School of Computer Science and Informatics
Cardiff University
2021

Predicting Information Flow and Survival of Malicious Posts Around Covid-19

Final Report

Author: Paige Lowe
Supervisor: Amir Javed

Table of Contents

1. Abstract

In recent years, social media platforms such as Twitter, Facebook and Instagram have largely increased their presence in our daily lives, with Twitter alone reporting an average of 500 million tweets daily. Though these platforms have a great positive impact on the connectivity we may feel in our lives, they unfortunately leave us open to many new forms of cyber-attack. As part of this research project, I will be looking at factors that influence propagation of malware on the social media platform Twitter, particularly account Covid-19. My proposed solution will focus on building a program to identify factors that are indicative of malicious tweets and that correlate to their size and survival. Survival being considered as the period of time a tweet is actively retweeted and size accounting for the tweets virality, measured in the number of retweets. The factors I will be considering will be a variety of account and content-based features such as tweet sentiment/language and the age of the posting account and its verification status. I hope the results of this research can later be used to aid identification of malicious content and prevent it's spread across social platforms.

2. Acknowledgements

I would like to thank my supervisor Amir, for his continued support and feedback throughout the duration of this project.

3. Introduction

In January 2021 there were estimated to be 3.78 billion [1] social media users across the globe, a value close to half of the world's population. This helps to conceptualise just how vast and diverse the audience for online social platforms really are. For many, the main attraction of these platforms is the ease at which they can share thoughts, experiences and communicate with friends. However, for those with malicious intent, this open sharing culture forms the perfect environment for the propagation of malware, which introduces a multitude of risk within the platforms. In order to minimise and eliminate these risks, it is essential to understand the ways social platforms are manipulated as a medium for malware propagation. Only by understanding this, can we begin to plan ways to prevent and protect against them.

In this project, I will be looking at drive-by download attacks on the social media platform Twitter. The motivation for this project stems from the fact that online social media platforms are becoming increasingly at risk of cyber-attack as those with malicious intent develop more advanced techniques to compromise users.[2] Twitter is an online social media platform that allows its users to send and receive short 'tweets' which other users can interact with by retweeting, liking or commenting. Unfortunately, the 280-character limit and automatic shortening of URLs means that twitter is particularly vulnerable to drive-by download attacks. A drive-by download is the unintentional download of malicious code to a computer or mobile device, in this case it is orchestrated by the user unknowingly clicking a malicious URL which takes them to web page where a script is executed that may cause harm to or exploit the user's device or data.[3] Twitter is particularly susceptible to this form of attack because attackers are readily able to hide malicious URLs in a shortened form inside

seemingly harmless content. By hiding malicious content in this way, the malware is then actively shared across the network unknowingly by users.

In order to prevent these forms of attacks, we first need to successfully identify and recognize the factors that aid their propagation across the network. Propagation being measure by their size and survival. Tweet survival is defined by the length of time a tweet is being actively retweeted and size is number of retweets a tweet receives which allows us to gage its virality. In this paper, I will focus on building a program to identify the factors that correlate with the propagation of these URLs across a social network. In particular, I will focus my research on how attackers make use of Covid-19 to spread malicious content. I have chosen to focus the research on a popular event as they tend to be a prevalent opportunity for attackers to hide their content at times of peak user activity. This assumption is confirmed by the fact that since the beginning of the Covid-19 pandemic there was a 50% increase in breached records. [4] As such I hope the event will provide a surpass of data that will reveal a true reflection of factors that affect tweet size and survival.

4. Background and Related work

4.1 Prediction of Malware Propagation within Communities in Social Media Based Events [5]

This paper was particularly interesting as it shared similarities with my own research, it looked into propagation of malware on Twitter around major events. A key difference however was that this paper focussed on drawing links between communities of users that support the propagation of malware, rather than looking at the individual user or content being shared. They looked at the relationships between users with the goal of identifying communities of accounts that were actively engaged in spreading malware. Based on this, it could be useful to consider the relationships between users sharing tweets but may be something that is outside the scope of this project. Another interesting point is that they used Capture HPC a high interaction client honeypot that analyses malicious URLs using a virtual machine to identify malicious tweets. This could be an alternative to my own planned implementation which would be to use VirusTotal to determine the nature of URLs. VirusTotal works by inspecting a URL with approximately 70 antivirus scanner and URL/domain blacklisting services to determine the nature of the URL.

4.2 Detecting Spam URLs in Social Media via Behavioural Analysis [6]

In this paper, they are researching spam detection in online social platforms. In particular, they consider how the specific behaviour of the user posting the URL and the user clicking the URL could be used to detect spam. They used this as opposed to traditional blacklist filters or analysing the URLs landing page directly. They found that analysing behavioural features resulted in very high precision of results. From this I can see it will be important to look closely at account-based features in my own research but a key difference here is the research focussed on identifying spam rather than looking at malware.

4.3 Tweet and Account Based Spam Detection on Twitter [7]

This paper's research focusses on developing spam detection on Twitter using machine learning. In order to train their model a variety of account and tweet-based factors were selected such as number of friends, followers, screen name, hashtags, mentions etc. The results of their experiment concluded that a combination of both tweet and account-based features led to the most accuracy in detection of spam. Understandably, spam is not a direct relation to malware. However, the fact that looking at account and tweet-based features increased the model's accuracy suggest that there may also be relationships between these factors and classifying tweet's which could be applied to malware and as such would be interesting to investigate their affect, if any, on its survival/size.

4.4 Emotions Behind Drive-by Download Propagation on Twitter [8]

Previous research has also looked directly into how sentiment can be used to understand and predict propagation of malware. In this paper, they discovered that the sentiment and emotion reflected in tweet content could be used as an indicator to its survival. They found that tweets classified as more positive correlated to a higher survival in non-malicious/benign posts, and tweets with a more negative sentiment where the emotion expressed highest was fear, were more likely to aid in the propagation of malicious content. Based on these findings, it can be assumed that analysing the sentiment of the tweet is a significant factor in predicting information flow of the tweet and as such a good metric to include in my own analysis. Evidence that the language used directly correlated to size/survival would also suggest that looking at other aspect of the tweet's language could be important, this could include things such as the number of nouns or verbs used, punctuation or number of emoticons. As emoticons are also an entity that can contextually imply a sentiment. Another interesting part of this paper is that from their data collection, they looked at how the factors they identified affected both malicious and benign tweets, with a comparison being drawn between how factors affect propagation of both. This could be useful in my own research and is something to consider.

4.5 Phishing Email Detection Based on Hybrid Features [9]

In this paper they show that the emotion and language used in phishing emails directly affects user engagement. They found that generally the emotion used in the wording of phishing emails is intended to induce stress and negative emotion. Doing so makes the recipients feel nervous, fearful, anxious and worried, enough so to break through the recipient's psychological defence. Understanding the psychological factors that cause users to engage with phishing emails is important when we consider it has also been found that negative sentiment correlates to survival of malicious content. As such it will be interesting to analyse psychological factors of tweets to see what this may reveal in terms of virality.

4.6 Conclusion of findings

In conclusion, I can see there has been a surpass of prior research into spam detection in online social networks. Of which, the research has been sufficient in concluding that account and content-based factors such as emotion analysis can be important factors in identifying such content. This finding introduces the question of how content and account-based analysis can be used to identify features of drive-by-download attacks and how these features affect the likelihood of successful propagation. This is a question that remains unanswered by previous research but will be the focus of this project by looking in depth at a variety of content/account-based factors and their correlation to tweet size and survival. Understanding

the identifying characteristics of drive-by-download attacks is a critical step towards its prevention and ensuring the online safety of users. Especially when we consider that this form of attack is considered one of the top 5 most common cyber-attacks in 2020. [10] One paper found that 1 in every 500 URLs on social media lead to a malicious site [11], this number becomes more astonishing when we consider that there are estimated to be 500 million tweets posted every day, a quarter of which contain a URL. [12]

5. Methodology

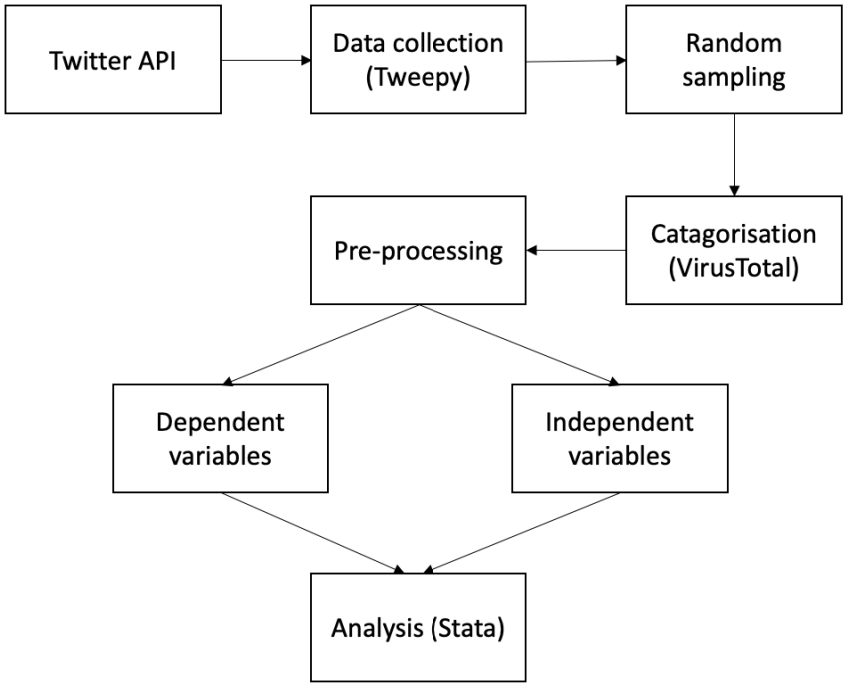

Figure 1. Experimental setup.

5.1 Twitter

Twitter was chosen as the data source for of this project because the research is focussed on understanding malware propagation, specifically by looking at factors of drive-by-download attacks, this is due to the fact they account for the vast majority of cybercrime in online social networks [13]. Drive-by-download is a form of attack to which Twitter, and its users, are particularly susceptible. This is due to the 280-character limit of tweets and automatic shortening of URLS which allows cyber-criminals to hide malicious URLs in seemingly harmless content, unbeknownst to the user. As a platform, Twitter allows authenticated users to stream and collect tweets in bulk via their API. In order to access this API, a Twitter developer account is required which provides authentication credentials. This can be attained by following an online application process detailing the nature of the project and the interactions required from the API – data collection only in this case. Once an application has been approved, the account receives the authentication credentials required to begin a successful connection with the twitter API.

5.2 Python

Python is an interpreted, high-level and general-purpose programming language that can be used for writing scripts. It is a powerful but user-friendly language with a relatively simply syntax and extensive set of supporting libraries. Due to the fact I have accounted significant experience working with this language over the course of my degree, I decided this would be a sensible option for use throughout the project.

5.3 Tweepy

Tweepy is an open-source Python package that offers a simple way to interact with the Twitter API. [14] Tweepy handles authentication, connection, creating/destroying the session and reading/partially routing incoming messages. It is a well-documented package that is actively maintained and offers a variety of classes and methods that can be used to stream and filter tweets by keyword and language. The individual tweets returned by the Tweepy Stream method, contain a JSON object (JavaScript Object Notation) with all the information about the tweet itself and the posting user, this is what will be stored as part of this experiment. As part of this research, we need to collect and analyse Covid related tweets. To ensure this is the case I will be filtering the Twitter stream by the keywords 'covid' and 'corona'. This will remove any tweets that do not include these words from the stream. The rationale behind selection of these two keywords is that at the time of this research, they are 2 of the 3 most trending hashtags on Twitter that are Covid based. [15] Therefore, filtering by these keywords ensures that I will be analysing covid related tweets and hence the results should reflect how the topic of Covid-19 has been used to propagate malware. Covid has been chosen specifically because I want to understand how attackers make use of significant events to spread malicious content. Large events such as Covid tend to be a prevalent opportunity for attackers to spread their content due to the large, diverse and multicultural audience it engages [16]. I will also be filtering the stream by language choosing only to look at tweets written in the English language. This is to avoid any problems further down the line that could be introduced with a dataset of multiple languages, such as sentiment analysis and reporting. The final filter applied to the stream will be that only tweets containing URL's will be considered, this is because the URL will be used later to be classify the tweet as malicious or benign.

The script 'tweets.py' was written and used for collection of tweets and functions as follows.

1. Define twitter authentication credentials
2. Define a Tweepy stream listener
3. Initialise Tweepy stream filtered by language (English) and keyword ('covid', coronavirus')
4. For each tweet received, if the tweet contains a URL store this tweet as part of the dataset

```python
class StdOutListener(StreamListener):
    def __init__(self):
        super().__init__()
    def on_status(self, status):
        if len(status._json['entities']['urls']) > 0 and not 'twitter.com/i/web/status/'
            file.write(json.dumps(status._json))
            file.write('\n')
        return True
    # return true on all as to not kill the stream
    def on_error(self, status):
        print(status)
        return True
    def on_timeout(self):
        print ('Timeout...')
        return True
    def on_disconnect(self, notice):
        print('disconnect')
        time.sleep(60)
        return True

if __name__ == '__main__':
    l = StdOutListener()
    # Authenticate to Twitter
    auth = tweepy.OAuthHandler(consumer_key, consumer_secret)
    auth.set_access_token(access_token, access_token_secret)
    # Create stream
    stream = Stream(auth, l, timeout=36000)
    while True:
        try:
            stream.filter(track=['covid','coronavirus'], languages=["en"])
        except Exception as e:
            # wait 15 minutes before re-connecting the stream
            time.sleep(15*60)
            continue
```

Figure 2. ‘tweets.py’

In more detail, tweets.py works as follows. Twitter credentials are required to authenticate the connection between Tweepy and the Twitter API. The StdOutListener is required to handle routing the incoming tweets through to the appropriate method. For example, it handles problems such as forcing a disconnect if the stream is falling behind. The Stream itself establishes a streaming session and routes messages to the StreamListener instance where the on status function handles each tweet that is received – writing tweets containing a URL to the output file in JSON format.

5.4 Data collection

Data collection began at midnight on the 03/03/21. This date was chosen in the hopes that there would be an increased level of activity around the topic of Covid following the health secretary’s coronavirus update statement earlier that day. [17] It was hoped that collecting data at a time of increased activity would mean the results of analysis more closely reflect how large events are used in the propagation of malware. Data collection ran continuously for 11 days and subsequently concluded at midnight on the 14/03/21. During this time, approximately 1 million covid related tweets were collected. A continuous collection approach was taken due to the nature of processing to be done at a later date, i.e., looking at how various tweets in the data set ‘survived’ and their ‘size’. Data was collected for 11 days but only data from the first 10 days of collection would be part of the sample for this experiment, the excess 24 hours of data collected on the 11th day was collected to ensure that

an accurate survival metric could be calculated for the tweets first seen on the 10th day. A summary of the tweets collected can be seen below.

Table 1. Number of tweets collected

Date	Number of tweets captured
04/03/21	119858
05/03/21	166430
06/03/21	124311
07/03/21	112823
08/03/21	149076
09/03/21	118279
10/03/21	138492
11/03/21	171239
12/03/21	151580
13/03/21	117956

To test that the collection process was working correctly, I created a script and ran it on a subset of the collected tweets to ensure that all tweets collected met the filter criteria. (Must contain the word 'covid' or 'coronavirus' and contain a URL) This script can be seen in the appendix - figure A.

5.5 Sampling

Out of the 1 million tweets collected, a random sample of 10,000 unique tweets will be analysed as part of this experiment. 1000 random and unique tweets from each of the 10 days of data collection, each tweet in the sample will also contain a unique URL to prevent duplicate analysis. Working on a smaller dataset than collected is due to the daily request limitation of the service used to analyse the URLs acting as a bottleneck. To ensure the selection of this sample is collected fairly and randomly, the script 'select_random.py' was used. This script uses the python module 'random' to generate 1000 random numbers in the range of tweets collected on that day. Those numbers can then be used to select the tweets to be analysed further. The Random module is a built-in module of Python and can be used to generate pseudo-random variables and as such we can confidently say that the sample selection was unbiased and truly random. The 'select_random.py' script also checked that for each tweet randomly selected, the URL and tweet had not been previously added to the dataset. Thus, ensuring no duplicate URL's or tweets would be analysed.

'select_random.py'

1. A list of 1000 random integers are generated ranging from 0 to the length of the file (all tweet collected in one day)
2. Two lists are maintained for URLs and tweets previously added to the sample
3. For each entry in the input file
4. If the current counter is equal to any of the random integers, and is not a duplicate URL or tweet
5. The tweet is written to the output file and the URL and tweet are appended to the lists maintaining duplicates

```
len_of_file = sum(1 for entry in open(input_file))
random_ints = random.sample(range(1, len_of_file), 1000)
added_tweets, added_urls = [], []\

with open(input_file) as f:
    x = 1
    for line in f:
        if x in random_ints:
            tweet = json.loads(line)
            # check for duplicte URLs and tweets
            if tweet['text'] in added_tweets or tweet['entities']['urls'][0]['expanded_url'] in added_urls:
                continue
            # if not duplicate, write to output file
            outputFile.write(json.dumps(tweet))
            outputFile.write('\n')
            added_tweets.append(tweet['text'])
            added_urls.append(tweet['entities']['urls'][0]['expanded_url'])
        x+=1
```

Figure 3. 'select_random.py'

In order to test that the random selection script was working correctly, the script found in the appendix - figure B was used. This script takes as input, a file generated by the script select_random.py, it then checks that all tweets in the file contain a unique URL – and that no duplicate URLs are contained in the sample, thus the sample is random.

5.6 VirusTotal

VirusTotal is a free service that allows users to analyse files and URLs to detect any malware it may contain. It does so by inspecting the item in question with over 70 antivirus scanners and URL/domain blacklisting services, in addition to a myriad of tools to extract signals from the content. [18]VirusTotal will be used to classify the randomly sampled dataset into two sub-categories, malicious and benign. Alternative methods for this classification were considered. In particular the approach of using a client honeypot to identify malicious sites. This approach would have used a dedicated virtual machine to interact with the potentially dangerous server and from the result of this interaction been able to determine the sites nature. The disadvantage and deciding factor in disregarding the honeypot approach is that a honeypot is only effective if it is able to deceive an attacker into thinking it is a normal computer system, and as technology progresses, hackers are becoming more aware of honeypots and how to work around them which could lead to the misclassification of malicious URL as benign.[19] VirusTotal overcomes this issue by scanning URLs with over 70 antivirus detectors and is able to classify a URL by identifying any threats resulting from these scans. It is important to note at this point, that due to the limited timescale and resource of this project, the threshold for classifying a URL as malicious is one threat being identified out of approximately 70. This coupled with the potential for false positives may cause misclassifications of benign URLs and if the research was taken further this may need to be considered. With more time and resource, it would be better to determine a more reliable and accurate method of identifying malicious URLs.

By identifying both malicious and benign tweets, I will be able to draw comparisons at a later stage about the differences between the factors that correlate to the survival of both.

In order to access the VirusTotal API a VirusTotal community account is required which provides an API access key. A limitation to the use of VirusTotal is that the public API has a daily request limit of 500 requests. Each URL requires 2 requests to process, one to send the URL to the VirusTotal scanning end point where it will be analysed and one to retrieve the report on any threats identified. This means one account could scan and process a maximum

of 250 URLs a day. By creating multiple accounts, I was able to process 1000 URLs a day between 4 accounts and API keys. The script used to run this process is named 'virusTotal.py' and works as follows.

1. Takes a json file of tweets as input
2. For each tweet append it to the list of tweets to be scanned
3. While number of requests less than daily limit
4. Take 4 tweets and post their URLs to VirusTotal scan end point
5. Wait for 60 seconds – VirusTotal has a max 4 request per minute limit
6. Then retrieve the scanned URL reports from VirusTotal
7. Based on the result either write the tweet to the malicious or benign output file

```
# while num requests less than 2000 (daily quota)
while request_count <= 2000:
  api_key = '938e86059bb865ff49be2898ec2e5e10d6e34a403d2ab2999c6b50076e92d44c'
  if request_count > 495 and request_count < 912: api_key='aa045709cf89b764e27c9969df56841d206d2ebb0d8eab7d8e3bc73e969b2fcf'
  elif request_count > 912 and request_count < 1488: api_key='93c0933ed7fdb547db9855f8419f815f469af3b81ce78c94d72e19dc1511330e'
  elif request_count > 1488: api_key='a79f397a7db308edf8878777094dfbef9d7264778227fd195f39b9942a1c035a'
  # scan and check response of 4 urls at a time (4 request per minute maximum)
  url_batch = url_list[x: y+1]
  response = scan(url_batch, api_key)
  time.sleep(60)
  report(response, api_key)
  time.sleep(60)
```

Figure 4. 'virusTotal.py'

```
def scan(url_batch, api_key):
  url = 'https://www.virustotal.com/vtapi/v2/url/scan'
  scan_url_list = []
  for URL in url_batch:
    expanded_url = URL['entities']['urls'][0]['expanded_url']
    try:
      params = {'apikey': api_key, 'url': expanded_url }
      response = requests.post(url, data=params)
      URL['scan_id'] = response.json()['scan_id']
      scan_url_list.append(URL)
    except ValueError as e:
      print("Rate limit detected:", e)
      continue
```

Figure 5. 'virusTotal.py'

```
def report(scan_id_list, api_key):
  url = 'https://www.virustotal.com/vtapi/v2/url/report'
  for entry in scan_id_list:
    try:
      params = {'apikey': api_key, 'resource': entry['scan_id'] }
      response = requests.get(url, params=params)
      # if threat identified, write tweet to file
      if response.json()['positives'] > 0:
        output_file.write(json.dumps(entry))
        output_file.write('\n')
      else:
        benign_output.write(json.dumps(entry))
        benign_output.write('\n')
    except ValueError as e:
      print("Rate limit detected:", e)
      continue
    except Exception:
      print("Error detected:")
      continue
```

Figure 6. ‘virusTotal.py’

After running this script, a total of 641 malicious tweets and 9359 benign tweets were identified, 10,000 in total. Classification of URLs as malicious and benign is a critical part of the experiment. To ensure that it was working correctly I ran the virusTotal.py script on a file containing known malicious and known benign URLs. The known benign URLs were taken from the Cardiff University site and the malicious URLs were taken from AV comparatives which provided public site URLs known to be malicious. [20] To double check each URL was malicious or benign, I used an external threat scanning tool – google transparency report, to scan the URLs to clarify their nature. [21] I then checked that after processing using the script virusTotal.py, that the URLs were categorised into the correct class.

5.7 Pre-processing

In order to continue the experimental investigation, we need to identify the factors that we will be investigating in regard to their effect on information size and survival of malicious content. This means identifying both independent and dependent variables of the experiment. The dependent variable is the variable being tested and measured in an experiment and is 'dependent' on the independent variable. The independent variables are the variables the we will be manipulating and is assumed to have a direct effect on the dependent variable. [22] The aim being to infer any correlation or relationship between the dependent and independent variable. In this case, identify any factors that correlate to the size/survival of malicious content.

5.7.1 Independent variables

The independent variables of this experiment fall into 2 main categories - Account and Tweet based factors. These categories were chosen as there has been previously supporting research to suggest that they can be used to successfully identify malicious content in online social networks. [23]

5.7.1.1 Account-based factors

These are features that originate from a user’s account and are specific to any one user and can be seen in table 2. The account-based features will be pulled from the user who posted the original tweet. The features may be defined by the user (e.g., user description) or automatically generated as the account builds relationships on the platform (e.g. number of followers).

Table 2. Account-based factors

Factor	Description	Type
Location	Presence of a user-defined location	User
URL	Presence of URL in the profile	User
Description	Presence of a user-defined UTF-8 string describing account	User
Default profile	Whether user has altered their profile theme or background	User
Verified	Whether the user has a verified account	N/A

Followers	Number of followers	Time in platform
Friends	Number of friends	Time in platform
Lists	Number of public lists a user is a member of	User controlled
Favourites	Number of favourited posts	User controlled
Statuses	Number of statuses posted	Time in platform
Age	Age of account at time of tweet – In days	Time in platform

The 'user' factors are factors that the user can control about their profile for other users to see. By analysing these factors, we aim to draw any correlations between them and malware propagation and could at a later date be used in identification of accounts more likely to spread malicious content. The 'time in platform' features are those generated from user engagement within the platform. They represent the network a user has built (e.g. Followers) or the statuses and age of the account.

5.7.1.2 Tweet-based factors

These originate directly from the content of the tweet and interactions with other users across the network, they can be seen in table 3. It involves analysis of the tweet itself to question whether the language or various media associated with the tweet has any relationship to the nature of the content.

Table 3. Tweet-based factors

Factor	Description	Type
Sentiment	Identification of anger, anticipation, disgust, fear, joy, sadness or surprise-based words in tweet	Language
Emotion	Identification of negative or positive language in tweet	Language
Nouns	Nouns present in tweet	Language
Verbs	Verbs present in tweet	Language
Adjectives	Adjectives present in tweet	Language
Length tweet	Length of tweet by characters	Language
Average word length	Average word length	Language
Repeated words	Count of repeated words in tweet	Language
Punctuation	Count of punctuation in tweet	Language
Sensitive	Whether the content of tweet has been classed by twitter as sensitive	Tweet
Reply status	Whether the tweet was a reply status	Tweet
Quote status	Whether the tweet was a quote status	Tweet
Reply count	Number of replies to tweet	Tweet
Favourite count	Number of favourites tweet receives	Tweet
Media	Number of images/Gifs in the tweet	Additional content
Hashtags	Number of hashtags in the tweet	Additional content
Mentions	Number of user mentions	Additional content
URLs	Number of URLs contained in the tweet	Additional content
Age	Age of tweet in days	Tweet
Hour posted	Time of day the tweet was posted	Tweet

A range of language-based factors have been chosen as is it has been previously shown that sentiment analysis can be used for improving accuracy of spam detection software [24] [25]. Therefore, it would be interesting to investigate how sentiment and other language factors correlates to the size and survival of malicious content. General tweet-based factors such as reply and favourite count were investigated as they are indicators or user engagement surrounding the tweet. As such it would be expected that a higher value of these features could directly correlate to survival. The 'additional content' factors have been chosen due to the fact there is previous research to suggest that this presence of these items statistically increases user engagement [26]. As such, it would be interesting to draw any connections between them and their effect on propagation of malware.

In order to extract these factors, the script 'analysis.py' was used. Which executes as follows.

1. For each tweet, identify if it's a retweet or original tweet
2. If tweet is truncated – run processing on extended tweet object
3. Extract features from tweet
    a. Sentiment – NRCLex
    b. Nouns/verbs/adjectives – NLTK
    c. Age of tweet and account (subtraction of start date from end date)
    d. Capture and convert non-integer attributes stored in tweet to 1 or 0 for true and false respectively. (user description presence)
4. Write the extracted features to a csv file

```
output_file = "mal-final-data.csv"

with open(output_file, 'a', newline='') as file:
    writer = csv.writer(file)
    # add field headings to csv
    writer.writerow(["anger", "anticipation", "disgust","fear","joy","negative","positive","sadness","suprise","trust","nouns","verbs",
    # for every tweet we are going to process
    with open(input_file) as f:
        for line in f:
            # row variable is a list that represents a row in csv
            row = []
            whole_tweet = json.loads(line)
            tweet = whole_tweet

            # if tweet is a retweete analyse the original tweet data
            if 'retweeted_status' in tweet:
                tweet = whole_tweet['retweeted_status']
            # if tweet is truncated the full text to be analysed and entities will be contained wihtin the extended tweet json object
            tweet_entities = tweet['entities']
            tweet_text = tweet['text']

            if tweet['truncated'] == True:
                tweet_text = tweet['extended_tweet']['full_text']
                tweet_entities = tweet['extended_tweet']['entities']
```

Figure 7. 'analysis.py'

NRCLex is a Python library that can be used to predict the sentiment and emotion contained in text. [27] It does so by relying on the underlying dataset offered by NRC Word-Emotion Association Lexicon which is a list of English words and their associations with eight basic emotions (anger, fear, anticipation, trust, surprise, sadness, joy, and disgust) and two sentiments (negative and positive). [28] Using this it is able to provide a basic sentiment and emotion analysis on a numerical scale for each tweets content.

```
text_object = NRCLex(tweet_text)
row.append(text_object.affect_frequencies['anger'])
row.append(text_object.affect_frequencies['anticip'])
row.append(text_object.affect_frequencies['disgust'])
row.append(text_object.affect_frequencies['fear'])
row.append(text_object.affect_frequencies['joy'])
row.append(text_object.affect_frequencies['negative'])
row.append(text_object.affect_frequencies['positive'])
row.append(text_object.affect_frequencies['sadness'])
row.append(text_object.affect_frequencies['surprise'])
row.append(text_object.affect_frequencies['trust'])
```

Figure 8. Emotion analysis of tweet using 'analysis.py'

Natural language toolkit (NLTK) is suite of Python libraries that can be applied for statistical natural language processing (NLP). [29] In the image below it is used to identify the number of nouns, verbs and adjectives present in the tweet text. It does so by first splitting text into smaller units called tokens, this process is known as tokenization. This tokenized text can then be part-of-speech (POS) tagged, which is a process of labelling each token based on its context and meaning. For example, the token words 'him, 'her' or 'herself' would be tagged as personal pronoun (PRP). To identify nouns, verbs and adjectives the tags 'NN', 'VB' and 'JJ' were in each tokenized tweet.

```
tokenized = nltk.word_tokenize(tweet_text)
nouns = [word for (word, pos) in nltk.pos_tag(tokenized) if(pos[:2] == 'NN')]
verbs = [word for (word, pos) in nltk.pos_tag(tokenized) if(pos[:2] == 'VB')]
adjectives = [word for (word, pos) in nltk.pos_tag(tokenized) if(pos[:2] == 'JJ')]
```

Figure 9. Identification of nouns/verbs/adjectives in tweet using 'analysis.py'

To be confident that the script was correctly in identifying and calculating the independent factors found I randomly sampled 10 tweets and manually calculated what the output for each tweet should be. I then processed these tweets using the script referred to in figure 7/8/9. Finally, I cross checked the results for each tweet and found no disparity between them and as such was confident that each factor was being calculated and outputted correctly.

5.7.2 Dependent variables

In order to identify the factors that correlate to malware propagation the experiment considers two dependent factors, tweet size and survival. Survival is defined as the time the tweet is being actively retweeted and size is defined by the number of retweets a tweet receives. We measure survival by looking at whether or not a tweet had been retweeted after 12 hours of its creation. A 12-hour window for measuring survival was chosen because we wanted to ensure that only tweets that were active on the network for a prolonged period were part of the model. Also, there had been previous research into the effect of a 24-hour survival window[30] and so we wanted to investigate the effect halving this window may have on the results. Size was considered as the number of retweets is a direct indication of user engagement with the content. Furthermore, size and survival are important factors when considering malware propagation as increasing either of these factors raises the threat that malicious content poses to users. A larger retweet count increases the potential number of users at risk from the malware and the longer the time the malware is being actively retweeted raises the time that these users are at risk.

We are able to calculate the retweet count for each tweet in the benign and malicious dataset using the script 'count_retweets.py'. Which works by first iterating over the entire dataset, creating a list of tweets that contain a retweet and their key values that change over time (such as retweet count and reply count). Tweets that contain a retweet are identified by the fact they contain a 'retweeted_status' object. [31] This 'retweeted_status' object holds all information about the original tweet including the posting user account.

```
for json_file in reversed(files):
    with open(json_file) as l:
        for entry in l:
            tweet_data = {}
            dataset_tweet = json.loads(entry)
            # if its a retweet pull out relevent fields
            if 'retweeted_status' in dataset_tweet:
                tweet_data['id'] =  dataset_tweet['retweeted_status']['id']
                tweet_data['created_at'] = dataset_tweet['created_at']
                tweet_data['retweet_count'] =  dataset_tweet['retweeted_status']['retweet_count']
                tweet_data['reply_count'] = dataset_tweet['retweeted_status']['reply_count']
                tweet_data['favorite_count'] =  dataset_tweet['retweeted_status']['favorite_count']
                all_data.append(tweet_data)
```

Figure 10. 'count_retweets.py'

The script then iterates over the entire dataset of malicious and benign tweets and for each tweet loops over the list of retweeted tweets created above. If a tweet is found matching the ID of a retweet and was created after the current created date, the values are updated to that of the time the tweet was last seen. By doing so, an up-to-date value for retweets, replies and favourites is stored at the time the tweet was last seen. Reply count and favourite count are collected as they are part of the independent variables of the experiment.

```
        # iterate over list of all retweets/ids/created
        for entry in all_data:
            retweet_created = datetime.strptime(entry['created_at'], '%a %b %d %H:%M:%S %z %Y')
            if entry['id'] == tweet_id and retweet_created > last_seen:
                last_seen = retweet_created
                last_seen_date = entry['created_at']
                retweets = entry['retweet_count']
                reply_count = entry['reply_count']
                favorite_count =  entry['favorite_count']
```

Figure 11. 'count_retweets.py'

In order to test the functionality of the script, I manually selected a random small sample of 10 tweets. I then processed these tweets using the script seen in figure 10 and 11. For each processed tweet I then located the last occurrence of the tweets ID in the dataset and ensured that the figures for retweet count, reply count, favourite count and last seen matched those stored in the latest occurrence of the tweet. As all values matched, I was confident that the script was pulling out the most up to date values for each field.

5.8 Analysis

Stata is a powerful statistical software that enables users to analyse data and produce statistical models. This software will be used to analyse and model the dataset collected and generate statistical tables identifying any correlations between the dependent and independent variables discussed above. Stata is a licenced software and in order to download and activate it, a set of authorisation credentials are required, which can be obtained by applying for an

account using Stata's website. [32] Alternative tools to Stata were considered, namely SPSS which is also a statistical modelling software package. From research, I found that SPSS is ideal for modelling complex and multivariate data, which was not a requirement for this project. I also found that the documentation for Stata was easier to follow allowing for a shallower learning curve and as such opted to use this tool.

5.8.1 Modelling tweet size

In order to identify factors that influence the number of retweets (size) in both benign and malicious tweets we will be using a count data model. A count data model is a model where a non-negative dependent variable can be considered against a range of independent variables allowing inference to be drawn on the ways the independent variable affects the value of the dependent variable. In the case of this experiment, retweet count is taken as the dependent variable. Because we are looking directly at malware propagation, it is important that our model considers 'viral' tweets, i.e. none that have 0 retweets as this would affect the accuracy of the model. There has been previous research that suggested 5 retweets is a good threshold for measuring a tweets propagation. [30] However, this does not consider the fact that even 1 retweet can have significant impact when considering the number of followers of the retweeting account. For example, the average user has 707 followers, and as such for every retweet a further 707 users are potentially exposed to the malicious content in their feed. [33] Because of this, a lower threshold of at least 3 retweets has been chosen for the experiment. After discounting any tweets whose retweets count was below the threshold, the final dataset contains 278 malicious tweets and 6,628 benign tweets. A summary of this data can be seen in Table 4 below. One thing to note is that the factors that are emotion or sentiment based (anger, negative, positive etc.) are scored between 0 and 1 according to the weight of their presence in the content of the tweet. 1 being extremely high and 0 being no presence of the emotion/sentiment. Furthermore, the factors reply status, quote status, possibly sensitive, location given, URL given, description given, default profile and verified are Boolean values stored as 1 for true and 0 for false.

Table 4. Summary of variables in final dataset

Variable	Mean		Std. Dev.		Range	
	Mal	Ben	Mal	Ben	Mal	Ben
Anger	0.04	0.03	0.08	0.1	0 - 0.5	0 - 1
Anticipation	0	0	0	0	0 - 0	0 – 0
Disgust	0.02	0.18	0.05	0.06	0 - 0.29	0 – 1
Fear	0.05	0.05	0.1	0.11	0 – 0.5	0 – 1
Joy	0.04	0.02	0.09	0.07	0 - 0.5	0 – 1
Negative	0.07	0.09	0.15	0.18	0 – 1	0 – 1
Positive	0.2	0.21	0.32	0.33	0 – 1	0 – 1
Sadness	0.03	0.04	0.08	0.93	0 - 0.5	0 – 1
Surprise	0.03	0.22	0.15	0.09	0 – 1	0 – 1
Trust	0.07	0.7	0.19	0.17	0 – 1	0 – 1
Nouns	10.72	8.3	5.56	4.23	2 – 38	1 – 47
Verbs	1.73	1.9	1.56	1.58	0 – 10	0 – 18
Adjectives	1.66	1.37	1.39	1.28	0 – 7	0 – 10
Age Tweet	48.3	80.65	83.73	517.90	0 – 860	0 – 9385
Survival 1/0	0.65	0.63	0.5	0.48	0 – 1	0 – 1
Hour posted	11.96	12.43	7.5	7.39	0 – 23	0 – 23

Reply status	0	0.05	0	0.22	0 – 0	0 – 1
Quote status	0.01	0.12	0.1	0.39	0 – 1	0 – 1
Possibly sensitive	0	0.01	0	0.08	0 – 0	0 – 1
Reply count	43.01	49.99	158.8	306.65	0 – 1590	0 – 15352
Retweet count	313.35	277.99	1115.46	5024.88	3 – 6384	0 - 311195
Favourite count	1086.91	1096.69	4268.68	22936.28	0 - 27653	0 - 1423372
Media	0.24	0.16	0.43	0.4	0 – 1	0 – 4
User mentions	0.53	0.23	1.94	0.94	0 – 17	0 – 16
Hashtags	1.27	0.43	3.32	1.96	0 – 20	0 – 22
URLs	1.07	1.03	0.32	0.27	1 – 3	1 – 10
Len of tweet	7.81	109.27	59.02	46.31	36 – 304	30 – 322
Avg word length	7.81	7.35	1.39	1.67	5 – 13	4 – 19
Repeated words	0.37	0.37	1.25	1.03	0 – 10	0 – 17
Punctuation	2.27	2.18	2.03	2.04	0 – 13	0 – 21
Location given	0.78	0.81	0.42	0.39	0 – 1	0 – 1
URL given	0.84	0.76	0.37	0.43	0 - 1	0 – 1
Description given	0.99	0.98	0.12	0.15	0 – 1	0 – 1
Default Profile	0.26	0.29	0.44	0.46	0 - 1	0 – 1
Verified	0.53	0.55	0.5	0.49	0 - 0	0 – 1
Followers	1010721	2050898	2329757	5393541	0 - 6288894	13648478
Friends	15387.85	8179.04	74215.8	37523.69	0 - 532880	0 – 649460
Lists	5065.82	10526.82	12616.65	25667.9	0 - 68879	0 – 867952
Favourites	21550.97	30726.39	87137.69	78162.04	0 - 862046	0 – 867952
Statuses	137089.4	162588.8	192329.2	217451.2	50 - 1003091	3 – 2453576
Age account	137089.4	3401.752	1435.673	1478.33	30 - 5105	3 – 5221

One option for modelling count data is the Poisson Probability Distribution, a discrete probability distribution that expresses the probability of a given number of events occurring in a fixed interval of time, if these events occur with a known constant mean rate and independently of the time since the last event. Unfortunately, a unique feature of the Poisson distribution is that it must satisfy the equidispersion property, which states the mean and variance of a Poisson-distributed variable should be the same.

$$E(y|x) = var(y|x) = \mu$$

Figure 12. Formula for testing equidispersion.

From the summary of variables in table 4 above, we can take the dependent variable retweets. The mean for the malicious data set was 313.35, however the variance (calculated using Stata) was 1244257. From the disparity between these values, we can see the formula is not satisfied, the variance is more than the mean, and as such there is overdispersion of the data. Further evidence of this dispersion can be seen in figure 13 and 14 below.

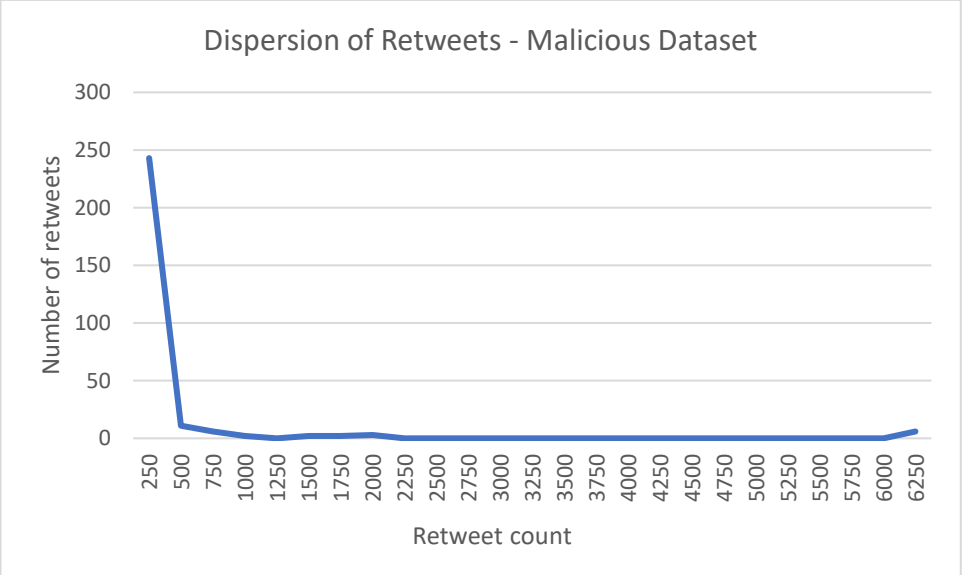

Figure 13. Chart showing the over dispersion of retweets among the malicious dataset.

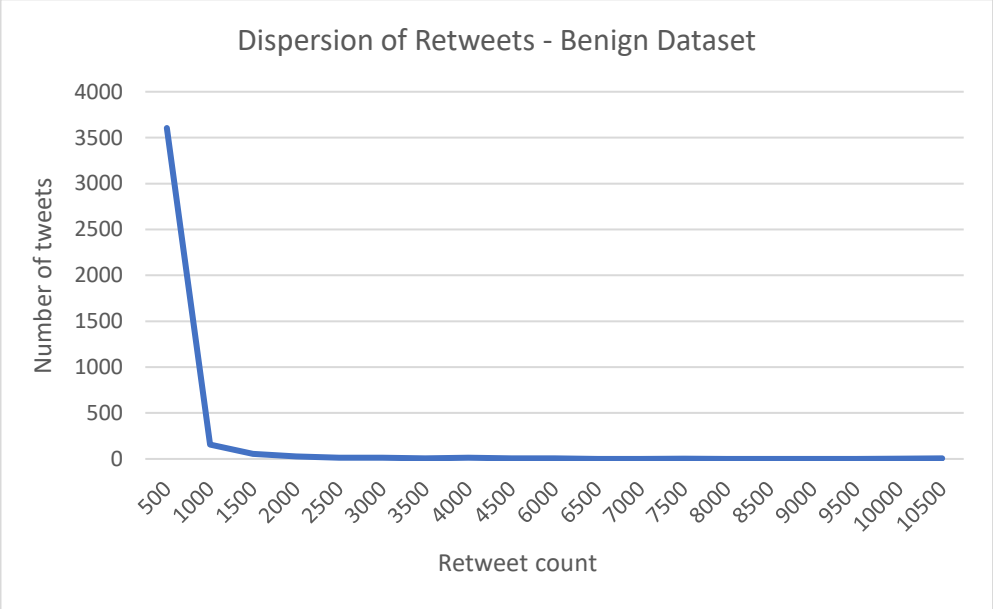

Figure 14. Chart showing the over dispersion of retweets among the benign dataset.

Because of the disparity concluded above, the more flexible negative binomial model was chosen to model the count data. The negative binomial model is a less restrictive model and does not hold the equidispersion property, it is able to do so as it holds one more parameter than the Poisson regression that is used to adjust the variance independently from the mean.

The negative binomial model is a discrete probability distribution for random variables, where the random variable is the number of repeated trials, X, that produce a certain number of successes. In other words, i.e. it models the number of failures before a success.

$$\Pr(X = x) = \mathrm{nb}(x; r, p) = \binom{x + r - 1}{r - 1} p^r (1 - p)^x \qquad \text{for } x = 0, 1, 2, 3, \ldots$$

Probability mass function

$$\mathrm{E}(X) = \frac{r(1 - p)}{p}$$

Expectation

$$\mathrm{Var}(X) = \frac{r(1 - p)}{p^2}$$

Variance

Figure 15. Probability mass function, expectation and variance formulas for the negative binomial distribution. [34]

A probability mass function (PMF) is a function that gives the probability that a discrete random variable is exactly equal to some value. [35] The PMF of the negative binomial model, figure 15, is a formula where r is the number of successes, x is the number of failures and p is the probability of success.

5.8.2 Modelling tweet survival

Survival modelling is a branch of statistics whereby the focus is to analyse the expected time until some failure event occurs. [36] In this case, survival is measured by whether a tweet is being actively retweeted after 12 hours of its creation. This 12-hour cut-off for retweets is the failure event for the model. The aim of using a survival model is to identify any correlation between our independent measures and tweet survival. For example, how does the sentiment of the tweet affect a tweets chance at survival. The specific model used in this experiment is the Cox proportional model. This model is a class of proportional hazard model which are commonly used to measure associations between survival time and a range of predictor variables. The effect on survival can be found by looking specifically at how independent factors affect the rate of failure occurring in the model. This rate is known as the hazard rate.

$$h(t) = h_0(t) \times exp(b_1 x_1 + b_2 x_2 + \ldots + b_p x_p)$$

Figure 16. Cox model expressed by hazard function h(t) [37]

The hazard function in figure 16 shows the formula to determine risk of failure at time t. It is dependent on covariates *x* which are the independent variables of this experiment. *b* are the coefficients which measure the impact of the independent variable as their size changes. *h0* represents the baseline hazard which is the hazard when all predictor variables are equal to 0. [38] Stata uses this underlying formula to determine the hazard ratio of our independent variables against tweet survival.

6. Results and Evaluation

6.1 Results

6.1.1 Results of Tweet size

Table 5 and 6 below are the results of the count data model of the benign and malicious dataset in regard to the independent variable - size. In these tables, the coefficient of the regression model is shown as incident rate ratios (IRR) rather than the logged form. This is because the interpretation as IRR is much simpler. The IRR is essentially the ratio of the expressed category to the base category, the base category being the dependent variable – size, and the expressed category being the independent variables. For experimental conclusions, it stands that an incident rate of more than 1 signifies the variables positive/increasing effect on the size variable, whereas a negative incident rate signified a decreasing effect on size/retweet count. For example, if an independent variable X has an IRR=1.45, this means for every unit increase in the dependent variable X the expected number of Y (dependent variable) would increase by 45% and so forth. For a negative incident rate of IRR=0.45, for every unit increase in X, Y would be 55% smaller.

It is also important to consider that from the tables, only variables whose significance (p-value) is less than 0.05 will be considered. In statistics, the null hypothesis is one that suggests there is no statistical relationship between an independent and dependent variable. The p-value of any variable represents the probability that this null hypothesis is true. It is the probability that random chance created the data rather than a significant correlation being identified. By only considering variables whose p-value is less than 0.05 we can assume that the variable is significant to the experiment. Naturally, a larger sample size tends to identify more significant relationships due to the fact that the chance of random error is reduced.

Table 5. Results of size model for malicious tweets

Predictor	IRR	Std. Err	Z	Sig.
Account Factors				
Location	0.8907886	0.1697737	-0.61	0.544
URL	1.287562	0.2615222	1.24	0.213
Description	0.6170105	0.3586092	-0.83	0.406
Default profile	0.7095076	0.1800022	-1.35	0.176
Verified	1.17933	0.237741	0.82	0.413
Followers	1	0.0000001	0.69	0.000
Friends	1.000001	0.0000014	0.58	0.564
Lists	0.9999642	0.0000163	-2.14	0.032
Favourites	1.000002	000000073	2.26	0.024
Statuses	0.9999984	0.00000074	-2.23	0.014
Age	0.9998486	0.0000791	-1.91	0.056
Tweet factors				
Anger	1.191788	1.233983	0.17	0.865
Anticipation	Omitted			
Disgust	53.48492	97.11625	2.19	0.028
Fear	0.2182442	0.19044	-1.74	0.081
Joy	3.900949	4.289566	1.24	0.216

Negative	2.961506	1.967209	1.63	0.102
Positive	1.407421	0.3513921	1.37	0.171
Sadness	0.10452	0.1357131	-1.74	0.082
Surprise	4.175863	2.015518	2.96	0.003
Trust	1.559633	0.5675084	1.22	0.222
Nouns	1.014304	0.0462784	0.31	0.756
Verbs	1.086293	0.0800414	1.12	0.261
Adjectives	0.8601301	0.0674682	-1.92	0.055
Age tweet	1.012459	0.0017775	7.05	0.000
Reply status	Omitted			
Quote status	1.350835	1.100449	0.37	0.712
Sensitive	Omitted			
Reply count	1.00416	0.0010177	4.10	0.000
Time posted	0.9968826	0.0091904	-0.34	0.735
Favourite count	1.000049	0.0000357	1.37	0.171
Media	0.6084061	0.1233284	-2.45	0.014
User mentions	0.8282545	0.0595978	-2.62	0.009
Hashtags	0.8430447	0.0439309	-3.28	0.001
URLs	0.6222543	0.1900267	-1.55	0.120
Len of tweet	1.004338	0.0051927	0.84	0.403
Avg word length	1.004338	0.0051927	-0.17	0.867
Repeated words	0.7744767	0.0761064	-2.60	0.009
Punctuation	0.9997557	0.0468728	-0.01	0.996

Significant account factors for malicious tweets

The account factors that were identified to have statistical significance in relation to the size of malicious tweets were number of *followers*, number of public *lists*, number of *favourited posts* and number of *statuses*. The number of followers had Z 0.69, $p<0.0$ and IRR=1. The IRR being exactly one tells us that though number of followers is significant in the model, it does not affect the size variable (retweets) as followers increases or decreases. Lists, had a Z of -2.14, $p<0.04$ and IRR=0.9999642, showing that as the number of public lists a user is a member of increases, the size of the tweet decreases by a very small amount (0.00003% approximately). Favourites, has an IRR=1.000002, Z=2.26 and $p<0.03$, which shows a positive correlation with tweet size as the number of favourited posts increases. Statuses was found to have IRR=0.9999984, Z=-2.23 and $p<0.02$, this IRR shows that as the number of statuses an account has posted increases, the size of the tweet decreases.

Significant tweet factors for malicious tweets

The tweet-based factors identified by the model to have a statistically significant effect on tweet size were the emotions - *disgust* and *surprise* and the number of *media*, *hashtags*, *user mentions* and *repeated words* contained within the tweet. Both emotions – disgust and surprise were found to have a positive effect on size, meaning that as the presence of these emotions increased, the likelihood of a higher retweet count occurred. Disgust had Z=2.19, $p<0.03$ and IRR=53.48492, indicating a 5248% increase in size and Surprise had had Z=2.96, $p<0.01$ and IRR=4.175863 indicating a 317% increase in tweet size as the presence of this emotion increased. These are very significant effects and suggest emotion has a big impact on

the size of a tweet. The other variables all had a negative effect on tweet size as their count increased. Media had Z=-2.45, p<0.02 and IRR=0.6084061 meaning a 39% decrease in tweet size as the number of media increased. User mentions had Z=2.62, p<0.01 and IRR=0.8282545 indicating a 17% decrease in tweet size as the number of mentions increased. Hashtags had Z=-3.28, p<0.01 and IRR=0.8430447, which shows a 15% decrease in tweet size for each new hashtag. Finally, repeated words had Z=-2.6, p<0.01 and IRR=0.7744767 which gives a 22% decreased in size as the number of repeated words increases.

Table 6. Results of size model for benign tweets

Predictor	IRR	Std. Err	Z	Sig.
Account Factors				
Location	0.9385501	0.0536786	-1.11	0.267
URL	1.48343	0.0817436	7.16	0.000
Description	1.175393	0.1625148	1.17	0.242
Default profile	1.464182	0.0858512	6.50	0.000
Verified	1.191323	0.0645973	3.23	0.001
Followers	1	0.00000001	-3.16	0.002
Friends	0.9999998	0.00000063	-0.31	0.759
Lists	1.000012	0.00000228	5.15	0.000
Favourites	1.000001	0.000000294	2.59	0.010
Statuses	0.999999	0.000000117	-8.32	0.000
Age	.9999766	0.000019	-1.19	0.236
Tweet factors				
Anger	0.7648597	0.154625	-1.33	0.185
Anticipation	Omitted from model			
Disgust	3.794491	1.463511	3.46	0.001
Fear	0. 818164	0.15843	-1.04	0.300
Joy	0.8035193	0.2474462	-0.71	0.477
Negative	0.9172713	0.1111824	-0.71	0.476
Positive	0.9250729	0.0599213	-1.20	0.229
Sadness	1.09409	0.2834104	0.35	0.728
Surprise	0.7831688	0.1822009	-1.05	0.293
Trust	1.256255	0.1470007	1.95	0.051
Nouns	1.026695	0.0108273	2.50	0.112
Verbs	1.105507	0.0212385	5.22	0.000
Adjectives	1.001214	0.0212062	0.06	0.954
Age tweet	1.00031	0.0000662	4.68	0.000
Reply status	0.8746888	0.0892705	-1.31	0.190
Quote status	1.094395	0.0633684	1.56	0.119
Sensitive	0.8495575	0.2084026	-0.66	0.506
Reply count	1.004486	0.000313	14.36	0.140
Time posted	0.9847958	0.0027508	-5.48	0.061
Favourite count	1.000216	0.000013	16.63	0.000
Media	0.8584157	0.0520709	-2.52	0.012
User mentions	0.8412095	0.0212234	-6.85	0.000
Hashtags	0.9843632	0.0132094	-1.17	0.240
URLs	1.302912	0.1079403	3.19	0.001
Len of tweet	0.9973207	0.0014579	-1.84	0.066

Avg word length	1.008068	0.0147535	0.55	0.583
Repeated words	0.9207511	0.022241	-3.42	0.001
Punctuation	1.006758	0.0122393	0.55	0.580

Significant account factors for benign tweets

The statistically significant account factors for benign tweets were *URL given*, *default profile*, *verified*, *followers*, *lists*, *favourites* and *statuses*. All factors bar statuses had a positive effect on tweet size as their value increased. URL given, is a Boolean attribute collected as 1 for True and 0 for False with Z=7.16, p<0.0 and IRR=1.48343. The IRR tells us that when a user gave a URL as part of their account profile, the effect on tweet size was a 48% increase. Default profile was another Boolean value, made true when the user had personalised their twitter account, the results from the model were Z=6.5, p<0.0 and IRR=1.464182. This IRR shows that when a user customised their profile, the tweet size was increased by 46%. Verified had Z=3.23, p<0.01 and IRR=1.191323, as another Boolean value, we can see that a verified account had a 19% increasing effect on tweet size. Number of followers had Z=-3.16, p<0.01 and IRR=1, as mentioned prior, and IRR of 1 signifies no particular correlation to the size variable as it increases or decreases. The Lists factor had Z=5.15, p<0.0 and IRR=1.000012 and Favourites had Z=2.59, p<0.01 and IRR=1.000001, both factors have a positive correlation with tweet size. The only account-based factor to have a negative correlation with tweet size was the number of statuses a user account had posted with Z=-8.32, p<0.00 and IRR=0.999999.

Significant tweet factors for benign tweets

The tweet-based factors identified by the model to have a statistically significant effect on tweet size were *disgust, number of verbs, favourite count, user mentions URLs* and *repeated* words. Disgust was found to have the most significant effect with Z=3.46, p<0.01 , IRR=3.794491. This IRR signifies a 279% in tweet size as the presence of disgust in the tweet language increases. The number of verbs (Z=5.2, p<0.00 , IRR=1.105507) was also found to increase tweet size by roughly 10% for each new verb in the tweet. Verbs are a descriptive word and so may suggest a more detailed tweet increases retweets in benign content. The favourite count of the tweet (Z=16.63=, p<0.00 , IRR=1.000216) and the number of URLS (Z=3.19, p<0.01 , IRR=1.302912) also increased tweet size, for each new URL attached to the tweets content, size increased by 30%. The other factors, Media (Z=-2.52, p<0.00 , IRR=0.8584157), User mentions (Z=-6.85, p<0.01 , IRR=0.8412095) and repeated words (Z=-3.42, p<0.01, IRR=0.920751), all negatively affected tweet size with both media and user mentions decreasing tweet size by 15% as they increased.

6.1.2 Results of Tweet survival

Below, table 7 and 8 depict the results of the Cox proportional hazard model for the dependent variable, survival. The effect of account and tweet-based factors can be surmised by their reported hazard ratio. The hazard ratio represents the risk of failure event based on the value of the independent factor, the failure event being the case that a tweet is not actively retweeted after a 12-hour period. It can be interpreted such that a hazard ratio of more than 1 is associated with increased risk of failure and decreased chance of tweet survival past the 12-hour threshold. Whereas a hazard ratio of less than 1 signifies a decreased risk of failure and

increased chance of tweet survival. Once again, only factors with statistical significance are considered. ($p<0.05$)

Table 7. Results of survival model for malicious tweets

Predictor	Haz. Ratio	Std. Err	Z	Sig.
Account Factors				
Location	1.237364	0.4016686	0.66	0.515
URL	2.327596	0.2438721	2.47	0.013
Description	1.788154	0.9060311	1.15	0.254
Default profile	1.52033	0.527422	1.21	0.220
Verified	6210493	0.150181	1.97	0.061
Followers	1	0.00000212	2.15	0.031
Friends	0.9999979	0.00000125	1.68	0.094
Lists	0.9999598	0.0000348	-2.47	0.078
Favourites	0.9924974	0.0000182	1.42	0.203
Statuses	1.000001	0.00000457	1.95	0.050
Age	1.000203	0.0001051	1.93	0.042
Tweet Factors				
Anger	0. 4704279	0.4741937	0.75	0.045
Anticipation	Omitted			
Disgust	0.0005296	0.0016682	-2.39	0.097
Fear	1.550732	1.564238	0.43	0.662
Joy	7.682571	14.38559	1.09	0.276
Negative	0.4076936	0.2689619	1.36	0.179
Positive	0.5576993	0.15667	2.08	0.138
Sadness	23.37994	0.508958	1.83	0.040
Surprise	1.221015	1.129025	0.22	0.820
Trust	1.001811	0.4267692	0.00	0.992
Nouns	0.9666424	0.0534406	0.61	0.534
Verbs	0.9608001	0.084889	0.45	0.657
Adjectives	1.022347	0.0893491	0.25	0.803
Reply status	1.728938	1.59683	0.59	0.556
Quote status	0.5461027	0.3318688	1.00	0.327
Sensitive	Omitted			
Reply count	0.9810209	0.0109438	1.72	0.081
Time posted	0 .9704837	0.0123272	-1.76	0.079
Favourite count	0.9924974	0.0007123	-3.99	0.000
Media	1.250254	0.3099878	0.90	0.360
User mentions	1.047513	0.0955696	0.51	0.616
Hashtags	1.032532	0.0584294	0.57	0.573
URLs	1.042678	0.4129078	0.11	0.918
Len of tweet	1.004348	0.0063576	0.69	0.492
Avg word length	0.994706	0.0828172	0.06	0.942
Repeated words	0.970842	0.1135083	0.25	0.804
Punctuation	1.086654	0.0599664	1.51	0.138

Significant account factors for malicious tweets

The factors identified in the model as statistically significant were *URL given, number of statuses, followers* and *age of account*. Statuses and Age had p<0.05 and Haz. Ratio of 1.000001 and 1.000203 respectively. A hazard ratio of over one signifies a positive correlation with time to failure and as such a negative correlation with survival. Meaning both factors increased presence in a tweet, negatively affected its chances of survival. It is important to note that both hazard ratios are close to 1 and as such the effect on survival would be less than -1%. Followers was found to have p<0.04 Haz. Ratio of 1, which suggest neither a positive or negative correlation with tweet survival as the number increases or decreases. Interestingly, URL given had p<0.02 and Haz ratio of 2.327596. A hazard ratio this high suggest that in the case that malicious content is posted by an account with a URL provided, it is 132% less likely to survive for 12 hours.

Significant tweet factors for malicious tweets

Of the tweet-based factors, 3 were identified as statistically significant - *anger, sadness* and *favourite count.* Interestingly two of these factors involve the emotion of the tweet. Sadness was found to have p<0.05 and Haz. Ratio=23.37994. This high a hazard ratio suggest sadness is a very significant emotion that negatively effects the chance of a tweet surviving for more than 12 hours. On the other hand, we found the factor anger to have p<0.04 and Haz. Ratio=0. 4704279 which shows that tweets with a higher anger score were 53% more likely to survive past 12 hours. The contrast in these emotions and their effect on a tweet's survival may be directly linked to their malicious nature. Finally, the number of favourites a tweet received was found to have p<0.00 and Haz. Ratio=0.9924974. This hazard ratio suggests that a higher number of favourites decreases a tweets chance of survival.

Table 8. Results of survival model for benign tweets

Predictor	Haz. Ratio	Std. Err	Z	Sig.
Account Factors				
Location	1.014324	0.0558061	0.26	0.796
URL	0.9149823	0.0462831	1.76	0.079
Description	1.125085	0.1323746	1.00	0.316
Default profile	1.010136	0.0562711	0.18	0.856
Verified	1.202803	0.037303	2.33	0.020
Followers	1	0.000000140	1.63	0.102
Friends	0.9999974	0.000000583	4.49	0.047
Lists	0.9999955	0.00000278	1.63	0.103
Favourites	0.999999	0.000000317	3.26	0.101
Statuses	1	0.000000117	2.38	0.067
Age	1.000058	0.000019	3.06	0.010
Tweet Factors				
Anger	0.9332348	0.197353	0.33	0.744
Anticipation	omitted			
Disgust	0.717154	0.2567685	0.93	0.353
Fear	0.866584	0.1600812	0.78	0.438
Joy	0.4316722	0.1291415	2.81	0.105
Negative	1.190292	0.1382033	1.50	0.134
Positive	1.010556	0.00660526	0.16	0.872
Sadness	0.9445512	0.2175737	0.25	0.104

Surprise	1.6249	0.2435687	0.18	0.010
Trust	0.9936557	0.1188461	0.05	0.958
Nouns	0.9535638	0.0109427	4.14	0.073
Verbs	0.9505828	0.0198208	2.43	0.095
Adjectives	1.008582	0.0206568	0.42	0.677
Reply status	1.138556	0.1039217	1.42	0.155
Quote status	1.652964	0.0407518	6.33	0.000
Sensitive	0.8962523	0.1729841	0.57	0.570
Reply count	0.9618724	0.0033769	-5.6	0.000
Time posted	1.003558	0.0027706	1.29	0.198
Favourite count	0.9974237	0.0001244	5.30	0.190
Media	0.9834252	0.0579866	0.28	0.777
User mentions	1.077904	0.0316348	2.56	0.011
Hashtags	0.9444362	0.0152907	-2.41	0.016
URLs	1.1537	0.1318273	1.25	0.211
Len of tweet	1.002714	0.0014935	1.82	0.069
Avg word length	0.9915446	0.0135483	0.62	0.534
Repeated words	0.9984922	0.0269117	0.06	0.955
Punctuation	1.01223	0.0115127	1.07	0.285

Significant account factors for benign tweets

For the benign dataset, 4 account factors were found to be statistically significant in their effect on tweet survival, these were *verified, number of friends, favourites and age of account*. Verified was found to have p<0.02 and Haz. Ratio=1.202803, suggesting that benign tweets posted by verified accounts were 20% less likely to survive past the 12-hour threshold. Number of friends (p p<0.00, Haz. Ratio=0.9999974) and favourited posts (p<0.01 and Haz. Ratio=0.999999) also correlated positively with tweet survival by approximately 1%. The only factor found to negatively affect tweet survival in benign tweets was the age of the account with p<0.01 and Haz. Ratio=1.000058, a hazard ratio of less than 1.01 suggests that the negative effect increasing age of account has on survival is minimal.

Significant tweet factors for benign tweets

In the benign dataset, a multitude of tweet-based factors were statistically significant in regard to tweet survival. These were *Surprise, quote status, reply count, mentions* and *hashtags.* From these, user mentions (p<0.02 and Haz. Ratio=1. 077904) negatively affected a tweets survival as it increased, the hazard ratio suggests for every user mention found in a tweet, the survival chance is decreased by 7%. The only emotion with statistical significance was surprise with p<0.01 and Haz. Ratio=1.6249. Suggesting that the detection of surprise in a tweet decreased a tweets chance at survival by 62%. The final negative factor against survival was quote status with p<0.0 and Haz. Ratio=1.652964 suggesting quote tweets were 65% less likely to survive for 12 hours. Reply count (p<0.00 and Haz. Ratio=0.9618724) and Hashtags (p<0.00 and Haz. Ratio=0.9993403) were found to positively affect a tweets likelihood of survival.

6.2 Evaluation

From the results obtained, we are able to see a multitude of account and tweet-based factors that correlate to tweet size (re-tweetability) and survival (the likelihood a tweet is being actively retweeted for more than 12 hours). Several of the findings are supported by previous research while others reveal new factors affecting tweet propagation. In order to answer the focus of the project – understanding information flow and survival of malicious posts around Covid-19 – it is important to look closely at how the factors identified in each model differ between benign and malicious posts. These differences will allow conclusions to be drawn on the factors that are specific to malware propagation and those that are related to general tweet propagation.

6.2.1 Evaluation of tweet size results

For the malicious dataset, four account-based factors were identified to have a correlation with tweet size. However, all of these relationships had less then 1% effect on tweet size, which suggest that in the case of malicious content, the likelihood of a tweet engaging users and receiving retweets is not highly dependent on the profile of the posting account. This is supported by previous literature that found the likelihood of a user sharing malicious content is not influenced by authoritativeness of the source but more dependent on any pre-existing attitude the user held regarding the content [39]. As such this could be why there is little significance in the nature of the posting account. This theory is supported when we consider the fact that out of those 4 significant account-based factors, two of them (statuses and favourites) had the same effect on tweet size in the benign data set. Suggesting they may be factors common to retweetability in general rather than the retweetability of malicious content. Furthermore, in the benign dataset, factors of a customised profile (a non-default account with a URL provided) positively correlated to tweet size, with both factors found to increase tweet size by approximately 46%. Unfortunately, these factors were not statistically significant in the malicious size data model so solid conclusion on how these factors affect malicious tweet size cannot be drawn. Both the benign and malicious size models found that number of followers had no effect on tweet size which suggests that virality is dependent on tweet content rather the number of followers held by the account. Again, this is supported by previous literature that found the content of tweet is much more deterministic in the number of retweets a tweet might receive than relationships held by the posting account. [40] From the evidence above we can see malware retweetability tends not to rely heavily on account-based features.

From the tweet-based factors, for both malicious and benign we found that tweets containing extra features such as hashtags, media and user mentions, were less likely to be retweeted. The effect of the presence of these factors in malicious tweets was however greater, with media causing a 39% decrease in malicious contents retweetability as opposed to 15% in benign. This disparity suggests that the retweetability of malware is not reliant on a surpass of extra features that can be added to a tweet. This does not align with previous research that found tweets containing an image, are 35% more likely to attain higher retweets. [41] From this I can see that it may perhaps have been a better hypothesis to test the effect of retweetability in the presence and absence of extra features, rather than how the specific count of each feature affects retweetability. For example, the skew in results may be caused by the fact that tweets with 10 images are less retweeted due to the fact users are less likely to be engaged by the surpass of content.

A factor that was found to have a significant effect on retweetability in both malicious and benign tweets were the emotions conveyed in the tweets text. This aligns with previous literature that found emotion to be a crucial part of retweeting behaviour. [42] In particular, tweets containing high levels of disgust received substantially more retweets than those without. In benign tweets, disgust increased retweetability by 279%, whereas in malicious tweets it increased retweetability by 5248%. This shows that the emotion is extremely significant in engaging users. Previous research supports this finding in stating that disgust originates from fear [43] and that fear is a known common driving factor in causing users to share and engage with content – especially malicious content. [44] It seems that those who want to spread malware rely heavily on the human instinct of revulsion/disgust to provoke a strong feeling and moral reaction to some content in order to drive them to share content, and as seen in the results this is highly effective. The second emotion found to effect retweetability was surprise, which increased retweets by 317%. A very significant effect. Surprise is a second emotion that is strong and can cause a reader an immediate reaction, this automatic response causes the user to act impulsively and share the content with others in their social environment. Thus, the emotions go hand in hand in driving people to share and retweet content. Given the fact the data collected is focussed around Covid-19, it is easy to see how disgust and surprise could be incorporated into tweets as new and unheard-of situations arise every day. To look closely into the data to see if this is reflected word clouds of the common language used in each dataset were created and can be seen in Figure 17 and 18 below. The content of which further supports the conclusion that hackers are able to hide their malware in disturbing and shocking headlines that cause users to interact and share their posts. In particular, the benign dataset tends to contain more covid neutral or positive expressions such as ‘help’, ‘relief’, ‘new’, ‘plan’ and ‘tests’ which in the context of Covid appear to be supportive and structured descriptive languages. Whereas the malicious dataset contained words that invoked more negative feelings such as ‘death’, ‘pandemic’, ‘restrictions’ , ‘vaccinations’, ‘stay home’ and ‘variant’. All of which reflect a disgustful and surprise-based tone, which is being used to ensure a reaction and engagement from the reader.

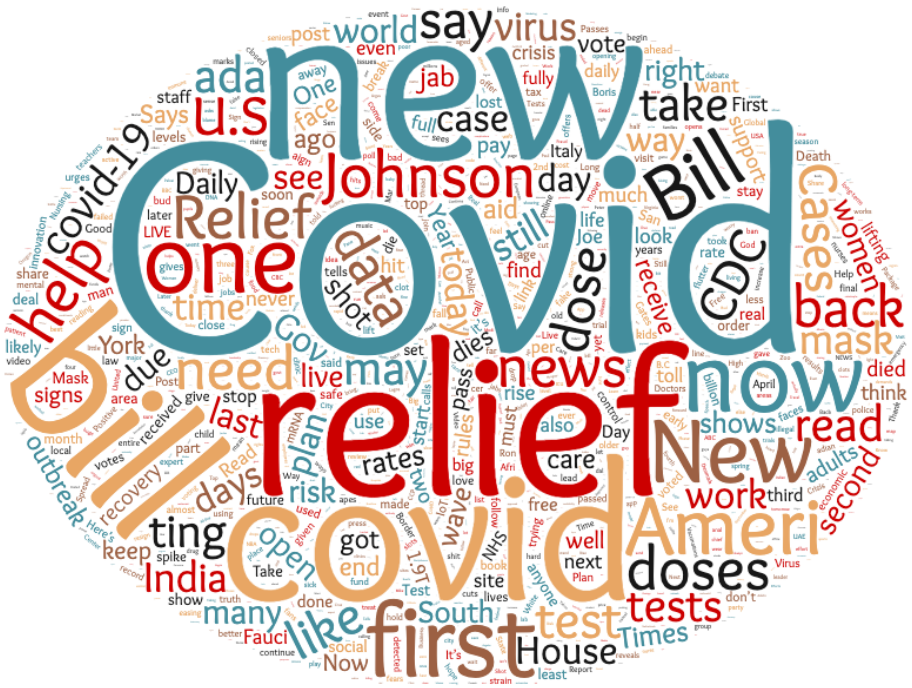

Figure 17. Word cloud of language used in the benign dataset

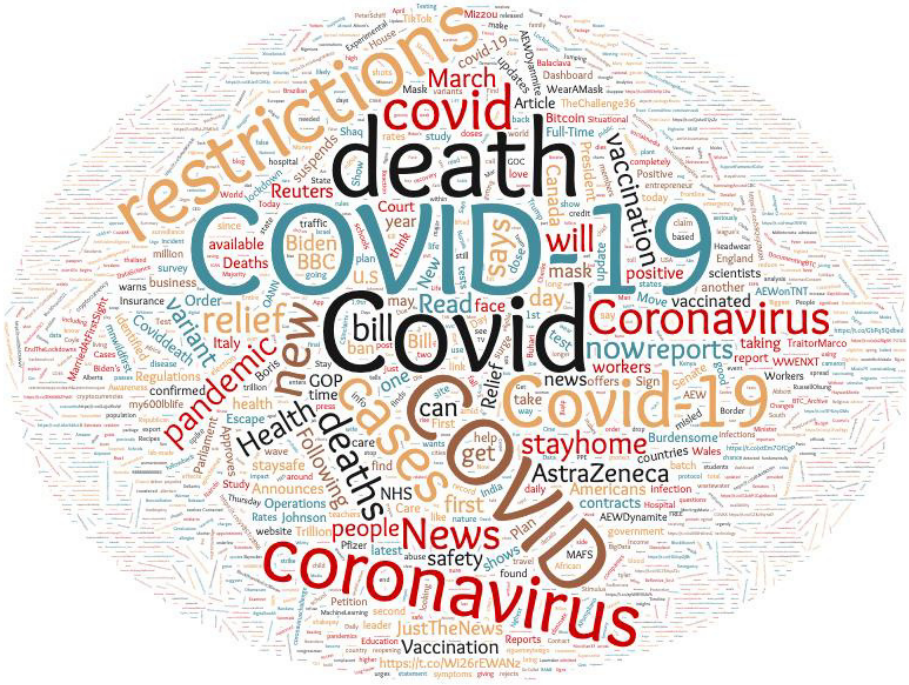

Figure 18. Word cloud of language used in the malicious dataset

6.2.2 Evaluation of tweet survival results

From the account-based analysis of tweet survival a number of significant findings were obtained. Firstly, in the malicious model, the URL given factor was found to decrease a tweets chances of survival by over 132%, this great an effect supports our previous conclusion that cyber-criminals are not reliant on features of the posting account to propagate malware. Especially where survival is considered as here it has a drastically negative effect. Unfortunately, this factor was not significant in the benign survival model so a straight comparison cannot be drawn. It was, however, significant in the benign size model and was found to increase retweetability by 48%. This suggest further that it is a factor more favoured to aid propagation of non-malicious content. In both the benign and malicious survival models, the age of account was found to decrease the likelihood a tweet would survive for 12 hours. This may be something favourable by cyber-criminals as Twitter has a policy of suspending accounts that are in breach of the twitter rules. [45] Due to the nature of the account sharing malicious content, it is likely they will be reported at some point and have their account suspended. At which point they would create a new account. We also found that the number of followers an account holds, has no direct effect on tweet size or survival in the case of benign of malicious tweets. This is upheld by a previous study that found out of a multitude of factors, follower count has less than expected impact on retweets. [46]

Of the tweet factors analysed, favourite count was found to increase a tweets likelihood of survival, this may be caused by the fact that people view the number of favourites a tweet receives as an indication of its value to others, be that in reliability/trustworthiness or as

containing interesting content. Either way, it is understandable that this would cause more engagement with a tweet over time, increasing the lifetime of the tweet. When we look at tweet-based factors of the benign dataset, we found that the presence of hashtags increased tweet survival. Previous research found that this may be related to the fact that tagging a tweet with some topic, ensures the tweet is shared with an audience interested in the topic, which would increase the longevity of a tweet, attracting a larger audience than a follow base alone. Relying on this external source for retweets rather than friends/followers is backed up by the finding that user mentions were found to decrease the likelihood of tweet survival by 7%. This could be because where user mentions tend to occur, the tweet is driven towards a particular user or group of users and not a wider audience of engagement. When we consider the fact that the tweets collected all focus around the topic of covid, it could be considered that tweets with user mentions may be more personal around the subject, minimising the audience and likelihood of prolonged active retweeting.

From the investigation into the effect of emotion/sentiment on tweet survival we found that for malicious posts, anger increased the chance of tweet survival by 53% and that sadness decreased the chance of survival by over 100%. The effect anger has on tweet survival supports the earlier conclusion that certain emotions can be used to make a person feel strongly, which drives them to act impulsively or feel the need to share the content with others. Causing the tweet to be shared for a longer period. As mentioned, prior research into the emotions that drive sharing found that though strong negative emotions tend to cause user engagement, this does not necessarily apply for sadness, in fact research concluded that the presence of sadness has a negative effect on a user's likelihood to share. [47] Interestingly, in the benign dataset, the emotion surprise was found to negatively affect tweet survival, by approximately 62%. Linking back to the malicious size model results, we found that surprise correlated positively with the retweetability of malicious content and as such we can see the emotion appears to have adverse effects on benign and malicious content. Which would suggest that surprise is an emotion that cyber-criminals are aware will propagate their content.

7. Conclusions and Future work

7.1 Conclusion

The main goal of this project was to understand and predict information flow of malicious content around Covid-19. To do so, a dataset of tweets was collected, both benign and malicious, from which a multitude of account and tweet-based factors were identified. These factors were then statistically analysed to identify any correlation they may have regarding the size and survival of tweets. Based on the results, we were able to conclude that retweeting of benign content is highly influenced by the characteristics of the posting account, as opposed to having very little effect on engagement with malicious content. Thus, showing that propagation of malware is more reliant on content-based features such as the emotion of the tweet. In particular, strong negative emotions such as anger and disgust were prevalent factors associated with successful propagation. Unfortunately, due to the time constraints of the project, the dataset for the malicious tweets was substantially smaller than I would have hoped, with a larger dataset there is the possibility more conclusions could be drawn. Nonetheless, with further work the findings of this experiment could be used to train machine learning models for detection of malicious content and incorporated into various social platforms to prevent and detect malicious content before it is spread.

7.2 Future work

The findings of this project were interesting but given more time there are a variety of factors I would have delved further into. One being the categorisation of malicious tweets. The project relied on VirusTotal to analyse tweet URLs, with a threshold of one threat identified to classify a tweet as malicious. Unfortunately, this does not take into consideration the possibility of false positives meaning there is chance that some tweets may have been misclassified which would affect the statistical results. In the future, I would consider using multiple services to classify URLs ensuring confidence in the classification of tweets.

A second factor I am aware of is the limitation of validity in small datasets. To overcome this, I would have extended the data collection period and analysed a larger quantity of tweets in the hopes of building a larger dataset of malicious tweets. This is because the number of malicious tweets (X=278) was substantially smaller than the benign data set (X=6,628) and smaller datasets can lead to lower statistical power and reliability of results. By building the malicious statistical models from a larger dataset I could have eliminated the chance that any findings were the result of randomness. With a larger malicious dataset, I may also have set the threshold for minimum number of retweets to be higher in the size data model. Doing so may have changed the results and indicated factors that show more substantial correlations in tweets with a higher retweet rate.

8. Reflection of learning

Through completing this project, I have widely expanded my skillset and overall understanding of autonomous working. Having full responsibility for the direction and focus of the project seemed daunting at first but thorough research in the early stages and a clear map of project milestones allowed me to stay on track throughout. In the early stages of the project, I had limited experience using APIs but since then have learnt how to work efficiently with them by making use of the documentation to guide my learning. I also learnt how to use statistical modelling tools and interpret both count and survival data models. This was an interesting growth of my knowledge in the area, having not studied statistics in depth since A-level maths and was a stage of the project I found particularly enjoyable.

Besides, development of my technical skills, the project also allowed me to advance a variety of soft skills, particularly responsibility and time management in overseeing the project and exercising self-discipline to ensure I reached the weekly goals set out by myself and supervisor. These were skill's that developed as the project progressed as in the early stages I fell behind the deadline for completion of data collection and had to evaluate my handling of the project.

Admittedly, there were challenges along the way. The one most impactful was the daily rate limit of scanning tweets using VirusTotal. My initial plan to get around this was to set up multiple accounts and scan tweets in parallel. Unfortunately, VirusTotal detected this and blocked my IP address which acted as a major bottleneck for the project as I fell behind on processing tweets. Fortunately, making use of a variety of VPN's allowed me to overcome this issue.

Overall, I found managing the project to be a rewarding but complex task and feel it has prepared me to handle a variety of challenges, which is a crucial life skill as I conclude my degree and begin my career.

9. Appendices

9.1 Appendix 1: Figure A – test collection script

test_collection.py > ...

```python
import json

input_file = "tweets_12.json"

with open(input_file) as f:
    for line in f:
        tweet = json.loads(line)
        # check entire tweet contains at least one occurence of the word 'covid' or 'coronavirus' and contains a URL
        string_object = str(tweet).lower()
        if ('covid' not in string_object and 'coronavirus' not in string_object) or len(tweet['entities']['urls'][0]['expanded_url']) == 0:
            raise ValueError('Tweet contains invalid data')

```

9.2 Appendix 2: Figure B – test sampling script

test_sampling.py > ...

```python

input_file = "sample-12th.json"

collected_urls = []
with open(input_file) as f:
    for line in f:
        tweet = json.loads(line)
        # check if the url has already been seen in the sample, throw error if so
        url = tweet['entities']['urls'][0]['expanded_url']
        if url in collected_urls:
            raise ValueError('Duplicate URL detected')
        collected_urls.append(url)

```

10. References

[1] Oberlo.co.uk. 2021. *How Many People Use Social Media in 2021 [Updated Jan 2021]*. [online] Available at: <https://www.oberlo.co.uk/statistics/how-many-people-use-social-media#:~:text=The%20latest%20figures%20show%20that,jump%20in%20just%20five%20years.> [Accessed 27 May 2021].

[2] TitanHQ. 2021. *Social Media Platforms Double as Major Malware Distribution Centres*. [online] Available at: <https://www.titanhq.com/blog/social-media-platforms-double-as-major-malware-distribution-centres/> [Accessed 27 May 2021].

[3] www.kaspersky.com. 2021. *What Is a Drive by Download*. [online] Available at: <https://www.kaspersky.com/resource-center/definitions/drive-by-download> [Accessed 27 May 2021].

[4] Irwin, L., 2021. *2020 cyber security statistics - IT Governance UK Blog*. [online] IT Governance UK Blog. Available at: <https://www.itgovernance.co.uk/blog/2020-cyber-security-statistics> [Accessed 27 May 2021].

[5] Sowriraghavan, A. and Burnap, P., 2015. Prediction of Malware Propagation and Links within Communities in Social Media Based Events. *Proceedings of the ACM Web Science Conference*,.

[6] Cao, C. and Caverlee, J., 2015. Detecting Spam URLs in Social Media via Behavioral Analysis. *Lecture Notes in Computer Science*, pp.703-714.

[7] Güngör, K. and Erdem, A., 2021. Tweet and Account Based Spam Detection on Twitter. *Artificial Intelligence and Applied Mathematics in Engineering Problems (pp.898-905)*, [online] Available at: <https://www.researchgate.net/publication/338359769_Tweet_and_Account_Based_Spam_Detection_on_Twitter> [Accessed 27 May 2021].

[8] Javed, A., Burnap, P., Williams, M. and Rana, O., 2020. Emotions Behind Drive-by Download Propagation on Twitter. *ACM Transactions on the Web*, 14(4), pp.1-26.

[9] Yang, Z., Qiao, C., Kan, W. and Qiu, J., 2019. Phishing Email Detection Based on Hybrid Features. *IOP Conference Series: Earth and Environmental Science*, 252, p.042051.

[10] Graham, A., 2021. *The 5 most common cyber attacks in 2020 - IT Governance UK Blog*. [online] IT Governance UK Blog. Available at: <https://www.itgovernance.co.uk/blog/different-types-of-cyber-attacks> [Accessed 27 May 2021].

[11] Zetter, K., 2021. *Trick or Tweet? Malware Abundant in Twitter URLs*. [online] Wired. Available at: <https://www.wired.com/2009/10/twitter-malware/> [Accessed 27 May 2021].

[12] Internet live stats. 2021. *Twitter Usage Statistics*. [online] Available at: <https://www.internetlivestats.com/twitter-statistics/> [Accessed 27 May 2021].

[13] Laing, B., 2021. *Drive-By Downloads and How to Prevent Them*. [online] Lastline. Available at: <https://www.lastline.com/blog/drive-by-download/> [Accessed 27 May 2021].

[14] GitHub. 2021. *tweepy/tweepy*. [online] Available at: <https://github.com/tweepy/tweepy> [Accessed 27 May 2021].

[15] Ritetag.com. 2021. *55 coronavirus hashtags popular on Twitter and Instagram | RiteTag: Find the best hashtags*. [online] Available at: <https://ritetag.com/best-hashtags-for/coronavirus> [Accessed 27 May 2021].

[16] Trend Micro. 2020. *Developing Story: COVID-19 Used in Malicious Campaigns*. [online] Available at: <https://www.trendmicro.com/vinfo/us/security/news/cybercrime-and-digital-threats/coronavirus-used-in-spam-malware-file-names-and-malicious-domains> [Accessed 27 May 2021].

[17] Gov.scot. 2021. *Coronavirus (COVID-19) update: Health Secretary's statement - 3 March 2021 - gov.scot*. [online] Available at: <https://www.gov.scot/publications/coronavirus-covid-19-update-health-secretarys-statement-3-march-2021/> [Accessed 27 May 2021].

[18] VirusTotal. 2021. *How it works*. [online] Available at: <https://support.virustotal.com/hc/en-us/articles/115002126889-How-it-works> [Accessed 27 May 2021].

[19] Faculty.nps.edu. 2021. *Measuring the Effectiveness of Honeypot Counter-Counterdeception*. [online] Available at: <https://faculty.nps.edu/ncrowe/honeypot_hcss05.htm> [Accessed 27 May 2021].

[20] AV-Comparatives. 2021. *Free public sources of malicious URLs*. [online] Available at: <https://www.av-comparatives.org/freepublic-malicious-url-sources/> [Accessed 27 May 2021].

[21] Transparencyreport.google.com. 2021. *Google Transparency Report*. [online] Available at: <https://transparencyreport.google.com/safe-browsing/search?hl=en_GB> [Accessed 27 May 2021].

[22] Mcleod, S., 2021. *Independent and Dependent Variables | Definitions & Examples | Simply Psychology | Simply Psychology*. [online] Simplypsychology.org. Available at: <https://www.simplypsychology.org/variables.html> [Accessed 27 May 2021].

[23] Güngör, K. and Erdem, A., 2021. Tweet and Account Based Spam Detection on Twitter. *Artificial Intelligence and Applied Mathematics in Engineering Problems (pp.898-905)*, [online] Available at: <https://www.researchgate.net/publication/338359769_Tweet_and_Account_Based_Spam_Detection_on_Twitter> [Accessed 27 May 2021].

[24] Ezpeleta, E. and Mendizabal, I., 2020. Novel email spam detection method using sentiment analysis and personality recognition. *Logic Journal of the IGPL, Volume 28, Issue 1, February 2020, Pages 83–94*, [online] Available at: <https://academic.oup.com/jigpal/article/28/1/83/5680435> [Accessed 27 May 2021].

[25] Paris, H. and Alqatawna, J., 2017. Spam profile detection in social networks based on public features. *International Conference on Information and Communication Systems (ICICS)*, [online] Available at: <https://ieeexplore.ieee.org/document/7921959> [Accessed 27 May 2021].

[26] Patterson, M., Hou, Z., Hrach, A., Hrach, A. and Griffiths, J., 2021. *How to Double Your Social Engagement With Images*. [online] Content Marketing Consulting and Social Media Strategy. Available at: <https://www.convinceandconvert.com/social-media-strategy/double-social-engagement-with-images/> [Accessed 27 May 2021].

[27] PyPI. 2021. *NRCLex*. [online] Available at: <https://pypi.org/project/NRCLex> [Accessed 27 May 2021].

[28] Saifmohammad.com. 2021. *NRC Emotion Lexicon*. [online] Available at: <https://saifmohammad.com/WebPages/NRC-Emotion-Lexicon.htm> [Accessed 27 May 2021].

[29] Nltk.org. 2021. *Natural Language Toolkit — NLTK 3.6.2 documentation*. [online] Available at: <https://www.nltk.org/> [Accessed 27 May 2021].

[30] Javed, A., Burnap, P., Williams, M. and Rana, O., 2020. Emotions Behind Drive-by Download Propagation on Twitter. *ACM Transactions on the Web*, [online] 14(4), pp.1-26. Available at: <http://orca.cf.ac.uk/132117/1/post-print%20Emotions%20behind%20drive-by%20download.pdf>.

[31] Developer.twitter.com. 2021. *User object*. [online] Available at: <https://developer.twitter.com/en/docs/twitter-api/v1/data-dictionary/object-model/user#:~:text=The%20User%20object%20contains%20Twitter,can%20be%20grouped%20into%20lists.> [Accessed 27 May 2021].

[32] Packages, O., Power, S., Output, and US, A., 2021. *IDRE Stats – Statistical Consulting Web Resources*. [online] Stats.idre.ucla.edu. Available at: <https://stats.idre.ucla.edu/> [Accessed 27 May 2021].

[33] Brandwatch. 2021. *60 Incredible and Interesting Twitter Stats and Statistics*. [online] Available at: <https://www.brandwatch.com/blog/twitter-stats-and-statistics/#:~:text=As%20of%20Q1%202019%2C%2068m,the%20average%20number%20of%20followers.> [Accessed 27 May 2021].

[34] Slideplayer.com. 2021. *Discrete Random Variables - ppt video online download*. [online] Available at: <https://slideplayer.com/slide/3953669/> [Accessed 27 May 2021].

[35] En.wikipedia.org. 2021. *Probability mass function - Wikipedia*. [online] Available at: <https://en.wikipedia.org/wiki/Probability_mass_function> [Accessed 27 May 2021].

[36] En.wikipedia.org. 2021. *Survival analysis - Wikipedia*. [online] Available at: <https://en.wikipedia.org/wiki/Survival_analysis> [Accessed 27 May 2021].

[37] Sphweb.bumc.bu.edu. 2021. *Cox Proportional Hazards Regression Analysis*. [online] Available at: <https://sphweb.bumc.bu.edu/otlt/MPH-Modules/BS/BS704_Survival/BS704_Survival6.html#:~:text=In%20a%20Cox%20proportional%20hazards,up%20to%20a%20specific%20time.&text=As%20a%20result%2C%20the%20hazard%20in%20a%20group%20can%20exceed%201.> [Accessed 27 May 2021].

[38] Sthda.com. 2021. *Cox Proportional-Hazards Model - Easy Guides - Wiki - STHDA*. [online] Available at: <http://www.sthda.com/english/wiki/cox-proportional-hazards-model> [Accessed 27 May 2021].

[39] Buchanan, T., 2020. Why do people spread false information online? The effects of message and viewer characteristics on self-reported likelihood of sharing social media disinformation. *PLOS ONE*, 15(10), p.e0239666.

[40] Zarrella, D., 2021. *The Science of ReTweets*. [online] Mashable. Available at: <https://mashable.com/2009/02/17/twitter-retweets/?europe=true#:~:text=The%20most%20obvious%20factor%20that,a%20less%20than%20expected%20impact.> [Accessed 27 May 2021].

[41] Blog.twitter.com. 2021. *What fuels a Tweet's engagement?*. [online] Available at: <https://blog.twitter.com/en_us/a/2014/what-fuels-a-tweets-engagement.html> [Accessed 27 May 2021].

[42] Chen, J., Liu, Y. and Zou, M., 2017. User emotion for modeling retweeting behaviors. *Neural Networks*, 96, pp.11-21.

[43] Psychology Today. 2021. *Fear + Disgust = Entomological Horror*. [online] Available at: <https://www.psychologytoday.com/us/blog/the-infested-mind/201512/fear-disgust-entomological-horror> [Accessed 27 May 2021].

[44] Sussman, B., 2021. *5 Emotions Used in Social Engineering Attacks [with Examples]*. [online] Secureworldexpo.com. Available at: <https://www.secureworldexpo.com/industry-news/5-emotions-hackers-use-social-engineering-attacks> [Accessed 27 May 2021].

[45] Help.twitter.com. 2021. *Why might your Twitter account be suspended and how to unsuspend*. [online] Available at: <https://help.twitter.com/en/managing-your-account/suspended-twitter-accounts> [Accessed 27 May 2021].

[46] Zarrella, D., 2021. *The Science of ReTweets*. [online] Mashable. Available at: <https://mashable.com/2009/02/17/twitter-retweets/?europe=true#:~:text=The%20most%20obvious%20factor%20that,a%20less%20than%20expected%20impact.> [Accessed 27 May 2021].

[47] Business.twitter.com. 2021. *The psychology of shareable content*. [online] Available at: <https://business.twitter.com/en/blog/psychology-of-shareable-content.html#:~:text=When%20readers%20experienced%20strong%20positive,emotions%2C%20like%20anxiety%20or%20arousal.> [Accessed 27 May 2021].