Comparing the Effectiveness of Machine Learning Algorithms in Classifying Google Alerts about Distributed Denial of Service

Author: André Khreiche
University of Twente
P.O. Box 217, 7500AE Enschede
The Netherlands

ABSTRACT,
Distributed denial of service (DDoS) attacks are attempts to make computer or network resources unavailable to its intended users. They cause firms and other organizations significant economic and reputational harm and have risen in frequency and strength over the course of the past years. In order to contribute to the understanding of DDoS attacks, this study explores machine learning as a tool to classify Google alerts about DDoS. I try to answer which machine learning algorithms can improve and simplify the process of retrieving news reporting a DDoS event. Several machine learning algorithms are tested on a dataset and compared in terms of effectiveness. I find the multinomial Naive Bayes algorithm with the bag of words model to be the most effective out of the ones I tested. Furthermore, I explore some applications for the Word2vec algorithm to provide information about semantic features of DDoS related Google alerts.

Graduation Committee members:
First supervisor: Prof. dr. M. Junger
Second supervisor: Abhishta

Keywords
Cybercrime, DDoS, Distributed Denial-of-Service, Machine Learning, Classification, Supervised Learning,

11th IBA Bachelor Thesis Conference, July 10th, 2018, Enschede, The Netherlands.

1. INTRODUCTION

1.1 Distributed Denial of Service

A Distributed Denial of Service (DDoS) attack has been defined as a malicious attempt from multiple systems to make computer or network resources unavailable to its intended users, usually by interrupting or suspending services connected to the internet. Attacks can cause websites or internet services to respond very slowly or even crash them completely, for the duration of the attack. DDoS attacks pose a threat to a wide range of internet-based services. Some of these can include e-commerce, banking, server hosting or online gaming, among many others. But other areas can be susceptible to these kinds of cyber-attacks as well, for instance, medicine, transportation, media or education (Wueest, 2014). In recent years DDoS attacks have seen an increase in strength, frequency and sophistication (Behal & Kumar, 2017; Jonker et al., 2017). This increase is the result of a higher availability of tools to perform such attacks. Nowadays, performing a DDoS attack does not require sophisticated knowledge of computer networks. Instead, one can just get a provider of DDoS-as-a-Service to attack any website (Santanna & Sperotto, 2014). Even though this kind of service is illegal, it is, in principal, not difficult to obtain. Furthermore, the price for this kind of service is quite low. Santanna & Sperotto mention a price of 5 US Dollars for 25Gbps of DDoS traffic, which is enough to do some damage to most hosts and services on the internet.

Several studies have examined the economic impacts which DDoS attacks have on firms (Anderson et al., 2013). Anderson et al. have provided a framework for analyzing cybercrime. They differentiate between direct losses, indirect losses and defense costs. Direct losses refer to things like paid ransom, financial damage caused by downtime, costs of reimbursement or lost customer traffic. Indirect losses can be reputational damage or loss of trust, which can for instance result in weakened stock prices. Defense costs entail various types of security products and services. Other authors have taken the vantage point of the attacker (Segura & Lahuerta, 2010) and concluded economic motives to be one of the main pushing forces for attacks. In a study conducted by Abhishta et al, the impact of negative news associated with DDoS attacks on firms' stock prices has been examined. They found that, in the cases they investigated, DDoS attacks – and news about those attacks – are often followed by a period of weakened stock prices. (Abhishta, Joosten, & Nieuwenhuis, 2017)

1.2 News about DDoS

When searching for DDoS related news on the internet, the results are plentiful. They range from reports about specific DDoS incidents over blog posts to general reports on statistics about the phenomenon of DDoS. Examining such news can help improve our understanding of several factors surrounding DDoS attacks, including things such as nature and timing of attacks, ties to specific political, cultural or economic events or who preferred targets of DDoS attacks are. Many DDoS incidents receive some media attention and reports about them can be found in various online outlets. Such reports bring about implications for victimized individuals or organizations, which go beyond the direct, technical or financial consequences. Goth describes them as "soft" elements surrounding DDoS attacks (Goth, 2007). In his study, he describes how DDoS attacks on the Estonian government added fire to a political conflict between Estonia and Russia. In this context, news about the attacks has played a crucial role. In the same way, impacts of news about DDoS attacks on firms can be examined. The aforementioned study by Abhishta et al. is one example for this. Furthermore, there are several previous studies which have examined impacts of internet security breaches on stock prices of firms (Campbell, Gordon, Loeb, & Zhou, 2003; Hovav & D'Arcy, 2003).

1.3 Research project motivation

As previous work on DDoS and other kinds of internet security breaches have shown, there can be value in examining DDoS related incidents. It can help shed light on the phenomenon of DDoS, or other kinds of attacks. Information about past attacks can be presented in a way, which is informative and rather easy to understand, for instance, by means of visual illustration. However, in order to provide meaningful insights, a certain amount of data is required. Analyzing and classifying news manually requires a lot of labor power and qualitative analysis of text data (news, reports, and blogs).

The occurrence of DDoS attacks is of interest for people in many different scientific and practical areas. These range from concerns related to cyber security in sectors such as government, finance, server hosting or online gaming to analytical activities, such as data mining or data science, to name just a few.

The key motivation for this study is to contribute to the process of efficiently retrieving data about DDoS attacks and processing the data, in order to provide some insight into past attacks. Automating this process to some extent, could help mitigate or solve this problem. Machine learning could be a helpful approach to simplify the procedure and perhaps enable more data to be processed more quickly and cheaply. Furthermore, I attempt to contribute to a better understanding of long or mid-term reputational damage to an organization, caused by DDoS attacks.

1.4 Google Alerts

For people and organizations who are interested in patterns and factors surrounding DDoS incidents, Google can be a valuable resource. It can provide a

lot of data centered around DDoS attacks. In this research the "Google alerts" service is used, in order to collect data related to the topic of DDoS. I will explain in some detail the functionality of the Google alerts service and how it is used for this study.

Google alerts is a content change detection and notification service. Users of the service can select a search term or a combination of terms and whenever the service finds new results, which match these terms, the user is notified via email. For this research, the search terms are "Distributed Denial of Service" and "DDoS".

However, in order to filter out those alerts that contain information regarding actual incidents of DDoS attacks, one has to sift through a large amount of articles and news. An approach to make this process less labor intensive is by utilizing machine learning.

1.5 Machine Learning

A common definition of machine learning is as follows:

> "A computer program is said to learn from experience E with respect to some class of task T and performance measure P, if its performance at task in T, as measured by P, improves with experience E." (Mitchell, 1997)

Which, more simply put, means that if a computer program can improve how it performs at a certain task, based on past experience, then it has learned. In our example the task T would be classifying DDoS related Google alerts, performance P would be things like accuracy with which alerts are classified correctly and experience E would be a database of Google alerts. It is also important to note that the computers or programs are not being explicitly programmed to perform the task or solve the problem.

There are several key benefits, which make machine learning a powerful tool in text categorization. Compared to the manual classification of texts, machine learning yields very good effectiveness, considerable savings in terms of expert labor power and straightforward portability to different domains (Sebastiani, 2002). Sebastiani defines text categorization as the activity of labeling natural language texts with thematic categories from a predefined set.

1.6 Research project objective

Ultimately, the objective of this research project is to improve the **process of retrieving** and assessing data, and to give **insight into DDoS attacks** and some of the factors associated with them. The intention is to explore how machine learning can be utilized to that end. This entails finding suitable algorithms to produce accurate results in classifying DDoS related alerts, which appear on Google. The classifications are along the lines of whether an alert reports on a DDoS attack, which has occurred, or not. Several algorithms, which are commonly used in the field of machine learning, specifically supervised machine learning for classification tasks, are examined. Thus, some insight into which algorithms are suitable for this task are provided.

2. RESEARCH QUESTION(S)

The research question I would like to answer pertains to how machine learning could contribute to automating and simplifying the process of retrieving data about DDoS incidents and what insights about DDoS attacks can be gained. More specifically, it is of interest how different machine learning algorithms can be used to that end. For this reason, I came up with the research question:

Q1: What is the efficiency of the present machine learning algorithms in separating attack-reporting Google alerts from non-attack-reporting Google alerts?

> Sq1: Which machine learning algorithms can be used for text classification?
>
> Sq2: How effective are these algorithms in separating attack-reporting alerts from non-attack-reporting alerts, in the given dataset?
>
> Sq3: Which algorithm is most effective in determining whether a Google alert reports on a DDoS attack on not?

Q2: What insights about DDoS attacks can be gained from the resulting dataset?

3. THEORETICAL FRAMEWORK

There are several types of machine learning methods. In this research project, a supervised learning method is used. In supervised learning, the aim is to predict a target variable using predictor variables. Furthermore, there is a distinction between regression tasks and classification tasks. In regression tasks, the target variable is continuous, whereas in classification tasks, the target variable consists of categories. (Kotsiantis, S. B., Zaharakis, I., & Pintelas, 2007). This research project will examine a classification task. More specifically the categories (target variables) are 'Being an attack-reporting alert' or 'Being a non-attack-reporting alert'. The features (predictor variables) are related to content of the alerts, in this case this means the most important feature for the analysis is the text of the alerts.

There are several learning algorithms and models, which have been used in supervised machine learning in the past. Some of these are decision trees or support vector machines (Kotsiantis, S. B., Zaharakis, I., & Pintelas, 2007). Another algorithm, which is tested, is the Naive Bayes algorithm. This algorithm has been used for the classification of text-based data in the past. McCallum and Nigam

pointed out that this algorithm has been used for text classification by numerous researchers (McCallum & Nigam, 1998).

3.1 Evaluation Technique

In order to evaluate the results produced by the learning algorithms, several metrics are important. It is not enough to simply look at the rate at which the different algorithms correctly predict whether an alert reports on an attack or not. This would not give much indication about whether the algorithm can actually identify attack-reporting alerts. Sebastiani states that the experimental evaluation of a classifier usually measures its *effectiveness*, rather than its efficiency, that is, its ability to take the *right* classification decisions (Sebastiani, 2002). Efficiency, in this context, would also be related to things such as computational speed. These kinds of performance measures have been studied in some previous studies about classification as well (Williams, Zander, & Armitage, 2006), but will not be elaborated on in this study. For this reason, I will use confusion matrices, which indicate the number of true positives, false positives, false negatives and true negatives. The matrices displayed (Appendix) have this format:

Category		Algorithm Judgment		
		Attack	Non Attack	
Actual Class	Attack	TP	FN	TP+FN
	Non Attack	FP	TN	FP+TN
		TP+FP	FN+TN	n

Table 1: Format of confusion matrices

Here, n is the sum of all alerts, FP (false positives) is the number of alerts incorrectly classified as attack-reporting alerts; TP (true positives), TN (true negatives) and FN (false negatives) are defined accordingly. Thus, information on two additional metrics, rather than just accuracy, can be provided, precision and recall. These metrics have been used by other researchers in past studies on machine learning algorithms for classification problems (Williams et al., 2006). So the three performance metrics I employ are:

- Accuracy: The number of correctly classified alerts divided by total number of alerts.

$$\frac{TP + TN}{n}$$

- Precision: The number of correct positive identifications divided by the total amount of positive identifications.

$$\frac{TP}{TP+FP}$$

- Recall: The number of correct positive identifications divided by the total amount of actual positives

$$\frac{TP}{TP+FN}$$

Moreover, I use k-fold cross validation. In this validation method, the dataset is randomly split into *k* subsets, which are equal in size. Subsequently, each of the *k* subsets is used as the test set and the other *k* subsets form the training set. The aforementioned performance metrics are then calculated across all *k* trials.

4. METHODOLOGY

In this study, a classification model for DDoS related alerts on Google is provided, differentiating between alerts, which report on a DDoS attack, and those, which do not. Different algorithms are tested, in order to find out which ones work well in predicting the correct classification.

4.1 Data Collection

The data used in this study has been gathered using the "Google alerts" content change detection and notification service, which notifies a user via email, whenever an alert on a selected topic appears on Google. In this case, the topics are "Distributed Denial of Service" and "DDoS". The data has been collected over a period of roughly two and a half years, starting on August 20, 2015 and ending on March 22, 2018. Overall, it contains 67831 Google alerts. It is part of a working paper on characteristics of DDoS attacks (Abhishta, Nieuwenhuis, Junger, & Joosten, 2017). The dataset contains information about the date of the alert, the alert type (blog, news or web), the link to the web page and a content column with the title of the alert and some of the content (about 2-3 sentences of the article).

4.2 Selection and Sampling

The Google alerts have to be reviewed manually, in order to determine whether they report on a DDoS attack. For this, I have randomly selected 1856 alerts of the "News" alert type. The justification to focus on the "News" alerts is that these produce more reliable and trustworthy accounts of DDoS related alerts, whereas "Web" alerts often tend to be very informal and less trustworthy. For these 1856 alerts, I have assigned each one with the labels "Attack", "Non Attack" or "Unknown". "Attack" indicates an attack-reporting alert, "Non Attack" indicates a non-attack-reporting alert and "Unknown" indicates an alert, which is inconclusive or is written in a language, which is foreign to me. As there were only 6 alerts to which I assigned the "Unknown" label, I decided to omit these 6 from the

selection. This is advantageous because it leaves me with a binary classification, which is easier to handle for some of the tested algorithms.

Furthermore, a portion of the data is used as a training set, D-train, and another portion is used as a test set D-test. 75% of the data is randomly selected to be the training set and the remaining 25% is used for testing. This is a distribution which has been used in many studies in the domain of machine learning (Rodrigues, Lourenco, Ribeiro, & Pereira, 2017; Zhou, Hu, & Wang, 2018). Applying this split leaves us with a training set containing 1387 alerts and a testing set containing 463 alerts.

4.3 Text Representation

In order to provide some explanation as to how the algorithms can be used to analyze text data, I will briefly mention how text data is represented. For machine learning algorithms to function, usually, numerical data is required. So the text data has to be transformed (Lewis, 1998). This can for instance be done by means of word counts. This model is often called bag of words. In this model, each word is assigned an index and the number of times each word shows up in the corpus is counted. Thus, word order and grammar are completely disregarded. This produces a format, which is compatible with machine learning algorithms. Another technique is the Tf idf (term frequency-inverse document frequency) transformation. This model puts more weight on words, which occur rarely. Whereas, in the bag of words model, words, which are large in number, will dominate the results (Kibriya, Frank, Pfahringer, & Holmes, 2004).

4.4 Machine Learning Algorithms

In this section, I explain the machine learning algorithms tested in this study:

- Naive Bayes
  - Multinomial NB
  - Gaussian NB

4.4.1 Naive Bayes

Naive Bayes has been referred to as one of the most efficient inductive algorithms for machine learning and data mining (Zhang, 2004). Naive Bayes methods entail several supervised learning algorithms based on Bayes' theorem with the assumption of independence between every pair of features. With a class variable c and a dependent feature vector t, Bayes' theorem states:

$$\Pr(c|t_i) = \frac{\Pr(c)\Pr(t_i|c)}{\Pr(t_i)}, \qquad c \in C$$

One of the variations of the Naive Bayes is the multinomial model. Along with the Bernoulli model, the multinomial model is one of the two commonly used models for text classification (Kibriya et al., 2004). Previous research has found the multinomial model outperforming the Bernoulli model. I examine the multinomial Naive Bayes and in addition, I test the Gaussian Naive Bayes. I feed training data to these algorithms and, based on the resulting model, I have the algorithms predict the classification for the test set. Then the predictions are compared to the actual classes of the alerts.

4.5 Word2vec

4.5.1 The Word2vec Algorithm

One of the more recent machine learning algorithms in text classification is Word2vec. It is different from of the previously used algorithms in the field, because it converts words and phrases into a vector representation (Lilleberg, 2015). Word2vec preserves syntactical information of the words in the corpus. This is in contrast to the models I have described in section 4.3. Some commonly used examples to illustrate the functionality of the algorithm look like this:

king – man + woman = queen

Or: Moscow – Russia + France = Paris

Referring to the first example, this can be understood in terms of subtracting the vector of the word 'man' from the vector of the word 'king'. This could be thought of as something like monarch, leaving out any gender connotation. Adding to that the vector of the word 'woman' would give us the vector of the word 'queen'. So when using Word2vec, emphasis is put on the linguistic context of the words in the corpus.

For the Word2vec algorithm, I will not be building a model to predict whether an alert reports on an attack or not. There are very few studies, in which prediction models with Word2vec have been built. Lilleberg has combined support vector machines and Word2vec, in order to build such a model (Lilleberg, 2015). Instead, I use the algorithm to analyze the text data in terms of importance of certain words and other features, related to linguistic properties.

4.5.2 Producing Word Vectors with Word2vec

In this section, I go over the individual steps I took in transforming the words in my corpus into vectors. At first, some pre-processing of the text was necessary. I removed all the punctuation marks from the text, since I am only interested in the linguistic properties of the actual words, rather than the various punctuation marks. The only punctuation marks left in the corpus after this step are the dots, which signify the end of one alert and the start of the next one. Thus, one sentence in the model represents one alert. Subsequently, I made use of a function, which removes stop words. This means, I removed very common words such as "the", "a", "of" and so on. These words do not carry a lot of meaning. Now, the individual words can be tokenized and assigned with a vector. This process resulted in a dictionary, containing 7465 words. This dictionary contains every word in the corpus, except for the stop words, with a vector assigned to it. With these steps completed, the model can be

trained. I will explore some results, which can be obtained with it in section 5.2.

5. RESULTS AND DISCUSSION

5.1 Naive Bayes

5.1.1 Combinations of tested Algorithms

As I have described in section 4.4, I tested two machine learning algorithms. The first one is the multinomial Naive Bayes and the second one is the Gaussian Naive Bayes. For each of these two algorithms, I used the bag of words model and the Tf idf model. This results in four combinations: multinomial Naive Bayes with bag of words, multinomial Naive Bayes with Tf idf, Gaussian Naive Bayes with bag of words and Gaussian Naive Bayes with Tf idf. In the Appendix, the confusion matrices for each of the four combinations can be found. There are five confusion matrices for each combination, each of them contains the results for one of the five trials from the k-fold cross validation. In the following section, I present the resulting accuracy, precision and recall scores for each combination. To give a quick summary of the overall results, the multinomial Naïve Bayes with bag of words has outperformed all the other combinations quite clearly. It is followed by the two Gaussian Naive Bayes combinations. The differences between Gaussian Naive Bayes with bag of words and with Tf idf are very small. The worst performer out of the four combinations, is the multinomial Naïve Bayes with Tf idf. It has produced only one identification of an attack reporting alert over all the k-trials. In the next two sections, I present the results in some more detail and discuss them.

5.1.2 Experimental Results

In this section, I present the experimental results, comparing the performance metrics of the machine learning algorithms. As explained in section 3.1, in addition to the accuracies of the respective algorithms, I will also provide information on precision and recall. I have performed a 5-fold cross validation and created a confusion matrix for each of the five trials. The three aforementioned performance measures, accuracy, precision and recall are calculated across all k-trails for each algorithm. In other words, the presented scores are averaged over the five trials.

These are the scores for Multinomial Naive Bayes (MNB) and Gaussian Naive Bayes (GNB), with Bag of words (BOW) and Tf idf for each of them.

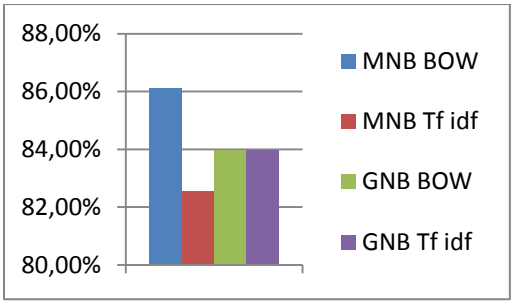


Figure 1: Mean accuracy scores

Figure 1 plots the accuracy score for each of the tested algorithms. We can see that for the examined dataset, the multinomial Naive Bayes with bag of words has performed best, in terms of accuracy. Compared to the bag of words model, Tf idf has resulted in a decrease in accuracy. For the Gaussian Naive Bayes, accuracy scores did not differ between bag of words and Tf idf.

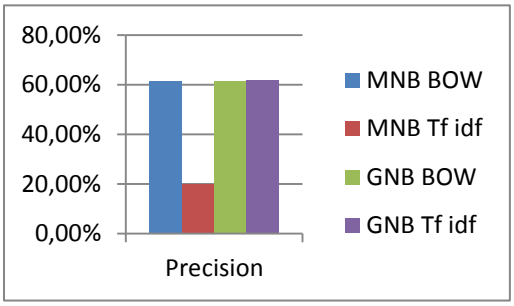


Figure 2: Mean precision scores

In Figure 2 the precision score for each algorithm is displayed. Looking at the multinomial Naive Bayes, a clear decrease in performance can be observed when the Tf idf model is used. Looking at the corresponding confusion matrices reveals that the Tf idf model for multinomial Naive Bayes has yielded almost no positive identifications. Merely one alert, out of all the k-trials, has been classified as an attack-reporting alert. That one has been correctly classified, which over five trials results in a 20% precision score. For the Gaussian Naive Bayes both, the bag of words and the Tf idf model, precision scores are just over 60% with a very small difference between the two.

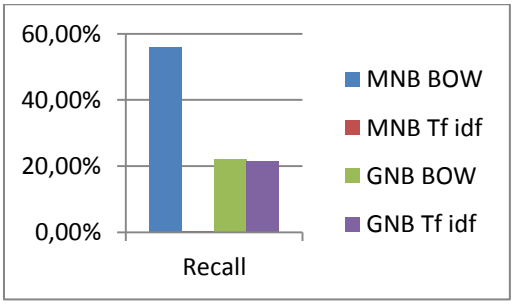


Figure 3: Mean recall scores

Figure 3 displays the recall score for each algorithm. Again, a striking difference between the

bag of words and Tf idf model for the multinomial bag of words, in favor of the bag of words model, can be observed. The recall score of just over 0.3% accentuates the fact pointed out in the previous paragraph, that there was only one positive identification. Looking at the Gaussian Naive Bayes, there is very little difference between the bag of words and Tf idf models. However, a difference in performance between the multinomial model and the Gaussian model becomes apparent. It appears that the multinomial model with bag of words performs better, than the Gaussian model.

5.1.3 Discussion

When assessing these results it is important to keep in mind that they are specific to the dataset at hand and do not necessarily predict performance of the tested algorithms in other contexts. Furthermore, some additional steps related to pre-processing of the data could presumably influence the results. Having said that, it appears that, for the data used in this study, the multinomial Naive Bayes has been shown to be the most effective machine learning algorithm, out of the ones which have been tested. However, with an accuracy of just over 86%, precision of about 61.4% and recall of 55.87% there is certainly room for improvement. Perhaps some more pre-processing of the text data could be beneficial. I will elaborate on this some more in the context of future research. The results indicate rather poor performance of the Tf idf model, compared to the bag of words model, certainly for the multinomial Naive Bayes. But I am inclined to be careful in assigning too much weight to this result. It is very possible that more processing of the data, than I was able to perform in the context of this research project, was required, to properly assess the effectiveness of the Tf idf model. Nevertheless, I think that the characterization of the Naive Bayes as one of the most adequate and easiest to implement machine learning algorithms in text categorization, has been confirmed.

5.2 Word2vec

The nature of my work with Word2vec is rather exploratory. One of the tools made available by the model I created, is to have it indicate which words are most similar to a certain word. For instance, according to the model, the 10 most similar words to the word ‘ddos’ are:

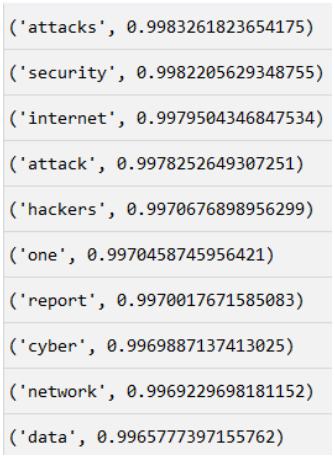
```
('.', 0.9999063014984131)
('attack', 0.9998792409896851)
('security', 0.9998787641525269)
('one', 0.9998695850372314)
('internet', 0.9998628497123718)
('attacks', 0.9998571872711182)
('hackers', 0.9998553395271301)
('network', 0.999850332736969)
('websites', 0.9998410940170288)
('data', 0.9998334646224976)
```

Graphic 1: Most related words to 'ddos'

The number one most similar word is the dot. This is true for many words because the dot appears at the end of every alert. Other than that, words produced by this input seem to make sense in terms of linguistic proximity to the word ‘ddos’ in this context. Another potentially interesting insight can be provided by splitting the corpus. I took only attack-reporting alerts and made a model out of it, as described in section 4.5.2.. Then, I did the same for non-attack-reporting alerts. Now we can look for the most similar word to ‘ddos’ again and compare the results for attack-reporting and non-attack-reporting alerts. I chose the word ‘ddos’ again because it is present in every single alert, regardless of it being attack-reporting or non-attack-reporting. This comparison makes apparent that some words are very high on the list in both models and others are more important in one of the models. For instance, the words ‘websites’, ‘company’, ‘service’, ‘anonymous’, and ‘targeted’ seem to correlate with attack-reporting alerts. This could be explored for any word and one could choose to look at more than just 10 words.

```
('attack', 0.9713090062141418)
('internet', 0.9639599919319153)
('attacks', 0.9548821449279785)
('websites', 0.9433374404907227)
('hackers', 0.9431532025337219)
('company', 0.9416540265083313)
('service', 0.9415541887283325)
('massive', 0.9357107281684875)
('anonymous', 0.9336929321289062)
('targeted', 0.9295684695243835)
```

Graphic 2: Most related words to 'ddos' in attack model



Graphic 3: Most related words to 'ddos' in non-attack model

6. CONCLUSION

Test classification can have a vital role in a wide range of tasks related to information retrieval. This study has explored a few algorithms, which can be of use in the context of separating attack-reporting Google alerts from non-attack-reporting Google alerts. Based on the data used for this study, the multinomial Naive Bayes with the bag of words model has been shown to be a candidate for effective classification of Google alerts about DDoS. The Tf idf model in combination with the multinomial Naïve Bayes has not yielded very good results in this study. In almost all cases, alerts have been classified as non-attack-reporting. The results in this research project suggest that the bag of words model is superior to the Tf idf model, for the multinomial Naive Bayes. However, these results should not be interpreted as a final evaluation of the two models.

The Gaussian Naïve Bayes has not performed very well in identifying alerts reporting on DDoS attacks. Only a little over twenty percent of attack-reporting alerts, have been categorized as such. For the examined alerts, there has been very little difference between the bag of words model and the Tf idf model. Again, one should be careful in assigning too much weight to these findings, especially in terms of implications for other contexts outside of this data set or domain.

The Word2vec model I have built, allows for some insights into semantic features of the Google alerts. It can be used to uncover words, which are correlated to one class of alert. Some words, which are more present in attack-reporting alerts include ‘websites’, ‘company’, ‘massive’, ‘targeted’ and ‘anonymous’. In non-attack-reporting alerts the words ‘security’, ‘report’,’ network’ and ‘data’ are among the more prevalent words.

7. FUTURE RESEARCH

This paper does not represent an exhaustive examination of machine learning algorithms applied to DDoS related text data, or even of the dataset. Thus, there are many possible directions of future research. First, there are several other machine learning algorithms for text classification, which could be tested in a similar way as I have done in this project. Furthermore, there are some steps which could be taken to pre-process the text data. In the context of my examination of the Word2vec algorithm, I have made use of stop words, to eliminate words such as “the”, “of” or “a”, which do not bear a lot of relevance, in order to differentiate attack-reporting from non-attack-reporting alerts. Moreover, word stemming could be considered. Here, inflected or derived words are reduced to their word stem.

Another important point to make, concerning this dataset, is that, within the time limits of this research project, I only made use of a relatively small fraction of the dataset. I ended up looking at 1856 Google alerts in total and the entire dataset contains 67831 Google alerts (as of the beginning of this research project).

An interesting prospect for future research could be to extrapolate the findings about effectiveness of machine learning algorithms to other sources of information. In the same way as Google alerts has been used for this study, other databases could be used to gather news reporting a DDoS attack. The performance of, say, the multinomial Naive Bayes with bag of words, could be tested against those other databases. One interesting alternative might be the LexisNexis Academic Knowledge Center, to name just one.

Finally, other domains related to business and cybercrime could be examined. Results of this study and other studies in the broader context of this research on DDoS could be extrapolated to keywords other than “DDoS” or “Distributed Denial of Service”. Some phenomena, which are often closely related to DDoS, such as extortion or disruption of cryptocurrency exchange, might be interesting candidates for studies, which are similar to this one.

I have also shown some of the work I have done related to the Word2vec algorithm. In the same way as I have described possibilities for further research into prediction models, future research could be done on the Word2vec model. It might be applied more to this dataset, to other databases and even to other domains related to cybercrime.

In addition to splitting the corpus into attack-reporting alerts and non-attack-reporting alters, one could use other features to split the corpus by. Specific time periods could be examined and compared, to see whether differences become apparent.

8. ACKNOWLEDGEMENTS

I would like to take this opportunity to thank my supervisors Abhishta and prof. dr. Marianne Junger for their impeccable support and supervision over the course of my thesis. I would also like to thank Abhishta for allowing me to use his dataset in the context of my thesis.

9. REFERENCES

Abhishta, Nieuwenhuis, L. J. M., Junger, M., & Joosten, R. A. M. G. (2017). *Characteristics of DDoS attacks: An analysis of the most reported attack events of 2016*.

Abhishta, Joosten, R., & Nieuwenhuis, L. J. M. (2017). Analysing the Impact of a DDoS Attack Announcement on Victim Stock Prices. In *2017 25th Euromicro International Conference on Parallel, Distributed and Network-based Processing (PDP)* (pp. 354–362). IEEE. https://doi.org/10.1109/PDP.2017.82

Anderson, R., Barton, C., Böhme, R., Clayton, R., van Eeten, M. J. G., Levi, M., … Savage, S. (2013). Measuring the Cost of Cybercrime. In *The Economics of Information Security and Privacy* (pp. 265–300). Berlin, Heidelberg: Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-39498-0_12

Behal, S., & Kumar, K. (2017). Characterization and comparison of DDoS attack tools and traffic generators - a review. *International Journal of Network Security*, *19*(3), 383–393. https://doi.org/10.6633/IJNS.201703.19(3).07

Campbell, K., Gordon, L. A., Loeb, M. P., & Zhou, L. (2003). The Economic Cost of Publicly Announced Information Security Breaches : Empirical Evidence from the Stock Market. *Journal of Computer Security*, *11*(3), 431–448. Retrieved from http://iris.nyit.edu/~kkhoo/Spring2008/Topics/Topic10/EconCostSecurityBreaches2003.pdf

Goth, G. (2007). The Politics of DDoS Attacks. *IEEE Distributed Systems Online*, *8*(8), 3–3. https://doi.org/10.1109/MDSO.2007.50

Hovav, A., & D'Arcy, J. (2003). The Impact of Denial-of-Service Attack Announcements on the Market Value of Firms. *Risk Management Insurance Review*, *6*(2), 97–121.

Jonker, M., King, A., Krupp, J., Rossow, C., Sperotto, A., & Dainotti, A. (2017). Millions of targets under attack. In *Proceedings of the 2017 Internet Measurement Conference on - IMC '17* (pp. 100–113). New York, New York, USA: ACM Press. https://doi.org/10.1145/3131365.3131383

Kibriya, A. M., Frank, E., Pfahringer, B., & Holmes, G. (2004). Multinomial Naive Bayes for Text Categorization Revisited. In *Proceeding AI'04 Proceedings of the 17th Australian joint conference on Advances in Artificial Intelligence* (pp. 488–499). https://doi.org/10.1007/978-3-540-30549-1_43

Kotsiantis, S. B., Zaharakis, I., & Pintelas, P. (2007). Supervised machine learning: A review of classification techniques. In *Emerging artificial intelligence applications in computer engineering* (pp. 3–24).

Lewis, D. D. (1998). Naive (Bayes) at forty: The independence assumption in information retrieval (pp. 4–15). https://doi.org/10.1007/BFb0026666

Lilleberg, J. (2015). Support Vector Machines and Word2vec for Text Classification with Semantic Features. *IEEE 14th International Conference on Cognitive Informatics & Cognitive Computing (ICCI*CC)*, 136–140. https://doi.org/10.1109/ICCI-CC.2015.7259377

McCallum, A., & Nigam, K. (1998). A Comparison of Event Models for Naive Bayes Text Classification. *AAAI/ICML-98 Workshop on Learning for Text Categorization*, 41–48. https://doi.org/10.1.1.46.1529

Mitchell, T. M. (1997). *Machine learning. WCB*.

Rodrigues, F., Lourenco, M., Ribeiro, B., & Pereira, F. C. (2017). Learning Supervised Topic Models for Classification and Regression from Crowds. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *39*(12), 2409–2422. https://doi.org/10.1109/TPAMI.2017.2648786

Santanna, J. J., & Sperotto, A. (2014). Characterizing and Mitigating the DDoS-as-a-Service Phenomenon. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* (Vol. 8508 LNCS, pp. 74–78). https://doi.org/10.1007/978-3-662-43862-6_10

Sebastiani, F. (2002). Machine learning in automated text categorization. *ACM Computing Surveys*, *34*(1), 1–47. https://doi.org/10.1145/505282.505283

Segura, V., & Lahuerta, J. (2010). Economics of Information Security and Privacy. https://doi.org/10.1007/978-1-4419-6967-5

Williams, N., Zander, S., & Armitage, G. (2006). A preliminary performance comparison of five machine learning algorithms for practical IP traffic flow classification. *ACM SIGCOMM Computer Communication Review*, *36*(5), 7–15. https://doi.org/10.1145/1163593.1163596

Wueest, C. (2014). The continued rise of DDoS attacks, 1–31. Retrieved from http://www.symantec.com/content/en/us/enterprise/media/security_response/whitepapers/the-continued-rise-of-ddos-attacks.pdf

Zhang, H. (2004). The optimality of naive Bayes. https://doi.org/10.1016/j.patrec.2005.12.001

Zhou, Y., Hu, Q., & Wang, Y. (2018). Deep super-class learning for long-tail distributed image classification. *Pattern Recognition*, *80*, 118–128.

10. APPENDIX

K-Fold Confusion Matrices GNB bag of words

Category		Algorithm Judgment		
		Attack	Non Attack	
Actual Class	Attack	14	51	65
	Non Attack	11	295	306
		25	346	371

Category		Algorithm Judgment		
		Attack	Non Attack	
Actual Class	Attack	13	52	65
	Non Attack	8	297	305
		21	349	370

Category		Algorithm Judgment		
		Attack	Non Attack	
Actual Class	Attack	14	51	65
	Non Attack	10	295	305
		24	346	370

Category		Algorithm Judgment		
		Attack	Non Attack	
Actual Class	Attack	17	48	65
	Non Attack	8	297	305
		25	345	370

Category		Algorithm Judgment		
		Attack	Non Attack	
Actual Class	Attack	14	51	65
	Non Attack	8	297	305
		22	347	370

K-Fold Confusion Matrices GNB TFIDF

Category		Algorithm Judgment		
		Attack	Non Attack	
Actual Class	Attack	14	51	65
	Non Attack	11	295	306
		25	346	371

Category		Algorithm Judgment		
		Attack	Non Attack	
Actual Class	Attack	12	53	65
	Non Attack	7	298	305
		19	351	370

Category		Algorithm Judgment		
		Attack	Non Attack	
Actual Class	Attack	14	51	65
	Non Attack	9	296	305
		23	347	370

Category		Algorithm Judgment		
		Attack	Non Attack	
Actual Class	Attack	17	48	65
	Non Attack	8	297	305
		25	345	370

Category		Algorithm Judgment		
		Attack	Non Attack	
Actual Class	Attack	13	51	64
	Non Attack	8	297	305
		21	348	369

K-Fold Confusion Matrices MNB bag of words

Category		Algorithm Judgment		
		Attack	Non Attack	
Actual Class	Attack	37	28	65
	Non Attack	19	287	306
		56	315	371

Category		Algorithm Judgment		
		Attack	Non Attack	
Actual Class	Attack	39	26	65
	Non Attack	23	282	305
		62	308	370

Category		Algorithm Judgment		
		Attack	Non Attack	
Actual Class	Attack	35	30	65
	Non Attack	25	280	305
		60	310	370

Category		Algorithm Judgment		
		Attack	Non Attack	
Actual Class	Attack	34	31	65
	Non Attack	26	279	305
		60	310	370

Category		Algorithm Judgment		
		Attack	Non Attack	
Actual Class	Attack	36	28	64
	Non Attack	21	284	305
		57	312	369

K-Fold Confusion Matrices MNB TFIDF

Category		Algorithm Judgment		
		Attack	Non Attack	
Actual Class	Attack	0	65	65
	Non Attack	0	306	306
		0	371	371

Category		Algorithm Judgment		
		Attack	Non Attack	
Actual Class	Attack	0	65	65
	Non Attack	0	305	305
		0	370	370

Category		Algorithm Judgment		
		Attack	Non Attack	
Actual Class	Attack	1	64	65
	Non Attack	0	305	305
		1	369	370

Category		Algorithm Judgment		
		Attack	Non Attack	
Actual Class	Attack	0	65	65
	Non Attack	0	305	305
		0	370	370

Category		Algorithm Judgment		
		Attack	Non Attack	
Actual Class	Attack	0	64	64
	Non Attack	0	305	305
		0	369	369