UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

# COS 720: Computer and Information Security Project Documentation


Armand Maree 120 178 00
Charl Meyers 14024633
Marc Antel 12026973

Department of Computer Science, University of Pretoria

# Contents

# 1 Introduction

Spam and malicious emails is a world wide problem wasting billions of dollars every year [1]. Since 90% of all email traffic is spam, a lot of network bandwidth, processing power, storage and electricity goes toward the processing and transportation of these emails [1]. Due to this large amount of unwanted email traffic workers waste time everyday filtering these out. Not only that, but malicious emails could cause massive financial loss to an individual and/or company. For these reasons there have been numerous studies done to identify and filter out these emails. One such approach is header based spam filtering [2]. This approach requires the analysis of large datasets in order to accurately classify emails [2].

# 2 Data Acquisition

On March 26, 2003 one of the most widely used email corpuses was published on the Internet by the Federal Energy Regulatory Commission (FERC) [3]. This was of course the Enron email corpus [4]. There were two reasons why this data was chosen for this project, the first is due to this data set's wide and successful use in previous research, and the second reason is due to the massive number of "real-world" emails that it provides.

# 3 Data Cleaning

There were multiple steps that played a role in the data cleaning process. The main purpose of this step was to place the data in a CSV format that could be used for further analysis. These steps will be discussed in the next sections.

## 3.1 Email Format

Due to previous research and individuals like Melinda Gervasio [4] the emails have been clean up drastically and reformatted to follow a more standard layout. The emails start out with all the headers and followed by the email body. These two are separated by a blank line. Since we are only interested in the email headers, we will focus on this section more extensively. The general header layout consisted of the header followed by a colon and then the header's value. Occasionally the value spans over multiple lines since a line is not allowed to be more than 998 characters long (although 78 characters is recommended) excluding the CRLF characters [5]. Any data after the blank line that separated the headers and the body was ignored.

## 3.2 Data Extraction

### 3.2.1 Header Acquisition

Each line of the header section of the email was scanned and split by the colon character based on the RFC specification for email layouts [5]. Any text before the colon is seen as the header key that follows the colon is seen as the header value. As already mentioned, there were cases where the header value wrapped across multiple lines. This is allowed as long as the new line starts with a white space character [5]. However, there were a few emails that did not start the new line with a white space character. These emails were flagged by the cleaning program and were corrected manually. Header values that spanned across multiple lines were combined into a single line.

The headers (with descriptions) that were found in the complete set of emails were:

- Received1 (fabricated header)
  Contained the address of the receiving server. Example: *by 184.168.221.41 with SMTP id hUFS1mAtWD54sRsA; Mon, 14 May 2001 16:39:00 -0700 (PDT).*
- Received2 (fabricated header)
  Contained the address of the sending server as observed by the receiving server. Example: *from enron.com (enron.com [184.168.221.41]) by mx.google.com with ESMTPS id IA15iqgM3n4tZP; Mon, 14 May 2001 16:39:00 -0700 (PDT) (version=TLS1_2 cipher=AES128-GCM-SHA256 bits=128/128); Mon, 14 May 2001 16:39:00 -0700 (PDT).*

- X-Mailer (fabricated header)
  This field was left blank in the emails but usually contain the email program that was used to compose the program [6].

- Message-ID
  The message ID of an email consists of a global unique ID to identify this email. It usually, although not in the case of the Enron emails, ends with the domain of the sending server. Example: *<18782981.1075855378110.JavaMail.evans@thyme>*.

- Date
  The time stamp the email was sent. This time stamp has a very specific format which is: Day of the week, Day Month Year Hour:Minute:Second +/-Time zone offset. Example: *Mon, 14 May 2001 16:39:00 -0700 (PDT)*

- From
  The sender email address. Example: *piet.pompies@enron.com*.

- To
  A list of one or more recipient email addresses separated by a comma. Example: *johan.kotze@enron.com, koos.opperman@enron.com*.

- Subject
  The subject of the email as specified by the sender.

- Mime-Version
  Multipurpose Internet Mail Extensions (MIME) is a header that indicates that the message has been formatted in accordance with the MIME format standard [7]. At the moment there is only one MIME version (version 1.0) [7] and there hasn't been a need to call for a new official version.

- Content-Type
  Also referred to as the media type [7], and is used to allow the received to choose the correct mechanism to display the email to the user. The default for this field is *text/plain; charset=us-ascii* [7].

- Content-Transfer-Encoding
  The primary purpose of this field is to allow the receiver to decode binary data that was converted to a text format. This field specifies how this data was encoded [7]. The default value is *7bit* [7].

- X-From
  As stated in [8] any header starting with an "X" is considered application specific and will not be overwritten by a future standard. This has since been deprecated [9] but only after the Enron emails have been sent.
  In our data set this field contained most often contained the sender's name and surname, their email address and/or other random information. Example: *Mass, Frans </O=ENRON/OU=NA/CN=RECIPIENTS/CN=FSAYRE>*.

- X-To
  Similar to the X-From header.

- X-cc
  Similar to the X-From header.

- X-bcc
  Similar to the X-From header.

- X-Folder
  This header appears to store the folder the email was placed in, in the user's email program.

- X-Origin
  Most often this field contained the sender's surname and firstname's initial. Example: *Mass-F*

- X-FileName
  The Note Storage Facility (NSF) or Outlook Personal Folders (PST) file name.

- Cc
  A list of one or more recipient email addresses of individuals that was added to the carbon copy list separated by a comma. Example: *johan.kotze@enron.com, koos.opperman@enron.com.*

- Bcc
  A list of one or more recipient email addresses of individuals that was added to the blind carbon copy list separated by a comma. Example: *johan.kotze@enron.com, koos.opperman@enron.com.*

#### 3.2.2 Removed Emails

Complete removal of emails were left as a last resort in order to maintain the large numbers of emails. The only emails that were stripped completely were emails that did not contain a header value for the "To" header key. The reason for this is because these emails were seen as draft emails that were not sent out.

#### 3.2.3 Data Preparation

Some of the fields were reformatted in order to provide a simpler data set which would in turn allow for a more successful data mining process. This includes the date field which was reformatted to the yyyy-MM-ddThh:mm:ss format. Missing headers were added with an empty value. Quotes and commas were stripped from the fields to allow CSV compatibility. For the Received1 and Received2 fields, only the IP addresses were extracted of the corresponding servers.

Some additional data was also gathered from the emails and added as additional fields. This includes:

- Message-ID-Server
  If the message ID contains a domain name then it is extracted and placed into this field.

- From-IP
  The actual IP address were requested for the sending email address's domain. Some domains' IP addresses could not be retrieved and in this case the domain name was placed in this field.

- File-size
  The size of the email in bytes.

#### 3.2.4 Data Conversion

In order to allow the machine learning phase to be able to more easily mine the data, the text field were converted to numerical representation and a mapping file (for our reference) was maintained. This includes the following fields:

- Message-ID-Server
- Mime-Version
- Content-Type
- Content-Transfer-Encoding

Other fields had some aggregation performed in order to get a numerical value. Like the recipient fields (listed below) were converted the number of recipients in that fields. For instance, a "To" field that contained 3 recipient email addresses has it's value replaced by the number 3.

We thus maintained two copies of the data, and either could be used by any phase of the project. The first copy is the original string formatted data and the second is the numerical data.

## 4 Exploratory Data Analysis

In order to gain a more in depth understanding of how the data looks we performed some Exploratory Data Analysis (EDA). EDA is a method used in research to gain knowledge of the data that could in turn generate ideas

and provide a possible path of investigation to follow [10].

A script was written to extract data from the cleaned data file and draw some Excel graphs to provide a visual mechanism of interpreting the data. The following analysis was performed as part of the EDA:

- Number of emails sent per user (figure 1)
- Number of emails received per user (figure 2)
- Number of emails sent during each hour of the day (figure 3)
- What content types were used (figure 4)
- What MIME-Types were used (figure 5)
- What transfer encodings were used (figure 6)
- What domains were listed as part of the message ID (figure 7)
- Some descriptive statistics were also performed to indicate what the max, min and average number of emails per user was (figure 8)

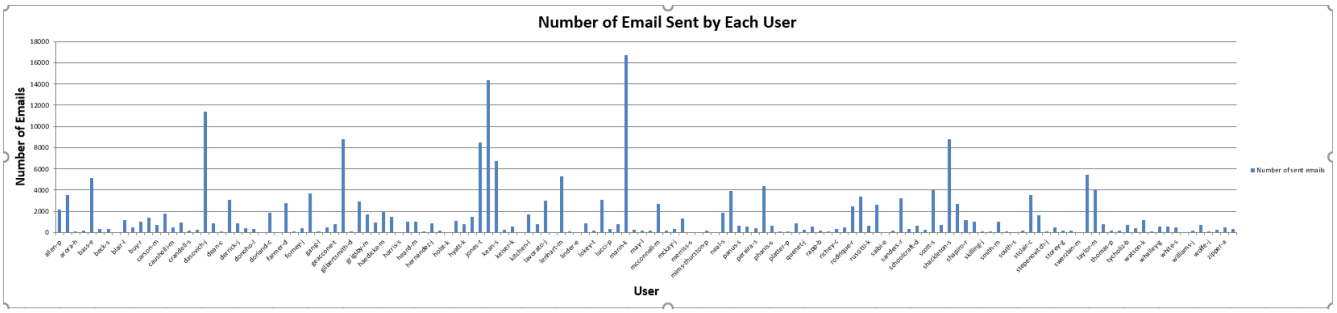


Figure 1: Number of emails sent per user.

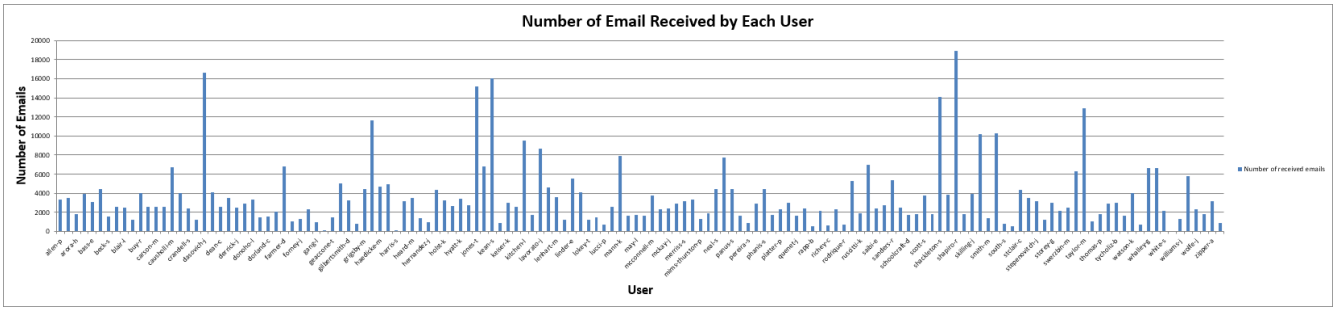


Figure 2: Number of emails received per user.

# 5 Attribute Identification

After analyzing the results from the EDA phase and a more in-depth understanding of the "landscape" of the data was obtained we performed attribute identification. This involved picking certain attributes that could (after some potential remodeling) be used by the machine learning tools to be able to more accurately classify the data.

The following attributes of the headers were identified, which would potentially aid in detecting abnormal or malicious emails. [11] [12]

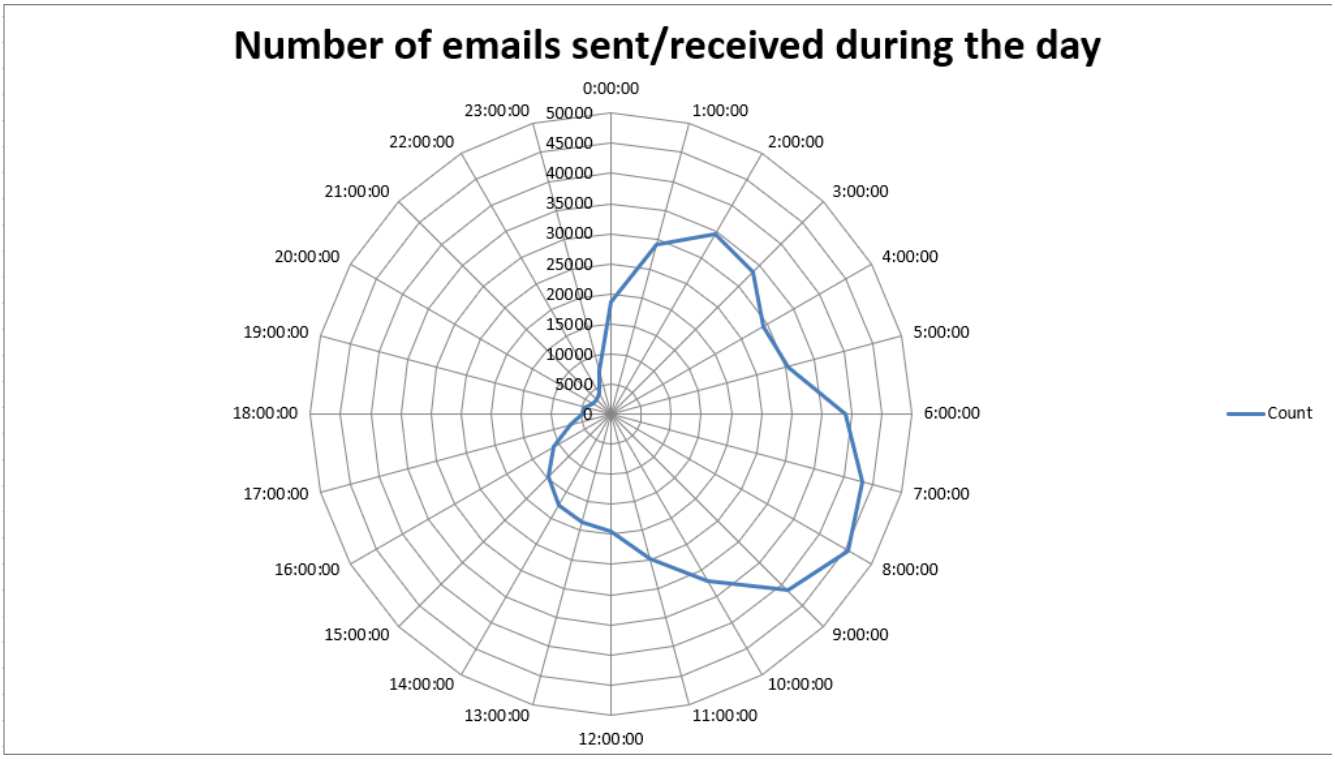


Figure 3: Time of the day emails were sent/received.

- Received1
  Can be used to identify if the receiving server has a malicious IP address by using a predefined list of malicious blacklisted IP addresses.
- Received2
  Can be used to identify if the sending server has a malicious IP address by using a predefined list of malicious blacklisted IP addresses.
- Message-ID
  In the Enron emails the domain of the message ID do not match the sender's email address domain. It is suspected that this was changed during the clean up of previous research since the two usually matches. Therefore, we did not utilize this technique ad only focuses on whether the domain specified is malicious or not.
- From
  The from field will be used to check if it is in anyway suspicious, or has any obvious indicators of a fabrication. An example of this would be a domain in the email address that is similar to a legitimate domain, like "claims@netbank.co.za" instead of "claims@nedbank.co.za". Unfortunately, due to time constraints this was omitted from our tests.
- To
  Mass spam and malicious emails often contain a large amount of recipients. This field can thus be used to check the number of addresses in the "To" field.
- Subject
  The Subject can be used to detect phishing emails based on the presence of certain characteristics, like the presence of special characters.
- X-To
  The X-To field can be used similar to the To field, to check how many other recipients there are to that email.
- X-cc
  The X-cc field can be used similar to the To field, to check how many other recipients there are to that email.

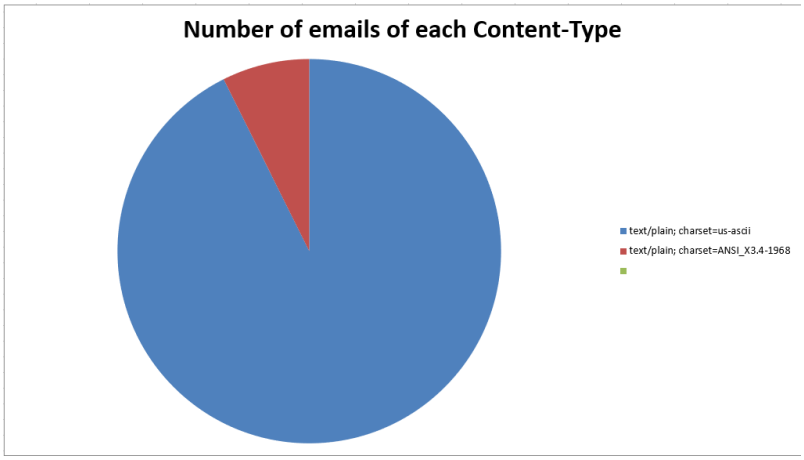


Figure 4: Content type used in emails.

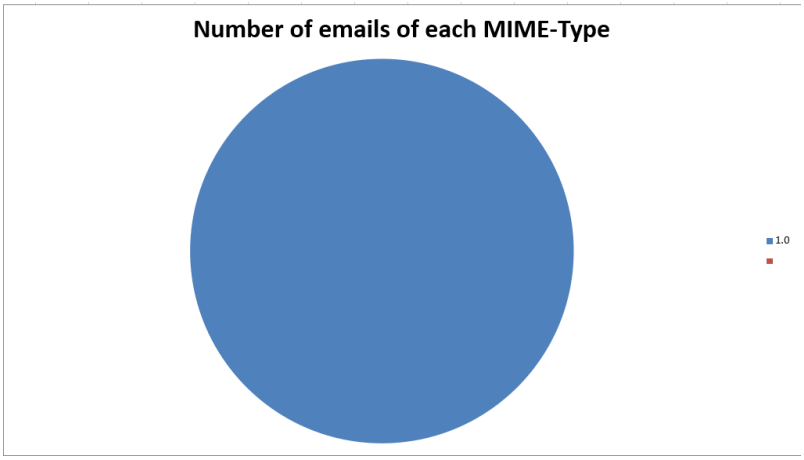


Figure 5: MIME type used in emails.

- X-bcc
  The X-bcc field can be used similar to the To field, to check how many other recipients there are to that email.
- Cc
  The Cc field can be used similar to the To field, to check how many other recipients there are to that email.
- Bcc
  The Bcc field can be used similar to the To field, to check how many other recipients there are to that email.

# 6 Feature Engineering

Feature Engineering was done by using the attributes identified above and processing them into values that can be interpretable for machine learning, as well as useful in detecting possible malicious emails.[11] [12] [13]

- To
  This gives the number of "To" contacts in the email.
- X-To
  This gives the number of "X-To" contacts in the email.
- X-cc
  This gives the number of "X-cc" contacts in the email.

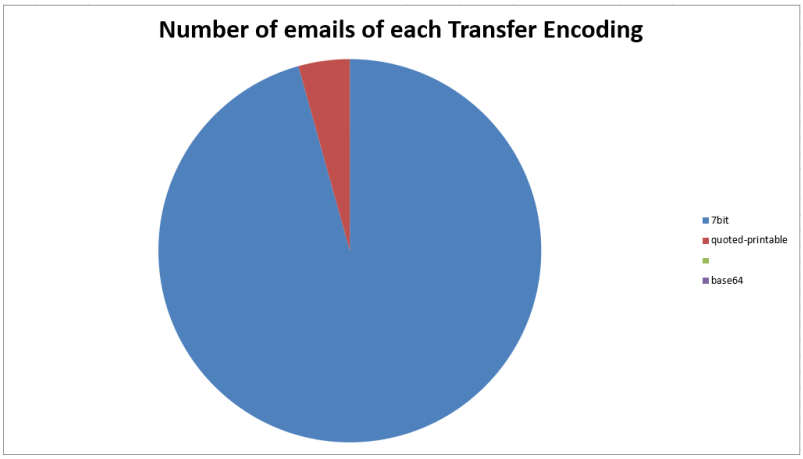


Figure 6: Transfer encoding used in emails.

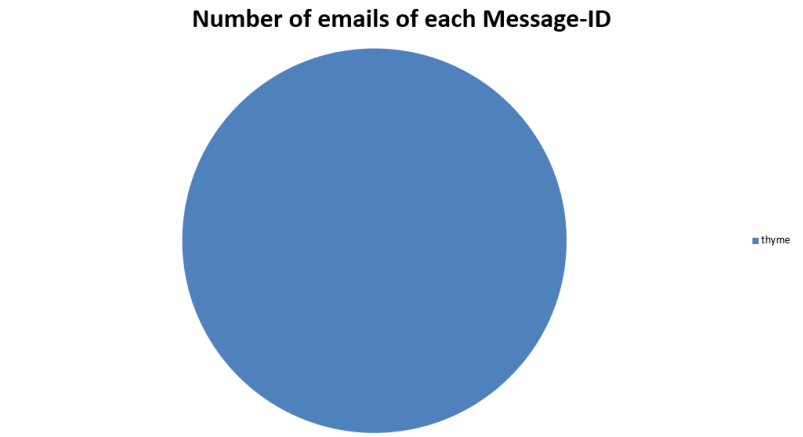


Figure 7: Message ID domain used in the emails.

- X-bcc
  This gives the number of "X-bcc" contacts in the email.
- Cc
  This gives the number of "Cc" contacts in the email.
- Bcc
  This field gives the number of "Bcc" contacts in the email.
- Possibly-Spam-Subject
  The Possibly-Spam-Subject is set to true if the subject is found to have characteristics of Spam mail.
- Bcc-Larger-Than-CC
  This field is a boolean value which is set to true if the number Bcc contacts is larger than the contacts found in the CC field. Otherwise it's set to false.
- Bcc-Larger-Than-To
  This field is a boolean value which is set to true if the number Bcc contacts is larger than the contacts found in the To field. Otherwise it's set to false.
- Blacklisted-IP-Address
  The Blacklisted-IP-Address field is a boolean which is set to true if the IP addresses found in the Received fields or in the From-IP field, matches an IP address in the blacklist of known servers found to be used for malicious emails.

| | A | B |
|---|---|---|
| 1 | Max emails sent | 16735 |
| 2 | Min emails sent | 0 |
| 3 | Average emails sent | 1470.267 |
| 4 | | |
| 5 | Max emails received | 18917 |
| 6 | Min emails received | 51 |
| 7 | Average emails received | 1470.267 |

Figure 8: Descriptive statistics on sent/received emails.

- Special-Chars-Subject
  The Special-Chars-Subject is a count of special characters such as "$' and "!' in the subject line.
- Count-Uppercase-Chars-Subject
  The Count-Uppercase-Chars-Subject is a count of how many characters are uppercase in the subject line.
- Subject-Length
  The Subject-Length is the length of the subject line. Together with Count-Uppercase-Chars-Subject the subject line is checked to see if the entire subject line is capitalized.

# 7 Using Machine Learning to Detect Abnormal Behavior

## 7.1 Decision tree

### 7.1.1 Setup

The following was done to prepare the machine learning system.

- Create training dataset
  - A training dataset with a size of 74 entries was created.
  - The training dataset consists of non-malicious emails,as well as malicious emails of various nature.
  - The training dataset is based findings from the feature engineering section, and has an additional boolean field named 'Possibly-Malicious', which is used as the target value for the training set.
- Create full dataset
  - A dataset containing the data of all the emails was created to test the accuracy of the model.

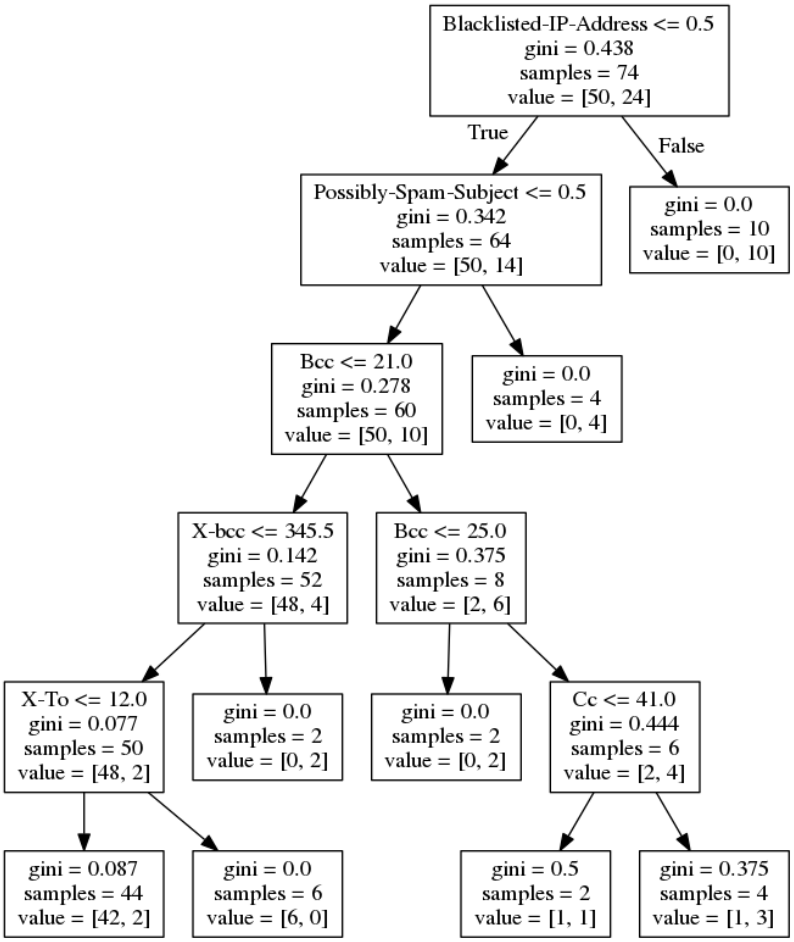


Figure 9: The decision tree model that was generated using the training dataset

#### 7.1.2 Results and interpretation

- Interpreting the model
  Figure 9 shows the decision tree model which was generated from the training dataset. Interpreting the model shows us that if Blacklisted-IP-Address is found to be true (>0.5), then the email will be found to be malicious. However, if it is found to be false (<=0.5) it will check the value of Possibly-Spam-Subject. If Possibly-Spam-Subject is true, then emails may also be found malicious. However if it's found to be false, then it moves to the next criteria, by check the values of Bcc. If the value was <=21 then the X-bcc and X-To will be checked. If it was >21 then the Bcc will be checked again as well as the Cc.

- Accuracy results on the full dataset
  The accuracy yielded after running a prediction on the full dataset was 99.6%. This is an indication that the model generated from the learning dataset is a good prediction on malicious emails in the full dataset.

### 7.2 Comparing rule induction and Naive Bayes

RapidMiner was used to setup a performance measure between Rule Induction machine learning, Naive Bayes and Decision Tree. The Confusion matrix and ROC of both algorithms are compared.

#### 7.2.1 Setup

- The data is first preprocessed to get more reliable data

  - First a subset of attributes are selected from the results of feature engineering
  - After that any rows with missing data is removed, this results in the dataset being reduced from 517 401 entries to 493 043 entries.

- The data is then sent to the machine learning algorithm and the performance is measured
  - First the data is split in two with a ratio of 70% and 30%
  - The 70% gets fed into the machine learning algorithm
  - The prediction column (Possibly-Spam-Subject) is removed from the 30% dataset for validation
  - The validation dataset is applied to the resulting model from the machine learning algorithm and compared to the input dataset to measure the performance. Figure 10 is an example of how the model looks like in RapidMiner.

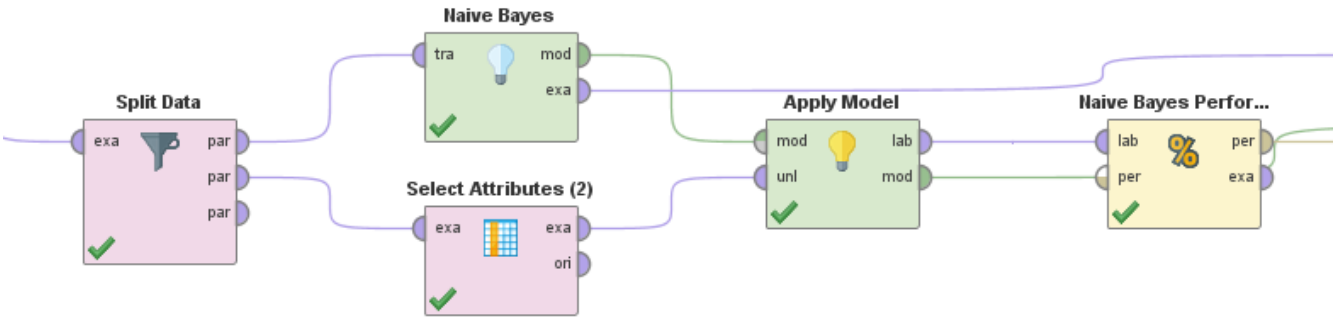


Figure 10: The resulting model in RapidMiner for one of the algorithms

### 7.2.2 Results and interpretation

The overall accuracy for rule induction is 96.2% and Naive Bayes is 99.64%. Thus far it looks like Naive Bayes is the better algorithm to use between Rule Induction and Naive Bayes. Further investigation is done to prove the previous statement.

Looking at the ROC curve between the algorithms in figure 11 we can see that Rule Induction jumped straight to the top left corner. This indicates that Rule Induction only predicted one outcome for all emails. It seems that Rule Induction did not correctly classify malicious emails. Upon further investigation we look at the confusion matrix for Rule Induction.

Looking at the confusion matrix of rule induction in figure 12 the class recall is 100% for actual value "false" yet the algorithm only achieved 96.2% prediction rate for value "false", and the algorithm did not predict true at all. We suspect that the algorithm split the data up in such a way that the validation set had no fields to predict true, or that the algorithm was not suitable to predict malicious emails correctly.

Comparing the ROC curve of Naive Bayes in figure 11, the curve for Naive Bayes algorithm looks more realistic. The confusion matrix for Naive Bayes in figure 13 also has more realistic prediction accuracies of 99.7% and 97.01% for value "false" and "true" respectively. This may indicate that the selected attributes from the training data may have not stated clear enough rules for rule induction and resulted a bias towards false prediction, causing the rule induction operator to be very inaccurate.

The results show that Naive Bayes has more realistic performance measures compared to Rule Induction. Therefore Naive Bayes may be more suitable for training on email header data. Compared to Decision tree it seems like decision tree has higher accuracy and therefore may be the best performing algorithm.

# 8 Cyber Criminal Profiling

The machine learning models generated above, assisted in creating various cyber criminal profiles.

- Profile A

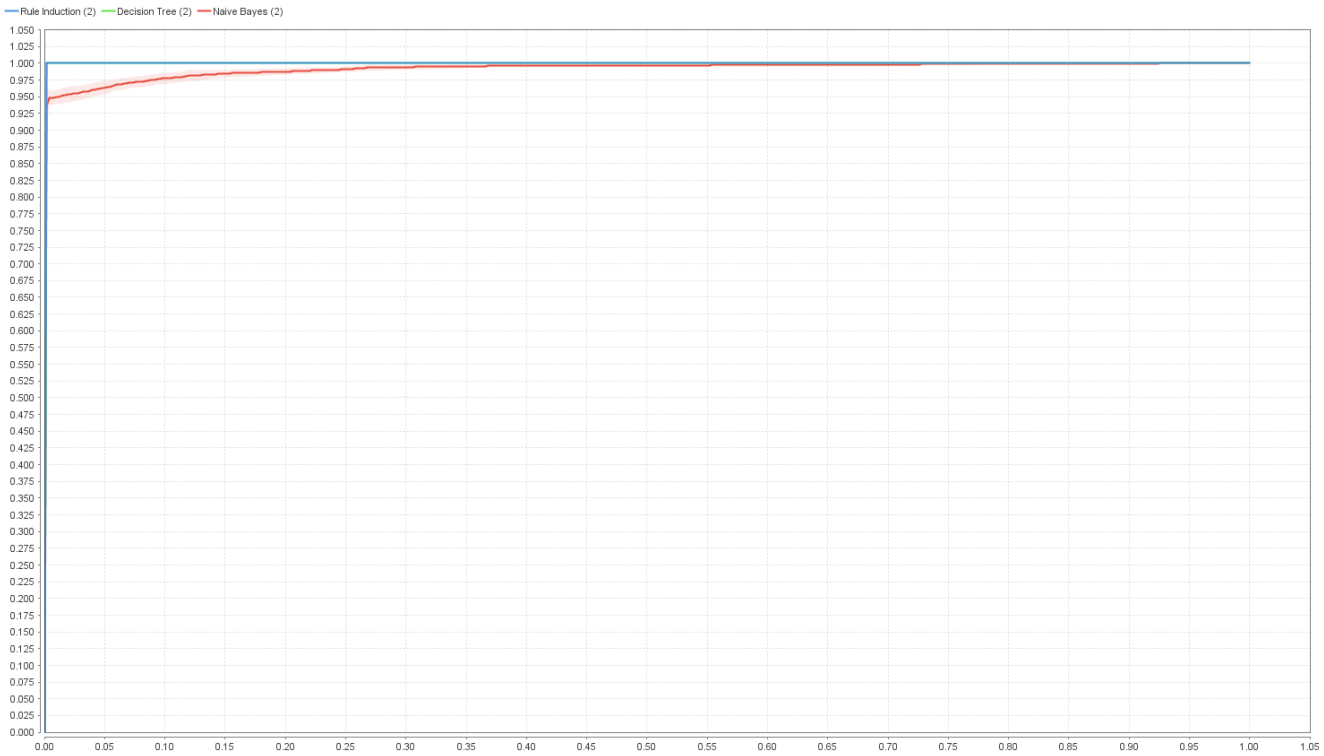


Figure 11: ROC graph comparing the algorithms

**accuracy: 96.20%**

| | true False | true True | class precision |
|---|---|---|---|
| pred. False | 69720 | 2757 | 96.20% |
| pred. True | 0 | 0 | 0.00% |
| class recall | 100.00% | 0.00% | |

Figure 12: Rule induction confusion matrix

  - This criminal uses an email server with an IP-address that is blacklisted and was immediately detected.

- Profile B
  - This criminal was not detected using an email server with an IP-address that is blacklisted.
  - However the Subject header was detected as a possible spam-subject.

- Profile C
  - This criminal was not detected using an email server with an IP-address that is blacklisted.
  - The Criminal's Subject line was not picked up as a spam-subject.
  - The Criminal sends the email to a large number of recipients in Bcc, Cc and To fields.

The differing Profiles indicate that there could be a difference in experience from one cyber criminal to another. With some criminals more experience and able to cover their tracks, or their targets being easier to convince.[14]

# 9 Anonymization

The following steps were followed in order to anonymize the email databset:

accuracy: 99.64%

| | true False | true True | class precision |
|---|---|---|---|
| pred. False | 142124 | 371 | 99.74% |
| pred. True | 162 | 5256 | 97.01% |
| class recall | 99.89% | 93.41% | |

Figure 13: Naive Bayes confusion matrix

- The process of anonymization starts with removing the email body in an attempt to reduce the chances of personal information leaked that may be contained in an email body.
- Thereafter any 6 to 16 digit numbers in the subject line are masked as ten 0's (0000000000).
- The account part in all the email addresses are hashed using the MD5 hashing algorithm. The reason for hashing the account part of the email addresses is to keep all masked email addresses the same to facilitate cross matching. The domain part of each email is kept as is to facilitate comparing the emails to known malicious email addresses.
- Any extra strings in the X-headers are also hashed as they may contain the full name of the owners of their respective email addresses. This is to protect the identities of the owners of each email address.
- Additional processing is done to put all email lists inline instead of having each email address in a list on it's own line. This is done for easier processing later on.
- The anonymized emails are saved to another folder after they have been anonymized.

# References


[1] Daniel Castro. How to stop the billions wasted annually on email spam, July 2013.

[2] Omar Al-Jarrah, Ismail Khater, and Basheer Al-Duwairi. Identifying potentially useful email header features for email spam filtering. In *The Sixth International Conference on Digital Society (ICDS)*, volume 30, page 140, 2012.

[3] Jessica Leber. The immortal life of the enron e-mails, July 2013.

[4] Carnegie Mellon University. Enron Email Dataset, May 2015.

[5] Internet Engineering Task Force. Rfc2822: Internet message format, April 2001.

[6] Nicole Martinez. What is a x-mailer header?

[7] Internet Engineering Task Force. Rfc2045: Multipurpose internet mail extensions (mime) part one: Format of internet message bodies, November 1996.

[8] Internet Engineering Task Force. Rfc822: Standard for the format of arpa internet text messages, August 1982.

[9] Internet Engineering Task Force. Rfc6648: Deprecating the "x-" prefix and similar constructs in application protocols, June 2012.

[10] Victoria Cox. Exploratory data analysis. In *Translating Statistics to Make Decisions*, pages 47–74. Springer, 2017.

[11] Hem Karlapalem. Threat hunting through email headers, August 2017.

[12] Jason Faulkner. What can you find in an email header?, September 2016.

[13] Omar Al-jarrah, Ismail Khater, and Basheer Al-duwairi. Identifying Potentially Useful Email Header Features for Email Spam Filtering. *The Sixth International Conference on Digital Society*, 2012.

[14] Dr Jason R. C. Nurse Dr Maria Bada. Profiling the cybercriminal, April 2016.