UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA
Denkleiers • Leading Minds • Dikgopolo tša Dihlalefi

COS700
Research Report

An Active Mode Content Based Stylometric Detection Framework for The Mitigation of Phishing Attacks

Student number: 14163285

Supervisor(s):
Prof Hein S Venter
Dr. Adeyemi R Ikuesan

An Active Mode Content Based Stylometric Detection Framework for The Mitigation of Phishing Attacks

Abstract

Public Cloud Computing is used to deliver applications through the internet as a service to the consumers. Public cloud resources from Cloud Providers such as Googles services are open to using to the general public. Therefore the Public Cloud is susceptible to cybercrime attacks, particularly Phishing attacks. Spear Phishing attacks in the cloud, are targeted attacks whereby targeted end-users release sensitive information to the attacker, through the attackers use of impersonation of the trusted organization. This attack has a higher success rate than the other phishing techniques. This research paper, therefore, describes a Stylometric Detection Framework, which is derived from an Anti-Spear Phishing Content-based Authorship Identification (ASCAI) framework, to detect and block mismatched emails from a received email body and the studied stylometric style of the original author who the account in the public cloud belongs to. The exploration into the Summary of Current Solutions channels the methodology to be followed in this research. The previous ASCAI solution was highly susceptible to false positives due to the limited implementation of the data extraction and the classification accuracy of 87% which could be improved. The use of supervised learning techniques is used to build a model that focuses on the behavioral biometrics of the content, in particular, the stylometrics, similar to the ASCAI module. This is then integrated into a software component that can be used to block emails if the Stylometric model detects an anomaly in the email body. Ultimately, the proposed approach will lead to the decrease in spear phishing attacks and increase the detection rate of spear phishing attacks in the public cloud

Keywords:

Stylometrics, Public Cloud Computing, Phishing Attack, Email, Detection, Authorship Identification, Machine Learning

1 Introduction

1.1 Cloud Computing

Computing and technology continuously advance in the Information Technology industry to improve processes that assist in meeting business and consumer objectives and improving consumer satisfaction. However, innovated resources do tend to have higher upfront costs. Therefore, not many businesses or consumers can afford to purchase these resources. In addition, with the emergence of the internet in this generation, the expense of storage as well as the consumption of power by computing components has increased [DKCE+16]. Thus, research into the idea of utilizing resources from an innovative and powerful vacant computer remotely immerged. This concept was referred to as Cloud Computing.

Cloud Computing according to the National Institute of Standards and Technology (NIST) is a computing model that exists for the enablement of ”ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction” [MG11]. In hindsight, cloud computing can be viewed as applications that are delivered through the internet as a service to the consumer as well as the reference to hardware and software of data centers that supply these services [ASZ+10]. The services are categorized into the following different models according to NIST [MG11]:

- Software as a service (SAAS), is applied through either a program interface or a web browser to use the supplier’s applications that are running on the infrastructure of the cloud.
- Platform as a Service (PaaS), is applied where the consumer deploys a consumer-created application that is supported by the supplier, whereby the user has control of application configuration
- Infrastructure as a Service (IaaS), is applied when the consumer is equipped with deployment and running of arbitrary software and can, therefore, control the deployed software and storage, in addition, can also have limit control to network components.

For the purpose of this research, we will look at Software as a Service in closer detail.

Cloud computing model is subdivided into 4 deployed infrastructures:

- Public Cloud, exists in the premises of a cloud provider which is open to use of the general public.
- Private Cloud, can exist in either the premises of an exclusive organization or off on a third party organization, which is used exclusively by a single organization or jointly with a third party organization. As long as it's not open to use by the public.

- Community Cloud, is utilized by organizations that share common interests, security or concerns. The environment exists either on the premise of either of the organizations or a third party organization rendering the cloud services to this community of organizations.

- Hybrid Cloud, is the combination of either of the previously stated deployed infrastructures distinctively.

The public cloud is of particular interest to the purpose of this paper, especially when looking into the commercial service platform that exists in the public cloud such as Google, Amazon, and Microsoft. The public cloud service infrastructure has rapidly grown, therefore has also brought forth an increase in unavoidable security issues. In essence, cloud security is a vital vulnerability that is of great concern, and thus even today we still find many organizations, that have not utilized the available cloud services. According to Gartner, a researcher that conducted an investigation around information security issues that are considered in cloud computing listed that privileged access is one of the issues highlighted that public cloud vendors need to consider [SH11]. For the end consumer, sensitive data should only be available to them, however, cloud vendor administrators should have access to a cloud environment for maintainability and security of the environment. Thus sensitive data may also need to be available to administrators.

1.2 Impersonation

To enable the availability to end user, Cloud computing service needs to provide an identity management. This is, however, a great concern in the public cloud environment as the cloud providers identity management are not consistently integrating their platforms with identity services [Bro5]. This is also due to the constant evolution of SaaS. This opens to a renowned cybercrime attack that is foreseeable in exposing the vulnerability. This crime is known as Phishing. Phishing is executed by someone impersonating as a legitimate and reputable organization or individual targeting any individual or organization to providing sensitive data [ASRS16]. Therefore, attackers performing Phishing attacks usually perform them through email, where they prompt target into giving sensitive information. Attackers also target public cloud infrastructure through PaaS where they inject links that users using Cloud platform as a SaaS would click the link that would redirect the user to a fraudulent web page.

Anti-phishing tools have been developed and utilized by public cloud providers in an attempt to prevent Phishing attacks. Tools like Envelope content splitting (ECS) tool is used to authenticate the attacker sending a link to a fraudulent website, and so protects the recipient from phishing [ASRS16]. There exist tools that also detect Phishing attacks, such as a Phishing tank whereby a database containing Phishing sites are used by other anti-phishing tools to detect phishing. The literature review will focus on some of the anti-phishing tools that were researched for the purpose of detecting Phishing, particularly in the Public Cloud. This will be followed by the identifying the gap in the detection of Phishing attacks that this research paper will address. Authorship detection is the proposed solution based on a Stylometric analysis approach that this paper will investigate.

1.3 Behavioural Biometrics

Behavioral biometrics is the study of users behavior [SSP18]. The type of behavioral biometry that is of interest in this paper is stylometry, formally known as Authorship Analysis, as it is the study of users linguistic style of writing by extracting and analyzing stylometric features. Stylometry has been applied in different frameworks in Cloud platforms with the aid of detecting a phishing attack through attribute authorship, and hence this paper will look to exploiting this feature. Stylometric analysis can be applied to three perspectives, namely stylometric identification, verification, and characterization. In relation to this research, the focus will be the application of stylometric verification, which in hindsight is determining the target body of work belongs to the author that claims the authorship of the body of work[BTSW13].

2 Problem Statement

Public cloud environments are susceptible to Spear phishing attacks as it is thought to be the easiest ways for criminals to acquire sensitive information from the end-user as the end-user is seen as the weak point in the system. It takes approximately over 229 days to detect an intrusion with the latest detection models employed in Cloud environments [VEE$^+$15]. This is due to the attackertextscfls knowledge of impersonating as a legitimate source by mimicking the style of writing format of the organization. Thus, people who have accounts that exist in the public cloud could find themselves, unsuspectedly, giving out their personal information to the perpetrator quicker than the current spear phishing detectors that could detect the anomalies that exist in the communication between the victim and the perpetrator. Although there are quicker solutions, such as Email Header Analysis, these solutions require very little information and therefore, the results are not highly accurate to detect a possible Spear Phishing attack [CLJ$^+$17].

3 Literature Study

A paper was done by Nazmul Islam, Mohammed Moshiul Hoque and Mohammad Rajib Hossain on the ”Automatic Authorship Detection from Bengali Text using Stylometric Approach” [IHH17] investigates writing styles of Bengali writters by collecting writings and blogs from their sample writers. Through analysis, they discovered n-gram features that were useful to detect certain authors. N-grams, also reffered to as shingles, is an adjacent sequence of n items from a given sample of text [BGMZ97]. Thus can be viewed as a probabilistic model in a language base that predicts the next item in a sequence derived from the Markov model [Bri95]. The investigation used unigram, bigram as well as trigram together with parts of speech features such as conjunctions and pronouns on the sample sets of texts provided. Then the use of three machine learning algorithms on the final dataset, namely Naive Bayes, Decision Tree and Random Forest Classifier, were used, with the explanation of how each step in the Random Forest was conducted and the results that got passed into a new document. This is due to the 96% accuracy they received from their research in the Random Forest Classifier. The strength of this research was dataset was reliable was they had selected texts randomly to minimize biasness of the data. It also reached a very high accuracy of 96% from the 3125 literary passages of a sample of 10 prominent writers. However, the approach has a limitation in that was narrowly focussed on blog writings and so proposed method has still need to address writings found in news articles, emails, tweets and other texts found in public cloud environments that are of interests to this research paper.

Rakesh Verma and Nirmala Rai proposed the ”Phish-IDetector: Message-Id Based Automatic Phishing Detection” [VR15] which primarily focuses on email headers. The research focussed on observing less than 10 legitimate emails and phishing emails as the research were more drawn to the Message-ID field that is a universally unique string. The research utilises Machine Learning algorithms together with the properties of Message-IDs onto the n-gram analysis of the Message-IDs. The Random Forest Classifier algorithm performed the best in the research and so results found were utilised with the SMO algorithm (The Sequential Minimal Optimization). The researchers ensured that 100% of the experimental dataset legitimate emails had Message-IDs. This experiment relies on the Message-ID field and so if not existing the experiment will not work, fortunately, it will still raise red flags to the email security. The research reached 99% True Positive Rates. It is important to note that in literature, there is no phishing detection of 100%, therefore the result of 99% was due to the smaller finite sample set of data, and is indeed noted that the exponential increasing of the sample email set together with a higher order of n-grams is difficult to run with different Machine Learning classifiers without the specialized use of big data approaches. This, therefore, shows a limit in the automatic detection system.

Following from this research, Rakesh Verma and Ayman El Aassal state its better to include users in the detection process or send warning to users to warn them of possible attack and thus include user training, when they were introducing ”A Correlation-based Analysis and User Participation method for Detecting Phishing Email” [VEE^{+}15]. In this research, a comprehensive method to determine to phish off an email was designed, whereby information extracted from the email header is relevant to the information contained in the email body. This introduced method was executed using two algorithms, namely Header Analysis Algorithm and Matching Algorithm. Within the Matching Algorithm, multiple header fields and sender identity are authenticated using digital signatures in the Domain-Key Identified Mail [Her09]. A sender Policy Framework is imployed to enable verification that the sending mail server is authorized in the domain that appears in the ”mail from” address. Other methods like URL Analysis and Semantic Analysis are also used in the research. The research primarily focuses on the header analysis and so if the results pass the rules set for the header, then the email is passed off as legitimate. This deduction, however, is not legitimate as the attacker can still get hold of a legitimate email address by hacking into the organisationtextscfls mail server and impersonate as a member of the organisation. Hence future work involves the analysis of email body.

A paper on a Content-Based Authorship Identification Framework that is used for the detection of spear phishing [KIJ11]. The paper introduces a novel framework called the Anti-Spear phishing Content-based Authorship Identification which analyses the message body of the message sender without relying on the sender's ID. Its important to note that this is used specifically for Spear Phishing attacks. Bulk Phishing attacks are generic and target many users, and so many cloud providers contain several software classifiers that have the ability to detect generic nature of bulk phishing attacks. Spear Phishing attacks are targeted and thus difficult to detect to detect due to unique nature. The paper highlights a key point in its motivation into using Content based, that is User ID-based authentication is not helpful in detection as Users read the text in parallel and not sequentially, thus can fall into typo-squatting and cousin-naming. Users also introduce weak authentication in systems, particularly in cloud infrastructures whereby they set very weak passwords as its easier to remember for them but also easy for a brute force algorithm to hack. In addition, users passwords can be stolen through Keyloggers, and so attackers can impersonate as the user in the system if they have the users credentials. The research aims to provide a software framework, that was otherwise not introduced in the previously mentioned literature. The research also takes a whitelist approach based off of email mining techniques constructed from emails sender stylometric profiles. Thus a unique approach compared to other literature that was a derivation of the black-list approach. 2 proposed modes to calculate similarities between the claimed and the predicted identities are namely the passive and active mode. In the passive mode, both identities are presented to the end-user. In the active mode, in a scenario where a mismatch between identities is found, the content of the message is blocked to the end-user with a warning of a mismatch occurred. The issue with this research is false positives can result if multiple users contribute to an email which the writeprints can be altered from original senders writeprint. Active mode was not explored in this paper as it increases implementation complexity and Software might not have been as accurate as the end-user. The paper followed the Security Content Automation Protocol (SCAP) methodology [MMM+11] due to its high achievement of 100% of classification accuracy on datasets as it makes use of byte-level n-grams and hence useful for natural languages. For this research, a dataset of 289 emails from 12 authors was used. An Accuracy rate was used to measure performance as dataset was evaluated with the use of 10-fold cross-validation. The results found in the research show a maximum accuracy of 83% for a non-greedy n-gram ranking method and 87% maximum accuracy rate of 87% with a focus on n-gram ranking methods. This framework had setbacks and limitations such as the writeprint extraction whereby if no message was previously sent by the original author then the SCAP would attempt to map it to the closest author profile match, which is not desired.

3.1 Aim of this study

Based of the ASCAI framework that the research paper conducted [MMM+11], this papers objective is to implement the ASCAI framework in active mode to develop a white-list of authorized authors stylometric identities. Furthermore, this white-list will be used during the authentication of the daily emails to detect identity impersonation. This is to compare which model provides the best results. Furthermore, this framework is developed to integrate mail directories from the public cloud. In particular, Googletextscfls Gmail service is used, as it is the most used email service in the world [XLM+13], is used to produce our dataset to see how the framework performs on real data. In addition, the software should be able to detect and inform in real time to address the issue that current solutions take more time to detect the anomaly.

4 Framework Design

The process workflow shown at the bottom gives an illustration on a high-level perspective of how the framework has been designed to function. It encompasses 7 Modules that are integrated with a Python Flask API. This API is used by the softwaretextscfls front-end Web API that facilitates the retrieval of credentials and displays the accuracy results. The 7 modules are:

I Data Acquisition

II Data Cleaning

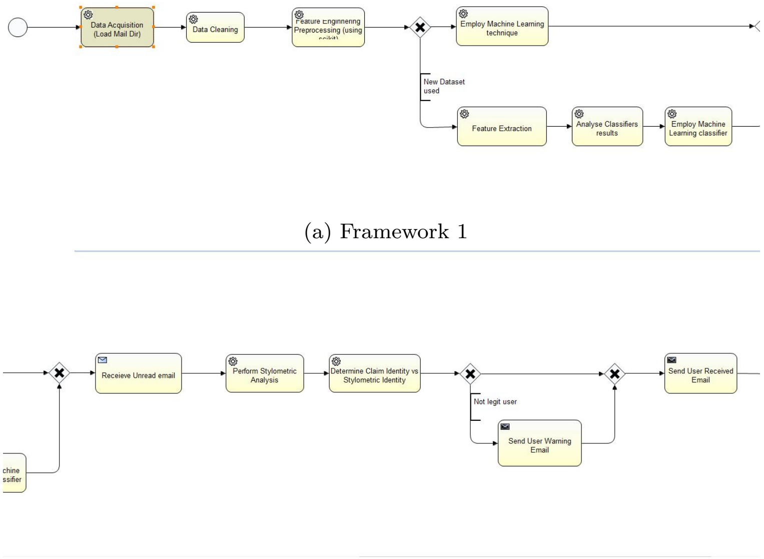
III Feature Engineering

IV Feature Analysis

V Classification

VI Claim vs Identity Checker

VII Warning Email Sender


Data Acquisition (Load Mail Dir)
Data Cleaning
Feature Enginnering Preprocessing (using scikit)
Employ Machine Learning technique
New Dataset used
Feature Extraction
Analyse Classifiers results
Employ Machine Learning classifier
Receieve Unread email
Perform Stylometric Analysis
Determine Claim Identity vs Stylometric Identity
Not legit user
Send User Warning Email
Send User Received Email
chine
ssifier

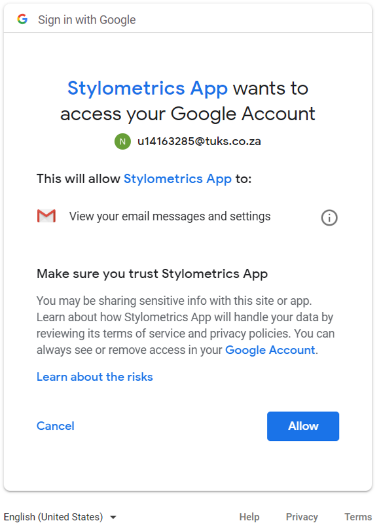
(a) Framework 1

(b) Framework 2

Figure 1: Framework Process Workflow

4.1 Data Acquisition

The framework design begins with the Data Acquisition Module. This is where the software retrieves the mail directory of the user from the public cloud. This software has been designed to interact with Googletextscfls Gmail API to acquire a usertextscfls inbox as their mail directory. The mail directory retrieved will be used as a training dataset for the classifiers in the Feature Analysis module. This Module firstly retrieves the username and password credentials from the user who entered these values on the software front-end GUI when they were prompted to log in upon opening the software. The Data Acquisition Module will the trigger the Google API to open a page whereby the user approves permissions for the software to retrieve, read and modify their email, as shown below:



Sign in with Google

Stylometrics App wants to access your Google Account

u14163285@tuks.co.za

This will allow Stylometrics App to:

View your email messages and settings

Make sure you trust Stylometrics App

You may be sharing sensitive info with this site or app. Learn about how Stylometrics App will handle your data by reviewing its terms of service and privacy policies. You can always see or remove access in your Google Account.

Figure 2: Google permisions

Once they have approved the permit to gain access, the software then requests to the Google API to retrieve the mail Inbox. Once retrieved, the module saves the result on to a CSV file. This will be shared with the integration module, for other modules to access this CSV file, this would ensure the performance of the system is optimum as the modules would not have to request the Google API to retrieve the mail directory for their prescribed use.

The number of emails retrieved is dependent on the account billing status. If the account is a free account, then the API only allows 100 emails to be retrieved. Thus to build the dataset, the module does check if the email has already been saved. Therefore, duplicates are not existent on the software that could potentially skew results.

4.2 Data Cleaning

In the Data Cleaning module, the CSV file, created by the Data Acquisition module, is extracted into an array of objects, which each object is an author with all the email messages the author has sent. During this process, the author is set as the sender from the header, the other additional fields in the headers are stripped away. Blank email bodies are ignored. This sanitized consolidated dataset is returned to the integration module which will feed it off to the feature engineering module.

4.3 Feature Engineering

n the Feature Engineering Process, Preprocessing of the data provided occurs to create Datasets. The preprocessing procedure evaluated the number of features that would be used for our supervised machine learning technique found in the feature analysis. The feature identified were:

- Average Number of characters in emails
- Average Number of words in emails
- Average Word lengths in emails
- Average Number of sentences in emails
- The ratio of short words in emails
- Email Sentiment Polarity
- Average Noun Phrase
- Numeric Ratio Density
- Average number of paragraphs in emails
- The frequency of punctuation marks

The dataset that results from these features extracted is saved onto a CSV file for the feature analysis module to use.

4.4 Feature Analysis

The datasets created from the feature engineering module are loaded onto the Feature Analysis module. Thereafter, two classifiers are constructed, namely the Decision Tree Classifiers and the NaÃŕve Bayes Classifiers from the Text Blob Library. The Text Blob library contains the necessary Natural Language Processing Toolkit to process the Content compiled [Lor18]. The N-Gram analysis is also done using features created from the feature engineering. The dataset sent in for Machine Learning is derived from the dataset created by the Data Cleaning Module. A new dataset is created me through each of the emails, assigning them on a list to the author they state to be assigned to. However, the last email of each author is used in the test dataset to determine the accuracy of the classifier. The accuracy of each of the classifiers is returned to the Integration Module once completed.

4.5 Classification

The Classification module uses the most accurate classifier detected and applies it to any new email. This Classification module originally was designed to be a listener event for any new unread email. But since sending requests to the Google Gmail API, to return any new unread email is computationally expensive, the module was redesigned to be triggered by the Integration Module. This is because the Integration Module links with the software's Front-End

Interface, which contains a button titled ”Check unread email” that alerts the Integration Module to trigger the Classification Module. Once triggered, the module requests to the API and receives, at max, two of the latest unread emails. The restriction of two emails was set for testing purposes but ideally, up to 100 emails could be retrieved. The email received is sanitized by extracting the email author through the email header and extracting the email body contents. The email author is stored for later use by the Claimed vs Identity Checker. The email contents are sent into the classifier and return’s the predicted author.

4.6 Claimed vs Identity Checker

The Claimed vs Identified Checker looks at the resultant identified author and compares with the claimed identity found on the email header as the sender. If there exists an anomaly whereby the authors do not match, the module checks if the claimed author identity exists as an author in the mail directory. If this author does exist it, therefore, indicates that it’s a spoofed email and therefore the user should be informed immediately about the anomaly. Thus, the Warning Email Sender module is triggered by notifying the integration module to alert the Warning Email Sender event handler to trigger to notify the user.
Note that if the author does not exist in the mail directory then the Warning Email Sender is not triggered.

4.7 Warning Email Sender

To fulfill the Active Mode approach, this module sends a warning email to the user, notifying them that an anomaly exists from the latest unread email the user had received. This module retrieves both the Claimed and the identified email and therefore explicitly explains to the user that the system identified the detected author and does not coincide with the claimed identity. This warning email is sent through the Google Gmail API. The credentials and session token are retrieved through the token.json and credentials.json stored on the system.

5 Implementation Challenges

The emails extracted from the mail directory are prone to adjustments made by services, such as embedded HTML and thus accurate extractions of emails are limited. The other most notable challenge was normalizing the email dataset to an equal number of emails per users. As we aim to predict accurately, the training set needs to have an equal number of training data mapping to an author, however there is no guarantee that an author has emailed the user logged in over a certain threshold, thus the classifier is prone to be training in a biased manner to the emails passed the threshold. Catering for forwarded emails is a limitation that exists. This could be resolved by looking at the message body and checking if the email contains the word ”Forwarded”, and if the word exists then the system should write it off, but this raises two defects:

1. Normal emails that were not forwarded but did contain the word in the text as part of the conversations would be written off. This won’t affect the training of data only if there existed more emails belonging to the author

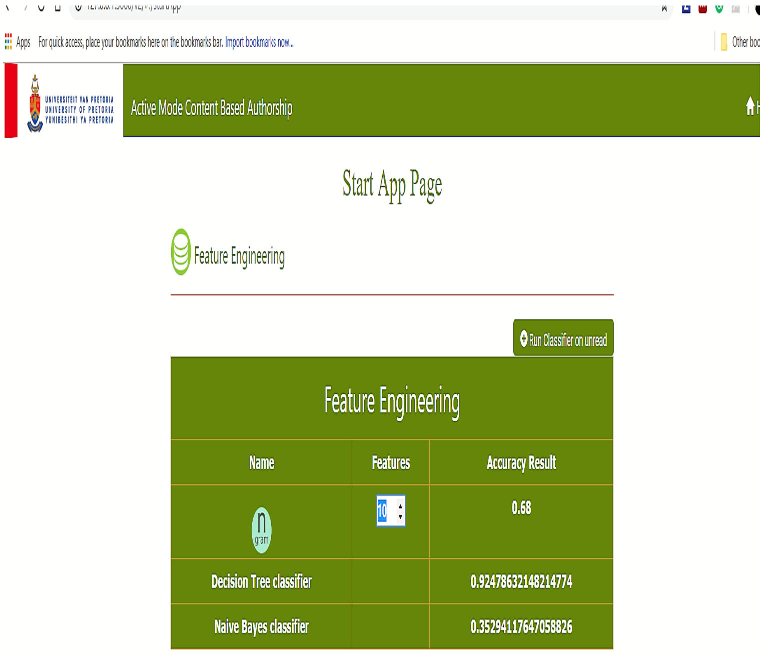
that may assist in training the classifiers. However, if there exists very few emails belonging to the author, the accuracy in stylometric detection related to the author might be limited and overpowered by the writeprints of the other authors that had more mail in the directory when training.

2. For unread emails, if the perpetrator knows about this vulnerability, they could use the word in every email communication they send, and so the perpetrator could never be detected as their email would be written off

Thus, based on defects, the system does not include a write-off functionality, which warrants the possibility that the forwarded email could be used in the training set, that affects the accuracy. The challenge that exists in the data acquisition module, whereby the number of emails retrieved were limited to 100 emails. Therefore, to grow the dataset, the user has marked the emails in the inbox, that we want to train, with the label UNREAD. The software in the Data Acquisition Module requested for all the emails in the inbox that were unread. Once completed the Integration Module posts an update through the Google Gmail API to remove the UNREAD label off the emails. That way, when the software is reloaded, the Data Acquisition Module will make the same call, but then a new set of unread emails are returned and added to our existing mail directory. Thus, this brings an extensive amount of overhead, which affects the performance.

6 Evaluation Report of Active Mode Software Solution

To evaluate the system, a test case was derived to evaluate the accuracy received and when an anomaly is detected, an email is sent with the right information. To begin, the account that had existed on Gmail since 2014 was used to test the software. This account was chosen because it had a decent amount of emails that existent in the inbox of the mail directory. Thus, giving a variety of authors that have sent an email. This is important because as the tester it would have been more challenging to compose different emails. When the software was launched, and the user logged in, the data acquisition extracted a dataset consisting of 115 emails and 17 Authors from the user's email inbox. Following the workflow process to the point where the Data Analysis module occurred, the following results were achieved:



Apps
For quick access, place your bookmarks here on the bookmarks bar. Import bookmarks now...
Other boo
UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA
Active Mode Content Based Authorship
Run Classifier on unread

Start App Page

Feature Engineering

Feature Engineering

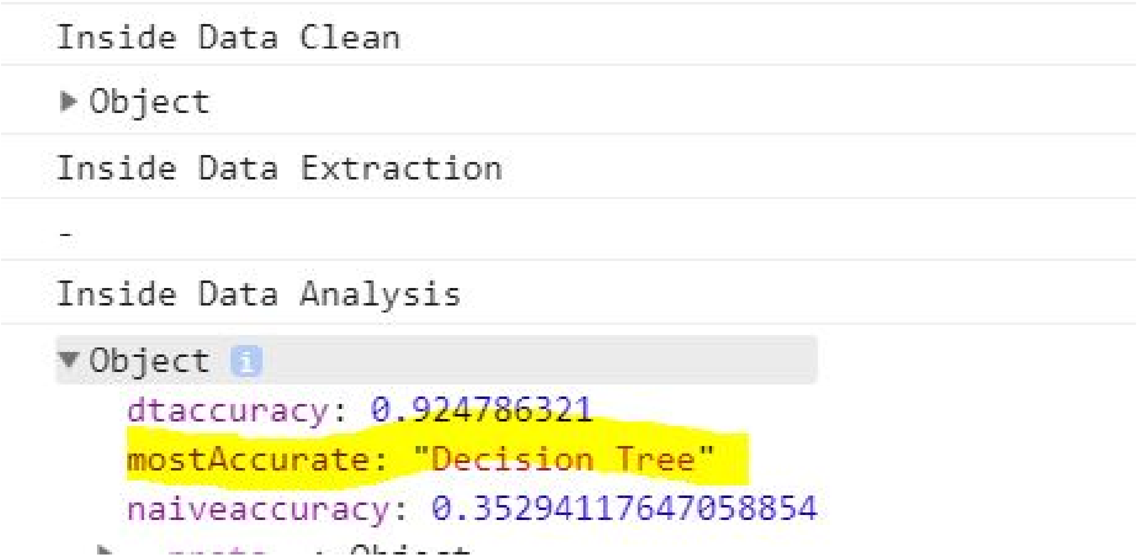
Name	Features	Accuracy Result
n gram	10	0.68
Decision Tree classifier		0.92478632148214774
Naive Bayes classifier		0.35294117647058826

Figure 3: Framework Process Flow Part A.

The N-gram classifier utilized a 10-fold cross validation, whereby just like in the ASCAI paper, the dataset sent in were split into 10 equal folds, thereafter the software took 9 folds and trained the classifiers, thereafter tested with 1 fold. This brought the best accuracy in the n-gram ranking of 68

The Decision tree classifier, which is facilitated by the TextBlob library used a dataset that consisted of the author and the email message on each element in the list. This dataset was used to train the classifier. Only the last email of each author was put in the dataset used to test the accuracy of the classifier. From the result, we can find that the Decision Tree Classifier produced an accuracy of 92

The NaÃŕve Bayes Classifier follows similar steps as the Decision Tree Classifier, this classifier produced an accuracy of 35%. The data Analysis module does send back the most accurate classifier to use for the unread email. This can be justified by looking at the console.



```
Inside Data Clean
▶ Object
Inside Data Extraction
-
Inside Data Analysis
▼ Object
    dtaccuracy: 0.924786321
    mostAccurate: "Decision Tree"
    naiveaccuracy: 0.35294117647058854
  ▶ __proto__: Object
```

Figure 4: Most Accurate Classifier in Integration Model.

The Decision Tree was sent back so that the Integration Module can send this value to the Classification module when the module is triggered.

To test the detection on the classification, an email by Neo Thokoa is sent to the test user. To ensure the result is an anomaly, an email written by Martijn from data camp was copied and sent as the sender been Neo Thokoa. Martijn exists in the test mail directory and thus its expected that an anomaly is to be detected.

Once the Classification Module was triggered, the result in the following figure shows that there exists an anomaly. In addition, a flag is also sent with the results that shows the author does exist.

The evaluation was repeated to evaluate consistency. In all iterations, the same constant value was retrieved from all respective classifiers.

Finally, the time taken to analyze the unread email was evaluated. In all iterations, it took approximately 4.47 seconds to retrieve a response from the module. Thus, it stands a good chance on mitigating a spear phishing attack, if the user is alerted before they can have a chance to open the email.

7 Result Issues

The Naive Bayes accuracy result was low due to the large difference in a number of emails that each author had in the dataset. Over time the larger the dataset grows results in a higher accuracy for all three classifiers.

8 Future Works

In future, the research will include a process to conduct on how users respond to a warning message received. This experiment will help us better understand the behaviour of users when they are notified of a possible threat. This way we get to see if it is possible to Mitigate Spear Phishing attacks through the users, or we should rather look at ways to prevent the possible attack before it gets through to the user. Our sample for this experiment will include professionals from universities, students as well as parents.

We will also look at using the Natural Language Processing Tool Kit (NTLK) library, as its the core library that offers more classifiers, particularly Deep learning algorithms, such that it can improve our accuracy as well as uses less time in the process.

Lastly, we would like to expand to other public cloud platforms, such as Amazon, and employ the framework to illustrate its use in the industry.

9 Conclusion

The use of Active Mode Content Based Authorship framework, that is a derived framework from the ASCAI) framework, on the public cloud mail directory, has shown good promise based off of the high accuracy it achieves, to mitigate Spear Phishing.

The use of Decision Tree, showcases why it is the leading classifier in the digital age of Machine Learning.

References

[ASRS16] R. Aravindhan, R. Shanmugalakshmi, K. Ramya, and C. Selvan. Certain investigation on web application security: Phishing detection and phishing target discovery. *ICACCS 2016 - 3rd International Conference on Advanced Computing and Communication*

Systems: Bringing to the Table, Futuristic Technologies from Arround the Globe, 2016.

[ASZ+10] Michael Armbrust, Ion Stoica, Matei Zaharia, Armando Fox, Rean Griffith, Anthony D. Joseph, Randy Katz, Andy Konwinski, Gunho Lee, David Patterson, and Ariel Rabkin. A view of cloud computing. *Communications of the ACM*, 53(4):50, 2010.

[BGMZ97] Andrei Broder, Steven C Glassman, Mark S Manasse, and Geoffrey Zweig. Syntactic clustering of the Web IS (A) n S (B). 29, 1997.

[Bri95] Eric Brill. Transformation-Based Error-Driven Learning and Natural Language Processing : A Case Study in Part-of-Speech Tagging. *Computational Linguistics*, 21(4):543–565, 1995.

[Bro5] Jon Brodkin. problems with saas security. *Network World*, 27(18):1–27, 5.

[BTSW13] Marcelo Luiz Brocardo, Issa Traore, Sherif Saad, and Isaac Woungang. Authorship verification for short messages using stylometry. In *Computer, Information and Telecommunication Systems (CITS), 2013 International Conference on*, pages 1–6. IEEE, 2013.

[CLJ+17] Changhee Choi, Hwaseong Lee, Ilhoon Jung, Changon Yoo, and Hosang Yun. EMAIL HEADER ANALYSIS FOR AUTHOR IDENTIFICATION. pages 1–4, 2017.

[DKCE+16] Sevtap Duman, Kubra Kalkan-Cakmakci, Manuel Egele, William Robertson, and Engin Kirda. EmailProfiler: Spearphishing Filtering with Header and Stylometric Features of Emails. *Proceedings - International Computer Software and Applications Conference*, 1:408–416, 2016.

[Her09] Amir Herzberg. DNS-based email sender authentication mechanisms: A critical review. *Computers and Security*, 28(8):731–742, 2009.

[IHH17] Nazmul Islam, Mohammed Moshiul Hoque, and Mohammad Rajib Hossain. Automatic authorship detection from Bengali text using stylometric approach. *2017 20th International Conference of Computer and Information Technology (ICCIT)*, pages 1–6, 2017.

[KIJ11] Mahmoud Khonji, Youssef Iraqi, and Andrew Jones. Mitigation of spear phishing attacks: A content-based authorship identification framework. *2011 International Conference for Internet Technology and Secured Transactions, ICITST 2011*, (December):416–421, 2011.

[Lor18] Steven Loria. textblob Documentation, 2018.

[MG11] Peter Mell and Timothy Grance. The NIST Definition of Cloud Computing Recommendations of the National Institute of Standards and Technology. *National Institute of Standards and Technology, Information Technology Laboratory*, 145:7, 2011.

[MMM$^+$11] Isla S. Mackenzie, Brian J. Mantay, Patrick G. Mcdonnell, Li Wei, and Thomas M. Macdonald. Managing security and privacy concerns over data storage in healthcare research. *Pharmacoepidemiology and Drug Safety*, 20(8):885–893, 2011.

[SH11] Farhan Bashir Fb Shaikh and Sajjad Haider. Security threats in cloud computing. *2011 International Conference for Internet Technology and Secured Transactions*, (December):214–219, 2011.

[SSP18] Howard Shrobe, David L Shrier, and Alex Pentland. *New Solutions for Cybersecurity*. MIT Press, 1 edition, 2018.

[VEE$^+$15] Rakesh Verma, Ayman El, Aassal Ensias, Avenue Mohamed, Ben Abdellah, and Regragui Rabat. Comprehensive Method for Detecting Phishing Emails Using Correlation-based Analysis and User Participation. pages 155–157, 2015.

[VR15] R Verma and N Rai. Phish-IDetector: Message-ID based automatic phishing detection. *2015 12th International Joint Conference on e-Business and Telecommunications (ICETE)*, 04:427–434, 2015.

[XLM$^+$13] Fengyuan Xu, Yunxin Liu, Thomas Moscibroda, Ranveer Chandra, Long Jin, Yongguang Zhang, and Qun Li. Optimizing Background Email Sync on Smartphones. *Proc. of MobiSys*, page 55, 2013.