Enron Spam Analysis

Quinton Weenink, 13176545[1]
Bernhard Schuld, 10297902[1]
Nicaedin Suklal, 15207812[1]

[1]Department of Computer Science – University of Pretoria
Pretoria, South Africa

1. Introduction

The purpose of this paper is to consolidate the findings and implementation of our email data set analysis. The project was split into different sections that allowed us to approach the classification and analysis of a large email data set from many different perspectives. Through selection and data cleaning as well as proper data exploration, a proper data structure could be built for forensic investigation. For those sections that have a kernel, a Github link to the code can be found.

2. Data Acquisition

2.1. Email Dataset

In order to have enough data to utilize for our scripts, we decided to acquire an existing email dataset that is publicly available rather than generate our own. We used the May 7, 2015 version of the Enron Email dataset available at the Carnegie Mellon University's website (https://www.cs.cmu.edu/ enron/)

The email dataset contained 517 401 emails.

2.2. Structuring the data

Having to perform input and output operations for each and every email file would be extremely time consuming and strenuous on our computers if we wanted to perform big data operations on the files. We thus opted to write all the data into one .csv file, namely enron.csv

This initial step requires a significant amount of time to perform, but allows later operations to be performed at a much quicker pace. Reading the emails into one file takes a rough average of about 30 minutes.

After this initial data acquisition, each script reads the .csv file into a Pandas DataFrame and performs it's relevant functionality on the data.

2.3. Spam Training Data

In order to train our model we had to have a set of data that was confirmed to be spam emails. For this we used a freely available spam archive that was available online (http://untroubled.org/spam/). We utilized the 2001 dataset as it was at the end of that year that Enron declared bankruptcy and one can make an assumption that it might help better identify spam during the Enron era as the words utilized in spam change over the years as languages evolve. We used the emails contained in this archive to train our model to recognize spam, which will be discussed in more detail in section 7

3. Data Cleaning - kernel

Specific attributes were identified to be not needed and dropped from further use.

4. Privacy Preservation - kernel

As it is illegal to utilize a person's data without their explicit consent, we had to anonymize any personally identifiable information. To this end we opted to perform a hashing algorithm on any header information we deemed might contain personally identifiable information. We utilized a SHA256 alogrithm to hash the following fields from the header:

- From
- To
- X-From
- X-To
- CC
- Bcc
- X-cc
- X-bcc
- X-Folder
- X-Origin
- X-FileName

It should be noted that because of the initial setup explained in section 2.2, the execution of the privacy preservation script takes less than 60 seconds.

5. Exploratory data analysis - kernel

In this section we identify what data is useful to use and interesting patterns that are present in the data such as how unique it is represented in 2

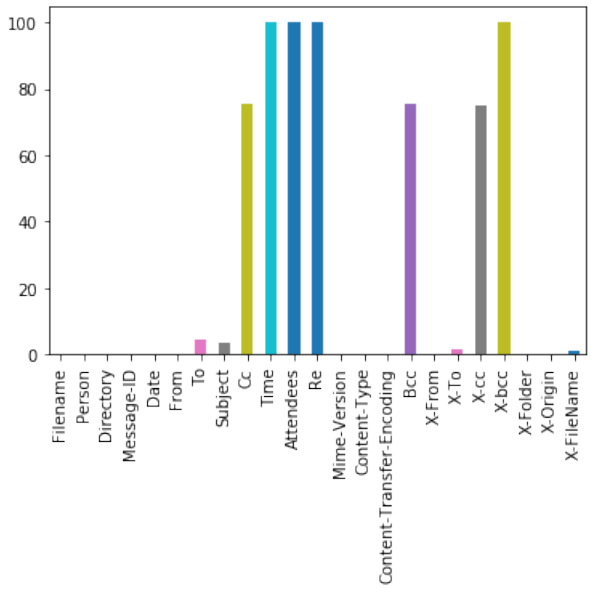

Figure 1. Headers per missing data

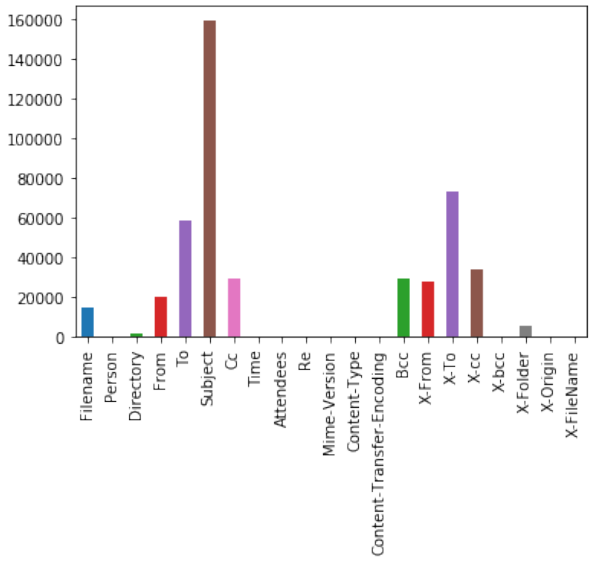

Figure 2. Uniqueness / stochasticity of data

6. Attribute identification and Feature Creation - kernel

6.1. Attributes

The identification of the attributes are centred around the discovery of each email header. After the enron.csv file was created with the headers stored, the attributes of each header were identified and then formed a basis of what information would be needed for digital forensic and authorship analysis purposes. The attributes are represented in 3.

6.2. Features

Features were defined from the attributes identified above, these are put in place to allow for the forensic evaluation and authorship analysis to progress with more efficiency as potentially suspicious emails are made easier to flag via a combination of uniquely identifying attributes.

The Features were then used in example applications in order to show how effectively they allow the discovery of potentially malicious emails. The example features can be shown in 4 and in the notebook link included at the top of this section.

```
['Filename',
 'Person',
 'Directory',
 'Message-ID',
 'Date',
 'From',
 'To',
 'Subject',
 'Cc',
 'Time',
 'Attendees',
 'Re',
 'Mime-Version',
 'Content-Type',
 'Content-Transfer-Encoding',
 'Bcc',
 'X-From',
 'X-To',
 'X-cc',
 'X-bcc',
 'X-Folder',
 'X-Origin',
 'X-FileName']
```

Figure 3. Attributes

```
authSus['Message-ID']
authSus['Date']
authSus['From']
authSus['To']
authSus['Bcc']
authSus['Subject']
authSus['Person']
authSus['X-Origin']
authSus['Content-Type']
```

Figure 4. Features

7. Machine Learning - kernel

A Neural Network was trained on the Enron data set as well as a set of known spam emails.

The subject of each email was broken up into words and added to a list of words with the number of times that it occurred. Only words that occurred more than 30 times were used.

7.1. Bag of words

When an email was processed the subject's words, if contained in the accepted word list, were mapped to a 1 while words that were not in the subject were mapped to a 0.

7.2. Training

5000 positive and 5000 negative samples were randomly obtained and used to train the neural network. Each of the positive samples had a target of 1 while the negatives samples have a target of 0.

A learning rate of 0.001 was used in order to prevent the neural network from overfitting the training data.

8. Cyber Criminal Profiling - kernel

At the end of the spam classifier kernel we are able to use the trained model to identify emails as spam or not spam.

In some cases the classifier seems to identify emails that are not spam as spam.