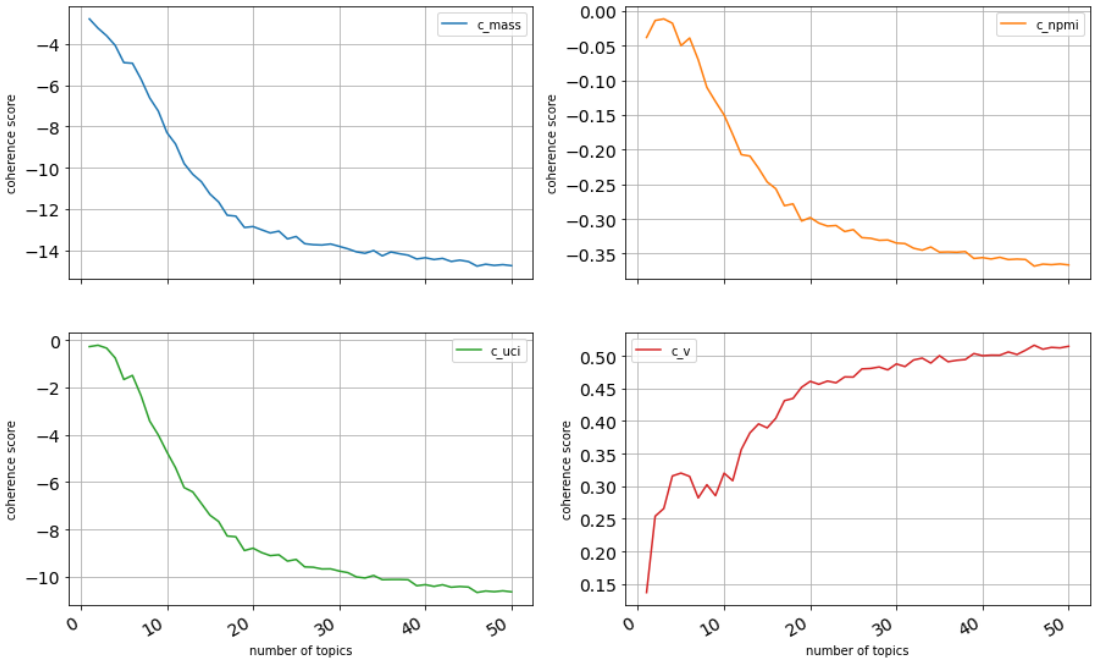


Figure 4.10: Coherence values of LDA for different topics. Notice the elbow points with a number of topics at 5, 12, 14, 20.

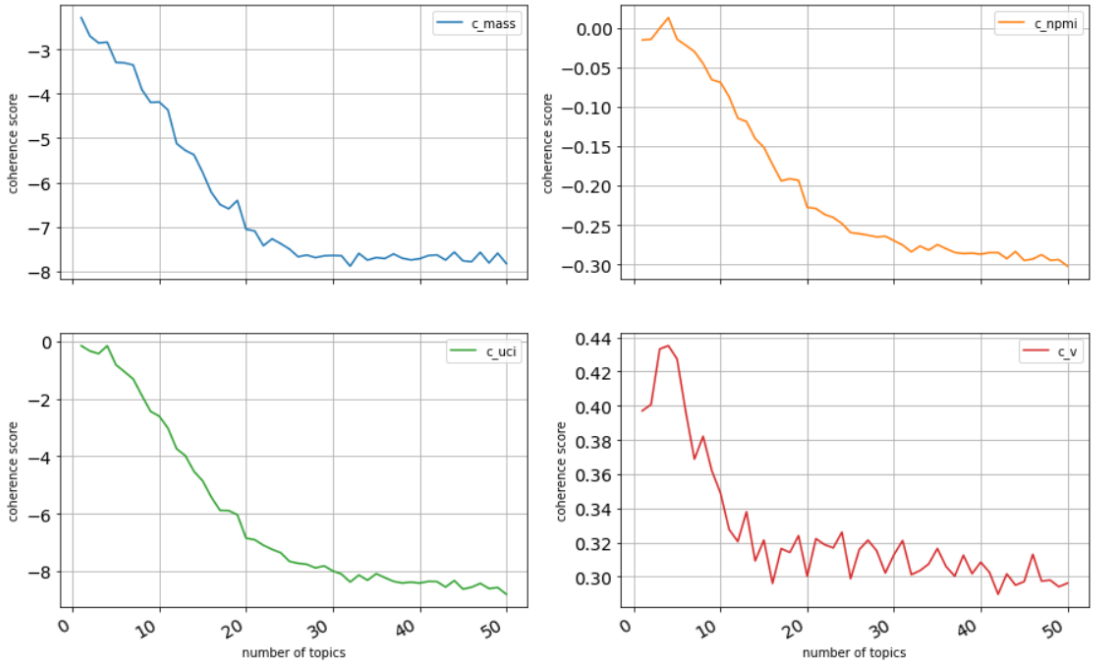


Figure 4.11: Coherence values of LDA-U for different topics. Notice the elbow points with a number of topics at 4, 8, 13, 15.

4.2.3 Topic Modeling Results

In this subsection, we represent the topic results generated by our LDA and LDA-U models with the selected topic number. For each model, we show the generated topics with their top 10 words(the words with maximal probability to be generated within the topic).

LDA Model Result In Figure 4.4 and Figure 4.5, 12 topics with 10 top words are shown. In Figure 4.6, the percentage of corpus assigned with the 12 topics is shown. It can be seen that the topic of debit cards, mixed topics,s, and topic of credit cards are the three most popular topics. In Table 4.7, the evaluation result for the 12 topics using four coherence scores is shown. Evaluating with the C_V, we can see that topic 9 has the highest score, which indicates that topic 9 is most coherent statistically regarding this metric. Figure 4.12 shows the visualization of the LDA model by using the *pyLDAvis*[2] software. It can be seen from the figure that the distribution of topics is not overly concentrated (blue circles do not overlap each other too much), which indicates that this model is not over-fitting.

LDA Topic List, Part A					
Topic 1	Topic 2	Topic 3	Topic 4	Topic 5	Topic 6
email	service	card	money	app	secure
payment	busy	work	abl	pay	today
debit	people	stop	fund	branch	sent
last	support	change	deposit	close	access
order	never	credit	line	open	code
though	complaint	month	crypto	cheque	log
lloyd	live	find	natwest	post	option
santand	care	lost	withdraw	website	digit
direct	thought	old	scam	update	key
cancel	away	paid	block	barclay	manag

Table 4.4: Topic list of LDA, Part A. The top 10 words of each topic are shown in the list. Some words are not understandable such as 'abl' in Topic 4, because in the process of word stemming, the words have been transformed to their root form, which is irreversible.

[2]https://pyldavis.readthedocs.io/en/latest/readme.html

LDA Topic List, Part B					
Topic 7	Topic 8	Topic 9	Topic 10	Topic 11	Topic 12
phone	account	call	time	well	number
amp	online	hour	week	good	receive
sort	current	wait	address	reply	probleme
ethic	move	hold	transaction	cash	hsbc
right	passport	minute	fraud	mortgage	message
matter	transfer	answer	speak	response	first
fix	student	person	ago	morning	set
listen	virgin	contact	application	text	check
value	save	team	request	absolute	long
poll	application	chat	accept	point	mobile

Table 4.5: Topic list of LDA, Part B. The top 10 words of each topic are shown in the list.

LDA Percentage of Corpus		
Topic Number	Topic Name	Percentage of Corpus
1	debit card and payment	9.8%
2	mixture issue	9.2%
3	credit card	8.5%
4	crypto currency natwest	8.4%
5	local branch close, online banking	8.4%
6	password	8.4%
7	mixture issue	8.4%
8	student account	8.3%
9	hotline waiting	8.2%
10	fraud transaction	7.6%
11	mortgage	7.4%
12	mixture issue	7.2%

Table 4.6: Percentage of the corpus by different topics in LDA. The topic name is base on the observation of respective top words. The percentage of the corpus is based on classifying tweets based on the dominant topics.

LDA Coherence Score per Topic				
Topic Number	UMass	UCI	NPMI	C_V
1	-6.644	-3.422	-0.104	0.237
2	-7.658	-4.502	-0.148	0.282
3	-9.064	-5.368	-0.178	0.322
4	-7.730	-4.222	-0.116	0.318
5	-5.886	-3.101	-0.091	0.288
6	-6.955	-3.987	-0.120	0.297
7	-10.41	-6.517	-0.187	0.229
8	-8.153	-4.219	-0.118	0.278
9	-5.083	-1.358	-0.009	0.414
10	-6.781	-3.735	-0.121	0.239
11	-10.086	-6.192	-0.206	0.333
12	-5.677	-2.485	-0.0714	0.260

Table 4.7: LDA Coherence Score per topic. Topics in LDA are labeled with respective coherence scores evaluated by four metrics.

Figure 4.12: Visualization of the LDA model by using the *pyLDAvis* software. Topics are visualized by blue circles. The correlation between topics is represented by the distance of the circles.

LDA-U Model Result In Figure 4.8 and Figure 4.9, 8 topics with 10 top words are shown. In Figure 4.10, the percentage of corpus assigned with the 8 topics are shown. It can be seen that topic of hotline waiting, the topic of credit card and the mixed topic of waiting time and online banking are the three most popular topics. In Table 4.11, the evaluation result for the 8 topics using four coherence scores is shown. Evaluating with the C_V, we can see that topic 1 has the highest score, which indicates that topic 1 is most coherent statistically regarding this metric. This result is consistent with the result from LDA model. Figure 4.13 shows the visualization of LDA model by using the *pyLDAvis* software.

LDA-U Topic List, Part A			
Topic 1	Topic 2	Topic 3	Topic 4
call	card	service	account
phone	credit	time	branch
hour	people	wait	close
hold	debit	online	pay
number	receive	address	open
minute	order	letter	cheque
answer	cancel	staff	santander
chat	month	absolute	current
min	direct	sort	local
line	text	change	charge

Table 4.8: Topic list of LDA-U, Part A. The top 10 words of each topic are shown in the list.

LDA-U Topic List, Part B			
Topic 5	Topic 6	Topic 7	Topic 8
app	time	amp	money
work	account	week	payment
hsbc	barclay	put	natwest
log	mobile	ago	lloyd
support	well	great	block
problem	done	account	secure
message	person	start	binance
student	tell	manage	stop
update	name	first	scam
error	app	home	crypto

Table 4.9: Topic list of LDA-U, Part B. The top 10 words of each topic are shown in the list.

LDA-U Percentage of Corpus		
Topic Number	Topic Name	Percentage of Corpus
1	hotline waiting	14.4%
2	credit and debit card	13.1%
3	hotline waiting	12.4%
4	local branch close	12.3%
5	online banking	12.2%
6	hotline waiting	12.0%
7	mixture issue	11.8%
8	crypto currency binance	11.7%

Table 4.10: Percentage of the corpus by different topics in LDA-U. The topic name is base on the observation of respective top words. The percentage of the corpus is based on classifying tweets based on the dominant topics.

LDA-U Coherence Score per Topic				
Topic Number	UMass	UCI	NPMI	C_V
1	-2.457	0.199	0.061	0.631
2	-3.665	-1.097	-0.010	0.299
3	-3.099	-0.930	-0.030	0.386
4	-3.234	-0.557	0.011	0.470
5	-6.116	-3.572	-0.097	0.271
6	-2.918	-0.953	-0.033	0.400
7	-3.654	-2.223	-0.075	0.301
8	-3.142	-0.586	0.025	0.437

Table 4.11: LDA-U Coherence Score per topic. Topics in LDA-U are labeled with respective coherence scores evaluated by four metrics.

Figure 4.13: Visualization of the LDA-U model by using the *pyLDAvis* software. Topics are visualized by blue circles. The correlation between topics is represented by the distance of the circles.

4.2.4 Case Study

In this subsection, we show a case study by integrating the result from subsections 4.2.2 and 4.2.3. In Figure 4.8, the outlier points in time interval from August 3 to August 8 occurs significantly. Therefore, we choose this time span to show how LDA and LDA-U can be applied when abnormal events are detected.

Data exploration is the first step before applying topic modeling methods. Through analysis, there are a total of 896 tweets in this time interval, in which there are 418 tweets with positive sentiment, 233 tweets with negative sentiment, and 245 tweets with the neutral sentiment. The number of tweets sent from clients is 549, in which 214 tweets are positive, 139 tweets are negative and 196 tweets are neutral. After data aggregation for LDA-U, there are 355 tweets from clients.

Then We apply then applied LDA and LDA-U to extract the dominant topics of the tweets in this data set. The two methods are applied to tweets sent from clients. The reason why we only analyze tweets from clients is that most of the tweets from banks do not contain many topic-related words. The result of topic modeling by LDA and LDA-U are shown in Table 4.12 and Table 4.13.

LDA As shown in Table 4.12, the top topic of tweets in this time period is the hotline waiting issue. After reading the tweets, we found that those customers who can't get through the bank's customer service call sent these complaint tweets. Topic 0 and topic 3 are in the third and fourth places with 14.4% and 12.2% of the corpus. This may explain the reason why the number of tweets peaked in this period. In addition to these top topics, topics 4 of cryptocurrency, topics 11 of mortgage, and topic 10 of fraud transaction are also noteworthy.

Topic Number	Topic Content	Percentage of Tweets
9	hotline waiting	15.3%
2	mixture issue	15.3%
1	debit card and payment	14.4%
3	credit card	12.2%
4	crypto currency natwest	7.3%
8	student account	6.2%
10	fraud transaction	5.8%
12	mixture issue	5.8%
11	mortgage	4.9%
7	mixture issue	4.4%
5	local branch close, online banking	4.4%
6	password	4.0%

Table 4.12: Case Study: topic distribution by LDA

LDA-U As shown in Table 4.13, the top topic in this time period is online banking with 28% of whole tweets. In second place are the topic relating to hotline waiting with 14%. This top topic distribution in LDA-U is similar to the distribution in LDA. In addition, topics 8 of cryptocurrency and topic 4 of local branch closing are also noteworthy.

Topic Number	Topic Content	Percentage of Tweets
5	online banking	28.2%
1	hotline waiting	17.7%
4	local branch close	13.2%
2	credit card	12.0%
8	crypto currency binance	7.8%
6	online banking	7.5%
7	mixture issue	7.2%
3	hotline waiting	6.3%

Table 4.13: Case Study: topic distribution by LDA-U

In this case study, we think the topics relating to "online bank", "credit card", "debit card", "payment" can best explain why the number of tweets during this period exceeds the normal value. From these topics, we can infer the following events: During this period of time, due to the large-scale payment system failures, customers called the hotline for help. However, the hotline could not get through after waiting for a long time, because many other customers has the same situation, so too many calls resulted in insufficient manpower to answer the telephone hotline at the same time. Therefore the customer chose to use tweets to report the situation. Also through the analysis of top topics from LDA and LDA-U, the reasons for the number of tweets peaked in this time period are consistent. Online banking failure events have strong timeliness. Once it happens, it will trigger a large scale of phone calls and tweets to report this information, which leads to a dramatic increase in the number of tweets during this time.

We can also find the phenomenon that, in abnormal time intervals, the percentage of tweets of event-related topics increase significantly compared to the whole corpus. In Table 4.6, the topic 9 of LDA only take 8.2% of the whole corpus, and topic 1 of LDA takes 9.8%, while they take 15.3% and 14.4% in Table 4.12 respectively. It is the same for topic 5 and topic 1 in LDA-U, shown in Table 4.13 and Table 4.10.

Through this case study, we believe that the model with a combination of time series analysis and topic modeling is proven to be able to detect the occurrence of financial emergent events to a certain extent.

4.3 Part B: Event Classifier Evaluations

The event classifier is trained by the labeled data. The feature vector is obtained by using four text vectorization methods: BOW, TF-IDF, Word2Vec, and Doc2vec. The vector is the X in supervised learning. The label y consists of an emotional label and a topic label. The classifier we build is designed to classify the tweets on the topic of "hotline waiting" with negative sentiment. Therefore we label the tweets with dominant topic 9 and the sentiment score with "Negative" as 1 and the rest of the tweets as 0. In this way, we transform the data set to a binary labeled data set which is suitable for training a binary classifier.

F1 Scores of Classifiers				
	BOW	TF-IDF	Word2Vec	Doc2Vec
Logistic Regression	0.722	0.708	0.587	0.257
SVM	0.737	0.732	0.575	0.552
Random Forest	0.730	0.713	0.320	0.088
XGBoost	0.739	0.705	0.398	0.399

Table 4.14: F1 scores results of the four machine learning algorithms

As shown in Table 4.14, there are four machine learning algorithms and four text vectorization methods. Therefore, we got 16 results of F1-score by the combination. Since the data set is uneven between positive and negative data points, we use F1-score to evaluate the performance of classifiers with a balance between recall and precision. In the table, the BOW achieves the best performance within the four vectorization methods in all four machine learning combinations. The TF-IDF achieves the second-best performance. Word2Vec and Doc2Vec are the third and fourth respectively. Theoretically, Word2Vec and Doc2Vec should over-perform the other two approaches because they are more complicated and can capture more information from text, but the result is reversed. We explain this result as the problem of the way we label the data. Topic labeling and Sentiment labeling are based on LDA and Lexicon-based approaches, which are based on the BOW model. Therefore, it is natural that the classifier with BOW achieves the best performance. However, in the practice, these two methods are not accurate as humans through the investigation of our data set. Therefore as described in the section of Future work, we advise using the manually labeled data to train the classier, the human is more sensitive to the topic and sentiment in comparison to the LDA and Lexicon-based approaches through our observation.

The structure of this part B is proposed for future work. Once we have the resource of manually labeled data, we can use the framework of part B to train a better classifier.

Chapter 5

Conclusion and Future Work

Conclusion The main goal of this thesis is to monitor and detect abnormal events based on bank-related tweets. To achieve this goal, two parts of projects (Part A and Part B) are completed. In part A, time series anomaly detection based on decomposable time series model is applied on the tweets data set to detect unusual time intervals. Then two topic modeling methods based on LDA are applied on the relevant tweets to extract the dominant topics. In part B, classifiers are built to classify tweets that are relevant to spesific topics with spesific sentiments. Four text vectorization methods: BOW, TF-IDF, Word2Vec and Doc2Vec are performed on tweets for getting the feature vectors; the LDA model trained in part A and *TextBlob* software are used to label the tweets for getting the target value in supervised learning. Four machine learning methods: Logistic Regression, SVM, Random Forest and XGBoost are applied to build classifiers. The main results obtained in this theis are:

- The combination of time series analysis and topic modeling shows a promising possibility to detect abnormal events, such as sudden transaction difficulties or large-scale offline banking mobile applications. The anomaly can be reflected in the number of tweets in a given time interval.
- Although LDA has been criticized for not being suitable for processing short texts, in this project LDA shows good performance to a certain extent.
- Classifier based on machine learning methods shows its application possibility on tweets classification. The quality of training data is the key factor to determine the performance of the model. For topic classification and sentiment analysis, the manually labeled data may be of higher quality than the automatic labeling which are implemented in this article.

Future Work Because of the limitation of time, computational resources, human resources, and geographic barriers, this project can be improved in many areas in the future.

Firstly, we will use other advanced topic modeling methods in future research. There are models that are specifically designed for processing short text, such as the Biterm model and W2V-GMM models in Jónsson and Stolee (2015). Bidirectional Encoder Representations from Transformers (Devlin et al. (2018)) is a state-of-art technique that we plan to apply to tweets processing. In addition, other time series analysis models can be evaluated for detecting the outliers in the number of tweets. For example Long Short-Term Memory Model (Hochreiter and Schmidhuber (1997)) or ARIMA model (Wang (2011)).

Second, the quality and integrity of tweets collection can be improved. The connection to Twitter Streaming API is not very stable in this thesis. The automatic disconnection of the Twitter Streaming API results in a lack of data in the time series. For real application scenarios in the future, if the data can be collected with higher quality and integrity, the model will be more accurate for detecting the outliers in time series.

Last but not the least, we will use manually labeled data to train the classifier if the condition is allowed in the future. In this thesis, automated labeling is the second-best choice for obtaining the labeled data with limited human power, because the lexicon-based sentiment analysis is not accurate enough through the examination with the data set in this thesis. If the classifier can be feed with manually labeled data for training, the performance of the classifier will be improved significantly.

Bibliography

Blei, David M., Andrew Y. Ng, Michael I. Jordan. 2003. Latent dirichlet allocation. *J. Mach. Learn. Res.* **3**(null) 993–1022.

Breiman, Leo. 2001. Random forests. *Machine Learning* **45**(1) 5–32. doi:10.1023/A:1010933404324. URL `http://dx.doi.org/10.1023/A%3A1010933404324`.

Brown, Peter F., Stephen A. Della Pietra, Vincent J. Della Pietra, Jennifer C. Lai, Robert L. Mercer. 1992. An estimate of an upper bound for the entropy of English. *Computational Linguistics* **18**(1) 31–40. URL `https://aclanthology.org/J92-1002`.

Brownlee, Jason. 2020. A gentle introduction to the bag-of-words model URL `https://machinelearningmastery.com/gentle-introduction-bag-words-model/`.

Chang, Jonathan, Jordan Boyd-Graber, Chong Wang, Sean Gerrish, David M. Blei. 2009. Reading tea leaves: How humans interpret topic models. *Neural Information Processing Systems*. URL `http://umiacs.umd.edu/~jbg//docs/nips2009-rtl.pdf`.

Chen, Tianqi, Carlos Guestrin. 2016. Xgboost: A scalable tree boosting system. *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*. 785–794.

Darling, William M. 2011. A theoretical and practical implementation tutorial on topic modeling and gibbs sampling. *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies*. 642–647.

Devlin, Jacob, Ming-Wei Chang, Kenton Lee, Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* .

Gilks, W.R., S. Richardson, D. Spiegelhalter. 1995. *Markov Chain Monte Carlo in Practice*. Chapman & Hall/CRC Interdisciplinary Statistics, Taylor & Francis.

Goutte, Cyril, Eric Gaussier. 2005. A probabilistic interpretation of precision, recall and f-score, with implication for evaluation. *Proceedings of the 27th European Conference on*

Advances in Information Retrieval Research. ECIR'05, Springer-Verlag, Berlin, Heidelberg, 345–359. doi:10.1007/978-3-540-31865-1_25. URL `https://doi.org/10.1007/978-3-540-31865-1_25`.

Harvey, A., S. Peters. 1990. Estimation procedures for structural time series models. *Journal of Forecasting* **9** 89–108.

Hochreiter, Sepp, Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Comput.* **9**(8) 1735–1780. doi:10.1162/neco.1997.9.8.1735. URL `https://doi.org/10.1162/neco.1997.9.8.1735`.

Hong, Liangjie, Brian D Davison. 2010. Empirical study of topic modeling in twitter. *Proceedings of the first workshop on social media analytics*. 80–88.

Hsu, Chih-Wei, Chih-Chung Chang, Chih-Jen Lin, et al. 2003. A practical guide to support vector classification.

Jónsson, Elıas, Jake Stolee. 2015. An evaluation of topic modelling techniques for twitter. *University of Toronto* .

Laptev, Nikolay, Saeed Amizadeh, Ian Flint. 2015. Generic and scalable framework for automated time-series anomaly detection. *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*. 1939–1947.

McCullagh, P., J. A. Nelder. 1989. *Generalized linear models (Second edition)*. London: Chapman & Hall.

Mikolov, Tomas, Kai Chen, Greg Corrado, Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781* .

Mikolov, Tomas, Quoc V Le, Ilya Sutskever. 2013b. Exploiting similarities among languages for machine translation. *arXiv preprint arXiv:1309.4168* .

Mikolov, Tomas, Ilya Sutskever, Kai Chen, Greg S Corrado, Jeff Dean. 2013c. Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems*. 3111–3119.

Mimno, David, Hanna Wallach, Edmund Talley, Miriam Leenders, Andrew McCallum. 2011. Optimizing semantic coherence in topic models. *Proceedings of the 2011 conference on empirical methods in natural language processing*. 262–272.

Newman, David, Jey Han Lau, Karl Grieser, Timothy Baldwin. 2010. Automatic evaluation of topic coherence. *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*. HLT '10, Association for Computational Linguistics, USA, 100–108.

Röder, Michael, Andreas Both, Alexander Hinneburg. 2015. Exploring the space of topic coherence measures. *Proceedings of the Eighth ACM International Conference on Web Search and Data Mining*. WSDM '15, Association for Computing Machinery, New York, NY, USA, 399–408. doi:10.1145/2684822.2685324. URL https://doi.org/10.1145/2684822.2685324.

Sammut, Claude, Geoffrey I. Webb, eds. 2010. *TF–IDF*. Springer US, Boston, MA, 986–987. doi:10.1007/978-0-387-30164-8_832. URL https://doi.org/10.1007/978-0-387-30164-8_832.

Taboada, Maite, Julian Brooke, Milan Tofiloski, Kimberly Voll, Manfred Stede. 2011. Lexicon-based methods for sentiment analysis. *Computational linguistics* **37**(2) 267–307.

Taylor, Sean J., Benjamin Letham. 2018. Forecasting at Scale. *The American Statistician* **72**(1) 37–45. doi:10.1080/00031305.2017.138.

Ting, Kai Ming. 2017. *Confusion Matrix*. Springer US, Boston, MA, 260–260. doi:10.1007/978-1-4899-7687-1_50. URL https://doi.org/10.1007/978-1-4899-7687-1_50.

Wang, Chi-Chen. 2011. A comparison study between fuzzy time series model and arima model for forecasting taiwan export. *Expert Systems with Applications* **38**(8) 9296–9304.

Wang, Rong, Wenlin Liu, Shuyang Gao. 2016. Hashtags and information virality in networked social movement: Examining hashtag co-occurrence patterns. *Online Information Review* .

Appendix A

Details of Tweets and Models

```
] model_w2v.wv.most_similar(positive="crypto")

[('boycottbarclay', 0.5705076456069946),
 ('fiat', 0.5685356259346008),
 ('decentralis', 0.5404370427131653),
 ('defi', 0.5369781255722046),
 ('solid', 0.5303640365600586),
 ('leverag', 0.5183134078979492),
 ('cryptotrad', 0.5174787044525146),
 ('binanc', 0.5159591436386108),
 ('corrdin', 0.51348477602005),
 ('itsourmoney', 0.5123952031135559)]

model_w2v.wv.most_similar(positive="fraud")

[('lvc', 0.48169639706611633),
 ('cpa', 0.47740107774734497),
 ('plead', 0.4416187107563019),
 ('sophist', 0.4294194281101227),
 ('retard', 0.4283182621002197),
 ('uncompens', 0.4276801347732544),
 ('joie', 0.4255749583244324),
 ('coincid', 0.424640029668808),
 ('detect', 0.42210114002227783),
 ('becuas', 0.41998621821403503)]

model_w2v.wv.most_similar(positive="barclay")

[('paym', 0.4073803424835205),
 ('void', 0.4072750210762024),
 ('allegedli', 0.38910651206970215),
 ('leagu', 0.3813632130622864),
 ('pingit', 0.3780866265296936),
 ('blacklist', 0.3750053644180298),
 ('onward', 0.3702782690525055),
 ('gut', 0.36663684248924255),
 ('customerexperi', 0.36639147996902466),
 ('miser', 0.363864004611969)]
```

Figure A.1: Most similar words in *Word2Vec*. In the figure, we choose three words: 'crypto', 'fraud' and 'barclay' as input, and the output is the most similar words in the corpus trained in our tweets dataset. The similarity is calculated by using cosine similarity between word vectors

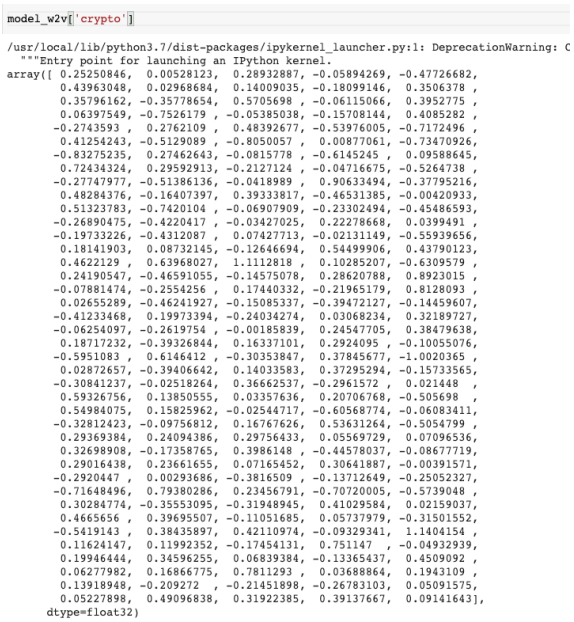


Figure A.2: Vector representation of word 'crypto' in the *Word2Vec* corpus with embedding length of 200.