DETECTING TARGETED MALICIOUS EMAIL THROUGH SUPERVISED CLASSIFICATION OF PERSISTENT THREAT AND RECIPIENT ORIENTED FEATURES

by Rohan Mahesh Amin

B.S. Computer and Telecommunications Engineering, University of Pennsylvania, 2002
M.S. Telecommunications and Networking, University of Pennsylvania, 2002

A Dissertation submitted to

The Faculty of
The School of Engineering and Applied Sciences
of The George Washington University
in partial fulfillment of the requirements
for the degree of Doctor of Philosophy.

January 31, 2011

Dissertation directed by

Julie J.C.H. Ryan
Associate Professor and Chair of Engineering Management and Systems Engineering

ProQuest LLC
789 East Eisenhower Parkway
P.O. Box 1346
Ann Arbor, MI 48106-1346

The School of Engineering and Applied Science of The George Washington University certifies that Rohan Mahesh Amin has passed the Final Examination for the degree of Doctor of Philosophy as of September 24, 2010. This is the final and approved form of the dissertation.

DETECTING TARGETED MALICIOUS EMAIL THROUGH SUPERVISED CLASSIFICATION OF PERSISTENT THREAT AND RECIPIENT ORIENTED FEATURES

Rohan Mahesh Amin

Dissertation Research Committee:

Julie J.C.H. Ryan, Associate Professor and Chair of Engineering Management and Systems Engineering, Dissertation Director

Enrique Campos-Nanez, Assistant Professor of Systems Engineering, Committee Member

Johan René van Dorp, Associate Professor of Engineering Management and Systems Engineering, Committee Member

Bhagirath Narahari, Professor of Computer Science, Committee Member

Gregory Rattray, Partner, Delta Risk, Committee Member

To Rishi

Acknowledgements

There are many people whose contributions were critical in helping me complete this dissertation. I'd like to first acknowledge my advisor Dr. Julie J.C.H. Ryan. She has been very patient with me over the past few years and her advice and counsel throughout this process have been much appreciated.

I also want to thank Anne Mullins and Tom Gordon, my bosses at my place of employment. Anne and Tom were very understanding of my personal goal; as the defense date neared my vacation hours increased and both of them were supportive enabling me to balance my work and personal commitments. I also want to thank Jim Connelly and Angeline Chen, who provided the approval for me to use the data in this dissertation. The data was absolutely central to the research and without it I would have not been able to conduct this study. I also want to extend a big thanks to several colleagues: Eric Hutchins, Mike Cloppert, Charles Smutz, Samuel Wenck, Michael Poddo and Mike Gordon. I am thankful for their collaboration and sanity checks whenever I needed reassurance. These individuals are the most talented information security professionals that I know and they do incredibly important work for their organization and for this nation. Thanks also goes to Elisabeth Frye who made sure I had multiple terabytes of storage available for all of the data used in this study.

Finally, and most importantly, a special thanks to my wife and the rest of my family for their unending support. There were many nights when I was mentally exhausted but my family encouraged me to press forward. My biggest thanks goes to my wife Sejal, who was expecting our first child during the bulk of my dissertation writing. She still managed to take care of absolutely everything on the home front, allowing me to stay focused on this study. I could not have completed this without her daily support, love and encouragement.

Abstract

Detecting Targeted Malicious Email through Supervised Classification of Persistent Threat and Recipient Oriented Features

Targeted email attacks to enable computer network exploitation have become more prevalent, more insidious, and more widely documented in recent years. Beyond nuisance spam or phishing designed to trick users into revealing personal information, targeted malicious email (TME) facilitates computer network exploitation and the gathering of sensitive information from targeted networks. These targeted email attacks are not singular unrelated events, instead they are coordinated and persistent attack campaigns that can span years. This dissertation surveys and categorizes existing email filtering techniques, proposes and implements new methods for detecting targeted malicious email and compares these newly developed techniques to traditional detection methods. Current research and commercial methods for detecting illegitimate email are limited to addressing Internet scale email abuse, such as spam, but not focused on addressing targeted malicious emails. Furthermore, conventional tools such as anti-virus are vulnerability focused examining only the binary code of an email but ignoring all relevant contextual metadata.

This study first documents the existence of TME and characterizes it as a form of malicious email attack different than spam, phishing and other conventional illegitimate email. The quantitative research is conducted by analyzing email data from a large Fortune 500 company that has been subjected to these targeted emails. Persistent threat features, such as threat actor locale and weaponization tools, along with recipient oriented features, such as reputation and role, are leveraged with supervised data classification algorithms to demonstrate new techniques for detection of targeted malicious email. The specific tools, techniques, procedures, and infrastructure that a threat actor uses characterize the level and capability of a threat; the recipient's role and repeated targeting speak to the intent of the threat. Both sets of features are used in a random forest classifier to separate targeted malicious email from non-targeted

malicious email. Performance of this data classifier is measured and compared to conventional email filtering techniques to demonstrate the added benefit of including these features. Performance evaluations are focused on false negative reduction since the cost of missing a targeted malicious email is far greater than the cost of mistakenly flagging a legitimate email as malicious.

Several findings are made in this study. First, targeted malicious email demonstrates association to persistent threat features as compared to non-targeted malicious email that does not. Second, targeted malicious email demonstrates association to recipient oriented features as compared to non-targeted malicious email that does not. Finally, detection of targeted malicious email using persistent threat and recipient oriented features results in significantly fewer false negatives than detection of targeted malicious email using conventional email filtering techniques. This improvement in false negative rates comes with acceptable false positive rates.

Future research can expand upon the features introduced in this study. For example, additional persistent threat features can be harvested from file level metadata (e.g. author names, document path locations) and additional recipient oriented features can be incorporated from organization databases. In this study, a binary outcome is defined: emails are either targeted malicious or non-targeted malicious. Future work can explore multi-class outcomes that pair specific threat actor campaigns and targeted recipients.

Table of Contents

List of Figures

List of Tables

Chapter 1: Introduction

Email has long been an Internet 'killer application' used by individuals, businesses, governments and other organizations for the purposes of communicating, sharing and distributing information. Basic email shares a common design flaw with postal mail: the sender is not necessarily authenticated and can be falsified in a message sent to a recipient. However, a fundamental difference is that there is a cost associated with sending postal mail while the marginal cost of sending magnitudes more email is virtually zero. Email has a near zero incremental cost when sending greater volumes of email resulting in illegitimate email dominating the vast majority of the Internet's email traffic (MAAWG, 2008).

There is a range of illegitimate email that corresponds to a spectrum of capabilities and intentions behind the threat actors responsible for sending those emails. Certain actors, such as those associated with spam, will use email to send mass unsolicited advertisements to persuade individuals to purchase products that will generate revenue. Other actors, such as those behind phishing, will use email as a means to steal personal information and co-opt a victim's identity. This identity theft allows an actor to open a line of credit, make a string of purchases or empty a bank account. Still other actors will use email as a vehicle to gain access to computer networks with sensitive information or to disrupt computer network operations.

Since as early as email made its Internet debut, so did an entire industry that has been focused on methods to detect and prevent illegitimate email from arriving into a user's inbox. There is a wealth of literature in the area of detecting unwanted emails with approaches ranging from simple rule-based filtering techniques to complicated machine learning algorithms. Numerous factors need to be considered with an email filtering implementation including impact to the user, false positive and false negative ratios, performance, longevity and applicability across a broad base of users. Most implementations of email filtering are focused on addressing Internet scale email abuse such as spam and phishing, tradecraft typically associated with profit motivated actors such as criminals. However, there is very little research on filtering methods

applicable to targeted malicious email sometimes associated with higher-order threats such as national governments or others who do not solely have immediate financial motivations.

1.1 Statement of the Problem

Several articles, industry reports and congressional testimonies document the existence of targeted malicious email (TME) sent by malicious threat actors not necessarily motivated by profit alone. These malicious emails have been targeted at company executives, government personnel and other individuals with access to sensitive information useful by an opposing party to advance a cause. Current research and commercial methods for detecting illegitimate email are limited to addressing Internet scale email abuse such as spam, none seek to address targeted malicious emails.

For organizations targeted by these emails, detection is critically important since these emails can enable the installation of malicious software on the targeted user's computer system. This malicious software can contain a backdoor that allows a malicious threat actor to gain entrance to an organization's network and its sensitive information. Whereas conventional unwanted email, such as spam, is sent in bulk to a large number of people on the Internet, TME is sent to very specific individuals. The techniques that malicious threat actors use to craft and send these targeted emails are different from the techniques used by spammers. Furthermore, since the targeted emails are sent to specific individuals, the characteristics of the recipient are relevant whereas with spam, they are less relevant. This dissertation exploits the differences between spam and TME by capturing features of TME and TME recipients and incorporating them into a decision classifier. The classifier is an algorithm that categorizes a given email as either TME or non targeted malicious email (NTME). This categorization allows an organization to decide whether to accept or reject the email coming into their network environment.

This dissertation surveys and categorizes existing email filtering techniques, proposes and implements new methods for detecting targeted malicious email and compares these newly developed techniques to conventional detection and prevention

methods. This dissertation answers these key questions:

1. What are the various email filtering techniques currently available?
2. Is there a separate class of email, targeted malicious email (TME), that is different than spam or phishing?
3. How do current filtering techniques address the spectrum of threat actors' capabilities and intentions behind illegitimate email?
4. How can the inclusion of persistent threat features improve email filtering over currently available techniques?
5. How can the inclusion of recipient oriented features improve email filtering over currently available techniques?

1.2 Outline of Dissertation

This dissertation is organized into six chapters. Chapter one introduces the problem and relevant background. Chapter two provides a review of current literature which covers current techniques in the area of email filtering. Chapter three outlines the research goals and hypotheses. Chapter four proposes the detailed research method and approach, including all of the statistical tests used in this study. Chapter five covers all of the results of the experiments described in this study. Chapter six summarizes all of the findings and provides recommendations for additional areas of study.

1.3 Background

There are different classes of malicious threat actors each with different intent and capability that a network defender might encounter. Conventional computer network attacks involve exploitation of network-based listening services such as a web server, whereas targeted attacks often leverage social engineering through vehicles such as email to enhance attack effectiveness. Email is an effective attack technique given

that nearly all organizations allow email to enter their network. Threat actors need not expend resources to defeat advanced firewall systems, they can instead leverage an ingress avenue already authorized in most networks. Additionally, these threat actors leverage the Hyper-Text Transfer Protocol (HTTP), typically used to browse the Internet, for Command and Control (C2) and to remove data from a network.

In June and July 2005, the U.K. National Infrastructure Security Co-ordination Centre (NISCC) and the U.S. Computer Emergency Response Team (US-CERT) issued technical alert bulletins describing targeted trojan email attacks leveraging social engineering to exfiltrate sensitive information. The attacks were specifically crafted for recipients with subject lines and email content relevant to the recipient to increase the appearance of legitimacy. The trojans, once installed on a compromised system, connect outbound to threat actors' servers using commonly available outbound ports such as port 80 (typically used for HTTP). Perhaps most troubling is that the targeted email attacks evaded conventional anti-virus and email filtering capabilities (UK-NISCC, 2005; US-CERT, 2005).

In November 2005, iDefense revealed additional information on these targeted email attacks and reported that some of the targeted emails were destined for military personnel and leveraged trojaned Microsoft Word document attachments. The documents were relevant to the recipient and the email was "from" someone the recipient trusted with content in the email relevant to work being performed by the target recipient (iDefense, 2005). This level of targeting and sophistication suggest a patient adversary with the resources to reconnoiter a target environment and craft emails relevant to the recipients. Clearly, this sort of advanced attack can not be performed on an Internet-wide scale and is only used by a determined adversary intent on gaining access to very specific information. Jakobsson (2005) establishes "context-aware phishing" which is an email based attack with information in the email relevant to the recipient. However, Jakobsson's examples suggest use by a common class of adversary, a more traditional criminal actor motivated by money.

Several testimonies to the U.S. Congress and public U.S. government documents characterize the nature of targeted email attacks and respective implications for

national security. Before the U.S. House Armed Services Committee Subcommittee on Terrorism, Unconventional Threats and Capabilities, James Andrew Lewis of the Center for Strategic and International Studies testified that attacks occurred against various government agencies in 2007 including the Department of Defense, State Department and Commerce Department where information collection was the intent of the malicious actors. Lewis also discussed the capability and intent of national government level adversaries such as China and Russia who have the resources, experience and skill to wage cyber warfare against the United States (Lewis, 2008). In testimony to the U.S. House Permanent Select Committee on Intelligence, Paul Kurtz described the extent to which government and private sector networks have been targeted and intellectual property stolen by both state and non-state actors (Kurtz, 2008). With even more specificity about the nature of computer network operations emanating from China, the 2008 and 2009 reports to Congress of the U.S.-China Economic and Security Review Commission summarizes open source reporting of targeted attacks against U.S. military, government and contractor systems. Again, threat actors were motivated by a desire to collect sensitive information (U.S.-China Economic and Security Review Commission, 2008, 2009). Additional corroboration is provided in the 2008 and 2009 Annual Report to Congress by the Office of the Secretary of Defense on the Military Power of the People's Republic of China (U.S. Department of Defense, 2008, 2009). With very specific details of a particular targeted email attack, a March 2008 US-CERT alert bulletin describes a targeted email attack that may have been directed at US government employees and defense contractors (US-CERT, 2008). Finally, in a report prepared for the U.S.-China Economic and Security Review Commission, Krekel (2009) profiles an advanced cyber intrusion with extensive detail including documenting the initial attack vector, targeted malicious email.

Numerous open source reports and industry based work also document the existence of targeted attacks that sometimes use email as a social engineering attack vector. Waterman (2008) describes malicious emails targeted at US think tanks that spoof the email addresses of known colleagues with attachment names relevant to the recipient.

Extensive reporting by BusinessWeek in 2008 also reveals targeted email attacks designed to extract sensitive information. Grow et al. (2008) reports an example of a targeted email sent to a Booz Allen Hamilton Vice President that spoofs a known email contact with highly relevant email content to the recipient (see Figure 1.1). The payload of the email was a malicious trojan designed to capture and log keystrokes, according to the report. Grow et al. also describe numerous other examples of targeted email attacks using malicious Microsoft Word documents, PowerPoint presentations and Access database files. Epstein and Elgin (2008) of BusinessWeek describe targeted

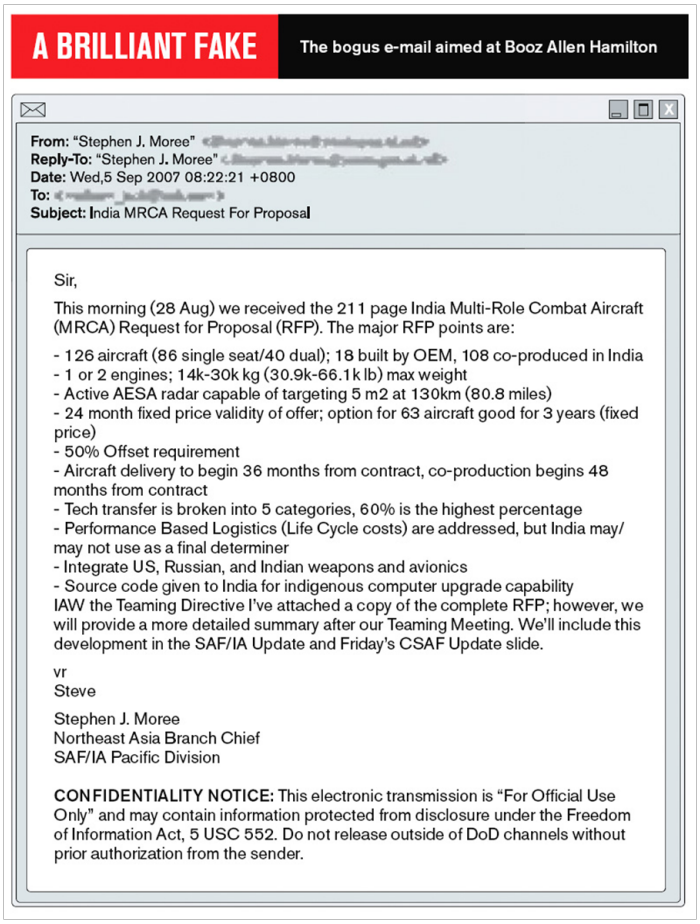


Figure 1.1: Screenshot of a targeted malicious email directed at Booz Allen Hamilton executives (Grow et al., 2008)

emails that fooled individuals at NASA and facilitated unauthorized access to its networks. Barnes (2008) documents another example of targeted attacks against US military networks in US Central Command reportedly with traces to Russian

involvement. Targeted email attacks, with malicious Adobe Portable Document Format (PDF) attachments, have also been levied against foreign correspondents based in China (Villeneuve and Walton, 2009). A fairly comprehensive summary of open source reporting on advanced attacks against the US military complex can be found in Fritz (2008).

Industry and vendor based work has also uncovered evidence of targeted attacks by advanced threats. Pro-Tibet groups reported being targeted by malicious email sent by trusted sources containing information related to recent events in Tibet (Claburn, 2008). Anti-Virus vendor F-Secure provides screenshots and details of the malicious code found in these Pro-Tibet targeted emails and notes that the attacks are advanced and designed to evade detection tools (F-Secure, 2008). Other security vendors such as MessageLabs also revealed the existence of targeted attacks, with detailed examples of malicious email sent to Olympic athletes and national sporting organizations. Many of these attacks leverage either malicious websites or malicious attachments in order to provide unauthorized access to data or networks (MessageLabs, 2007a,b,c, 2008a, 2009). In February 2010, iSec Partners released a report on the current nature of advanced targeted attacks. Their findings noted that current approaches such as Anti-Virus and patching are not sufficient, end users are directly targeted, and threat actors are after sensitive intellectual property or software source code (Stamos, 2010).

Through all of these targeted email attack examples, the capabilities and intentions of the threat actors differ from traditional criminals primarily interested in immediate financial gain. For these advanced persistent threats, acquisition of valuable data is the real intention; while many victims of illegitimate email have money, only certain organizations have valuable information, the type of information that yields long term strategic advantage. The nature of these targeted malicious emails and the capabilities and intentions of the advanced threat actors behind them can be summarized as follows:

1. Email as a form of social engineering is a popular vehicle for conducting targeted attacks.

2. Email based attacks may evade conventional anti-virus, anti-spam and anti-phishing detection mechanisms.

3. Targeted emails are usually crafted such that they are relevant to the recipient - email addresses, subject lines and content are tailored to increase the interest of the intended target enticing them to open the email.

4. Targeted email attacks might be low in volume intended to evade detection.

5. Certain classes of users, such as executives or military personnel, appear to be targeted together in waves of targeted email attacks.

6. Threat actors may repurpose previously sent emails and append them with a malicious attachment for a new attack.

7. Targeted emails use both malicious attachments as well as malicious web links in emails to facilitate unauthorized access to networks and data.

8. Often, the goal behind targeted email attack is acquisition of sensitive information.

In this study, the term "Targeted Malicious Email", abbreviated by "TME", will be used to describe advanced email attacks levied at certain recipients in an attempt to gain access to sensitive information. All other email will be referred to as "Non-targeted Malicious Email", abbreviated by "NTME". Figure 1.2 proposes an email taxonomy to put different types of commonly used email types into context. Email attacks can be evaluated across two dimensions: the first, motivation, characterizes a threat actor's objective; the second, distribution, describes the level of targeting. Motivation is important to understand because it defines the level of recipient engagement needed. If the threat actor's motivation is immediate financial gain, the email attack only needs to trick the user into providing some information (e.g. bank account information) that can be used immediately for financial gain. If the threat actor's motivation is the acquisition of sensitive data, the threat actor is likely going to need a foothold on the

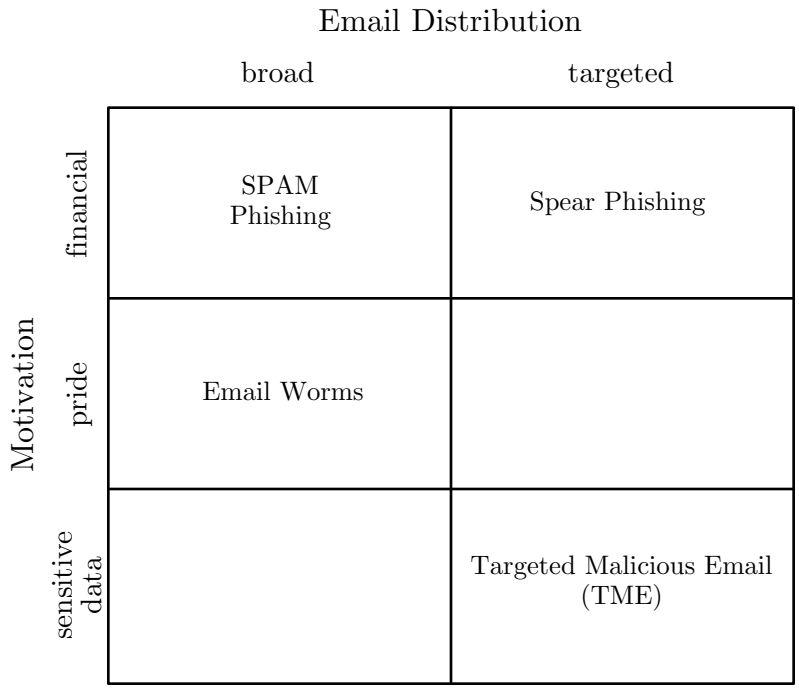


Figure 1.2: Email taxonomy

victim's machine. This foothold will likely require the execution of some malicious software on the target system.

Because of the unique nature of targeted malicious email, newer detection methods are needed. Conventional methods designed to detect traditional actors may not be well suited for actors with different capabilities and intentions. Chapter two will survey the landscape of current email filtering methods and identify shortcomings of these approaches with respect to targeted malicious email.

1.4 Purpose

The purpose of this research is three-fold: a) to characterize and document the existence of TME as different from spam or phishing, b) to demonstrate a correlation between persistent threat and recipient oriented features and targeted malicious email, and c) to develop a classifier that is able to detect targeted malicious email better than conventional email filtering techniques such as anti-spam or anti-virus.

1.5 Significance

The significance of this research is to create an email detection algorithm specifically tuned to uncover TME. TME can result in threat actor presence and exploitation activity on an organization's network leading to significant data loss. Current email filtering techniques such as anti-spam and anti-virus are ill-suited to detect this different class of email. The models developed in this study are able to better detect TME than conventional email filtering techniques.

1.6 Scope and Limitations

This research is scoped to study targeted malicious emails that have been received by one large organization during a finite time period. These targeted malicious emails are used to develop a classifier that is able to better detect these malicious emails than conventional email filtering techniques.

Detection of malicious email can use many features of email and numerous classification algorithms can be used on these features. The classifiers developed in this study do not have any features related to content of attachment; aside from basic attachment information, the attachments are ignored. This is very different from conventional anti-virus where attachment content forms the basis for malicious code detection. Incorporating features related to file attachment content is an area for future study. Additionally, only one classification algorithm is refined in this study. There are numerous classification algorithms that can be applied to the collected data but only one was necessary to demonstrate the improvement over conventional email filtering techniques. Other algorithms may yield even better results but that is another area for future study.

Chapter 2: Literature Review

This chapter will provide a primer on email format, a summary of the threat actor spectrum and the threat kill chain, a review of current email filtering techniques, and finally a summary of the weaknesses with current email filtering approaches. When considering the spectrum of attacks described in chapter one, it is important to understand how current techniques may fall short of defending against advanced threat actors. Current email filtering techniques can be broadly categorized into five classes: authentication, contextual, characterization, reputation and resource consumption. Many of the techniques incorporate elements from multiple classes.

2.1 Email format primer

To understand email filtering techniques, a working knowledge of email structure and format is required. The structure of email is defined in RFC 5322 (Resnick, 2008). Like traditional postal mail, there is an envelope and a letter. Users typically never see the envelope because email systems throw away the envelope just before delivering the letter (e.g. the email message) to the user. An example of an envelope is shown in Figure 2.1. Additional envelope recipients would be defined with additional *RCPT*

```
EHLO col0-omc4-s17.col0.hotmail.com
MAIL From:<rohanamin@live.com> SIZE=26103 BODY=8BITMIME
RCPT To:<rohan@rohanamin.com>
```

Figure 2.1: An example of an email envelope

To: lines. The letter consists of two major parts, the header and the body. The body can include text and also attachments. Figure 2.2 shows an example email including header and body with attachments. Notable features of the email are as follows:

1. *Delivered-To* - The email address the message will be delivered to.

```
Delivered-To: rohan@rohanamin.com
Received: by 10.150.206.13 with SMTP id d13cs119935ybg;
        Sat, 6 Mar 2010 14:55:14 -0800 (PST)
Received: by 10.151.58.8 with SMTP id l8mr904856ybk.59.1267916114523;
        Sat, 06 Mar 2010 14:55:14 -0800 (PST)
Return-Path: <rohanamin@live.com>
Received: from col0-omc4-s17.col0.hotmail.com (col0-omc4-s17.col0.hotmail.com [65.55.34.219])
        by mx.google.com with ESMTP id 38si7218005ywh.70.2010.03.06.14.55.13;
        Sat, 06 Mar 2010 14:55:14 -0800 (PST)
Received-SPF: pass (google.com: domain of rohanamin@live.com designates 65.55.34.219 as
permitted sender) client-ip=65.55.34.219;
Authentication-Results: mx.google.com; spf=pass (google.com: domain of rohanamin@live.com
designates 65.55.34.219 as permitted sender) smtp.mail=rohanamin@live.com
Received: from COL109-W36 ([65.55.34.201]) by col0-omc4-s17.col0.hotmail.com with Microsoft
SMTPSVC(6.0.3790.3959);
        Sat, 6 Mar 2010 14:53:13 -0800
Message-ID: <COL109-W36BCC9221F3B110DCC6153D2370@phx.gbl>
Return-Path: rohanamin@live.com
Content-Type: multipart/mixed;
      boundary="_53be5617-f8b0-4615-9a14-5ea2feef7c55_"
X-Originating-IP: [127.0.0.1]
From: Rohan Amin <rohanamin@live.com>
To: "Rohan Amin" <rohan@rohanamin.com>
Subject: Example Email
Date: Sat, 6 Mar 2010 17:53:12 -0500
Importance: Normal
MIME-Version: 1.0
X-OriginalArrivalTime: 06 Mar 2010 22:53:13.0280 (UTC) FILETIME=[CD16CC00:01CABD7F]

--_53be5617-f8b0-4615-9a14-5ea2feef7c55_
Content-Type: text/plain; charset="Windows-1252"
Content-Transfer-Encoding: quoted-printable



This is an example email with attachment.
                                =20
_________________________________________________________________
Hotmail: Trusted email with Microsoft=92s powerful SPAM protection.
http://clk.atdmt.com/GBL/go/201469226/direct/01/=

--_53be5617-f8b0-4615-9a14-5ea2feef7c55_
Content-Type: application/msword
Content-Transfer-Encoding: base64
Content-Disposition: attachment; filename="test.doc"

0M8R4KGxGuEAAAAAAAAAAAAAAAAAAAAAPgADAP7/CQAGAAAAAAAAAAAAAAABAAAAJgAAAAAAAAAA
AABSAGkA8/+zAFIADL0AAAAAAAAAAwAVABhAGIAbABlACAATgBvAHIAbQBhAGwAAAAcABf2AwAA
NNYGAAEKA2wANNYGAAEFAwAAYfYDAAACAAsAAAAoAGsg9P/BACgAAA0AAAAAAAAAAAcATgBvACAA
TABpAHMAdAAAAAIADAAAAAAAUEsDBBQABgAIAAAAIQBLnfYYAAEAABwCAAATAAAAW0NvbnRlbnRf
...
hFUJv4FV7DOyOxZhFdeWCiI9JIyjdkSk9MncEuBvJejXoXX4w77LxomLFIoe+HTewJxXkdv8oBXj
JPNhuzSNq9j35QGkKEZ7XPngu9ytEP0MccDpzHDfocQJ99nd4DYdOiZNEkS/GQkdS2jVTgdOaPqq

--_53be5617-f8b0-4615-9a14-5ea2feef7c55_--
```

Figure 2.2: An example email

2. *Received* - Every email server that handles the message will add a *Received* line entry which includes a time stamp.

3. *Return-Path* - The email address from which the message was sent.

4. *Received-SPF* - Sender Policy Framework (SPF) domain authentication results.

5. *Message-ID* - A unique number assigned by the sending mail server.

6. *Content-Type* - Defines the boundary string used to separate the Multipurpose Internet Mail Extensions (MIME) parts of an email.

7. *X-Originating-IP* - The Internet Protocol (IP) address of the sending client.

8. *From* - Set by the sender's email program. *From* consists of a phrase and address (the phrase is the string before the email address). This does not have to equal the *MAIL From* line in the email envelope.

9. *To* - Set by the sender. *To* consists of a phrase and address. This does not necessarily have to equal the *RCPT To* line in the email envelope. If there are any *Cc* recipients they would appear in the *RCPT To* line in the email envelope. Any *Bcc* recipients would not be shown but would be in the email envelope as *RCPT To* recipients.

10. *Subject* - Set by the sender.

11. *Date* - Set by sender's email program. It includes the local time zone of the system used to send the email.

12. *Content-Type* - Defines the character set used by the email.

13. *Content-Disposition* - Includes some information about the attachment.

2.2 Threat actor spectrum and the threat kill chain

2.2.1 Threat actor spectrum

Computer and network systems are attacked today by a range of adversaries who vary in capability and intent. Key to understanding the nature of Computer Network Attack (CNA) and Computer Network Exploitation (CNE) activities are the motivations behind classes of threat actors, also referred to as malicious actors, adversaries or attackers (in the case of CNA). Simply put, different threat actors have different motivations, different levels of sophistication and different levels of resources all requiring an associated spectrum of effective defenses. This section summarizes some relevant publicly available characterizations of threat actors.

In a 2001 statement to the Joint Economic Committee of the United States Congress, Dr. Lawrence Gershwin, the U.S. National Intelligence Officer for Science

and Technology, outlines five major categories of malicious actors who threaten information systems in the United States: hackers, hacktivists, industrial spies and organized crime groups, terrorists, and national governments. He describes hackers as hobbyists without the tradecraft or motivation to pose a significant threat to national-level infrastructure. Hacktivists, Dr. Gershwin describes, carry out their activities for purposes of propaganda rather than to damage critical infrastructure but they still pose a medium-level threat able to carry out a limited but still severe attack. The next class of malicious actor, industrial spies and organized crime, also pose a medium-level threat to the United States but are primarily motivated by money; these criminals engage in targeted attacks to evade attention from law enforcement. Traditional terrorists, according to Dr. Gershwin, pose little threat to information systems since they will remain focused on more conventional attack methods such as bombs. Finally, Dr. Gershwin characterizes the national government or nation-state threat as the only class of malicious actor with the resources and time-horizon to cause significant damage to critical infrastructure. A lengthy time-horizon allows threat actors to wage persistent attack campaigns instead of singular attacks. He notes that specialized tools are needed for targeted attack defense and that these tools differ than those needed for defense against Internet wide exploitation (Gershwin, 2001).

A 2005 U.S. Government Accountability Office (GAO) report on the U.S. Department of Homeland Security (DHS) role in cyber security, delineates a spectrum of threats such as bot-network operators, criminal groups, foreign intelligence services, hackers, insiders, spammers, phishers, malware authors, and terrorists. The report describes that a few of these threats, such as foreign intelligence services, have the capability to impact national-level interests. DHS assembled this threat landscape using data from the Federal Bureau of Investigation, the Central Intelligence Agency and Carnegie Mellon's CERT/CC (U.S. Government Accountability Office, 2005).

Schudel and Wood (2008) focus on the cyber terrorist threat and note that a terrorist threat is not as sophisticated as a national government but still has significant resources to disrupt or degrade systems. Additionally, terrorist methods will be targeted and terrorists will only expend the minimum amount of resources necessary to accomplish

a mission, nothing more. These observations about this class of adversary imply that from a defensive standpoint, network defenders need to look for different tradecraft and methods of operation from a criminal who may operate on an Internet-wide scale primarily for monetary gain.

Finally, in February 2008, United States Director of National Intelligence Michael McConnell stated that, "nations, including Russia and China, have the technical capabilities to target and disrupt elements of the US information infrastructure and for intelligence collection. Nation states and criminals target our government and private sector information networks to gain competitive advantage in the commercial sector" (McConnell, 2008). McConnell additionally stated that different actors have different capabilities and different intentions which means that the associated defenses must differ and be based on the particular threat being opposed (McConnell, 2008).

2.2.2 Threat kill chain

The threat kill chain is the sequence of events that must occur for a threat to successfully achieve its objective. Figure 2.3 depicts the threat kill chain from the threat actor's perspective assuming computer network exploitation is the goal (e.g. acquisition of sensitive information). Different kill chains can be created depending on specific threat actor objectives; for example an attack against data integrity would have a different kill chain to model the threat actor's process (Hutchins et al., 2010).

Recognizing that a human is behind a keyboard executing email-based exploitation is key to exploring new methods for targeted malicious email detection. In cases of persistent activity there may be an institutional infrastructure driving email-based exploitation. New detection techniques can be developed by focusing on each component of the threat kill chain. This kill chain decomposition allows defenders to create detections based on the habits of individual threat actors and the processes of institutions. Detection and prevention anywhere along the kill chain is a success for defenders as long as the threat actor's final goal is not achieved. In the following threat kill chain decomposition the acquisition of sensitive information is considered to be the threat's intent with email being the primary exploitation vehicle. This

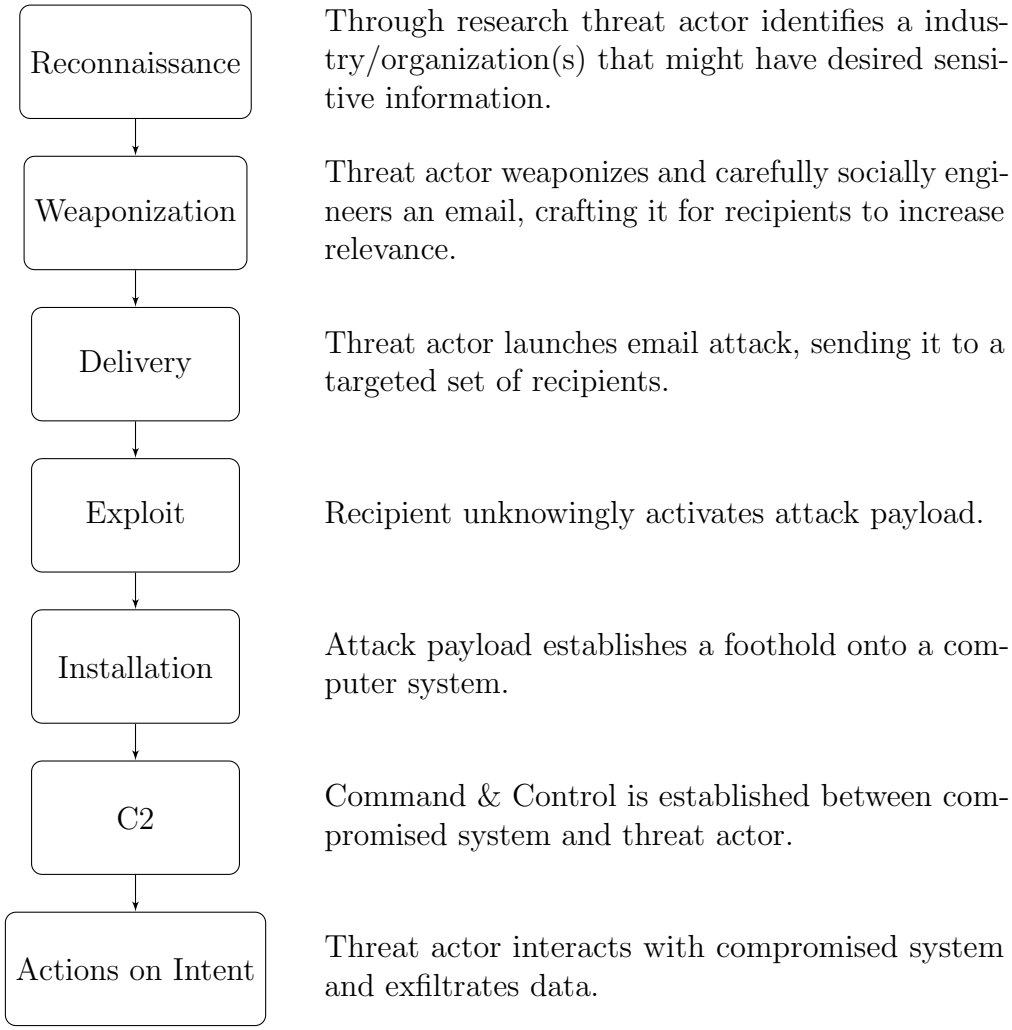


Figure 2.3: Example email kill chain (Hutchins et al., 2010)

study develops a framework for email analysis that can be used to automate detection along the entire threat kill chain. Through this framework the tactics, techniques, procedures and infrastructure behind threat actors can be decomposed and leveraged for significant gains in detection capability. This study primarily focuses on features that expose the reconnaissance, weaponization and delivery phases of the kill chain.

Reconnaissance

Threat actors gather as much information as possible to increase the likelihood of success during the reconnaissance phase of the kill chain. Assuming the threat actor has specific data collection tasking, he will need to understand where that information is located, what organizations have it and who specifically in the organization has access to it. To effectively target certain individuals using email, the threat actor will need the email address of the recipient and context about the recipient to make the email relevant. Careful planning during the reconnaissance phase will increase the chances of success for a threat actor. A network defense team should pay attention to details such as how Internet users access their organization's website, the search terms used to access their website and even language settings of web browsers being used to access their website. This sort of analysis maps back to specific techniques and infrastructure that threat actors employ.

Weaponization

The weaponization phase of the kill chain is when the threat actor creates and packages an email weapon that will eventually be delivered to targeted recipients. Typically a threat actor can weaponize an email through the use of a malicious attachment or a web link to a malicious website. In the case of a malicious attachment, there is typically the file container, the exploit and a backdoor. The file container could be a Microsoft Word document or Adobe Portable Document Format (PDF) for example with content that is relevant to the intended recipient targets. This file container might contain author information, file path information or other information about the host computer that was used to create the file. Inside the file container is typically an

exploit for a vulnerability in that software package. Finally, the file container contains a backdoor that will be installed on the recipient's system and provide unauthorized access for the threat actor. Internet searches for 'trojan pdf creator' turn up tools that can be used to package a file container with exploit and backdoor. These tools may leave signatures in the weaponized attachments that can be leveraged for detection purposes. Another important point to consider is that the person who creates the malicious payload may not be the same as the person who sends the malicious email. Email senders might need to acquire these malicious components from a supply chain of providers. This is analogous to an assassin who uses a gun and ultimately pulls the trigger but gets bullets from a supplier. In the case of a link to a malicious website, the threat actor needs to host malicious code on a website. The website can either be under the complete control of the threat actor or could be on an existing legitimate website that has been compromised. The latter approach typically results in a more legitimate looking link being included in an email but also has a higher cost for the threat actor to establish.

Delivery

Once the weaponization phase is complete, a threat actor needs to send an email to a targeted set of recipients. There are a number of delivery elements to consider including the email addresses and email content, tools used to send email, and the distribution method.

When sending an email a threat actor can choose to either use legitimate or illegitimate sender information. Since conventional email does not require any authentication of the sender, threat actors generally opt to misrepresent their identity typically through falsification of the sending *From* address in the email header. Threat actors can impersonate another legitimate email sender or choose to completely falsify the information with no intent to impersonate a recipient known legitimate sender. The text of the *From* address consists of an optionally provided Full Name (e.g. John Smith) and a required *From* email address (e.g. john.smith@example.com). If a threat actor is trying to increase the probability of a recipient opening a malicious email, the

From name and address might be chosen to be relevant to that particular recipient (e.g. a known colleague, friend, or business). A threat actor can also opt to falsify the sending system information. Normally, a sending system will add its identifying information such as its host name or IP address to the email headers. If an email is sent through a network of relays before arriving at its destination, relay systems that adhere to standards will also append their identifying information to the email headers. However, a threat actor can append a superfluous chain of email relays to a malicious email to obscure the true system of origin. User visible content of an email is important depending on the targeted recipients and intent of the threat actor. Both the *Subject*, in the header of an email, and the actual email *body* are elements of user visible email content. Email *body* content can either be blank, random, of generic relevance to the recipient, or of specific relevance to the recipient. Blank or random content might be completely irrelevant to a recipient resulting in the recipient immediately deleting such an email. However, if a threat actor's only intent is to build a dictionary of valid recipients by analyzing invalid recipient error messages from receiving mail systems, sending blank emails to a brute force generated list of email addresses might be the most appropriate method. Furthermore if a threat actor's intent is to confuse or influence the calibration of a target email system's filtering capability, an email with random text might be an appropriate method. When not reconnoitering or calibrating, a threat actor generally wants the malicious email to be opened which means the email will have content of either generic or specific relevance to the recipient. If a threat actor intends to send an email to a large population the content might consist of topics relevant to a large audience such as current news, current events, or pop culture. If a threat actor intends to send an email to a very specific and smaller population, the content might be highly relevant aligning with the recipients' affiliations, organization or role. The combination of identity misrepresentation and crafted email content can be very powerful for increasing the relevancy of a malicious email to a specific recipient. Threat actors may maintain databases of recipients and those recipients may be targeted multiple times. Different threat actors may have different databases and as such some recipients may only be targeted by some threat actors and not by

others.

Another important consideration from a detection perspective in the delivery phase of the kill chain is the type of tools used to send the targeted malicious emails. Typically users send email using software such as mutt[1], Microsoft Outlook[2], Mozilla Thunderbird[3], Foxmail[4] or web-based services such as Google Gmail[5], Microsoft Hotmail[6] or Yahoo! Mail[7]. A threat actor can certainly use traditional email software to send malicious emails, however, these tools do not easily facilitate misrepresenting one's identity or distributing large amounts of email; after all, these applications enforce some standards. Automated tools facilitate rapid generation and distribution of emails sometimes customized per recipient. For example, a single email can be sent individually to hundreds of email addresses instead of all email addresses being in the *To* line of a single email. Furthermore, these automated tools can also customize email content by including the recipient's name at the beginning of an email or by appending a unique string to the end of each email to hamper signature based email detection algorithms. These email tools may also include traces of language settings or character encodings used when the content was created. Locale information such as time zones might also be included by the email tool. In summary, tools will leave a footprint or signature in the email that can be used to identify usage of a particular tool by a threat actor or information about the threat actor himself.

Once a threat actor has created an email and defined its recipients, he needs to send the illegitimate email to the target(s). There are generally four ways a threat actor can distribute an email: directly, through one or many relays, through a public webmail provider, or through a large network of autonomous, compromised machines (commonly known as a "botnet"). If the threat actor has his own email server, he can choose to send his illegitimate email directly from his email server which will connect to the recipients' respective email servers. However, this approach will leave a

[1]mutt - http://www.mutt.org
[2]Microsoft Outlook - http://www.microsoft.com/outlook/
[3]Mozilla Thunderbird - http://www.mozilla.com/thunderbird/
[4]Foxmail - http://www.foxmail.com.cn/
[5]Gmail - http://gmail.google.com/
[6]Hotmail - http://www.hotmail.com
[7]Yahoo! Mail - http://mail.yahoo.com

clean trail back to the threat actor and may also diminish in effectivity over time as email filtering tools are updated with new known-bad servers. This approach might be useful if a threat actor wants to send a small amount of email over a short period of time. A second distribution vehicle involves one or many email relays on the Internet. A threat actor can configure their email software to connect to an open relay on the Internet that will send the email on his behalf. This will separate the threat actor from the recipient's email server but depending on the configuration used, the received email headers will still show a chain of email relays used to send the email. This might be enough obfuscation depending on the intent of the threat actor. A third distribution vehicle involves a threat actor creating an account with a public webmail provider such as Google Gmail. Sending an email this way can also obfuscate the source of the threat actor; to the recipient, the message originates with the webmail provider, not the threat actor. Some webmail providers, however, include the IP address of the system that connected to it in the email headers that are visible to the email recipient, some do not[8]. Tools are also available that facilitate mass creation of webmail accounts for malicious use (MessageLabs, 2008b). A final distribution vehicle for malicious email are large networks of autonomous, compromised machines known as botnets. Botnets can consist of upwards of hundreds of thousands of compromised systems worldwide which are in the control of a "botherder." These botnets, among other actions, can be directed to execute denial of service attacks against Internet websites or send massive amounts of email around the world. Traded like commodities, ownership or use of a botnet involves a cost to a threat actor but affords a measure of obfuscation and scale unmatched by other distribution vehicles. Again, depending on the specific intention of a threat actor, leveraging a botnet infrastructure may or may not be the most appropriate malicious email distribution vehicle (Rajab et al., 2006).

All of these delivery elements can be leveraged for detection purposes. Institutions engaging in systematic exploitation activities might have developed processes and procedures that can be detected. Given the relative difficulty it is conceivable that the

[8]As of the publication date of this study, Hotmail and Yahoo include the originating client IP address and Google Gmail does not

skill level of individuals sending malicious email might be lower than those developing weaponized packages. This skill level difference may result in more mistakes or more discernible habits that can be detected.

Exploit

To successfully have the targeted recipient's computer system perform a function desired by a threat actor, there are two exploit methods a threat actor can leverage: a threat actor triggered exploit, exploiting a vulnerability in the email software itself or a recipient triggered exploit, requiring the user to open an attachment or click on an Internet link. Threat actor triggered exploitation does not require the user to take any action since a vulnerability in the email software itself allows for automatic exploitation as long as the email software has not been fixed. There have been several email software vulnerabilities enabling threat actor triggered exploitation but over time they have been fixed by email software vendors Microsoft (2002); US-CERT (2007); Mozilla (2008). Recipient triggered exploitation largely depends on a technique known as social engineering, where a user is tricked or manipulated into taking an action in the threat actor's favor. User education and awareness is generally the only solution to these types of social engineering based attacks. To actively engage an email recipient, a threat actor can leverage either a malicious file attachment or Internet link. Both can enable a threat actor to direct an unsuspecting user to a website for selling a product, to download malicious software to compromise their system or to a website to collect personal sensitive information. Depending on the threat actor's intentions and capabilities, there are a number of considerations that favor one active exploitation technique over another. For example, manipulating a user to click on a link in an email might require a threat actor to compromise a legitimate Internet web site to install malicious code so a user believes they are going to a legitimate web site. A victim triggered exploit might still exploit a vulnerability in client software (e.g. a vulnerability with Microsoft Office software) but the exploit will not trigger unless the user opens the malicious attachment.

Installation

Once the malicious payload has been activated it will be installed on the recipient's system. The payload might get installed to certain folders on the system or it might create unique system registry keys. The payload might employ different persistence mechanisms such as execution via the system startup folder or through a startup service. It is possible the payload might not seek persistence at all but only run once on a recipient system. If an institutional threat actor is exploiting numerous organizations, the payload installation might have organization specific configurations to help effectively manage the compromised systems across different organizations. All of the these factors can be considered for detection purposes since different threat actors might have a preference for different approaches.

Command and Control (C2)

After malicious code is installed on a recipient system it typically communicates back to the threat actor for purposes of command and control (C2). This C2 channel is how the threat actor manipulates and controls the now compromised system. For example, the threat actor might direct a system to search for and return certain files. The C2 channel might use certain protocols, employ specific obfuscation techniques or might be destined for certain IP addresses on the Internet. These features can be associated with the particular backdoor used by a threat actor and can also describe the infrastructure on the threat actor's end. Additional persistent threat insight can be gained from analyzing the Internet domain registry information for the domains used by threat actors for C2 purposes.

Actions on Intent

Different threat actors may employ different techniques to achieve the final goals they wish to achieve. For example if data exfiltration (removal) is the threat actor's intent there are different methods to package and remove data. One threat actor might opt to exfiltrate files one at a time, another might opt to remove files in bulk using a

compressed archive file. To actually transfer the files, some threat actors might use traditional file transfer techniques such as File Transfer Protocol (FTP) and others might obfuscate data removal within an obfuscated C2 channel. These behaviors can be detected and used to provide a measure of threat actor attribution.

2.3 Current email filtering techniques

2.3.1 Authentication

Authentication based approaches to filtering email are designed to validate that an email was sent using a valid path for the advertised sending domain name and that the domain name is not being spoofed by a malicious actor. This method of filtering typically occurs very early in the email transmission when a sending email server first connects to a receiving email server. Authentication based filtering leads to a binary response: email is either be accepted or rejected based on the authentication result.

Wong and Schlitt (2006) and Lyon and Wong (2006) describe two methods of domain authentication called Sender Policy Framework (SPF) and Sender-ID. Both rely on the sending system publishing valid email server records in the Domain Name System (DNS). The receiving system is then able to verify that an email advertised as coming from a particular domain actually came from email servers authorized to send email on behalf of that domain. SPF and Sender-ID are very similar in approach and differ in the fields they use on the receiving end for the lookup.

Crocker et al. (2005), Otis et al. (2005), Leslie et al. (2005) discuss the components of Certified Server Validation (CSV) which is another authentication scheme leveraging DNS for domain validation. However CSV differs in that it uses the domain name in the Simple Mail Transfer Protocol (SMTP) HELO transaction. CSV first checks to ensure the server sending IP address matches the IP address in DNS for the domain used in the HELO transaction. Second, CSV verifies the reputation of that domain vs. the domain name advertised in the email headers. This difference is important when considering situations where individuals are sending email through a mail server where the *From* address and the mail server may not match (e.g. a travelling user using a

remote Internet Service Provider). This difference between CSV and SPF/Sender-ID results in a different approach for handling spoofing. With the former, spoofing controls need to be handled on the sending server side to ensure the server is only sending email that it is supposed to send. With the latter, spoofing controls need to be handled on the receiving server side to ensure that the sending IP address is valid for the advertised *From* domain in the email headers.

Another authentication mechanism, DomainKeys, is described by Delaney (2007), Allman et al. (2007) and Leiba and Fenton (2007). DomainKeys is different in that it leverages a public/private key cryptographic solution where the sending email server signs the email with a private key and the receiving email server validates the signature by retrieving the public key for the *From* domain in the email headers via DNS. This approach is similar to SPF/Sender-ID in that DomainKeys validates that a particular email server is authorized to send email for a domain advertised in the *From* email headers. DomainKeys differs from SPF/Sender-ID, however, because it does not require the sending domain to maintain lists of authorized email servers for the domain. The downside is that there is overhead associated with computing, maintaining and distributing the private/public key pair needed for DomainKeys. Taylor (2006) describes Google Mail's (Gmail) approach to establishing sender reputation and it is heavily based on both SPF and DomainKeys as complementary approaches because each has its strengths and weaknesses.

Other authentication like approaches include the Occam protocol described by Fleizach et al. (2007), the Single-Purpose Address (SPA) described by Ioannidis (2003) and Trustworthy Email Addresses (TEA) described by Seigneur et al. (2004). The Occam protocol works in real-time on a per-email basis where the receiving email server asks the sending email server to validate that it sent a particular email based on the email Message-ID field. SPA uses a cryptographic based email address which encapsulates the policy in the email address itself. Not designed for person-to-person interaction, someone looking to receive communication from a party at a later date (e.g. online retailer) would generate a SPA and give that to the party for their explicit use. The policy in the SPA defines an expiration date and authorized senders who are

allowed to use the SPA. Enforcement of this policy is done by the receiver. TEA is a challenge-response authentication scheme that uses hashes of previously exchanged email between two email addresses to authenticate that a new received email is being sent from the correct email server and not being spoofed.

All of these authentication approaches are designed to make sure that an email being received is actually being sent by a system or person authorized to send an email from the advertised email address. With respect to the threat spectrum described in chapter one, all of these techniques are well suited to address malicious actors who send large numbers of email spoofing various domains (e.g. spammers). A threat actor could register a new domain, equip it with the appropriate authentication capabilities and then send spam from that domain. However, Internet-wide real-time blocklists would be quickly updated and tag email from this domain as illegitimate. Trying to scale this sort of approach would introduce a non-trivial cost to the actor. These authentication approaches can also be used to prevent a more advanced threat actor from sending a targeted spoofed email. However, these techniques require all senders and all receivers to implement them in order to realize the benefit of being able to prevent targeted social engineering malicious email attacks. Furthermore, approaches like SPF, Sender-ID and DomainKeys, which are the predominant email authentication approaches in use today, only validate at a domain level and not on a per email address basis. Thus a more patient and resourceful adversary who is able to establish new email accounts at a public email provider such as Google, can use that account to send targeted email and the authentication mechanisms will simply validate that Google's mail servers were authorized to send email using Google's domain. This does not say anything about the intent of the user using the Google system. These approaches only help in situations where a threat actor is spoofing an email using another domain since these authentication mechanisms will flag those attempts. Short of full adoption, it only takes a malicious actor to identify one trusted sending domain name which does not use any email authentication systems in order to send spoofed and malicious email to countless other receiving organizations. Receiving organizations would need to be willing to drop email from domains that don't use email authentication approaches

but due to business drivers and the criticality of email communication, this approach would not normally be acceptable.

2.3.2 Contextual

The bulk of email filtering related research falls into the contextual analysis category. These are techniques which leverage the actual content of the email when making filtering decisions. Contextual approaches span from simple dirty word searches to machine learning to hash based techniques. The result of contextual analysis is usually a probabilistic answer with regards to the legitimacy or illegitimacy of an email instead of a binary answer typically associated with the authentication approaches described above. This allows for multiple techniques to be combined together to enhance a particular detection capability.

Basic approaches to contextual analysis include processing a set of rules, or heuristics, that assign a score to the presence of certain words or phrases in an email. Rules can be established using words or phrases commonly found in the types of email that are being sought. Graham (2002) shows that this sort of rules-based approach is feasible for detection of spam but is problematic since a high number of false positives result when trying to approach 100 percent detection of spam. Stone (2007) uses a rules-based approach based on Natural Language Processing (NLP) and is able to achieve a 75 percent detection rate using four rules for detecting phishing emails. Evading these types of filtering techniques is rather trivial since a threat actor only needs to craft emails to change words that avoid any of the rules in the defined rule set. In the case of a threat actor that might be repurposing legitimate email, such as the example in Figure 1.1, this sort of heuristics based approach to identify a bad email will certainly fail since the email content is legitimate. Email content is very easily changed by a threat actor and as such does not have great durability with regards to detection.

In seminal papers, Sahami et al. (1998) and Pantel and Lin (1998) describe machine-learning Bayesian based approaches for filtering spam. Interestingly, Sahami et al. incorporate additional properties in the classification vector for each email such as

whether an attachment is present. They note that most junk email does not have an attachment and is sent at night which based on more recent reports of advanced attacks covered earlier in this dissertation, may not be the case for detecting attacks from a different class of adversary. Different threats may have different behaviors. Androutsopoulos et al. (2000a) compare several Bayesian approaches to basic heuristic or keyword based approaches and note that the Bayesian approaches are superior even with a small amount of training. Other researchers have explored the use of Bayesian approaches including Schneider (2003) who finds that a multi-nomial model that incorporates word frequency information in the email classification vector is superior to multi-variate Bernoulli models that do not. Chen et al. (2007) finds that incorporating email headers in the Bayesian analysis, not just email body, improves classifying performance. Bayesian techniques are the most explored technique for email filtering in the literature and very often researchers will benchmark newly created algorithms against basic Bayesian approaches. It is an effective technique for identifying spam and is incorporated into open-source and commercial email filtering systems.

Additional machine learning contextual based approaches to filtering email include Support Vector Machines, Neural Networks and Memory Based Approaches. Drucker et al. (1999) first leverage Support Vector Machines (SVMs) for separating spam from non-spam email. Both sets of training email are mapped to a higher-dimensional space and a hyperplane is created that has a maximum distance between the sets. Once the SVM is trained, the hyperplane is used as a decision boundary for categorization of emails as spam or non-spam. Bergholz et al. (2008) also leverage SVMs but specifically for the detection of phishing emails. They incorporate basic email features in their email classification vector but also include two advanced features based on Dynamic Markov Chains and Latent Topic Models which show improvement over SVM approaches that do not include the advanced features. Clark et al. (2003) use a backpropagation neural network classifier for filtering and find that it outperforms Bayesian approaches but takes significant training time. Tzortzis and Likas (2007) use a Deep Belief Network, a Neural Network with more hidden layers, with comparable

results to SVMs and Sirisanyalak and Sornil (2007) use a backpropagation classifier that has a feature extraction technique based on artificial immune systems. Sakkis et al. (2001) use a memory based approach that does not create a model for each category of email but simply stores the training emails and computes the similarity of new emails to stored emails using a k-Nearest-Neighbor algorithm. They find that this approach yields comparable results to Bayesian techniques.

Several contextual based approaches are based on hashing and coding techniques that create representations of email using digests, hashes or codes and uses these representations for comparison purposes when trying to determine how to filter an email. Yoshida et al. (2004) introduces a technique that creates a set of hashes based on shifting substrings of the text in an email. These hashes are then compared to count the number of similar emails in a particular set. These researchers report this approach as having superior speed and classification accuracy than Bayesian, SVM and Memory Based filtering approaches. Zhou et al. (2005) introduce a learning approach that uses Huffman coding to create a representation of training data. Since their algorithm does not require the representation of earlier messages to be updated it can be applied in real-time which allows the classifier to adapt to changes as they occur. Delany and Bridge (2006) present a feature extraction free case-based approach which stores previously categorized emails and computes similarity using text compression as the distance metric. They report superior email classification performance compared to other case-based approaches but note that using text compression introduces added computing overhead even though there is no overhead associated with feature extraction. Damiani et al. (2004) create a Peer to Peer (P2P) architecture for collaborative filtering that uses digests to represent emails for comparison purposes. The added benefit for filtering spam or phishing is that the system is able to leverage a potentially world-wide knowledgebase when making filtering decisions.

A subset of contextual approaches examine the Internet links in email bodies to make filtering decisions. Kolesnikov et al. (2003) present an approach that mines search engines, such as Google, for categorization information about Internet Uniform Resource Locators (URLs) contained in emails. The category of any URLs found in

an email are used to determine the appropriate categorization of the email itself. Liu et al. (2006) developed an approach that is focused on phishing detection that also examines URLs contained in email but instead of looking at the category of the URL their technique examines the structure of the webpage. If the webpage includes a form for entering usernames, passwords, account or other sensitive information it will flag as a phishing webpage which in turn categorizes the email containing that URL as a phishing email. Chandrasekaran et al. (2006) extend this approach by not only extracting the types of form fields on a webpage but by mimicking a user response by providing fake data to the requesting website and analyzing the result using a set of rules.

For targeted malicious email, it is conceivable that contextual based approaches may provide some benefit if there are common content elements across emails from a particular malicious actor. Based on the low volumes of targeted malicious email, creating a large and relevant enough training set would be problematic. P2P and other collaborative approaches make a fundamental assumption that a large population will receive a particular email such that others in the distributed network could validate the categorization. In a targeted email attack scenario, if only one or two organizations receive a particular email, the distributed network may not have any references available to help make a filtering decision.

2.3.3 Characterization

Some email filtering techniques leverage network traffic or other behavioral characterization techniques designed to focus on the behaviors of the actors sending the email. Similar to contextual approaches, probabilistic answers are the result.

For the purposes of filtering spam from non-spam, Gomes et al. (2004) demonstrate that spam traffic has different characteristics than non-spam in the areas of email arrival times, email sizes and number of recipients per email. These differences can be used as the basis for making filtering decisions. Beverly and Sollins (2008) exploit the fact that spammers, in order to send large quantities of email, need to leverage large numbers of resource constrained hosts. They created a tool called "SpamFlow" which

has a classifier based on network transport layer properties such as packet Round Trip Time (RTT). For resource constrained hosts being used by spammers, they find that the network congestion or asymmetric nature of the links used by these hosts introduces significant packet level delay.

By studying the domains that spammers target instead of just using the spammers IP address, Ramachandran et al. (2007) are able to create a dynamic spammer blocklist by clustering similar sending patterns. They created a tool called "SpamTracker" that when used across several domains is effective in distinguishing spam from legitimate email even before spammers are listed on conventional blocklists. Focused primarily on filtering email borne viral propagation occurring via infected attachments, Bhattacharyya et al. (2002) created a tool called "MET" (Malicious Email Tracker) that leverages a client/server architecture to track statistics of email sent and received to determine if there are viral propagations occurring. Any identified viral emails can be filtered out once identified, and new viral propagations can be discovered early.

Alternative behavioral characterization approaches focus on identifying a fingerprint of the author of emails. (O'Brien and Vogel, 2003) leverage authorship identification techniques, specifically a Chi by degrees of freedom approach, for the purposes of filtering spam and find that it performs equal to or better than Naive Bayes. McCombe (2002) provides a good overview of authorship identification techniques which have applications far beyond email filtering. Finally, Calais et al. (2008) leverage a frequent pattern tree in order to uncover features that are common to multiple emails sent by a single spammer. The pattern tree exploits the fact that spammers may reuse elements from one email to the next, perhaps when using a single spam tool, in a campaign of spam. Using this tree based approach, they are able to identify roughly 16,000 spam campaigns across a data set of more than 97 million emails.

One of the challenges with characterization based approaches to filtering email is a general reliance on the defining characteristic of volume which is normally only suitable for filtering spam. Since targeted email attacks are generally low volume and mimic other normal email characteristics such as rate and message content, filtering using these mechanisms is problematic. A frequent pattern tree approach might be

relevant if the right headers and other email features are used to find commonality across targeted malicious email.

2.3.4 Reputation

Reputation based approaches to filtering email are based on maintaining lists of good vs. bad or calculating a level of trust through relationship linkages. Whitelists, blacklists and DNS-based Real-time Block Lists (DNS RBLs) are examples of list based reputation filtering. Leveraging social networks for establishing trust due to relationship linkages is another form of reputation analysis used in email filtering. In these approaches, the reputation of an email is calculated based on the sum of the component reputations; known bad senders can decrease an email's overall reputation whereas known good sending IP addresses can increase an email's overall reputation. The majority of the research is focused on sender, not recipient, reputation.

Erickson et al. (2008) use a combination of challenge-response and a persistent whitelist per user for filtering legitimate email. They find that even though there is some initial user-overhead required for managing the whitelist, over a period of two years very little spam appears in users' inboxes. However, they do make a fairly significant assumption that sender-based authentication services, described above, are a prerequisite to prevent simple spoofing. Duan et al. (2004) created DiffMail, an architecture that allows user to classify senders into allowed, unknown and not allowed. Using this classification, unknown emails are then left on the sending mail server with only header information being sent to the recipient. The recipient can then retrieve the rest of the email if they want. This approach requires spammers to maintain additional online resources if they want recipients to retrieve their emails. Jung and Sit (2004) analyze DNS based black lists and find that across spam analyzed over a roughly three year period, approximately 80 percent of spam sources are listed in at least one of seven popular DNS based black lists. However, they show that relying on only one or two lists is not sufficient since some lists are more conservative than others when determining which sources get listed.

Several reputation based approaches to filtering email leverage social networks to

establish relationship linkages or levels of trust. Golbeck and Hendler (2004) developed a tool called "TrustMail" that allows users to assign a reputation rating to people they know. Through a network of relationships, users are able to establish ratings for people they don't know. A slight variation, Garriss et al. (2006) designed "Re:" a system that automatically populates whitelists using friend-of-a-friend relationships and allows recipients to check if other recipients have whitelisted a particular sender. Chirita et al. (2005) and Lam and Yeung (2007) build ratings for senders based on social network graphs created from archives of email of a group of individuals. Boykin and Roychowdhury (2004) also use a graph based approach but only create the social network from a single user's archive of email. Their approach leaves nearly 50 percent of emails unclassified but they correctly categorize with 100 percent accuracy all spam in their test dataset. Instead of using closed social networks, Rivera et al. (2008) use the Google OpenSocial network alliance and set of interfaces for establishing sender reputation. This allows them to query Internet based social networks such as Facebook[9], MySpace[10], Hi5[11], orkut[12] and Friendster[13].

Reputation based approaches are typically sender oriented but Abaca Technology, a company in San Jose, CA, has developed a recipient reputation based filtering system. Their spam-focused technology, called ReceiverNet, aggregates the reputations of recipients of a message to determine if a message is legitimate or spam. Some email addresses are considered more likely to receive spam and others are considered less likely to receive spam. This approach is beneficial when a spammer sends an email to a large number of recipients (Abaca Technology, 2007).

One feature of targeted malicious email noted in chapter one is that threat actors will spoof known senders for a particular recipient. A fundamental assumption in most of the reputation based approaches for filtering email is that senders are authenticated, if not, a trivial spoofing of the email *From* address will significantly degrade the effectivity of these approaches.

[9] http://www.facebook.com
[10] http://www.myspace.com
[11] http://hi5.com
[12] http://www.orkut.com
[13] http://www.friendster.com

2.3.5 Resource Consumption

Instead of only passively filtering email, resource consumption based approaches actively increase cost to senders. Costs are increased through increased use of resources such as network bandwidth or computing power. Threat actors, for example, may need more time to send the same number of emails or they will have to identify new relays or targets to evade resource constraining email servers.

Tran and Armitage (2004) propose an approach that leverages reputation and contextual based approaches to detect spam early in the email transmission. Bandwidth is then reduced or latency increased to the network flow associated with that email. By doing this, connections take longer to complete and result in more resources being needed by the sender to send the same quantity of email when no delays are introduced. To be effective, emails have to be flagged as spam early to significantly impact sender resources. Ultimately, all email is delivered resulting in a much lower penalty for false positives than outright rejection but unwanted email appears in users' mailboxes at a lower rate. Li et al. (2004) also slow network flows associated with unwanted email but do so using a contextual rules-based approach that identifies spam early in the transmission and does not acknowledge received network packets or changes the transmission window back to the sender. This approach results in spammer throughput being reduced hundreds of times.

Marsono et al. (2007) describe a two queue mail system that classifies email as spam or not spam before handing it off to the appropriate queue. Modeling by using queueing theory, Marsono et al. show that they can reduce the load on email servers and slow the delivery rate of unwanted email by rejecting email when the unwanted email queue is full.

Two final resource consumption based approaches introduce cost to the sender by requiring senders to prove some extra computation was done before accepting an email. Back (2002) introduces *HashCash* a system that requires senders to solve a cryptographic puzzle to send email to a particular recipient. This puzzle requires computing time on the sender's side to be solved. A similar approach is presented

by Dwork et al. (2003) of Microsoft Research but these researchers use a memory bound approach instead of the CPU bound approach proposed by Back. Dwork et al. demonstrate that memory bound approaches may make the differences between older and newer systems in terms of CPU computational power less relevant.

Resource consumption based techniques for filtering email are largely focused on threat actors who send large amounts of email. This activity is typically associated with spammers. In all of the approaches, unwanted email is still ultimately delivered to the recipient but senders are not able to send the large volume of email they would normally send without resource consumption constraints. Additionally, the impact of these techniques is felt by the last email relay in a chain of email relays. If a particular email relay is being used for both legitimate and illegitimate email, it will suffer the same impact and the threat actor's system may incur no penalty because it is earlier in the email relay chain. Finally, large networks of compromised machines, such as botnets, may be able to overcome computational or bandwidth limitations through sheer brute force.

2.4 Existing weaknesses

Existing techniques for filtering email have limitations when applied to targeted malicious email. Authentication based techniques require receivers to enforce domain level authentication upon email receipt. Since these techniques are not fully adopted across the Internet, enforcing the authentication at all times is not possible. To complicate matters, the authentication is at the domain level, not on a per email address basis. Thus, a public webmail provider like Google, may be authenticated but a threat actor may have created an email account for the purposes of sending targeted malicious email. Contextual approaches typically focus on message content, making classification decisions largely on the words in the body of an email. From a threat actor's perspective, message content is the easiest to change and thus is not very durable across multiple email campaigns from the same threat actor. Furthermore, since targeted malicious emails often have message content very specific to the recipient, finding common words across emails is not as relevant as it is with spam. Characterization based

approaches to filtering email usually involve quantifying aspects of email volume; low volume attacks such as targeted malicious emails are likely to remain undetected. With targeted malicious email, known email addresses and names are used which hampers the effectivity of reputation based approaches. Finally, resource consumption based techniques for filtering email are largely focused on malicious actors who send large amounts of email, typically spammers.

Targeted malicious emails are low volume and directed at certain recipients, which is in contrast to spam which is often directed at numerous recipients and is of high volume. Existing approaches to filtering email are focused on specific attacks but do not leverage features that are more durable and possibly common across a set of attacks from a particular threat. As outlined in Figure 2.3, threat actors have to execute multiple steps to achieve their objective. Some of these steps, such as weaponization, are more complicated to make unique from attack to attack. Tools such as anti-virus typically intervene fairly late in the threat kill chain with little insight into steps such as reconnaissance. By focusing on steps in the kill chain that are more difficult for the threat actor to readily manipulate, greater detection capability for targeted malicious emails can be achieved.

Chapter 3: Research Goals and Hypotheses

3.1 Research Goals

The goal of this research was to develop new methods for filtering email that are specifically designed to address the threat posed by targeted email attacks employed by advanced threat actors. Conventional approaches to filtering email are well suited to Internet-scale email abuse, such as spam and phishing, but do not readily apply to targeted email attack scenarios. Introduction of these new methods required the creation of software modules to integrate into an existing email detection architecture. The effectivity of these new techniques are evaluated to determine if the introduction of these new techniques adds statistically significant improvement over conventional approaches. The primary focus of methods described in this research is high sensitivity (e.g. low false negatives) detection.

The goals of this research can be summarized as follows:

1. To develop a framework that incorporates an array of email features that can be applied to email filtering decision logic.

2. To identify any association between targeted malicious email and persistent threat or recipient oriented features of email.

3. To measure the effectivity of email filtering that leverages persistent threat or recipient oriented features of email as compared to conventional email filtering that does not.

3.2 Hypotheses

This research has the following associated hypotheses:

H1 Targeted malicious email demonstrates association to persistent threat features of email such as locale and tools as compared to non-targeted malicious email that does not show an association to persistent threat features.

H2 Targeted malicious email demonstrates association to recipient oriented features such as role, reputation, relationships and access as compared to non-targeted malicious email that does not show an association to recipient oriented features.

H3 Detection of targeted malicious email using persistent threat and recipient oriented features results in fewer false negatives than detection of targeted malicious email using conventional email filtering techniques.

Researching these hypotheses requires samples of both targeted malicious and non-targeted malicious email. These emails were obtained from a large organization that has been subjected to targeted email attacks.

Chapter 4: Research Method

In this study, new techniques for email filtering and detection of targeted malicious email (TME) are introduced and analyzed. Detecting TME requires building a classifier that incorporates a variety of email features not used in conventional email filtering techniques. One shortcoming of conventional email filtering techniques is their general reliance on sender controlled and easily manipulated parameters such as email content. By augmenting these conventional methods with new features that can not be readily manipulated by threat actors, such as recipient oriented features and indicators left by weaponization tools, detection methods are more durable. Additionally, it is important to remember that threat actors are humans that make mistakes; examining, in detail, the clues they leave behind leads to the development of a threat specific detection capability. As attacks become more targeted, less voluminous, and more surgical, appropriate defenses must also be tuned carefully to keep pace.

This chapter will cover: a description of the data used in this study, statistical methods to analyze the data, a thorough description of persistent threat and recipient oriented features of email, an outline of the software created to execute this study, a description of the classifier applied to the data, and an outline of the methods used to compare new and conventional detection techniques.

4.1 Data

Typically, data sets used to evaluate email filtering techniques are incomplete or an amalgamation of several different data sets. For example, the PU1[1] and ling-spam[2] corpora commonly used for evaluating the performance of spam filters are mixtures of known spam and known legitimate emails from different sources (Androutsopoulos et al., 2000a). Privacy concerns make it difficult to obtain legitimate email for analysis and to further complicate matters, data sets sometimes lack email header information

[1]PU1 Corpora - http://www.iit.demokritos.gr/skel/i-config/downloads/pu1_encoded.tar.gz

[2]Ling-spam Corpora - http://www.iit.demokritos.gr/skel/i-config/downloads/lingspam_public.tar.gz

or are sanitized to the point where useful information is lost. Since the bulk of email filtering research is in the text classification area, a lack of email headers has not generally presented a problem. Although, it is possible that the research has been focused on text classification due to the absence of data that would allow otherwise.

This study aims to measure the added value of including features of malicious email that are persistent threat and recipient specific. Since recipient specific features require additional context beyond what is available in an email itself, typical publicly available corpora do not suffice. The data used in this study, came from a large Fortune 500 company, with more than 100,000 users, which has been exposed to targeted malicious email. Additional recipient context, such as job function, was added from an internal company directory. In a security incident handling system, the company has recorded emails which have been manually identified as targeted and malicious.

4.1.1 Data use approvals

Use of this data was fully reviewed with the company's legal counsel and information security personnel. While research was conducted using actual data, all results have been sanitized to anonymize both the company and any users of the company's email system. Throughout the study, the company's name is substituted with "company" or "example" if used in the context of a domain name. Any sensitive features which would reveal too much about the company's detection capability are redacted for security purposes.

4.1.2 Data sets created and used

Multiple data sets were created and used in this study. The following sections will describe each one in detail.

User Information

To provide recipient context for email, attributes of the company's email users were mirrored from the company's directory service and inserted into a separate relational database for fast lookup. Table 4.1 contains all of the recorded attributes for each

user. Many of these attributes came directly from the internal company directory and a few, such as *google_search_count*, were added based on other collected data. The field, *business_area*, was manually created by normalizing the *company* field and providing a mapping of business units (i.e. *company*) to business areas. As with any large organization, not all of the data was consistent and had to be cleaned up to create a common format across the entire user population. The field, *title_short*, was also manually generated and based on the *title* field. The *title_short* field was used to group like disciplines, such as Finance or Systems Engineering, together. Email addresses for everyone in the company were checked against Google, using the Google API[3], and the number of search results for every email address was recorded in the field *google_search_count*. Finally, the field *num_tme_received*, contains a count of the number of known TME received by this user since the beginning of the study.

Table 4.1: Per user fields from company directory

Field Name	**Description**
id	Internal unique identifier. This does not come from the directory service
cn	Common Name for a user, typically "Last Name, First Name Middle Initial" (e.g. "Doe, John X.")
sn	Last Name
givenname	First Name
middlename	Middle Name
initials	Middle initials
sAMAccountName	User account name
mail	User email address
companyEmployeeID	Company internal employee ID
company	Business unit for this user
business_area	Business area for this user. Business areas comprise multiple business units
locationdescription	Facility or campus where user is located
st	State where user is located
title	Full title of user
title_short	Shortened title which groups like disciplines together
level	Job level of the user (e.g. level 1 is an entry level user, level 10 is the CEO)
userprincipalname	User account string including directory information
c	Country where user is located

Continued on next page...

[3]Google Code - http://code.google.com

Table 4.1 – Continued

Field Name	Description
extensionattribute2	Additional attributes about the user
manager	User's manager's name
manager_id	User's manager's internal employee ID
distinguishedName	Full directory identifier for this user
google_search_count	Number of times this user's email address appears in Google search results
num_tme_received	Number of times, since the study started, user has been the recipient of a targeted malicious email
memberOf	Recorded in a linked table, this field contains all of the group memberships that this user has recorded as part of their account information

Non-targeted malicious email data set

A non-targeted malicious email data set consisting of 20,894 random emails from a 2.5 month period from September 1, 2009 through November 20, 2009 was assembled. This data set only includes emails from the Internet to the company, no intra-company emails were included. This data set includes only those emails that were passed by a commercial anti-spam system. Thus, emails that were classified as generic Internet spam by the commercial anti-spam system were removed from this data set. Additionally, email in this data set was not processed by a commercial anti-virus system (the email collection point was pre anti-virus). Email collection was facilitated using a Linux-based collection system, connected to the network via a network tap. The Linux system ran the Vortex-IDS[4] which allowed reconstruction of email off the network wire into files. Table 4.2 summarizes this data set, referred to as *NTME1* in this study.

Table 4.2: *NTME1* - Non-targeted malicious email data set

Data set name	*NTME1*
Date Span	September 1, 2009 through November 20, 2009
Total Emails	20,894 (unique)

[4]Vortex-IDS - http://sourceforge.net/projects/vortex-ids/

Targeted malicious email data set

Through manual computer forensics and information sharing with a community knowledgeable with targeted malicious email, this company retroactively identified targeted malicious emails. These emails were manually reviewed to confirm their membership to this classification of email. This data set consists of 2,315 emails from April 16, 2009 through December 19, 2009. The time period is longer due to the very low number of targeted malicious emails received as compared to non-targeted malicious email. Table 4.3 summarizes this data set, referred to as *TME1* in this study.

Table 4.3: *TME1* - Targeted malicious email data set

Data set name	*TME1*
Date Span	April 16, 2009 through December 19, 2009
Total Emails	2,315 (unique)

Joint non-targeted malicious and targeted malicious email data set

For supervised learning, testing, and feature importance a joint *NTME1* and *TME1* data set is used. Table 4.4 summarizes this data set, referred to as *NTME1-TME1* in this study. *TME1* represents 9.97% of the joint data set.

Table 4.4: *NTME1-TME1* - Joint non-targeted malicious and targeted malicious data set

Data set name	*NTME1-TME1*
Date Span	April 16, 2009 through December 19, 2009
Total Emails	23,209 (unique)

Spam recipients data set

A spam recipients data set was constructed using spam log data from a 2.5 month period from September 1, 2009 through November 20, 2009. Full emails were not available, only transactional logs. Thus, this data set was used only for determining if

some recipient oriented characteristics differ in spam emails as compared to targeted malicious email. Table 4.5 summarizes this data set, referred to as *SP1* in this study.

Table 4.5: *SP1* - Spam recipients data set

Data set name	*SP1*
Date Span	September 1, 2009 through November 20, 2009
Total Email Addresses	666,602 (non unique)

Test only email data set

As an added step to measure the effectiveness of methods developed in this study, a separate email data set was created. This data set was not used in training at all and only for evaluation purposes once a classifier model had already been created. This data set consists of 1,457,729 emails from December 22, December 24 and December 30, 2009. This represents a full three days of post spam filtered email. These three days were selected since the company had discovered targeted malicious emails on these days as a result of internal intelligence analysis and sharing with industry partners. The company's security team identified 44 targeted malicious emails in these three days. Table 4.6 summarizes this data set, referred to as *TS1* in this study.

Table 4.6: *TS1* - Test only data set

Data set name	*TS1*
Date Span	December 22,24,30 2009
Total Emails	1,457,729
Total TME	44

4.2 Software and Database

4.2.1 Software

In this study free and open-source software was leveraged for data collection and classification, custom software was written for everything else.

Free and open-source software

To facilitate the collection of email, the Vortex-IDS was used. "Vortex is a near real time IDS and network surveillance engine for TCP stream data. Vortex decouples packet capture, stream reassembly, and real time constraints from analysis. Vortex is used to provide TCP stream data to a separate analyzer program" (Smutz et al., 2010).

To execute the classification algorithms needed in this study, the Waikato Environment for Knowledge Analysis (WEKA) data mining toolkit was used (Hall et al., 2009). WEKA is an open source Java based application that processes data using a variety of machine learning algorithms. In addition, the R Project for Statistical Computing was used as a supplementary tool for feature analysis and feature importance calculations (R Development Core Team, 2009).

Custom developed software

The software created for this study facilitated both data collection and data normalization. Software was created to perform the following functions:

- Directory Information - Perl[5] scripts were created to authenticate against the company's directory service as well as iterate through all person objects and extract the fields outlined in Table 4.1. The Lightweight Directory Access Protocol (LDAP) was used to interface with the directory. Several helper scripts were written to clean and normalize information (e.g. grouping titles into categorical short titles as seen in Table 4.1). Scripts were also created to insert the extracted information into a relational database.

- Email Features - Perl scripts were created to extract relevant features from email. A base interface to parse email was created by the company for its internal detection objectives. This interface was modified and extended to support the needs of this study. The scripts created converted email fields into feature vectors suitable for import into a classification tool.

[5]Perl - http://www.perl.org - Perl is an interpreted programming language

- Google Search - Leveraging the Google Search API[6], scripts were created to query Google for each user's email address and record the number of search hits for that email address. For this study, the number of Google search hits was recorded as of the time of data analysis not when the actual email was received. This information was recorded in Table 4.1.

4.2.2 Database

The database used in this study was MySQL[7] which is a popular open-source relational database. The database contained user information that was mirrored from the company's directory service. This mirroring was done to minimize the interaction with the company directory during the software development and test phase of the study. A database is not necessarily required as queries can be executed directly against the directory service using the Lightweight Directory Access Protocol (LDAP). A number of indices were created to increase query time for certain key fields such as the user's email address.

4.3 Statistical methods

The principal method of analysis for this research lay in the use of statistical testing. The application of specific inferential tests as well as machine learning applied statistical methods were used.

4.3.1 Inference for proportions

In later sections, the data sets in this study will be analyzed with respect to email feature proportions. Some of the data sets demonstrate proportion differences with certain email features that are relevant for enhancing email filtering. Devore (2004) describes how to compare population proportions with large samples and that method is adapted below for this study.

[6]Google AJAX Search API - http://code.google.com/apis/ajaxsearch/
[7]MySQL - http://www.mysql.com

Assume $p_1$and p_2 denote the proportions of email recipients in populations 1 and 2, respectively, who have a certain characteristic (e.g. job title). As an example, populations 1 and 2 could be spam recipients and targeted malicious email recipients. Further, assume there are a sample of m recipients from the first population and n from the second. Finally, let independent random variables X and Y represent the number of email recipients in each population sample having a certain characteristic. It is assumed that there are at least 10 spam and non-spam recipients, along with at least 10 targeted malicious and non-targeted malicious recipients. Additionally, the two samples are random samples that are less than 10% of their respective populations and the samples were selected independent of each other.

Hypothesis Testing

The null hypothesis for comparing two populations is that the two proportions are the same:

$$\begin{aligned} H_0 &: p_1 - p_2 = 0 \\ H_0 &: p_1 = p_2 \end{aligned} \tag{4.1}$$

Depending on the proportions being compared the alternative hypothesis can be one-sided (showing greater-than or less-than) or two-sided (simply showing the proportions are different):

$$\begin{aligned} H_A &: p_1 - p_2 > 0 \quad (p_1 > p_2) \\ H_A &: p_1 - p_2 < 0 \quad (p_1 < p_2) \\ H_A &: p_1 - p_2 \neq 0 \quad (p_1 \neq p_2) \end{aligned} \tag{4.2}$$

Since the population sizes are much larger than the sample sizes in this study, the distribution of X can be assumed binomial with parameters m and p_1 and Y can be assumed binomial with parameters n and p_2. The best estimate for the difference in population proportions, $p_1 - p_2$, is the difference in sample proportions $X/m - Y/n$.

Setting $\hat{p_1} = X/m$ and $\hat{p_2} = Y/n$, the estimator of $p_1 - p_2$ can be written as $\hat{p_1} - \hat{p_2}$. Since X and Y are assumed binomial, the expected values are $E(X) = mp_1$ and $E(Y) = np_2$. Equation 4.3 derives the formula for the expected value of the difference in population proportions.

$$\begin{aligned} E(\hat{p_1} - \hat{p_2}) = E\left(\frac{X}{m} - \frac{Y}{n}\right) &= \frac{1}{m}E(X) - \frac{1}{n}E(Y) \\ &= \frac{1}{m}mp_1 - \frac{1}{n}np_2 \\ &= p_1 - p_2 \end{aligned} \tag{4.3}$$

The variance for the difference in population proportion is shown in Equation 4.4 where $V(X) = mp_1q_1$ and $V(Y) = np_2q_2$.

$$\begin{aligned} V(\hat{p_1} - \hat{p_2}) = V\left(\frac{X}{m} - \frac{Y}{n}\right) &= V\left(\frac{X}{m}\right) + V\left(\frac{Y}{n}\right) \\ &= \frac{1}{m^2}V(X) + \frac{1}{n^2}V(Y) \\ &= \frac{p_1q_1}{m} + \frac{p_2q_2}{n} \end{aligned} \tag{4.4}$$

Since the standard deviation (σ) is equal to the square-root of the variance, the standard deviation of the proportion difference is:

$$\sigma(\hat{p_1} - \hat{p_2}) = \sqrt{\frac{p_1q_1}{m} + \frac{p_2q_2}{n}} \tag{4.5}$$

The proportions will be compared using z test statistic:

$$Z = \frac{x - \mu}{\sigma} \tag{4.6}$$

The test has to be carried out assuming H_0 is true which means that $p_1 - p_2 = 0$. Therefore, $E(\hat{p_1} - \hat{p_2}) = \mu = 0$. With $p_1 = p_2$ the Z test statistic becomes:

$$Z = \frac{\hat{p_1} - \hat{p_2}}{\sqrt{pq\left(\frac{1}{m} + \frac{1}{n}\right)}} \tag{4.7}$$

The null hypothesis, H_0, does not specify a common value of p_1 and p_2 so it has to be estimated. Since both populations are assumed to have the same p, both sample populations can be pooled together:

$$\hat{p}_{pooled} = \hat{p} = \frac{X+Y}{m+n} = \frac{m}{m+n}\hat{p_1} + \frac{n}{m+n}\hat{p_2} \tag{4.8}$$

Using this new pooled estimate of p and defining $\hat{q} = 1 - \hat{p}$ the Z test statistic becomes:

$$Z = \frac{\hat{p_1} - \hat{p_2}}{\sqrt{\hat{p}\hat{q}\left(\frac{1}{m} + \frac{1}{n}\right)}} \tag{4.9}$$

A hypothesis test can be performed using this Z test statistic to determine whether to accept or reject the null hypothesis when comparing two populations with respect to a certain characteristic. If the null hypothesis is that the two proportions are the same (e.g. $p_1 = p_2$) then the P-value is the probability that the Z test statistic is less than or greater than the Z critical value for that Z test statistic. This is known as a two-tailed test. To reject the null hypothesis at the $\alpha = 0.05$ level of significance, the P-value has to be less than 0.05 which corresponds to a Z test statistic greater than 1.96. To reject the null hypothesis at the $\alpha = 0.01$ level of significance, the P-value has to be less than 0.01 which corresponds to a Z test statistic greater than 2.58. However, if the null hypothesis is that one proportion is greater (or less) than another proportion (e.g. $p_1 > p_2$), then the P-value is the probability that the Z test statistic is greater (or less) than the Z critical value for that Z test statistic. This is known as a one-tailed test. To reject the null hypothesis at the $\alpha = 0.05$ level of significance, the P-value has to be less than 0.05 which corresponds to a Z test statistic greater than 1.64. To reject the null hypothesis at the $\alpha = 0.01$ level of significance, the P-value has to be less than 0.01 which corresponds to a Z test statistic greater than 2.33. The Z test statistics can be found in the Z-tables in most statistics textbooks.

Confidence Intervals

Confidence intervals provide an interval estimate for $p_1 - p_2$. The confidence interval can be calculated by:

$$\hat{p_1} - \hat{p_2} \pm Z \cdot \sigma = \hat{p_1} - \hat{p_2} \pm Z\sqrt{\frac{p_1 q_1}{m} + \frac{p_2 q_2}{n}} \tag{4.10}$$

For a two-tailed 95% confidence interval, $Z = 1.96$ and for a two-tailed 99% confidence interval, $Z = 2.58$. If only a lower bound or upper bound is needed, then a one-tailed confidence interval can be calculated. For a one-tailed 95% confidence interval, $Z = 1.64$ and for a one-tailed 99% confidence interval, $Z = 2.33$.

4.3.2 Inferences Based on Two Samples

At times it may be necessary to compare values obtained from two independent random samples to determine if there is a statistically significant difference between the values. Assuming $X_1, X_2, \ldots, X_m$ is a random sample from a population with mean μ_1 and variance σ_1^2, and $Y_1, Y_2, \ldots, Y_m$ is a random sample from a population with mean μ_2 and variance σ_2^2, a two-sample t test can be conducted to determine statistical significance. Devore (2004) outlines a two-sample t test where the sample sizes may be unequal and the population variances have unknown values.

Hypothesis Testing

The null hypothesis for comparing two means is that the two means are the same (e.g. $\mu_1 - \mu_2 = 0$).

$$\begin{aligned} H_0 &: \mu_1 - \mu_2 = 0 \\ H_0 &: \mu_1 = \mu_2 \end{aligned} \tag{4.11}$$

Depending on the test being performed, the alternative hypothesis can be one-sided (showing greater-than or less-than) or two-sided (simply showing the means are

different):

$$\begin{aligned} H_A &: \mu_1 - \mu_2 > 0 \quad (\mu_1 > \mu_2) \\ H_A &: \mu_1 - \mu_2 < 0 \quad (\mu_1 < \mu_2) \\ H_A &: \mu_1 - \mu_2 \neq 0 \quad (\mu_1 \neq \mu_2) \end{aligned} \tag{4.12}$$

The test statistic, where $\bar{x}$ and $\bar{y}$ are the means of X and Y, m and n are the sample sizes, and s_1 and s_2 are the standard deviations of X and Y, is:

$$t = \frac{\bar{x} - \bar{y}}{\sqrt{\frac{s_1^2}{m} + \frac{s_2^2}{n}}} \tag{4.13}$$

The degrees of freedom, *df*, is calculated by:

$$df = \frac{\left(\frac{s_1^2}{m} + \frac{s_2^2}{n}\right)^2}{\frac{(s_1^2/m)^2}{m-1} + \frac{(s_2^2/n)^2}{n-1}} \tag{4.14}$$

Confidence Intervals

The two-sample two-sided t confidence interval for $\mu_1 - \mu_2$ with confidence level $100(1-\alpha)\%$ is:

$$\bar{x} - \bar{y} \pm t_{\alpha/2,v}\sqrt{\frac{s_1^2}{m} + \frac{s_2^2}{n}} \tag{4.15}$$

The corresponding one-sided confidence interval is:

$$\begin{aligned} \text{Upper Bound} &: \bar{x} - \bar{y} + t_{\alpha,v}\sqrt{\frac{s_1^2}{m} + \frac{s_2^2}{n}} \\ \text{Lower Bound} &: \bar{x} - \bar{y} - t_{\alpha,v}\sqrt{\frac{s_1^2}{m} + \frac{s_2^2}{n}} \end{aligned} \tag{4.16}$$

4.3.3 McNemar test for comparing classifiers

In this study the McNemar test, using a χ^2 distribution, is used to compare whether two classifiers differ significantly in the ability to detect targeted malicious email.

Specifically, this statistical test will be used to evaluate whether detection of targeted malicious email using persistent threat and recipient oriented features results in fewer false negatives than detection of targeted malicious email using conventional email filtering techniques (see Section 3.2). As a non-parametric test the McNemar test, unlike the t-test, does not make any assumptions about distribution(Everitt, 1977). Given two targeted malicious email (TME) detection methods A and B, the following variables are defined: n_{00} is the number of TME missed by both A and B, n_{01} is the number of TME missed by A but not by B, n_{10} is the number of TME missed by B but not by A, and n_{11} is the number of TME detected by both A and B. The total number of TME being tested is n where $n = n_{00} + n_{01} + n_{10} + n_{11}$ (Salzberg, 1997; Dietterich, 1998). Table 4.7 summarizes these outcomes.

Table 4.7: McNemar Contingency Table

	A-Correct	A-Error
B-Correct	n_{11}	n_{01}
B-Error	n_{10}	n_{00}

Hypothesis Testing

The null hypothesis for comparing two detection methods is that there is no difference in the ability to detect targeted malicious email. Stated differently, the two methods should have the same error rate, which means that $n_{01} = n_{10}$. Under the null hypothesis, the expected counts are shown in Table 4.8. The two counts where the methods agree, n_{00} and n_{11}, are not relevant for determining the difference in detection ability between the two methods.

Table 4.8: McNemar Test: Null hypothesis expected counts

	A-Correct	A-Error
B-Correct	n_{11}	$(n_{01} + n_{10})/2$
B-Error	$(n_{01} + n_{10})/2$	n_{00}

The alternative hypothesis is that the two classifiers are different in ability to detect targeted malicious email. The test statistic is chi-square distributed with 1

degree of freedom:

$$\chi^2 = \frac{(|n_{01} - n_{10}| - 1)^2}{n_{01} + n_{10}} \tag{4.17}$$

The null hypothesis is rejected if χ^2 is greater than $\chi^2_{1,0.05} = 3.841$ at an $\alpha = 0.05$ level of significance or greater than $\chi^2_{1,0.01} = 6.635$ at an $\alpha = 0.01$ level of significance.

4.3.4 Correlation Analysis

Some of the features in the data sets show some correlation between them. To measure the extent of correlation between two quantitative variables, Pearson's Product Moment Correlation Coefficient, r, will be calculated. The value of r ranges from -1.0 to 1.0, where positive values indicate positive correlation, negative values indicate negative correlation, and values near 0 indicate little or no correlation.

$$r = \frac{\Sigma(X - \mu_X)(Y - \mu_Y)}{N\sigma_X\sigma_Y} \tag{4.18}$$

To determine if the correlation coefficient is significant, either a one-tailed or two-tailed t-test will be used. If there is a reason to hypothesize that X tends to increase (or decrease) with respect to Y then a one-tailed t-test should be used. If there is no good reason to expect positive or negative correlation a two-tailed t-test will be used. The null hypothesis for a one-tailed t-test is that the correlation coefficient is equal to 0 and the alternative hypothesis is that the correlation coefficient is not equal to 0.

$$\begin{aligned} H_0 &: r = 0 \\ H_A &: r \neq 0 \end{aligned} \tag{4.19}$$

For a two-tailed t-test the null hypothesis is that the correlation coefficient is equal to 0 and the alternative hypothesis is that the correlation coefficient is greater than zero

(to test for positive correlation) or less than zero (to test for negative correlation).

$$\begin{aligned} H_0 &: r = 0 \\ H_A &: r > 0 \quad \text{(for positive correlation)} \\ H_A &: r < 0 \quad \text{(for negative correlation)} \end{aligned} \tag{4.20}$$

To conduct the test, the significance level has to be set (e.g. $\alpha = 0.05$) and the critical value of r is needed from a table of critical values of the correlation coefficient. For a one-tailed test, the degrees of freedom, $df = n - 1$, for a two-tailed test, the degrees of freedom, $df = n - 2$, where n is the number of observations. If r is less than or equal to the critical value or r then the null hypothesis is accepted. If r is greater than the critical value of r then the null hypothesis is rejected and the alternative hypothesis is accepted.

4.4 Email analysis procedures

This study explores detection and filtering techniques incorporating two types of basic features:

1. Persistent Threat - During the *weaponization* stage of the kill chain (see Section 2.2.2), a threat actor needs to weaponize an email so that upon *delivery* it will result in a system compromise that facilitates unauthorized access. Weaponizing an email involves various tools and other procedures that leave fingerprints useful for detection purposes. Inevitably, threat actors resort to automation or other procedural techniques that can enhance detection across a number of attacks. There is a cost to creating an email weapon so threat actors may reuse weapons with different delivery vehicles to achieve a measure of reuse. The combination of tools, techniques, procedures, and infrastructure used by a threat actor measure its capability.

2. Recipient Oriented - During the *reconnaissance*, *weaponization*, and *delivery*

stages of the kill chain (see Section 2.2.2), a threat actor needs to define email recipients, ensure a measure of relevancy to the recipient, and deliver a malicious email to those recipients. The targeting frequency of recipients by threat actors speaks to the intent of the threat.

4.4.1 Persistent threat features

Locale and tool are two types of persistent threat features that were incorporated into email filtering techniques for this study. A broader set of persistent threat features can be found in Section 2.2.2.

Locale

When a threat actor is preparing and launching an email weapon, certain elements of the threat actor's locale may be left in the email itself. If the threat actor is using a malicious attachment as the malicious payload then the attachment may also have indications of the threat actor's locale. Locale can be inferred through language settings, character encoding, time zone settings, and Internet Protocol (IP) addresses and system host names.

Tool

Threat actors sometimes use automated tools to facilitate email weapon creation or delivery. These tools will often leave fingerprints that can be incorporated into an email filtering scheme. Some tools actually leave its name in an email and other tools leave other more subtle clues.

4.4.2 Recipient Oriented Features

There are numerous recipient oriented features that can be incorporated into email filtering techniques: role, relational, access and reputation. In this study, role and reputation based features were the focus. To illustrate various scenarios, fictitious individuals Alice Roberts (alice.roberts@example.com) and Bob Smith (bob.smith@example.com) who both work for Example.com will be used.

Role

A threat actor may send an email to a particular individual because of their role in an organization. For example, if Alice Roberts was the Chief Executive Officer (CEO) of Example.com, a threat actor may target her thinking her system may have sensitive information. In another scenario, Bob Smith who works in business development, might be prone to receiving targeted emails simply because as a function of his job his email address information is more readily available. Individuals in certain roles may also have sensitive data of interest to a threat actor. A recipient's role could be a job title or also a job level (e.g. an entry level 1 employee vs. a senior level 3 employee).

Relational

Targeted malicious email may use a spoofed *From* address that is relevant to the recipient. Assume that Alice is Bob's manager. An example of a targeted email using a spoofed *From* address relevant to the recipient would be a targeted email sent to Bob spoofed using Alice's real email address (alice.roberts@company.com). Another example would be a targeted email sent to Bob spoofed using Alice's name in an email address at a public email provider like Yahoo! (alice.roberts@yahoo.com or aliceroberts@yahoo.com). In either case, employee Bob may believe he is receiving a legitimate message from his manager.

Other relational characteristics include the proximity of two email addresses on a publicly available webpage. For example, if two email addresses appear together on a publicly available webpage, that could be indicative of a relationship between the users behind those two addresses and something a threat actor may want to exploit.

Access

Targeted malicious email may be sent to a particular recipient based on the desire of a threat actor to gain access to certain information. For example, if Bob Smith is an administrator for a large number of systems at Example.com or if he has access to sensitive information he may be more prone to receiving a targeted malicious email.

Since access to sensitive information is generally restricted to those who need it, access may be closely associated with particular projects or activities that an employee supports. If a threat actor is after information about 'Project X', someone who has access to 'Project X' information is probably someone who is supporting that project.

Reputation

Just as some sender focused reputation approaches to filtering email maintain lists of known bad senders, recipient reputation involves maintaining a list of recipients known to receive targeted malicious email. It is conceivable that threat actors maintain a database of email addresses for a specific target organization and that these email addresses may receive more than one targeted malicious email over time. Another dimension of reputation includes email visibility. Presumably, those email addresses that are more publicly known and available are likely to be targets of unwanted email. Email address visibility can be as straightforward as the number of times an email address appears in an Internet search engine (e.g. Google.com). Furthermore, employees who have left a company may continue to receive targeted malicious email to their no-longer-valid email address as their email address will still exist in threat actor databases or still appear on websites affiliated with a particular technology.

4.4.3 Detailed List of Features

In this study, a number of persistent threat and recipient oriented features were extracted and calculated from emails being analyzed. Decisions on features to expose for use in a classification algorithm were based on implementation complexity and problem set relevance. Table 4.9 contains a summarized listing of features extracted and exposed for all emails analyzed in this study. The columns for Table 4.9 are defined as follows:

- Feature Name - Unique assigned feature name.
- Category (Cat.) - Category of this feature. "T" for persistent threat feature and "R" for recipient oriented feature.

- Type - Type of this feature. "N" for numeric, "B" for binary, and "C" for categorical.
- Description - Short description of this feature.

To anonymize the name of the company whose data was used in this study, "company", "example" and "example.com" are used throughout the feature listing.

Table 4.9: Detailed List of Extracted Email Features: Category: "T" for persistent threat, "R" for recipient oriented; Type: "N" for numeric, "B" for binary, "C" for categorical

Feature Name	**Cat.**	**Type**	**Description**
attachment	T	B	1, if an attachment exists, 0 else.
attachment_doc	T	B	1, if a .doc (Microsoft Word) [8] attachment exists, 0 else.
attachment_htm	T	B	1, if a .htm attachment exists, 0 else.
attachment_mdb	T	B	1, if a .mdb (Microsoft Access) attachment exists, 0 else.
attachment_pdf	T	B	1, if a .pdf (Adobe PDF) [9] attachment exists, 0 else.
attachment_ppt	T	B	1, if a .ppt (Microsoft Powerpoint) attachment exists, 0 else.
attachment_xls	T	B	1, if a .xls (Microsoft Excel) attachment exists, 0 else.
cc_empty	T	B	1, if the Cc: line is present but empty, 0 else.
cc_no_example	R	B	1, if the Cc: line does not include someone from the company, 0 else.
char_encoding_base64	T	B	1, if the email uses base64 encoding, 0 else.
char_encoding_big5	T	B	1, if the email uses big5 encoding, 0 else.
char_encoding_gb2312	T	B	1, if the email uses gb2312 encoding, 0 else.
char_encoding_gbk	T	B	1, if the email uses gbk encoding, 0 else.

Continued on next page...

[8] http://office.microsoft.com
[9] http://www.adobe.com

Table 4.9 – Continued

Feature Name	Cat.	Type	Description
char_encoding_windows1252	T	B	1, if the email uses windows1252 encoding, 0 else.
date_header_timezone	T	C	Time zone offset (from Greenwich Mean Time) of the Date: header field.
dkim_header_defined	T	B	1, if a DKIM header is present, 0 else.
email_size	T	N	Size of the email in bytes.
envelope_recipients_invalid_percentage	R	N	The percentage of invalid recipients defined in the email envelope.
envelope_recipients_invalid_total	R	N	The number of invalid recipients defined in the email envelope.
envelope_recipients_total	R	N	The number of recipients (valid or invalid) defined in the email envelope.
envelope_recipients_valid_addresses_alpha_ordered	T	B	1, if the envelope recipients are listed in alphabetical order, 0 else.
envelope_recipients_valid_avg_google_search_count	R	N	For all valid envelope recipients, the average number of Google search hits for the respective email addresses.
envelope_recipients_valid_avg_job_level	R	N	For all valid envelope recipients, the average job level.
envelope_recipients_valid_avg_num_tme_received	R	N	For all valid envelope recipients, the average recipient reputation of previous targeted malicious emails received.
envelope_recipients_valid_total	R	N	The number of valid recipients defined in the email envelope.
envelope_recipients_valid_total_business_area_A	R	N	The number of valid envelope recipients in Business Area "A".
envelope_recipients_valid_total_business_area_E	R	N	The number of valid envelope recipients in Business Area "E".

Continued on next page...

Table 4.9 – Continued

Feature Name	Cat.	Type	Description
envelope_recipients_valid_total_business_area_E2	R	N	The number of valid envelope recipients in Business Area "E2".
envelope_recipients_valid_total_business_area_I	R	N	The number of valid envelope recipients in Business Area "I".
envelope_recipients_valid_total_business_area_S	R	N	The number of valid envelope recipients in Business Area "S".
envelope_recipients_valid_total_title_short_bus_devel	R	B	The number of valid envelope recipients with job title "Bus Devel" (Business Development).
envelope_recipients_valid_total_title_short_bus_devel_analysis	R	B	The number of valid envelope recipients with job title "Bus Devel Analysis" (Business Development Analyst).
envelope_recipients_valid_total_title_short_communications	R	B	The number of valid envelope recipients with job title "Communications".
envelope_recipients_valid_total_title_short_international_bus_dev	R	B	The number of valid envelope recipients with job title "International Business Development".
envelope_recipients_valid_total_title_short_program_management	R	B	The number of valid envelope recipients with job title "Program Management".
from_domain_aol	T	B	1, if the From: header email address domain is aol.com.
from_domain_gmail	T	B	1, if the From: header email address domain is gmail.com.
from_domain_gov	T	B	1, if the From: header email address domain is .gov.
from_domain_hotmail	T	B	1, if the From: header email address domain is hotmail.com.

Continued on next page...

Table 4.9 – Continued

Feature Name	Cat.	Type	Description
from_domain_example	R	B	1, if the From: header email address domain is example.com.
from_domain_example_invalid	R	B	1, if the From: header email address from example.com is invalid.
from_domain_example_similarity	R	N	Similarity score of the from address, if from example.com, to all company email addresses.
from_domain_mil	T	B	1, if the From: header email address domain is .mil.
from_domain_yahoo	T	B	1, if the From: header email address domain contains "yahoo".
from_header_encoding_big5	T	B	1, if the From: header uses big5 encoding.
from_header_encoding_gb2312	T	B	1, if the From: header uses gb2312 encoding.
from_header_phrase_contains_email_address	T	B	1, if the From: header email phrase [10] contains an email address.
from_header_phrase_contains_gov_email_address	T	B	1, if the From: header email phrase contains a .gov address.
from_header_phrase_contains_example_email_address	R	B	1, if the From: header email phrase contains a .example.com email address.
from_header_phrase_contains_mil_email_address	T	B	1, if the From: header email phrase contains a .mil address.
from_header_phrase_contains_user_of_address	T	B	1, if the From: header email phrase contains the user [11] portion of the email address.

Continued on next page...

[10] If the From: header has"John Doe" <john.doe@example.com>as a value, "John Doe" is the phrase of the From: header email address
[11] If the From: header has "John Doe" <john.doe@example.com>as a value, "john.doe" is the user of the From: header email address

Table 4.9 – Continued

Feature Name	Cat.	Type	Description
from_header_phrase_equals_email_address	T	B	1, if the From: header email phrase equals the From: header email address, 0 else.
from_header_phrase_exists	T	B	1, if the From: header email phrase exists, 0 else.
from_listserv	T	B	1, if the email came from an email listserver, 0 else.
link_exe	T	B	1, if the email contains a hyperlink to an .exe file, 0 else.
link_htm	T	B	1, if the email contains a hyperlink to an .htm file, 0 else.
link_zip	T	B	1, if the email contains a hyperlink to a .zip file, 0 else.
message_id_[redacted]	T	B	1, if the Message-ID contains "[redacted]", 0 else.
mime_boundary_2rfk	T	B	1, if the email contains a MIME boundary with the characters "2rfkindysadvnqw3nerasdf" [12].
received_line_[redacted]	T	B	1, if a Received: line contains "[redacted]" in the server name field, 0 else.
received_line_[redacted]	T	B	1, if a Received: line contains "[redacted]", 0 else.
reply_to_defined	T	B	1, if the Reply-To: header is defined, 0 else.
reply_to_from_address_notequal	T	B	1, if the Reply-To: email address is different than the From: email address.

Continued on next page...

[12]The MIME boundary, "2rfkindysadvnqw3nerasdf" is generally associated with the Foxmail email client located at http://www.foxmail.com.cn

Table 4.9 – Continued

Feature Name	Cat.	Type	Description
reply_to_gmail	T	B	1, if the Reply-To: email address is at gmail.com, 0 else.
reply_to_hotmail	T	B	1, if the Reply-To: email address is at hotmail.com, 0 else.
reply_to_example	R	B	1, if the Reply-To: email address is at example.com, 0 else.
reply_to_example_invalid	R	B	1, if the Reply-To: email address at example.com is invalid, 0 else.
reply_to_yahoo	T	B	1, if the Reply-To: email address is at yahoo.com, 0 else.
to_empty	T	B	1, if the To: header is defined but empty, 0 else.
to_gmail	T	B	1, if the To: header only contains a gmail.com email address, 0 else.
to_hotmail	T	B	1, if the To: header only contains a hotmail.com email address, 0 else.
to_no_example	R	B	1, if the To: header does not contain an example.com email address, 0 else.
to_yahoo	T	B	1, if the To: header only contains a yahoo.com email address, 0 else.
x_forwarded_to_defined	T	B	1, if the X-Forwarded-To: header is defined, 0 else.
x_mailer_aol	T	B	1, if the X-Mailer: header contains"AOL", 0 else.
x_mailer_aspnet	T	B	1, if the X-Mailer: header contains "aspnet", 0 else.
x_mailer_blat	T	B	1, if the X-Mailer: header contains "blat", 0 else.

Continued on next page...

Table 4.9 – Continued

Feature Name	Cat.	Type	Description
x_mailer_dreammail	T	B	1, if the X-Mailer: header contains "DreamMail", 0 else.
x_mailer_extreme_mail_express	T	B	1, if the X-Mailer: header contains "ExtremeMail Express", 0 else.
x_mailer_foxmail	T	B	1, if the X-Mailer: header contains "Foxmail", 0 else.
x_mailer_ghost_mail	T	B	1, if the X-Mailer: header contains "Ghost Mail", 0 else.
x_mailer_outlook_express	T	B	1, if the X-Mailer: header contains "Outlook Express", 0 else.
x_mailer_yahoomail	T	B	1, if the X-Mailer: header contains "YahooMail", 0 else.

4.4.4 Explanation of Features

The features of email that were exposed in this study were chosen based on implementation complexity and problem set relevance. Since this study aims to build a classifier than can detect attacks from specific threat actors targeted at a defined set of recipients, the chosen features were selected based on manual analysis of the corpus of targeted malicious email. The following sections provide more insight into why certain features were exposed.

Attachment

The two primary mechanisms for a threat actor to coerce a recipient to execute malicious code are use of a malicious attachment or use of a hyperlink that points to a malicious web page. The attachment related features are designed to expose whether an attachment is present in the email and if so, the type of attachment. The two predominantly exploited file types are Microsoft Office files (Microsoft Word, Microsoft Access, Microsoft Powerpoint and Microsoft Excel) and Adobe Portable Document Format (PDF) files. Due to embedded content, many legitimate emails contain .htm (Hypertext Markup) attachments and many targeted malicious emails do not. Table 4.10 shows a breakdown of the total number of emails with at least one attachment, a density representing the proportion of emails with at least one attachment, and densities representing the proportion of attachments of a certain file type.

Table 4.10: Attachment proportions in the *NTME1* and *TME1* data sets

Data set	Att Density	doc	htm	mdb	pdf	ppt	xls	other
TME1	0.46	0.36	0.00	0.00	0.46	0.05	0.04	0.09
NTME1	0.09	0.11	0.15	0.00	0.19	0.02	0.07	0.45

The attachment densities indicate that the proportion of attachments (Att Density) in targeted malicious emails is greater than the proportion of attachments in non-targeted malicious emails. A quick Z-test for proportions indicates this difference is significant at the $\alpha = 0.01$ level of significance. Additionally, doc, pdf, and ppt attachment types were significantly more prevalent in targeted malicious email (Z-test,

$\alpha = 0.01$). The htm and xls attachment types were significantly more prevalent in non-targeted malicious email.

Cc Header

The *Cc* header, commonly referred to as the Carbon Copy header, is used legitimately to add additional, non-primary, recipients to an email message. Standard email clients will not include a *Cc* header in an email if there are no *Cc* recipients. In both data sets the majority of emails do not use the *Cc* line and very few emails have someone in the company addressed on the *Cc* line. *TME1* appears to have slightly less usage of the *Cc* line. The *TME1* data set has significantly more empty *Cc* headers and

Table 4.11: *Cc* Header proportions between the *NTME1* and *TME1* data sets

Data set	empty	no company
TME1	0.99	0.99
NTME1	0.93	0.96

significantly more *Cc* addresses not addressed to the recipient's company (Z-test, $\alpha = 0.01$).

Character Encodings

Character encodings are a map between a character (such as the letter "A") and a number to facilitate transmission and display on a computing system. For the purposes of email, character encodings are threat specific and set during email creation, often unbeknownst to the user since they are set by the email client or tools used to create the email. Big5, GB2312 and GBK are all character encoding sets used primarily in the far east region of the world. Base64 encoding is a binary-to-ASCII encoder that obfuscates many text-based email analysis tools (e.g. regular expressions) but will still render cleanly in most recipient email clients. It is also used most times when there is an attachment to an email. The *TME1* data set has significantly more base64, big5, and gb2312 character encodings but the *NTME1* data set has significantly more windows1252 encodings (Z-test, $\alpha = 0.01$).

Table 4.12: Character encoding proportions in *NTME1* and *TME1* data sets

Data set	base64	big5	gb2312	gbk	windows1252
TME1	0.653	0.033	0.155	0.000	0.014
NTME1	0.166	0.009	0.001	0.000	0.025

Date Header

Emails typically include a *Date* header which is inserted by the mail client that sent the email. The mail client will typically include the current time zone of the computer running the mail client. The +0200, -0700, +0800, +0900 time zones, are more prevalent in the *TME1* data set than the *NTME1* data set (Z-test, $\alpha = 0.01$). The +0200 time zone is in Eastern Europe and Central and South Africa. The +0800 and +0900 time zones cover China to Japan. Interestingly enough, the -0700 time zone was more prevalent in the *TME1* data set even though this is the mountain time zone of the United States. Some of the time zones are not even valid; emails systems do not necessarily enforce format or compliance of this field to any standard.

DKIM

DomainKeys Identified Mail, or DKIM, is a cryptographic authentication mechanism used to verify the validity of a domain name associated with a particular email message (Delaney, 2007), (Allman et al., 2007), (Leiba and Fenton, 2007). An email with a valid DKIM signature is authenticated as originating from the advertised *From:* email domain. In this implementation, only the existence of a DKIM header is being checked instead of actually verifying the signature. Verifying the digital signature is expensive since it involves a lookup against Domain Name Services (DNS) records. Manual analysis did not reveal any spoofed DKIM signatures, any signatures if present were accurate. If spoofed DKIM signatures became prevalent, it is straightforward to convert this feature to one that actually verifies the signature. A majority of emails do not use DKIM, this is still a growing Internet standard. *TME1* has a significantly smaller proportion of DKIM signed emails than *NTME1* (Z-test, $\alpha = 0.01$). A number of the TME1 emails are sent from Google Gmail servers so they contain correct DKIM

Table 4.13: *Date* header time zone proportions in *NTME1* and *TME1* data sets

Time zone	TME1	NTME1	Timezone	TME1	NTME1
+0000	0.002	0.148	+0600	0.000	0.001
-0006	0.000	0.000	+0630	0.000	0.000
-0100	0.000	0.000	-0700	0.174	0.102
+0100	0.003	0.012	+0700	0.000	0.001
-0200	0.000	0.000	-0800	0.001	0.151
+0200	0.009	0.004	+0800	0.505	0.003
-0230	0.000	0.000	-0900	0.000	0.000
-0300	0.000	0.001	+0900	0.199	0.002
+0300	0.000	0.002	+0930	0.000	0.000
+0330	0.000	0.000	-1000	0.000	0.001
-0330	0.000	0.000	+1000	0.002	0.001
-0400	0.042	0.125	+1030	0.000	0.000
+0400	0.000	0.001	-1100	0.000	0.000
-0430	0.000	0.000	+1100	0.001	0.001
+0430	0.000	0.000	-1200	0.000	0.000
-0500	0.053	0.317	+1200	0.000	0.000
+0500	0.000	0.000	+1300	0.000	0.000
-0530	0.000	0.000	+1800	0.000	0.000
+0530	0.000	0.001	+1900	0.000	0.000
+0545	0.000	0.000	+2000	0.000	0.000
+0550	0.000	0.000	Unknown	0.000	0.003
-0600	0.009	0.122			

headers, even though the sending Gmail account might be false and malicious.

Table 4.14: DKIM proportions in the *NTME1* and *TME1* data sets

Data set	DKIM
TME1	0.181
NTME1	0.217

Email Size

Emails that include attachments are typically larger than emails without attachments. Thus, email size is a useful feature to differentiate targeted malicious emails with attachments from non-targeted malicious emails. Table 4.15 lists email sizes in bytes. While the *NTME1* data set has the largest email at 26MB, the average size of *TME* emails is larger at 276KB.

Table 4.15: Email size in the *NTME1* and *TME1* data sets

Data set	min	max	mean	stddev
TME1	1 KB	3.7 MB	276 KB	440 KB
NTME1	434 B	26.0 MB	95 KB	657 KB

Envelope Recipients

Emails when sent over the Internet are wrapped in an envelope that, similar to postal mail, lists the real recipients of an email. The envelope is handled at the system level and is not exposed to the user, instead a user sees the actual email. The actual email may accurately reflect the correct recipients or it may spoof the actual recipients. This group of mostly recipient oriented features describes the email recipients and various characteristics of recipients. Several of the envelope recipient features characterize the number of valid and invalid recipients. If threat actors are using a database of email addresses, it is possible that some of those addresses become invalid as employees are no longer associated with a company. Other characteristics of recipients exposed in these features include the average job level across the valid recipients (where a job level of 1 indicates an entry level employee and a job level of 10 indicates the Chief Executive Officer) and the number of recipients working in a particular business area. Similar to how anti-spam systems maintain a reputation of senders, there are two features designed to characterize the reputation of email recipients. First, there is one feature that calculates the average number of received targeted malicious email across all recipients. Second, there is a separate feature that calculates the average number of Google search hits for the respective email addresses. If threat actors are using a database of email addresses, recipients that have received targeted malicious email are likely to receive it again. Threat actors might seed a database of email addresses by using a search engine such as Google to extract email addresses for a particular email address domain. For example, executing a Google search for "john.doe@example.com" will return all web pages where that email address exists. Similarly, executing a Google search for "@example.com" will return web pages that contain email addresses from the example.com domain. Part of the data set used in this study includes the number

of Google search hits for all email addresses in the target company. Since an email may include multiple recipients, for a given email the average number of Google search hits is calculated. These recipient oriented features have a measure of durability since the threat actor can not change them without changing the targeted recipients. Figure 4.1 shows a histogram of the number of email accounts that have received a certain number of targeted malicious email at the company.

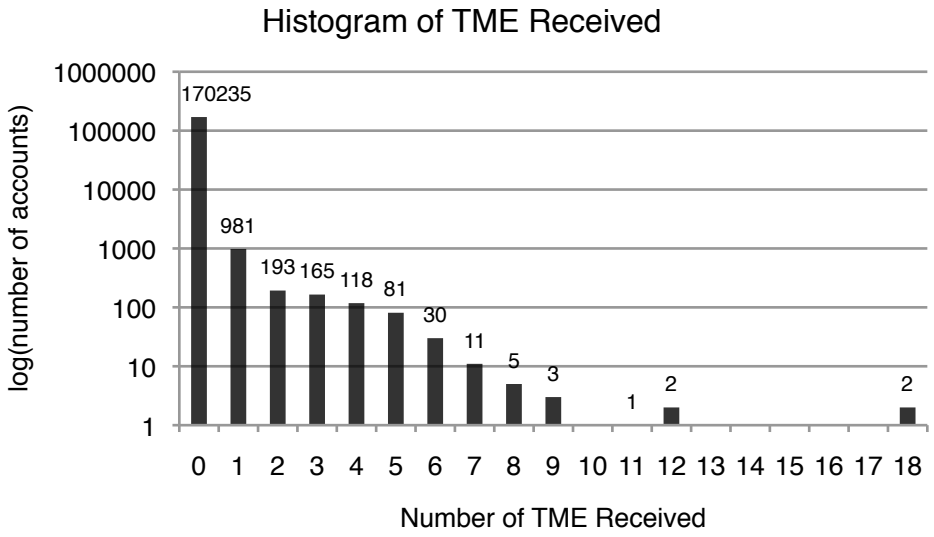
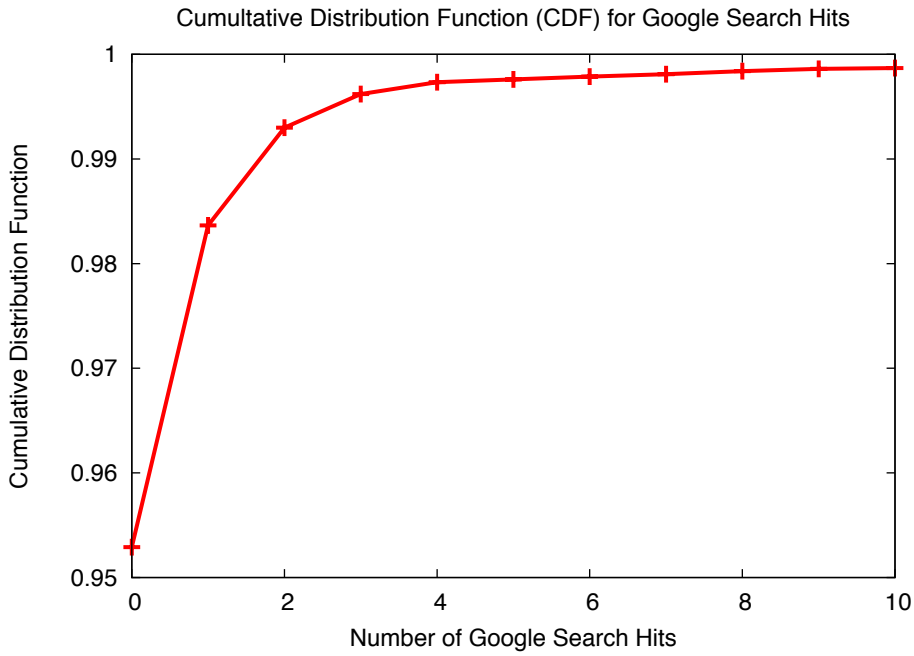
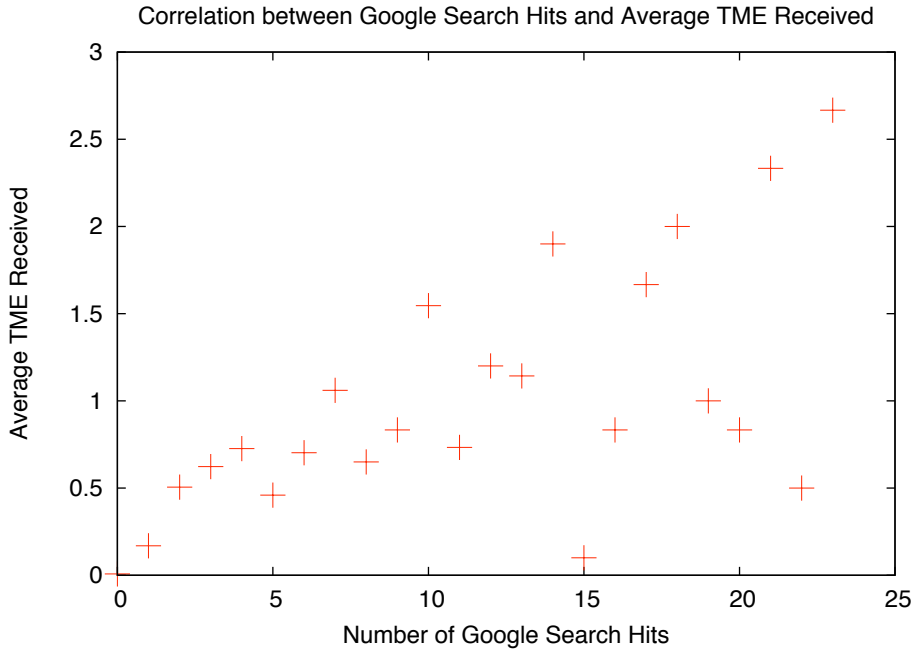


Figure 4.1: Number of TME received by accounts at company

Examining the relationship between the number of Google search hits and the number of targeted malicious emails that have been sent to that email address reveals an interesting trend. Figure 4.2a shows the cumulative distribution function of Google search hits in the company. A vast majority of email addresses are not listed in Google. Figure 4.2b shows a scatter plot of Google search hits against the average amount of targeted malicious email received by email accounts with that specific number of Google search hits. The x-axis only uses the first 24 Google search hit bins which accounts for 99.93% of the total population. Later bins are very sparsely populated. Visual inspection shows a positive correlation between these two variables. The critical values of the correlation coefficient, r, at the $\alpha = 0.05, 0.01, 0.005$ significance levels in a one-tailed significance test with 23 degrees of freedom are 0.3365, 0.4622, and 0.5052, respectively. Using equation 4.18 the calculated correlation coefficient for Google



(a) CDF for Google search hits



(b) Scatter plot of number of Google search hits vs. average amount of TME received

Figure 4.2: Analysis of Google search hits

search hits against average amount of TME received is $r = 0.621$. Thus, even at a α significance level of 0.005, the null hypothesis that there is no correlation is rejected and the alternative hypothesis that the correlation is positive is accepted. This shows that there is a positive correlation between the number of Google search hits and the average amount of TME received. Those email addresses with more search hits on Google have more targeted malicious email sent to them. Table A.1 in Appendix A lists all of the data for this analysis.

There are 2,379 unique job classes in the company. Not all individuals in the company have ready access to email (e.g. manufacturing jobs) and thus they all do not receive the same proportion of email. Table 4.16 shows the top 15 job classes in the company which represent 34.77% of the total population. Comparing the

Table 4.16: Top 15 job classes, by population, in the company

Job Class	**% of population**
Systems Engineering	10.00%
Software Engineering	5.75%
Program Management	2.32%
Embedded Software Engineering	2.00%
Mechanical Engineering	1.98%
Member Engineering Staff	1.58%
Multi Functional Finance	1.54%
Systems Integration & Test Engineering	1.51%
Quality Assurance Engineering	1.34%
Project Engineering	1.32%
Administrative Assistant	1.27%
Systems Integration	1.13%
Aeronautical Engineering	1.09%
Systems Administrator	1.02%
Electrical Engineering	0.94%

proportional amount of non-targeted malicious email (NTME) received by job classes in data set *NTME1* with the real job class population proportion (as shown in Table 4.16) reveals those job classes which proportionately receive more NTME than their population proportion. Table 4.17 shows the fifteen largest job class proportion differences between NTME and the actual population. In a sense, this list represents the job classes which use email most disproportionately from their actual population

proportion. For example, Employment Representatives which only account for 0.17% of the employee population received 0.37% of email in the *NTME1* data set. The Z value and 99% lower bound confidence interval (one-tail) is calculated using the formulas in section 4.3.1. If the null hypothesis is that the proportions are the same (e.g. $p_1 = p_2$), then at the α=0.01 level of significance the Z-test statistic must be greater than 2.33 (one-tail) to accept the alternative hypothesis that $p_1 > p_2$. All of the Z values in Table 4.17 are greater than 2.33 thus the alternative hypothesis can be accepted for these fifteen job classes. Additionally, all of the 99% lower bound confidence intervals are greater than 0 which further confirms the significance of these proportion differences. Several of the job classes which appear in Table 4.17 make sense given the type of work performed by individuals in that job class. For example, business development, subcontract administration, procurement and contracts negotiation individuals all heavily use Internet-based email to coordinate with their respective business partners. Comparing the proportional amount of spam received by job classes in the *SP1* data

Table 4.17: Fifteen largest job class proportion differences between NTME and actual population

Job Class	$p_1 - p_2$	**Z**	**99% CI LB**
Program Management	1.83%	29.29	1.70%
Administrative Assistant	1.49%	29.67	1.39%
Systems Engineering	1.08%	10.24	0.83%
Business Development Analysis	0.99%	27.41	0.91%
Subcontract Administrator	0.72%	19.89	0.64%
Procurement Representative	0.49%	17.24	0.43%
Project Engineering	0.48%	11.09	0.38%
Systems Integration	0.39%	9.73	0.30%
Business Development	0.38%	15.16	0.33%
Employment Representative	0.37%	17.02	0.33%
IT Program	0.33%	15.59	0.29%
Computer Systems Architect	0.31%	9.83	0.24%
Multi Functional Finance	0.28%	6.38	0.18%
Contracts Negotiator	0.26%	9.17	0.20%
Member Engineering Staff	0.26%	5.86	0.16%

set with the proportional amount of NTME received by job classes in the *NTME1* data set reveals those job classes which proportionately receive more spam than

NTME. Table 4.18 shows the fifteen largest job class proportion differences between spam and NTME. All of the Z values in Table 4.18 are greater than 2.33 thus the alternative hypothesis can be accepted for these fifteen job classes. Additionally, all of the 99% lower bound confidence intervals are greater than 0 which further confirms the significance of these proportion differences. The individuals in the job classes

Table 4.18: Fifteen largest job class proportion differences between spam and NTME

Job Class	$p_1 - p_2$	**Z**	**99% CI LB**
International Licensing	1.22%	50.37	1.18%
Multi Functional Manufacturing	0.37%	20.11	0.33%
Network Monitoring Technician	0.36%	28.46	0.34%
Materials Support Team Member	0.31%	20.04	0.28%
Strategic Planner	0.30%	20.20	0.27%
Manufacturing Planner	0.27%	17.23	0.24%
Administrative Support	0.24%	9.81	0.18%
Technician	0.23%	12.67	0.19%
Computer Support	0.20%	10.89	0.16%
Engineering Planner	0.19%	8.10	0.14%
Systems Engineering Field Technical Support	0.19%	10.75	0.15%
Manufacturing Engineering	0.18%	7.04	0.12%
Material Handler	0.17%	13.16	0.15%
Aeronautical Engineering	0.17%	6.38	0.11%
Configuration	0.17%	8.64	0.13%

listed in Table 4.18 proportionately receive more spam than NTME. This may be reflective of spam conducive computing practices employed by individuals in these job classes (e.g. more sharing of their email address, signing up at websites). Comparing the proportional amount of targeted malicious email (TME) received by job classes in the *TME1* data set with the proportional amount of NTME received by job classes in the *NTME1* data set reveals those job classes which proportionately receive more TME than NTME. Table 4.19 shows the fifteen largest job class proportion differences between TME and NTME. All of the Z values in Table 4.19 are greater than 2.33 thus the alternative hypothesis can be accepted for these fifteen job classes. Additionally, all of the 99% lower bound confidence intervals are greater than 0 which further confirms the significance of these proportion differences. With a high degree of confidence, the

true proportion difference between TME and NTME for the job classes listed in Table 4.19 is greater than the value listed in the column "99% CI LB". The individuals in

Table 4.19: Fifteen largest job class proportion differences between TME and NTME

Job Class	$p_1 - p_2$	Z	99% CI LB
Business Development Analysis	4.67%	20.83	3.66%
Program Management	4.11%	11.25	2.94%
International Business Development	3.41%	38.82	2.62%
Communications	1.80%	17.51	1.19%
Business Development	1.58%	10.39	0.95%
Project Specialist	1.56%	12.79	0.97%
Mechanical Engineering	1.47%	5.59	0.68%
Software Engineering	1.29%	3.20	0.25%
Fellow	1.29%	11.78	0.75%
Electronics Engineering	1.24%	6.64	0.61%
Project Engineering	1.21%	4.99	0.49%
Research Engineering	1.06%	7.75	0.52%
Communications Representative	0.97%	9.45	0.50%
Research Scientist	0.78%	8.71	0.36%
Field Engineering	0.63%	5.76	0.21%

the job classes listed in Table 4.19 proportionately receive more TME than NTME. If TME was no different than spam, then the proportional differences between TME and NTME would be similar to the proportional differences between spam and NTME. But, in fact, none of the job classes listed in Table 4.19 are listed in Table 4.18. Additionally only 4 of the 15 job classes listed in Table 4.19 are listed in the top 15 job classes by population in Table 4.16.

The average number of recipients in specific business areas is another envelope recipient characteristic that shows some differences between the *TME1* and *NTME1* data sets. Table 4.20 shows the average number of recipients per email in a specific business area, μ_{BA}, and the standard deviation, σ_{BA}. For each business area, a t statistic (Equation 4.13), and the degrees of freedom (Equation 4.14) is calculated. Except for Business Area "A", all of the other differences between the *TME1* and *NTME1* data sets are significant at a $\alpha = 0.001$ level of significance using a two-tailed test. The 99.9% confidence intervals of the difference between means (Equation 4.15)

supports the alternative hypothesis that the means are different.

Table 4.20: Envelope recipients by business area - $\mu_{BA}(\sigma_{BA})$

Data set	**"A"**	**"E"**	**"E2"**	**"I"**	**"S"**
TME1	0.189(0.556)	0.423(0.974)	0.172(0.466)	0.234(0.676)	0.307(0.720)
NTME1	0.195(0.483)	0.273(0.569)	0.118(0.383)	0.289(0.684)	0.164(0.511)
t	-0.517	7.397	5.557	-3.895	9.532
df	2348	2329	2345	2361	2337
99.9% CI -	-0.046	0.079	0.020	-0.104	0.091
99.9% CI +	0.034	0.221	0.088	-0.006	0.195

Examining the difference in the total number of valid envelope recipients in the *TME1* and *NTME1* data sets also yields a significant result. Table 4.21 shows the minimum, maximum, average and standard deviation of valid envelope recipients per email in the *TME1* and *NTME1* data sets. With a $\alpha = 0.001$ level of significance, the average valid number of recipients in *TME1* is greater than the average valid number of recipients in *NTME1*.

Table 4.21: Total valid envelope recipients

Data set	**min**	**max**	**mean**	**stddev**
TME1	0	36	1.354	2.352
NTME1	0	100	1.109	0.982

t	5.008
df	2321
99.9% CI -	0.075
99.9% CI +	0.415

Looking at the average total number of invalid envelope recipients per email, in Table 4.22 yields similar results as the average total number of valid envelope recipients. With a $\alpha = 0.001$ level of significance, the average invalid number of envelope recipients in *TME1* is greater than the average invalid number of envelope recipients in *NTME1*.

Ignoring the validity of a recipient email address, a significant difference still exists. Table 4.23 shows the average total number of envelope recipients per email in the *TME1* and *NTME1* data sets. With a $\alpha = 0.001$ level of significance, the average

Table 4.22: Total invalid envelope recipients

Data set	min	max	mean	stddev
TME1	0	15	0.165	1.155
NTME1	0	15	0.056	0.441

t	4.537
df	2320
99.9% CI -	0.025
99.9% CI +	0.193

total number of envelope recipients in *TME1* is greater than the average total number of envelope recipients in *NTME1*.

Table 4.23: Total envelope recipients

Data set	min	max	mean	stddev
TME1	1	50	1.544	3.547
NTME1	1	100	1.165	1.164

t	5.138
df	2318
99.9% CI -	0.122
99.9% CI +	0.636

Finally, examining the average job level for valid envelope recipients per email shows that with a $\alpha = 0.001$ level of significance, the average job level of valid envelope recipients in *TME1* is greater than the average job level of valid envelope recipients in *NTME1*. Table 4.24 shows this result.

Table 4.24: Average job level of valid envelope recipients

Data set	min	max	mean	stddev
TME1	1	9	4.92	1.511
NTME1	1	10	3.938	1.638

t	31.089
df	2368
99.9% CI -	0.872
99.9% CI +	1.092

From Header

In legitimate email, this field will contain the email address of the sender. In malicious email, this field can be set to an arbitrary value by a threat actor (e.g. spoofed email address). There are features that record whether the *From* address is advertised as coming from one of several public webmail providers (e.g. Google, Yahoo, Hotmail). There are also features that record whether the *From* address is advertised as coming from some top-level domain names such as .mil (used by the United States Military) or .gov (used by the United States Government). In an attempt to confuse users, threat actors will sometimes falsify the *From* address as being from the organization that is being targeted (e.g. sending an email spoofed "from" example.com to a user in example.com). There is a feature that captures if the target company's domain name is used in the *From* address and if so, if the email address in the *From* address is valid. Similar to the email character encodings, the *From* header can also designate language specific character encodings, so those are also captured and expressed as features. These are often associated with the threat actor's locale. A final set of features in this section are threat specific and have to do with abnormalities discovered in the construction of the *From* header. The email address in the *From* header actually consists of multiple parts, the address, the phrase (also known as a Display Name), and the comment (Resnick, 2008). Manual analysis of the targeted malicious email data set revealed numerous cases where the email address was simply replicated in the phrase portion of the email address (e.g. "john.doe@example.com" <john.doe@example.com>). This could be indicative of a procedure that some threat actors consistently follow. There were also cases where threat actors were trying to spoof legitimate US government or US military email addresses and wanted the recipient email client to show a spoofed email address. For example, if the threat actor used Google Gmail for sending the malicious email the *From* address looked like "john.doe@agency.mil" <john.doe@gmail.com>. Gmail will only allow a user to assign a non-Gmail *From* address if the user verifies ownership of that email address via clicking on a link sent to that address. Since a threat actor may not control a

desired email address, inserting the spoofed email address in the email phrase will show that email address in the recipient email client. This is enough to confuse most users into thinking the email legitimately originated from the spoofed email address. Table 4.25 shows the proportion of several *From* header domains between the *NTME1* and *TME1* data sets. *TME1* has a significantly greater proportion of emails with a *From* header domain of gmail.com, .gov, yahoo.com and the company's domain name with invalid email address than *NTME1* (Z-test, $\alpha = 0.01$). *TME1* has a significantly smaller proportion of emails with a *From* header domain of aol.com, hotmail.com and the company's domain name. There is no statistically significant difference in the proportion of *From* headers with .mil domain email addresses between *TME1* and *NTME1*. If the email address in the *From* header is from the company, than

Table 4.25: *From* header by domain proportions in the *NTME1* and *TME1* data sets

Data set	aol	gmail	.gov	hotmail	.mil	yahoo	other
TME1	0.006	0.521	0.089	0.000	0.024	0.063	0.289
NTME1	0.015	0.029	0.028	0.012	0.027	0.029	0.845

Data set	company	company invalid
TME1	0.006	0.005
NTME1	0.015	0.001

the similarity score match is used to compare that advertised email address to all email addresses in the company to see if there is a close match. The idea with this feature is to uncover spoofed email addresses that are similar to legitimate email addresses. The similarity score is obtained from a fulltext index search using MySQL and summary statistics are shown in Table 4.26. Just as character encodings are

Table 4.26: *From* header similarity score match when the domain is the company's domain in the *NTME1* and *TME1* data sets

Data set	min	max	mean	stddev
TME1	0	17.22	4.976	5.776
NTME1	0	40.28	4.866	5.676

part of email message bodies, they can be set in the *From* header as well. Table 4.27

shows the proportional difference in big5 and gb2312 character encodings between the *TME1* and *NTME1* data sets. *TME1* has a significantly greater proportion of emails with either the big5 or gb2312 character encodings (Z-test, $\alpha = 0.01$). Table 4.28

Table 4.27: *From* header encodings in the *NTME1* and *TME1* data sets

Data set	big5	gb2312
TME1	0.016	0.010
NTME1	0.000	0.000

summarizes the results of *From* header phrase differences between the *TME1* and *NTME1* data sets. *TME1* has a greater proportion than *NTME1* of emails with a *From* header phrase, email addresses in the *From* header phrase, a .gov email address in the *From* header phrase, a company email address in the *From* header phrase, a .mil email address in the *From* header phrase, and the user of the *From* header email address in the *From* header phrase (Z-test, $\alpha = 0.01$). *NTME1* has a greater proportion than *TME1* of emails where the *From* header phrase equals the *From* header email address.

Table 4.28: *From* header phrases in the *NTME1* and *TME1* data sets

Data set	exists	email	.gov email	company email
TME1	0.936	0.081	0.031	0.038
NTME1	0.878	0.027	0.000	0.015

Data set	.mil email	user of address	equals address
TME1	0.003	0.066	0.002
NTME1	0.000	0.004	0.008

Email list servers

Many users legitimately subscribe to Internet based email lists. Targeted malicious emails are not generally sent through a large email list server for distribution, they are sent directly to targeted recipients. Table 4.29 shows that *NTME1* has a significantly greater proportion of emails from email list servers than *TME1* (Z-test, $\alpha = 0.01$).

Table 4.29: Email list server proportions in the *NTME1* and *TME1* data sets

Data set	From List Server
TME1	0.000
NTME1	0.236

Hyperlinks

In addition to attachments, the second primary vehicle for a threat actor to exploit a user is via a malicious link included in an email that directs the user to a malicious webpage that exploits a vulnerability on the user's system. Analysis of the targeted malicious email data set revealed threat actor preference for including links that direct users to an executable file (.exe) or a compressed zip file (.zip). Often the .zip files would include a malicious executable inside that the user would execute on behalf of the threat actor, sometimes with no technical vulnerability being exploited other than user vulnerability. *TME1* had a greater proportion of emails with .zip hyperlinks than *NTME1* (Z-test, $\alpha = 0.01$).

Table 4.30: Hyperlink proportions in the *NTME1* and *TME1* data sets

Data set	exe	htm	zip
TME1	0.001	0.241	0.093
NTME1	0.015	0.453	0.010

Message-ID

Analysis of targeted malicious email revealed that a small percentage of TME emails contained a fixed string in the *Message-ID* field. The *Message-ID*, which is automat-

ically generated, is supposed to be a unique identifier for a given email. The exact string is redacted here at the request of the data owner.

Table 4.31: *Message-ID* proportions in the *NTME1* and *TME1* data sets

Data set	"[t: redacted]"
TME1	0.011
NTME1	0.000

MIME Boundaries

Multipurpose Internet Mail Extensions, or MIME, is an Internet standard that defines numerous extensions to standard email. These extensions support functionality such as binary formatted attachments and emails with multiple parts. Sometimes email clients insert MIME boundaries that are readily identifiable. Analysis of targeted malicious email revealed many messages that included a specific MIME boundary separator in the email. This specific MIME boundary is often associated with a specific email client or tool that threat actors may be using to craft targeted malicious email. Table 4.9 contains the exact MIME boundary string that is proportionally more present in *TME1* than *NTME1* (Z-test, $\alpha = 0.01$).

Table 4.32: MIME boundary counts and proportions in the *NTME1* and *TME1* data sets

Data set	"2rfk"
TME1	389 (0.168)
NTME1	2 (0.000)

Received Line

As emails travel through the Internet to their destination, a *Received* header entry is added by each email server that handles a message. False *Received* headers can be added to email, which often occurs with spam. Sometimes a mail client on a client computer sends email to a locally running mail server before sending it onto an Internet based host. In this case, the host name of the computer running the local

mail server may appear in the *Received* line. If threat actors are re-using a certain computer with local mail server setup to send targeted malicious email it may be possible to track the emails. The exact strings are redacted here at the request of the data owner. These strings are present in a greater proportion of *TME1* than *NTME1* (Z-test, $\alpha = 0.01$).

Table 4.33: *Received* line proportions in the *NTME1* and *TME1* data sets

Data set	**"[s: redacted]"**	**"[v: redacted]"**
TME1	0.007	0.006
NTME1	0.000	0.000

Reply-To Header

The *Reply-To* header of an email defines the email address to send a return email should a user decide to reply to an email. If no *Reply-To* is present, email clients default to the email address in the *From* header. Threat actors will sometimes set the *Reply-To* address to an email address in their control so that they can capture any replies from users. Many users may not notice a change in email address when replying to an email. This functionality allows a threat actor to spoof an email address in the *From* header that may be familiar to the recipient while still controlling the destination mailbox for replies. Also, if the email is purely spoofed and if the user replies, the user might get an error from the invalid account or the real person saying "I did not send this." There are features that record whether the *Reply-To* exists, whether, if present, it is equal to the *From* address, whether the *Reply-To* points to a public webmail provider such as Hotmail or Gmail, and if the *Reply-To* address points back to the company (to a valid or invalid address). In Table 4.34, the proportions for the public webmail providers and company are with respect to those emails where a *Reply-To* header exists. *TME1* contains a greater proportion of email with a *Reply-To* address from gmail, hotmail, yahoo or the company than *NTME1* (Z-test, $\alpha = 0.01$).

Table 4.34: *Reply-To* header proportions in the *NTME1* and *TME1* data sets

Data set	Exists	≠ From	gmail	hotmail
TME1	0.410	0.032	0.878	0.064
NTME1	0.433	0.261	0.010	0.002

Data set	yahoo	company	co. invalid	other
TME1	0.017	0.018	0.018	0.023
NTME1	0.010	0.009	0.000	0.969

To Header

The *To* header of an email address shows the user which recipients received the email message. The *To* header is for display in an email client and may or may not be the same as the actual recipients recorded in the envelope recipient list of an email. Some malicious emails have a *To* header defined but with no recipients listed (empty). Other malicious emails only addressed public webmail provider email addresses in the *To* header which means that the target company email addresses were included in the *Bcc* (Blind Carbon Copy) field on the threat actor's end. Still other emails do not have anyone from the target company in the *To* line, which means those recipients were possibly on *Cc* or *Bcc*. Email addresses that show up in the envelope recipient list but not on the *To* or *Cc* line were emailed via *Bcc*. *TME1* had a greater proportion of email with an empty *To* line than *NTME1* (Z-test, $\alpha = 0.01$).

Table 4.35: *To* header proportions in the *NTME1* and *TME1* data sets

Data set	Empty	gmail	hotmail	yahoo	no company
TME1	0.105	0.006	0.000	0.000	0.064
NTME1	0.020	0.006	0.001	0.003	0.134

X-Forwarded-To

The *X-Forwarded-To* header is used when a user has their email forwarded to another account. For example, if someone forwards their hotmail.com email to their example.com account, the forwarded email received at example.com may contain the *X-Forwarded-To* header along with the target example.com email address. *NTME1*

had a greater proportion of email with the *X-Forwarded-To* header than *TME1* (Z-test, $\alpha = 0.01$).

Table 4.36: *X-Forwarded-To* header proportions in the *NTME1* and *TME1* data sets

Data set	Proportion
TME1	0.000
NTME1	0.006

X-Mailer Header

Many email clients leave identification in sent emails. Often, this identification is done via the *X-Mailer* header. This is not a required field and not all emails include an *X-Mailer* header but various features were extracted based on analysis of X-Mailer headers in targeted malicious email. *TME1* had a greater proportion of email with "aspnet", "blat", "dreammail", "extreme mail", "foxmail", "ghostmail" and "outlook express" in the *X-Mailer* header than *NTME1* (Z-test, $\alpha = 0.01$).

Table 4.37: *X-Mailer* header proportions in the *NTME1* and *TME1* data sets

Data set	aol	aspnet	blat	dreammail	extreme mail
TME1	0.004	0.055	0.005	0.004	0.006
NTME1	0.011	0.004	0.000	0.000	0.000

Data set	foxmail	ghostmail	outlook express	yahoomail
TME1	0.152	0.001	0.518	0.027
NTME1	0.000	0.000	0.016	0.032

4.4.5 Summary of feature differences

Table 4.38 summarizes key feature differences between TME and NTME emails. The table outlines which features are more dominant in either type of email.

Table 4.38: Key feature differences between *TME1* and *NTME1* data sets

Feature	TME	NTME
Attachment	46% of TME, mostly .doc, .pdf, .ppt.	9% of NTME, mostly .htm, .xls
Cc Header	empty Cc lines	
Char Encodings	base64, big5, gb2312	windows1252
Date Header	Timezones: +0200, -0700, +0800, +0900	Timezones: -0500, -0600, -0800
DKIM	Limited, mostly TME sent using Google Mail	Signed
Email size	Average: 276 KB	Average: 95 KB
MIME	Boundary "2rfk" present in 389 emails	Boundary "2rfk" present in 2 emails
Reply-To	gmail, hotmail, yahoo, company	other
To	empty To lines	yahoo, no company
X-Mailer	aspnet, blat, dreammail, extreme mail, foxmail, outlook express	aol, yahoomail

4.4.6 Features to Vectors

Table 4.9 listed the features that were extracted from all emails. These features were either binary, numeric or categorical. If each feature is denoted as f, the number of features as F, and the number of emails as N, then the set of features for a specific email, E, can be represented as a vector $\theta_E = \{f_1, ...f_F\}$. Figure 4.3 shows a graphical depiction of how emails are represented as a vector of features.

4.5 Classification

According to Duda et al. (2000) pattern recognition is, "the act of taking in raw data and making an action based on the 'category' of the pattern." The objective of this study was to create a system that can accept raw email and classify the email as

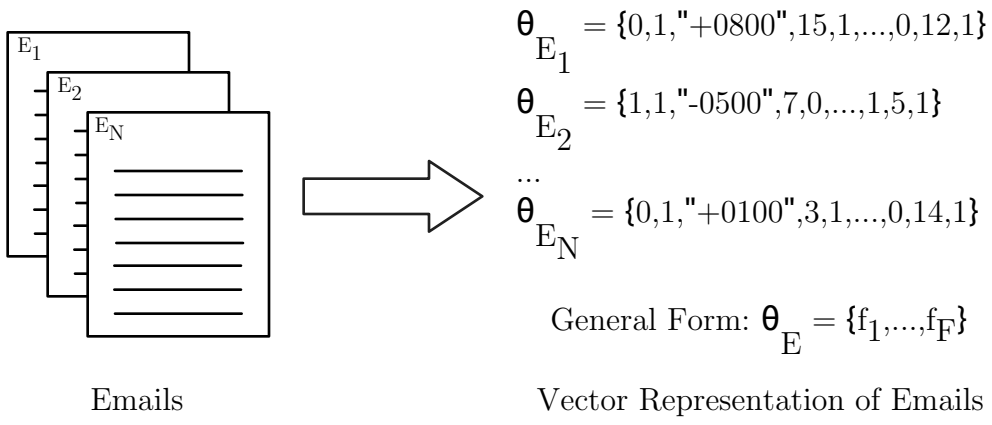


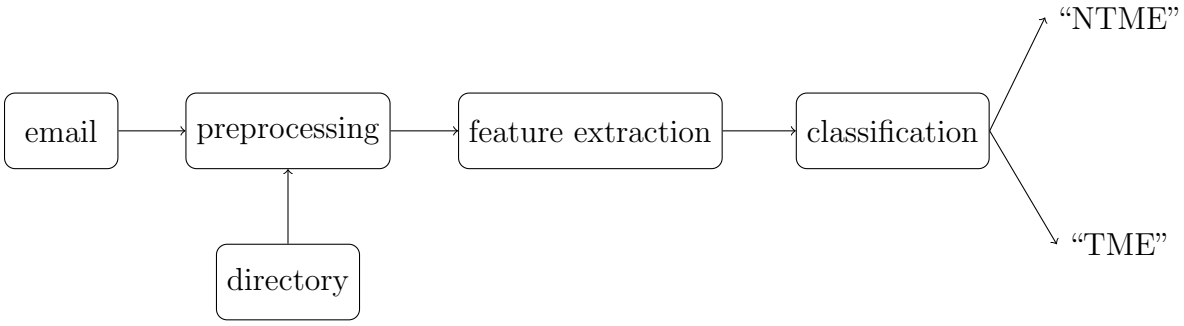
Figure 4.3: Feature representation of emails

belonging to either a category of "non-targeted malicious email" (NTME) or "targeted malicious email" (TME). Figure 4.4 depicts the high level process for determining the classification of a given email.

4.5.1 Random Forests

To separate "non-targeted malicious email" (NTME) from "targeted malicious email" (TME), the random forest (Breiman, 2001) classifier was chosen. There are several characteristics of this classifier that made it ideal for the data sets in this study: a) It can handle a high number of features; b) It can handle a large number of emails; c) It can handle a mixture of binary, numeric and categorical features; d) It generally does not overfit[13]; e) It can handle missing features; f) The algorithm is trivially parallelized to scale up for huge data sets; g) It can estimate which features are more important than others; h) It can handle unbalanced data sets (e.g. training data that consists of a much larger number of "non-targeted malicious email" than "targeted malicious email"). In traditional decision tree classification algorithms each node is split using the best split from all available features (where 'best split' provides the most amount of separation in the data). With random forests, each node is split using the best split from a randomly selected set of features at that node. In

[13]Overfitting is a situation where a classifier performs well on training samples but does not perform well on new patterns



Email	An email with unknown classification.
Directory	Corporate directory which includes information about email users such as job title.
Preprocessing	Email and directory information are combined to provide additional recipient context to emails.
Feature Extraction	Relevant features are extracted from the email and converted into a multi-dimensional vector with each element of the vector representing a feature.
Classification	The email is processed through a classifier that was trained with previously labeled data to determine the classification of the input email. "NTME" corresponds to "non targeted, malicious" email and "TME" corresponds to "targeted, malicious" email.

Figure 4.4: Classification process

addition, multiple decision trees are created using bootstrap (random selection with replacement) samples from the data set. These trees are created independently of each other and a classification decision is reached by a simple majority vote from the trees in the forest. The algorithm has two primary parameters k, the number of trees in the forest, and m, the number of random features to consider for node splitting. Details of the random forest algorithm can be found in Appendix B.

4.5.2 Types of Error

As seen in Figure 4.4, the final step in the classification process is for the classifier to predict the classification given an input email. In this study, binary classification will be performed such that emails will be classified as either "targeted malicious email"

(TME) or "non-targeted malicious email" (NTME) where the category of primary interest is targeted malicious email (TME). When the classifier correctly predicts a known TME as TME that is known as a True Positive (TP). When the classifier correctly predicts a known NTME as NTME that is known as a True Negative (TN). In cases where the classifier predicts a known NTME as TME that is known as a False Positive (FP) or Type I error. In cases where the classifier predicts a known TME as NTME that is known as a False Negative (FN) or Type II error. Table 4.39 shows a confusion matrix which is a visual representation of the different outcomes from the classifier.

Table 4.39: Confusion Matrix

	Actual TME	Actual NTME
Predicted TME	True Positive (TP)	False Positive (FP)
Predicted NTME	False Negative (FN)	True Negative (TN)

The false positive rate (FPR) is the proportion of NTME emails that were incorrectly classified as TME. The *specificity* is equal to $1 - fpr$. The FPR is:

$$fpr = \frac{FP}{FP + TN} \tag{4.21}$$

The false negative rate (FNR) is the proportion of TME emails that were incorrectly classified as NTME. The *sensitivity* is equal to $1 - fnr$. The FNR is:

$$fnr = \frac{FN}{FN + TP} \tag{4.22}$$

There is always a tradeoff between the false positive rate and false negative rate. An increased false positive rate results in a decreased false negative rate and an increased false negative rate results in a decreased false positive rate. These measures of performance will be combined with others for a full set of metrics that will be used to evaluate overall classifier performance.

4.5.3 Cost Sensitive Learning and Classification

Turney (2000) provides an excellent overview of the different costs associated with classification; the cost most relevant to this study is the cost of misclassification. There is also a teacher cost associated with creating the data sets used in this study but since it is a one-time cost with respect to this study, it will be ignored. A primary challenge with the data sets used in this study are that they are imbalanced. The "targeted malicious" class of email constitutes a minority of the data but it is the class of interest. He and Garcia (2009) provides a good review of the current research into learning from imbalanced data sets. According to Chen et al. (2004) there are two primary methods to address the imbalance when using a Random Forest classifier. One is based on cost sensitive learning where a high cost to misclassification of the minority class is assigned and the classifier is trained to minimize total error cost instead of the total number of errors. The second method, called stratification, is to use a sampling technique where either the majority class is under-sampled or the minority class is over-sampled or both. With imbalanced data sets either method improves performance over the default Random Forest algorithm (Chen et al., 2004). Both Margineantu (2000) and McCarthy et al. (2005) note improved classifier performance when using cost sensitive learning instead of stratification particularly for large data sets. Therefore, the cost sensitive classification method will be applied in this study to address the imbalanced data sets.

MetaCost

In WEKA, cost sensitive learning can be implemented using MetaCost (Domingos, 1999). MetaCost creates an ensemble of cost sensitive classifiers based on a base classifier, which in this study is Random Forest. Each one is based on a re-sample of the original data (bootstrap aggregation or bagging).

Cost model for targeted malicious email

The following section provides a conceptual framework for understanding the cost model for targeted malicious email. This cost model is based on the operations, as of this writing, of the company whose data is used in this study. The cost model described in Lee et al. (2006) was used as a base but adapted for the case of targeted malicious email (TME). Table 4.40 outlines the outcome costs associated with an email filtering system focused on three scenarios: 1) detection of targeted malicious email (TME-Detect), 2) blocking targeted malicious email (TME-Block), and 3) blocking of spam (Spam-Block). There is no corresponding Spam-Detect since the spam system is deployed only in a blocking configuration.

RCost is the response cost which varies based on the specific outcome and scenario. In the True Positive (TP) outcome, the *RCost* associated with TME-Detect is greater than the *RCost* associated with TME-Block since in a detect configuration a more robust incident response is needed to follow up on the positive detection. With TME-Block, the residual *RCost* is for gathering additional intelligence on the blocked email to aid future detection. In the False Positive (FP) outcome the *RCost* associated with TME-Detect, TME-Block and Spam-Block are the same since the cost to confirm the legitimacy of an email is the same. Due to the increased fidelity, *RCost* in a threat specific detection should be lower than *RCost* in a general detection since the response analyst knows precisely what to look at to confirm or refute the outcome; increased attribution of TME to specific threat actors would further drive down *RCost*. Furthermore, the closer a detection is to the earlier stages of the threat kill chain, the lower the *RCost*. Some organizations without good response mechanisms may inadvertently set *RCost* to zero; organizations with a good security intelligence capability realize that even true positives in a blocking-only configuration need to have a response.

BICost is the business impact cost which is realized when a certain outcome results in work stoppage or other form of business impact. With both TME-Block and Spam-Block, the *BICost* is the same. One example is the cost of not receiving a

business critical email (e.g. opportunity lost because of a blocked request for proposal email) because it was blocked by the email filter.

DCost is the damage cost which is realized when the email filter does not correctly detect or block a targeted malicious email. For example, this cost could be the loss of intellectual property due to threat actor presence and subsequent data exfiltration enabled by the malicious code in the targeted attack email. This cost could also be fines based on government regulations surrounding the protection of certain types of data. Perhaps more difficult to quantify, this cost can also encompass a loss of reputation. In attack scenarios involving the loss of sensitive information, *DCost* can be very high and is typically much higher than any *RCost* or *BICost*. In a detection-only scenario an organization with a rapid response capability can have a *DCost* of zero even with positive detection of TME.

Table 4.40: Conceptual cost model for various email filtering outcomes

Outcome	TME-Detect	TME-Block	Spam-Block
TP	*RCost*	*RCost*	0
FP	*RCost*	$BICost + RCost$	$BICost + RCost$
TN	0	0	0
FN	*DCost*	*DCost*	*UCost*

UCost is the cost associated with the loss of user productivity stemming from having to process an unwanted email. With spam, a false negative is a spam email that is misclassified as a legitimate email resulting in the user having to process that email manually.

When classifying conventional spam among non spam, the cost of a false positive (FP) is generally greater than the cost of a false negative (FN). Many current studies on email filtering focus on false positives being of greater importance than false negatives (Delany et al., 2005; Sakkis et al., 2003; Zhang et al., 2004; Koprinska et al., 2007). Specifically, if a non spam email is misclassified as spam (FP) that means that a possibly legitimate email a user was expecting may end up being filtered into a "junk" folder. However, if a spam email is misclassified as a non spam email (FN) that means that the user may see an unwanted spam email in their inbox. Generally users do

not want to miss legitimate email and are willing to handle a few misclassified spam messages, so in this case the cost of a false positive is greater than a false negative.

When classifying targeted malicious email (TME) among non-targeted malicious email (NTME), the classification costs are reversed due to the impact. A false negative means a TME was misclassified as NTME. Since a single TME can result in threat actor presence on a network, this false negative cost is much greater than the false negative cost in the conventional spam scenario. Presumably as the level of attack targeting and impact damage (DCost) associated with a TME increase, analysts and organizations will have a greater tolerance for false positives due to a desire for fewer false negatives. Conversely, a false positive means a NTME was misclassified as a TME. With a detection-only filter (TME-Detect), false positives may be more tolerable than with a protection (TME-Block) filter. In a business situation, a false positive could result in a business critical email not being delivered to a user if the intervening email filter is in block mode (vs. detection-only mode). Tool developers may be very false positive adverse if the tool can only be used in a block mode. In this study, a simplifying assumption is made that the costs for all false negatives are the same, when in practice the cost of a false negative differs based on the specific email and specific threat.

A useful analogy for the difference between the importance of false positives and false negatives can be found at every airport in the United States. Airport security screening is very much like detecting TME since the cost of a false negative (e.g. terrorist getting on a plane) is much greater than the cost of a false positive (e.g. unnecessarily screening a benign individual). Figure 4.5 breaks down the airport security screening process. Airport security screening typically consists first of a high sensitivity filter in the form of a metal detector followed by a highly specific manual pat down if needed. A highly sensitive filter is designed to minimize false negatives, and in the case of airport screening ensures that any possible suspected terrorist is flagged up front. It is acceptable if some individuals are unnecessarily flagged since the false negative cost is far greater than the false positive cost. If an individual passes the metal detector, they are allowed to fly on the plane. If an individual does not pass

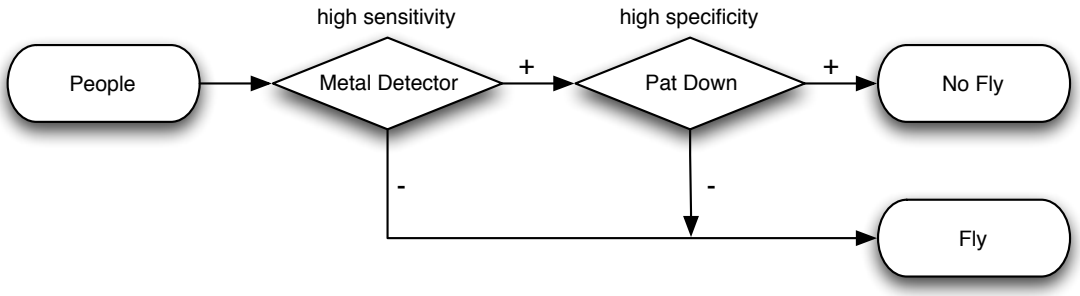


Figure 4.5: Airport Analogy

the metal detector, they are typically subjected to a more in-depth manual pat down. This second filter is designed to be highly specific, meaning that individuals who are truly terrorists will be detected. To put everyone through a manual pat down would result in even worse delays at the airport than we have today. The metal detector is a very fast test but generates false positives. The manual pat down is very slow but won't generate false negatives. Combined, this scheme balances the need for speed along with a need to reduce false negatives. When detecting TME, a multi-layer detection approach can be employed in a similar manner to minimize false negatives. A first layer would be highly sensitive and a slower more in-depth second layer would focus on minimizing false negatives.

Since quantitative cost data for the outcomes in Table 4.40 are not available, this study will focus on the ratio between false negatives and false positives. Table 4.41 outlines the false negative to false positive ratios for the outcomes in Table 4.40. For the purposes of this study, since $DCost$ can be very high given the seriousness of the threat, the cost of false positives will be set to 1.0 and the cost for false negatives will be set to $1.0 \cdot \lambda$. Thus the false negative to false positive ratio will be λ and this will be varied over a range to demonstrate the effects of various misclassification cost ratios. Table 4.42 shows a modified confusion matrix with costs associated with each possible classification outcome.

One final consideration is that the proportion of spam to legitimate email is significantly higher than the proportion of targeted malicious email to legitimate email.

Table 4.41: False negative to false positive ratios

TME-Detect	TME-Block	Spam-Block
$\frac{FN}{FP} = \frac{DCost}{RCost}$	$\frac{FN}{FP} = \frac{DCost}{BICost+RCost}$	$\frac{FN}{FP} = \frac{UCost}{BICost+RCost}$

Table 4.42: Cost Sensitive Confusion Matrix

	Actual TME	Actual NTME
Predicted TME	TP (1)	FP (1)
Predicted NTME	FN (λ)	TN (1)

According to MessageLabs (2009), for the traffic they monitor, global spam accounted for greater than 85 percent of all email at about 107 billion per day. In contrast, the average number of targeted malicious emails was 48 per day. This underscores the need for a cost sensitive approach for handling this imbalance.

4.5.4 Feature Importance and Cost

For every email a number of features are extracted and then supplied as input to a classifier to determine the type of email. Not all of the features have the same differentiating power to separate targeted malicious email from non-targeted malicious email. In this study, the most important features will be identified and classifier performance will be compared when using a full or reduced set of features. Practically, each feature to be extracted from an email increases the computation time to convert an email into its vector of features. This time in practice is minimal and does not exceed the rate at which emails arrive at the company. However, there is a software development and integration cost associated with extracting certain features. Some features are more trivial than others to implement. For example, extracting a recipient oriented feature such as the average number of Google search hits for all valid recipients involves external information not available in the email itself. The search hit information has to be extracted from Google in advance and exposed to the feature extraction software tools via a database. Trying to query Google in real time while extracting features from an email may introduce delay or confuse Google into

thinking your system is automated bot (thus resulting in blocking of your IP address). Therefore, an understanding of feature importance can help prioritize which features are the most important to focus software development and integration effort.

In this study, feature importance will be measured using mean decrease Gini. The mean decrease Gini measures the quality of a split in every node of the trees. According to Breiman et al. (1984), for a two class problem such as in this study, the Gini impurity, i, for a given node, t, is calculated by the following equation where $p(k|t)$ is the relative proportion of class k emails at node t:

$$i(t) = 2p(NTME|t)p(TME|t) \tag{4.23}$$

The Gini index for a given feature is the sum over all trees in the forest of the decrease in impurity after each split involving that feature. Averaging the Gini index across all trees yields the mean decrease Gini. Every time a split of a node is made on a feature the Gini impurity for the two descendent nodes is less than the parent node. The higher the mean decrease Gini, the greater a feature's importance.

4.5.5 Practical implementation

There are several practical considerations that have to made when implementing the approaches in this study in an operational environment. First, the ability of a classifier to detect targeted malicious email (TME) is related to the quality of features extracted from emails. Aside from the recipient oriented features, the other features have to be threat specific. Analysts have an array of tools at their disposal for detecting TME. As new relevant features of email are identified, it is important that feature extraction tools are updated to expose these new features to the classifier. As threat actors modify their techniques over time, the features may vary in their ability to detect TME. Regular review and update of features will help ensure maximum detection ability.

Second, the recipient oriented features must be updated at periodic intervals. In companies, employees are constantly hired, retired, fired or laid off. Individuals change

jobs and get promoted. Therefore, it is important to keep the relevant recipient oriented information up to date. Specifically, information mirrored from the company directory service and also search hit counts from Google should be updated on a periodic basis. Detection of targeted attacks requires targeted detection and as such these features have to be created and updated specific to the organization or company being targeted. Many features will translate from organization to organization but their relative importance may vary depending on the specific threat profile and organization population.

4.6 Evaluation

In this study, the primary research objective is to demonstrate that detection of targeted malicious email is enhanced when using persistent threat and recipient oriented features instead of conventional techniques. This section will describe the conventional techniques used in this study, the methods used to optimize random forest parameters, and the measures used to compare classifier performance.

4.6.1 Conventional Techniques

Common email filtering architectures include anti-spam filtering in addition to anti-virus filtering. In this study, the conventional techniques used for comparison purposes are Spamassassin and ClamAV. SpamAssassin[14] is a popular open-source spam filtering tool that largely uses email content for decision making. ClamAV[15] is an anti-virus toolset for Unix specifically designed for filtering email. Both tools will be executed against experimental data sets to determine their ability to detect targeted malicious email.

4.6.2 Parameter Optimization

As described in Section 4.5.1, the Random Forest classifier has two primary parameters: k, the number of trees in the forest, and m, the number of random features to consider

[14]The Apache SpamAssassin Project - http://spamassassin.apache.org/
[15]ClamAV - http://www.clamav.net/

for node splitting. In this study, these two parameters will be varied to maximize classifier performance. There is no standard for determining the optimal k and m values. Research by Khoshgoftaar et al. (2007) indicates that a value of $k = 100$ and $m = log_2 M + 1$, where M is the total number of features available, is a starting guideline. Hastie et al. (2008) suggest that $m = \sqrt{M}$ is a reasonable default value. As described in Section 4.5.3, there is a cost difference between false negatives and false positives. Thus the false negative, false positive ratio, λ, will be varied to understand the effect on classifier performance.

4.6.3 Measuring Classifier Performance

This section will describe the metrics to evaluate the performance of classifiers in this study. Androutsopoulos et al. (2000b) outlines measures to evaluate classification performance when filtering spam and non spam email. However, the calculations assume that false positives are more costly than false negative. However, in section 4.5.3 a conceptual model was presented for targeted malicious email (TME) where false negatives were more costly than false positives. New performance measures adjusted for TME follow.

Let N_{NTME} and N_{TME} be the total numbers of non-targeted malicious email and targeted malicious email, respectively, and $n_{Y \to Z}$ the number of emails belonging to classification Y that the classifier classified as belonging to classification Z $(Y, Z \epsilon \{NTME, TME\})$. This can be related to the outcomes described above as follows:

$$n_{TME \to TME} = TP \tag{4.24a}$$

$$n_{NTME \to NTME} = TN \tag{4.24b}$$

$$n_{TME \to NTME} = FN \tag{4.24c}$$

$$n_{NTME \to TME} = FP \tag{4.24d}$$

$$N_{TME} = TP + FN \tag{4.24e}$$

$$N_{NTME} = TN + FP \tag{4.24f}$$

$$N_{NTME} + N_{TME} = TP + TN + FN + FP \tag{4.24g}$$

Accuracy (Acc) is the number of correct classifications as a percentage of total classifications:

$$Acc = \frac{n_{TME \to TME} + n_{NTME \to NTME}}{N_{TME} + N_{NTME}} = \frac{TP + TN}{TP + FN + TN + FP} \tag{4.25}$$

The *error rate* ($Err = 1 - Acc$) is:

$$Err = \frac{n_{TME \to NTME} + n_{NTME \to TME}}{N_{TME} + N_{NTME}} = \frac{FN + FP}{TP + FN + TN + FP} \tag{4.26}$$

These two measures, however, assign equal weights to the false positive ($NTME \to TME$) and false negative ($TME \to NTME$) errors. As established in section 4.5.3, $TME \to NTME$ is λ times more costly than $NTME \to TME$. Accuracy and error rate can be made sensitive to this cost differential by treating each TME as if it were λ emails. This adjustment results in the following definition of *weighted accuracy* ($WAcc$) and *weighted error rate* ($WErr = 1 - WAcc$):

$$WAcc = \frac{\lambda \cdot n_{TME \to TME} + n_{NTME \to NTME}}{\lambda \cdot N_{TME} + N_{NTME}} = \frac{\lambda \cdot TP + TN}{\lambda(TP + FN) + TN + FP} \tag{4.27}$$

$$WErr = \frac{\lambda \cdot n_{TME \to NTME} + n_{NTME \to TME}}{N_{TME} + N_{NTME}} = \frac{\lambda \cdot FN + FP}{\lambda(TP + FN) + TN + FP} \tag{4.28}$$

For comparison purposes, the classifier will be compared to a "baseline" approach where no filter is present. The absence of a filter means that non-targeted malicious emails are, correctly, not detected and targeted malicious emails are, incorrectly, not detected. The *weighted accuracy* ($WAcc^b$) and *weighted error rate* ($WErr^b = 1 - WAcc^b$) of this baseline are:

$$WAcc^b = \frac{N_{NTME}}{N_{NTME} + \lambda \cdot N_{TME}} = \frac{TN + FP}{TN + FP + \lambda(TP + FN)} \tag{4.29}$$

$$WErr^b = \frac{\lambda \cdot N_{TME}}{N_{NTME} + \lambda \cdot N_{TME}} = \frac{\lambda(TP + FN)}{TN + FP + \lambda(TP + FN)} \tag{4.30}$$

A ratio between the baseline weighted error rate ($WErr^b$) and the weighted error rate ($WErr$), called the *Total Cost Ratio (TCR)*, allows the performance of a classifier to be compared to a baseline (no classifier) approach:

$$TCR = \frac{WErr^b}{WErr} = \frac{N_{TME}}{n_{NTME \to TME} + \lambda \cdot n_{TME \to NTME}} = \frac{\lambda \cdot (TP + FN)}{FP + \lambda \cdot FN} \tag{4.31}$$

Greater TCR values indicate better performance. If TCR <1, using no classifier is better than using the classifier. An intuitive definition of TCR follows: TCR measures the time and cost associated with responding to an incident due to a TME delivered to a user (N_{TME}), compared to the time needed for an analyst to review an email mistakenly flagged as TME ($n_{NTME \to TME}$) plus the time and cost associated with responding to an incident due to a TME mistakenly classified as a NTME ($n_{TME \to NTME}$). A higher TCR means a greater portion of an organization's response is dedicated to true incidents rather than errors.

With an imbalanced data set, *Accuracy* (*Acc*) can be misleading. For example, a classifier running against a data set with 95% non-targeted malicious email and 5% targeted malicious email can be 95% accurate if it simply identifies all email as non-targeted malicious. Thus, sometimes it is beneficial to look at the two classes of email separately.The True Positive Rate, TPR, sometimes referred to as sensitivity, describes how well the classifier recognizes all targeted malicious email.

$$TPR = sensitivity = \frac{n_{TME \to TME}}{n_{TME \to TME} + n_{TME \to NTME}} = \frac{TP}{TP + FN} \tag{4.32}$$

The True Negative Rate, TNR, sometimes referred to as specificity, describes how well the classifier recognizes all non-targeted malicious email.

$$TNR = specificity = \frac{n_{NTME \to NTME}}{n_{NTME \to NTME} + n_{NTME \to TME}} = \frac{TN}{TN + FP} \tag{4.33}$$

The False Positive Rate, FPR, is the proportion of non-targeted malicious emails that

were incorrectly classified as targeted malicious emails.

$$FPR = 1 - specificity = \frac{n_{NTME \to TME}}{n_{NTME \to NTME} + n_{NTME \to TME}} = \frac{FP}{TN + FP} \tag{4.34}$$

The False Negative Rate, *FNR*, is the proportion of targeted malicious emails emails that were incorrectly classified as non-targeted malicious emails.

$$FNR = 1 - sensitivity = \frac{n_{TME \to NTME}}{n_{TME \to TME} + n_{TME \to NTME}} = \frac{FN}{TP + FN} \tag{4.35}$$

Precision describes how well the classifier correctly identifies a targeted malicious email when it classifies an email as being targeted malicious.

$$precision = \frac{n_{TME \to TME}}{n_{TME \to TME} + n_{NTME \to TME}} = \frac{TP}{TP + FP} \tag{4.36}$$

A 100% highly sensitive classifier will correctly identify all targeted malicious email but in the process may incorrectly identify some emails that are not targeted malicious emails (false positives). In contrast, a 100% highly specific classifier will correctly identify all non-targeted malicious email but in the process may incorrectly identify some targeted malicious email. For the purposes of detecting targeted malicious email, a high sensitivity at the risk of having a lower specificity is desired.

Supervised classification involves training a classifier, in this study Random Forest, with pre-labeled emails and then executing the classifier against a test data set. In this study, two test methods will be used: 10-fold cross validation and independent test data. In a m-fold cross validation, the data set is divided randomly into m disjoint subsets of equal size n/m, where n is the number of emails in the data set. The classifier is then trained m times, each time withholding one set for testing. Errors are averaged across these m executions of the classifier (Duda et al., 2000). With an independent test data set, one data set will be used for training and a completely separate data set will be used for testing. None of the test samples will be in the training data set.

4.6.4 Summary of Analysis Procedures

The totality of analysis procedures from raw data to results are summarized in the steps below:

1. Data collection and feature extraction - The emails used in this research were collected using the tools outlined in Section 4.2.1. Raw emails were processed using Perl scripts and the features as described in Section 4.4.3 were extracted from those emails. Output files for TME and NTME email were created listing a unique identifier for each email along with the feature vector (as described in Section 4.4.6) for each email. The output files are formatted using the attribute-relation file format (ARFF) which is a format commonly used by data classification tools. As part of the email features, various characteristics of the email recipients were also recorded.

2. Supervised classifier training - Once the data sets were created, the next step was to train the random forest classifier. This was performed using the WEKA tool as outlined in Section 4.2.1. The WEKA tool imports the training data set into its database. The training emails are labeled which allows the classifier to associate certain features with either TME or NTME.

3. Classifier testing - Once the training data set is loaded, WEKA allows an analyst to execute the newly created classifier against a test data set. In this dissertation, two test data sets are used: the *NTME1-TME1* data set is used in a cross-validation approach (Section 4.6.3) and the *TS1* data set is used as a completely independent test set). WEKA takes each test email, runs it through the random forest and categorizes each email as either TME or NTME. The correct classification of all emails is known so WEKA is able to determine how many true positives, true negatives, false positives and false negatives result.

4. Optimization - In this research, there are two types of optimization. The first accounts for the difference in cost between false negatives and false positives. As outlined in Section 4.5.4 the classifier is trained assuming the cost of false

negatives and false positives are equal. Subsequent trainings set the cost of false negatives at a multiple of false positives. This cost differential forces the classifier to avoid making one type of error over another. The second optimization accounts for the random forest classifier itself. The random forest classifier, as outlined in Section 4.5.1 has two parameters. These two parameters are varied to see how the classifier performs with respect to false negatives and false positives.

5. Evaluation - The optimized random forest classifiers are then compared to conventional email filtering techniques to understand the difference in performance.

Chapter 5: Evaluation

This chapter presents the results of executing the methods outlined in this study. First, the Random Forest algorithm will be used to determine the importance of the features outlined in Table 4.9. Second, newer approaches as outlined in this study will be used to filter out targeted malicious email from different data sets. The newer approaches will be analyzed with respect to various configuration parameters to maximize detection strength. These newer approaches will also be compared to conventional techniques to determine if there is an overall improvement.

5.1 Feature Importance

Table 4.9 lists a total of 83 features used to represent each email through the classification process. These features are not equal with respect to classification strength; some features over others provide greater separation between targeted malicious email and non-targeted malicious email. From a practical implementation perspective, extracting the fewest number of features necessary is easier to implement and faster to process. Therefore, the fewest number of features that still achieves the desired classification strength is ideal. As described in section 4.5.4, the Random Forests algorithm supports calculating feature importance. Figure 5.1 shows the output of calculating feature importance using the *NTME1-TME1* data set; only the twenty-five most important features are shown.

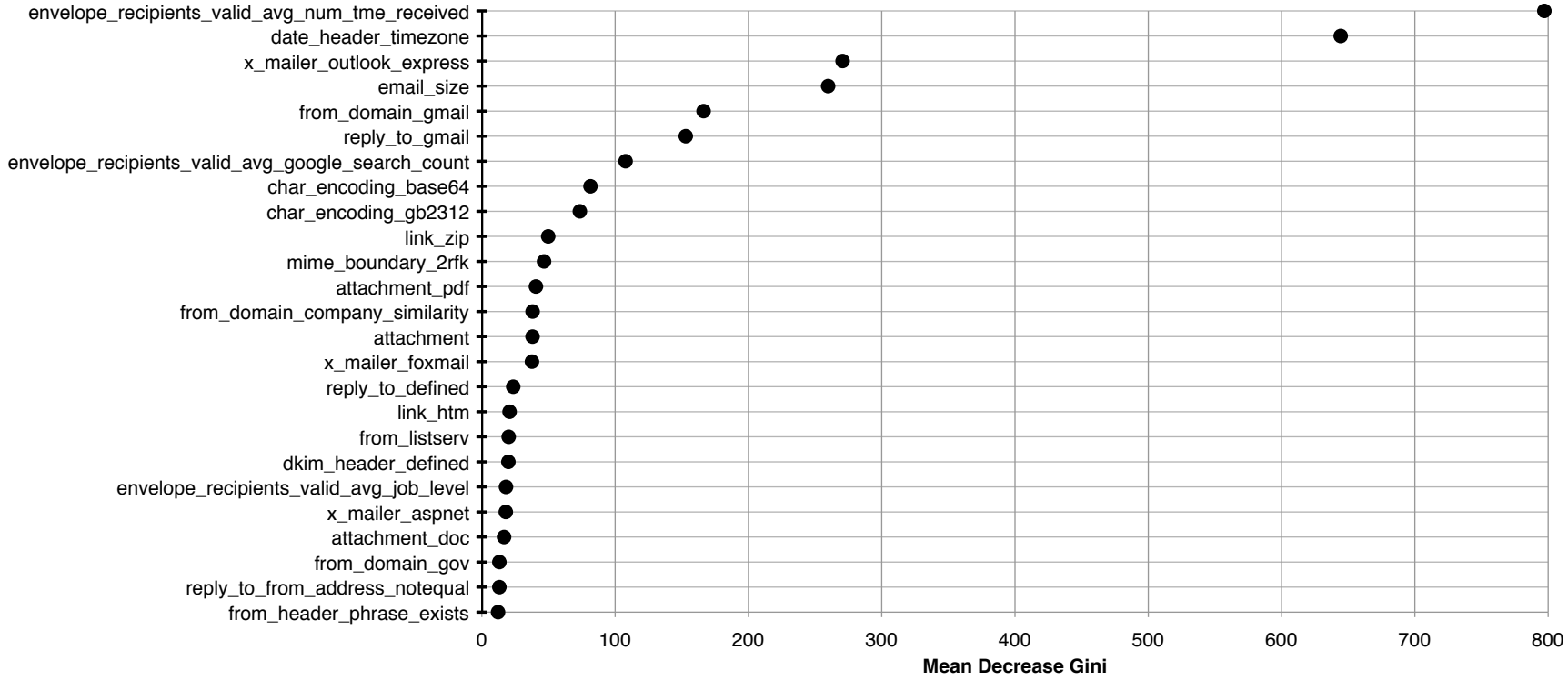


Figure 5.1: Feature Importance Using Mean Decrease Gini: The twenty-five most important features

Using the Mean Decrease Gini measure, the top twenty-five features are shown in Table 5.1.

Table 5.1: Top twenty-five features based on Mean Decrease Gini

Feature	Gini
envelope_recipients_valid_avg_num_tme_received	797.14
date_header_timezone	644.36
x_mailer_outlook_express	270.66
email_size	259.79
from_domain_gmail	166.35
reply_to_gmail	152.97
envelope_recipients_valid_avg_google_search_count	107.81
char_encoding_base64	81.5
char_encoding_gb2312	73.51
link_zip	49.78
mime_boundary_2rfk	46.65
attachment_pdf	40.57
from_domain_company_similarity	38.07
attachment	38.03
x_mailer_foxmail	37.6
reply_to_defined	23.54
link_htm	20.79
from_listserv	20.13
dkim_header_defined	19.91
envelope_recipients_valid_avg_job_level	18.11
x_mailer_aspnet	17.96
attachment_doc	16.66
from_domain_gov	13.13
reply_to_from_address_notequal	13.13
from_header_phrase_exists	12.15

5.2 Random forest classifier against the *NTME1-TME1* data set

This section presents the results of processing the *NTME1-TME1* data set using an optimized random forest classifier. First, conventional techniques will be assessed. Second, the random forest parameters will be optimized. Third, a cost sensitive random forest classifier will be assessed. Fourth, features will be successively reduced to determine how classification strength degrades with fewer and fewer features included in the random forest model. Finally, statistical tests will be conducted to

compare newer and conventional techniques. As described in Section 4.6.3, a 10-fold cross validation test method is used.

5.2.1 Conventional email filtering techniques

Two popular, conventional email filtering tools were applied against the *NTME1-TME1* data set, SpamAssassin and ClamAV. The following sections present the results of using these tools.

SpamAssassin

SpamAssassin was configured using the stock distribution of heuristics with no bayesian learning for email body content. Each heuristic in SpamAssassain has a score associated with it. Some scores are positive and some scores are negative depending on whether the heuristic is detecting a negative or positive attribute, respectively. By default, an email with a score over 5.0 is flagged as Spam. Table 5.2 contains the results of executing SpamAssassin against the *TME1* data set (reference Table 4.3). The false negative rate is calculated using Equation 4.22. Table 5.3 lists all of the heuristics

Table 5.2: Results of running SpamAssassin against the *TME1* data set

Outcome	# Emails
True Positives (TP)	626
False Negatives (FN)	1689
Total Emails	2315
False Negative Rate	0.73

that matched for emails in the *TME1* data set. A number of the heuristics are related to the body content of the email but a few are similar to features extracted for this study, such as base64 encoding. More than half of the True Positive detections are due to a heuristic that looks for an invalid Message-ID.

Table 5.3: Number of emails from the *TME1* data set that matched SpamAssassin heuristics

# Emails	Score	Heuristic	Description
1	0.0	HS_INDEX_PARAM	URL Link contains a common tracker pattern.
1	0.0	HTML_SHORT_LINK_IMG_1	HTML is very short with a linked image
1	0.0	UPPERCASE_50_75	Message body is 50-75% uppercase
1	0.1	HTTP_ESCAPED_HOST	URI Uses %-escapes inside a URL's hostname
1	1.1	WEIRD_PORT	URI Uses non-standard port number for HTTP
1	1.4	WEIRD_QUOTING	BODY Weird repeated double-quotation marks
1	1.5	DATE_IN_FUTURE_03_06	Date: is 3 to 6 hours after Received: date
1	1.5	HTTP_EXCESSIVE_ESCAPES	URI Completely unnecessary %-escapes inside
1	1.9	HTML_IMAGE_ONLY_04	BODY HTML: images with 0-400 bytes of words
1	4.2	TVD_STOCK1	BODY Message looks like its pushing a stock
2	0.0	FORGED_OUTLOOK_HTML	Outlook can't send HTML message only
2	0.0	HTML_MESSAGE	BODY HTML included in message
2	0.0	HTML_MIME_NO_HTML_TAG	HTML-only message, but there is no HTML tag
2	0.1	FORGED_OUTLOOK_TAGS	Outlook can't send HTML in this format
2	1.9	MIME_HTML_ONLY	BODY Message only has text/html MIME part
2	2.9	RCVD_ILLEGAL_IP	Received: contains illegal IP address
3	1.2	MIME_HEADER_CTYPE_ONLY	'Content-Type' found without required MIME
6	3.0	TVD_RCVD_SINGLE	Received: line contains localhost as a server name

Continued on next page...

Table 5.3 – Continued

# Emails	Score	Heuristic	Description
7	1.5	DATE_IN_PAST_06_12	Date: is 6 to 12 hours before Received: date
7	1.7	MIME_BASE64_TEXT	RAW Message text disguised using base64 encoding
7	1.8	HS_FORGED_OE_FW	Outlook does not prefix forwards with "FW & "
7	3.2	FROM_LOCAL_NOVOWEL	From: localpart has series of non-vowel letters
9	0.0	UNPARSEABLE_RELAY	Informational message has unparseable relay line
9	3.0	FORGED_MUA_OUTLOOK	Forged mail pretending to be from MS Outlook
10	0.1	MISSING_MIMEOLE	Message has X-MSMail-Priority, but no X-MimeOLE
10	1.9	SUBJ_ALL_CAPS	Subject is all capitals
11	0.0	NORMAL_HTTP_TO_IP	URI Uses a dotted-decimal IP address in URL
11	0.1	RDNS_DYNAMIC	Delivered to trusted network by host with Dynamic DNS
11	1.5	FH_HOST_EQ_PACBELL_D	Host is pacbell.net dsl
11	1.5	MSGID_FROM_MTA_HEADER	Message-Id was added by a relay
11	1.6	FROM_EXCESS_BASE64	From base64 encoded unnecessarily
11	2.9	HTTPS_IP_MISMATCH	BODY IP to HTTPS link found in HTML
12	3.3	RCVD_BAD_ID	Received: header contains a badly formatted ID parameter
13	0.0	UNPARSEABLE_RELAY	Informational message has unparseable relay lines
16	1.3	MSOE_MID_WRONG_CASE	Incorrect Message-ID label for Outlook Express
16	1.7	DEAR_SOMETHING	BODY Contains 'Dear (something)'

Continued on next page...

Table 5.3 – Continued

# Emails	Score	Heuristic	Description
21	1.4	DIET_1	BODY Lose Weight Spam
21	1.5	MISSING_HEADERS	Missing To: header
21	2.9	DC_GIF_UNO_LARGO	Message contains a single large inline gif
25	0.1	RDNS_NONE	Delivered to trusted network by a host with no reverse DNS
29	0.7	HTML_FONT_FACE_BAD	BODY HTML font face is not a word
35	1.3	DATE_IN_PAST_03_06	Date: is 3 to 6 hours before Received: date
38	0.0	MIME_HTML_ONLY_MULTI	Multipart message only has text/html MIME parts
38	1.4	PLING_QUERY	Subject has exclamation mark and question mark
72	0.1	FORGED_OUTLOOK_TAGS	Outlook can't send HTML in this format
75	0.0	FORGED_OUTLOOK_HTML	Outlook can't send HTML message only
77	0.0	HTML_MIME_NO_HTML_TAG	HTML-only message, but there is no HTML tag
79	3.0	BASE64_LENGTH_79_INF	Base64 should only be 76 chars long
100	3.0	FORGED_MUA_OUTLOOK	Forged mail pretending to be from MS Outlook
103	1.5	MPART_ALT_DIFF	BODY HTML and text parts are different
130	1.9	MIME_HTML_ONLY	BODY Message only has text/html MIME parts
143	1.7	MIME_BASE64_TEXT	RAW Message text disguised using base64 encoding
327	2.3	TVD_SPACE_RATIO	BODY High ratio of spaces to non-spaces
331	2.9	MSGID_OUTLOOK_INVALID	Message-Id is fake (in Outlook Express format)

Continued on next page...

Table 5.3 – Continued

# Emails	Score	Heuristic	Description
494	0.0	HTML_MESSAGE	BODY HTML included in message

Executing SpamAssassin against the *NTME1* data set provides little value as any positives reported by SpamAssassin may be Spam, not TME false positives. Thus, to calculate a Total Cost Ratio (TCR) for SpamAssassin against the *NTME1-TME1* data set, a false positive count of 0 and true negative count of $20,894$ (see Table 4.2) is used. Table 5.4 shows the TCR for SpamAssassin. The TCR does not change for increasing λ since there are no false positives. Assuming a false positive count of 0, 1.37 is the highest TCR SpamAssassin can achieve given its high false negative rate. To

Table 5.4: SpamAssassin Total Cost Ratio for *NTME1-TME1* data set

Outcome	# Emails
True Positives (TP)	626
False Negatives (FN)	1689
False Positives (FP)	0
True Negatives (TN)	20,894

Total Cost Ratio for λ=1,2,10,100	1.37

see how SpamAssassin handles NTME and TME differently, executing SpamAssassin against the *NTME1* data set results in 4,221 emails flagged as Spam. Out of a total of 20,894 emails in the *NTME1* data set, that means that SpamAssassin had a 20.2% positive detection rate. This compares to the 27.0% positive detection rate (see Table 5.2) from executing SpamAssassin against the *TME1* data set. Comparing these two proportions using a Z-test for proportions (see Section 4.3.1) results in a Z test statistic value of 7.68 which means that at the $alpha$=0.01 level of significance, you can accept the alternative hypothesis that the positive detection rate of SpamAssassin with the *TME1* data set is greater than the *NTME1* data set.

ClamAV

ClamAV was configured using the stock distribution with a virus definition file as of January 18, 2010 (the last TME in the *NTME1-TME1* data set was received December 19, 2009). Since ClamAV is an anti-virus tool its primary purpose in this study is to detect malicious software embedded within emails, most often in the form

of file attachments. In the *TME1* data set, there are 1,065 emails with at least one attachment. Table 5.5 contains the results of executing ClamAV against the *TME1* data set (reference Table 4.3). The false negative rate is calculated using Equation 4.22. A similar situation as SpamAssassin exists when considering the execution of

Table 5.5: Results of running ClamAV against the *TME1* data set

Outcome	# Emails
True Positives (TP)	223
False Negatives (FN)	2,092
Total Emails	2315
Total Emails with Attachment	1065
False Negative Rate	0.90
False Negative Rate (attachment only)	0.79

ClamAV against the *NTME1* data set. Any positives reported by ClamAV in the *NTME1* data set may be malicious in nature, not targeted malicious email (TME) but standard Internet worms and viruses that should be filtered in an email system. Thus, assigning a false positive count of 0 to ClamAV will yield the maximum TCR for ClamAV given its false negative rate. Table 5.6 shows the TCR for ClamAV. The TCR does not change for increasing λ since there are no false positives. With a false positive count of 0, 1.11 is the highest TCR ClamAV can achieve given its high false negative rate. To see how ClamAV handles NTME and TME differently,

Table 5.6: ClamAV Total Cost Ratio for *NTME1-TME1* data set

Outcome	# Emails
True Positives (TP)	223
False Negatives (FN)	2,092
False Positives (FP)	0
True Negatives (TN)	20,894
Total Cost Ratio for λ=1,2,10,100	1.11

executing ClamAV against the *NTME1* data set results in 2,097 emails flagged as

malicious. Out of a total of 20,894 emails in the *NTME1* data set, that means that ClamAV had a 10.0% positive detection rate. This compares to the 9.6% positive detection rate (see Table 5.5) from executing ClamAV against the *TME1* data set. Comparing these two proportions using a Z-test for proportions (see Section 4.3.1) results in a Z test statistic value of 0.61 which means that even at the $alpha$=0.05 level of significance, you are not able to reject the null hypothesis that the positive detection rate of ClamAV with the *TME1* data set is different than the *NTME1* data set. This could indicate that ClamAV treats TME and NTME the same and does not have a strong differentiating capability between the two.

SpamAssassin+ClamAV combined

Email filtering regimes in organizations will often consist of two stages: spam filtering and anti-virus filtering. To determine the joint detection power of the two conventional methods, email from the *TME1* data set was passed first through SpamAssassin and then through ClamAV to determine the aggregate ability to detect TME. Some emails slip past SpamAssassin but are detected by ClamAV resulting in a positive detection for the joint capability. After processing the *TME1* data set with SpamAssassin, 626 emails were correctly detected and 1,689 emails were missed. Passing those remaining 1,689 through ClamAV resulted in 131 additional detections. At the end, 1,558 TME emails were left undetected resulting in a 67% false negative rate for the joint SpamAssassin+ClamAV method. Table 5.7 summarizes the results. Again,

Table 5.7: Results of running SpamAssassin+ClamAV against the *TME1* data set

Outcome	# Emails
True Positives (TP)	757
False Negatives (FN)	1,558
Total Emails	2315
False Negative Rate	0.67

assuming no false positives for SpamAssassin+ClamAV, the TCR for this method is

1.49. Table 5.8 summarizes these results.

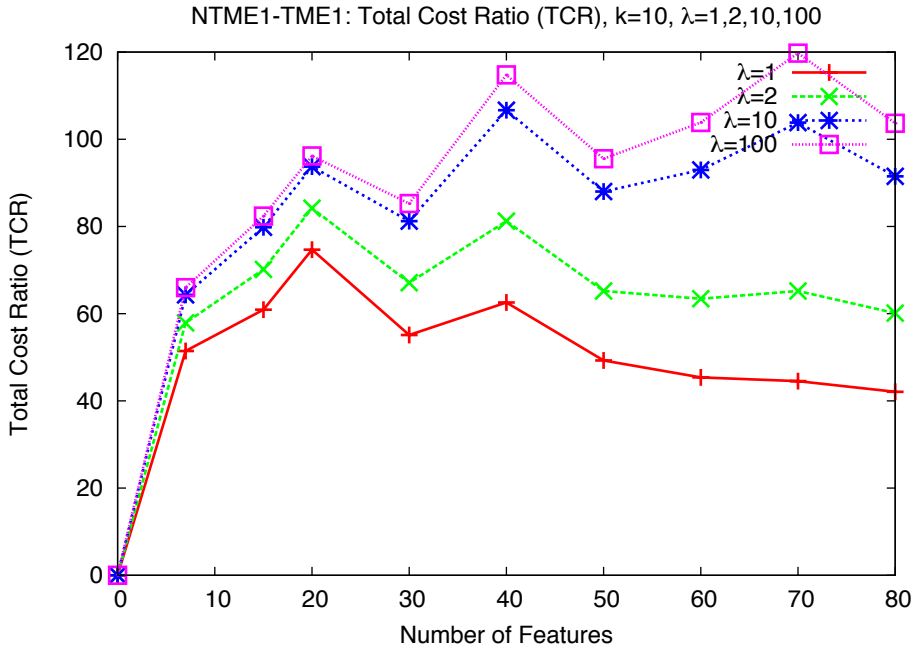
Table 5.8: SpamAssassin+ClamAV Total Cost Ratio for *NTME1-TME1* data set

Outcome	# Emails
True Positives (TP)	757
False Negatives (FN)	1,558
False Positives (FP)	0
True Negatives (TN)	20,894
Total Cost Ratio for λ=1,2,10,100	1.49

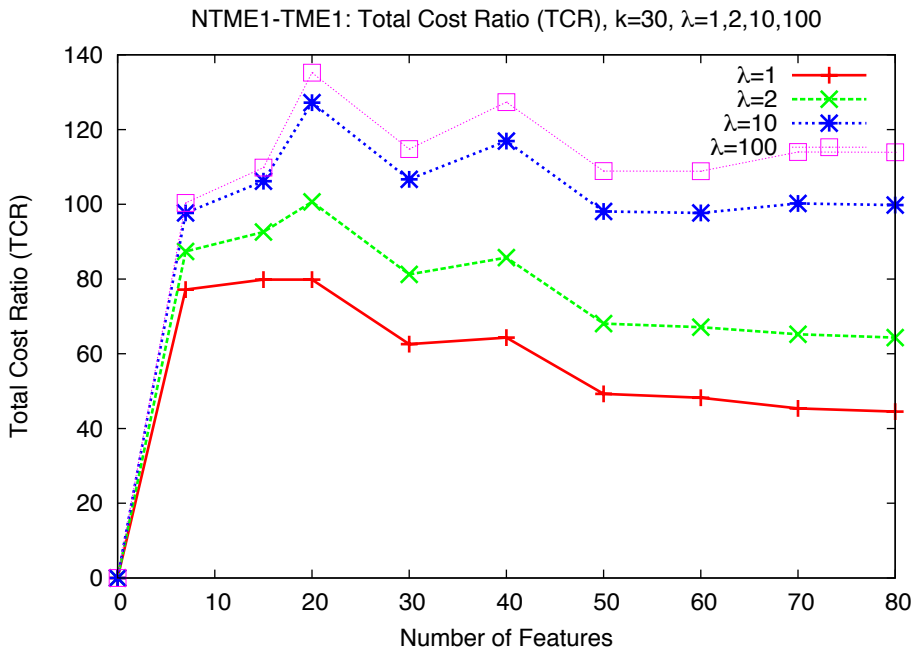
5.2.2 Random forest parameter optimization

As described in Section 4.5.1, the Random Forest classifier has two primary parameters: k, the number of trees to use in the forest, and m, the number of random features to consider for node splitting. This section compares the results of varying these two parameters. Figure 5.2 shows Total Cost Ratio (TCR) graphs for varying number of trees when using increasing subsets of features for node splitting. Time to model a random forest increases as the number of trees increase but since this is a one-time cost it is ignored. From a practical standpoint, new random forest models can be built whenever there are new email samples to justify creating a new model.

A λ=1 means the cost of a false positive and false negative is the same. However, as described with the cost sensitive evaluation in Sections 4.5.3 and 4.6.3, λ can be increased to change the cost ratio between false negatives and false positives. Assuming false negative and false positive misclassification costs are equal, the Random Forest has the highest TCR of 92.60 with $k = 500$ and $m = 15$. With large forest sizes, TCR is relatively stable even as more features are used for node splitting. Assuming false negatives cost twice as much as false positives, the Random Forest has the highest TCR of 105.23 with $k = 50$ and $m = 20$ or $m = 30$. Assuming false negatives cost ten or one-hundred times as much as false positives, the Random Forest has the highest TCR of 141.16 and 152.91, respectively, with $k = 50$ and $m = 30$. A random forest with parameters $k = 50$ and $m = 30$ will be used for comparison purposes when

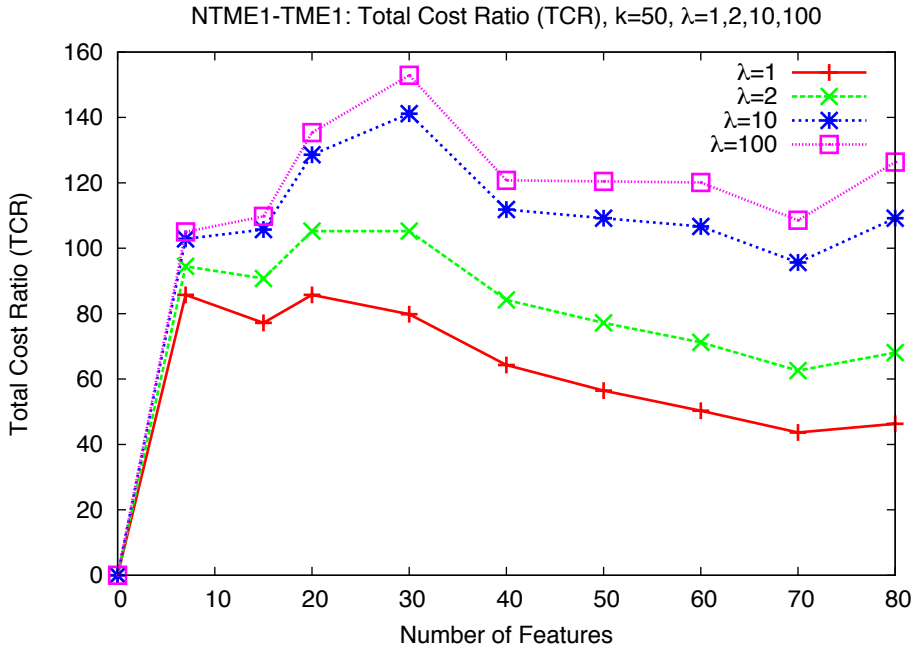


(a) TCR at k=10, λ=1,2,10,100

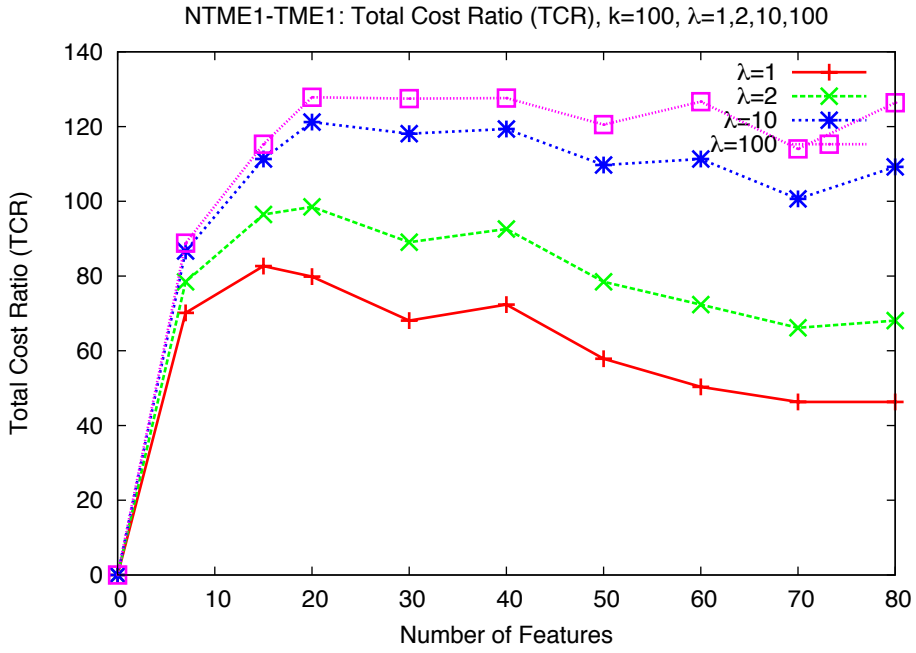


(b) TCR at k=30, λ=1,2,10,100

Figure 5.2: Total cost ratio optimization for random forest using the NTME1-TME1 data set (continued...)

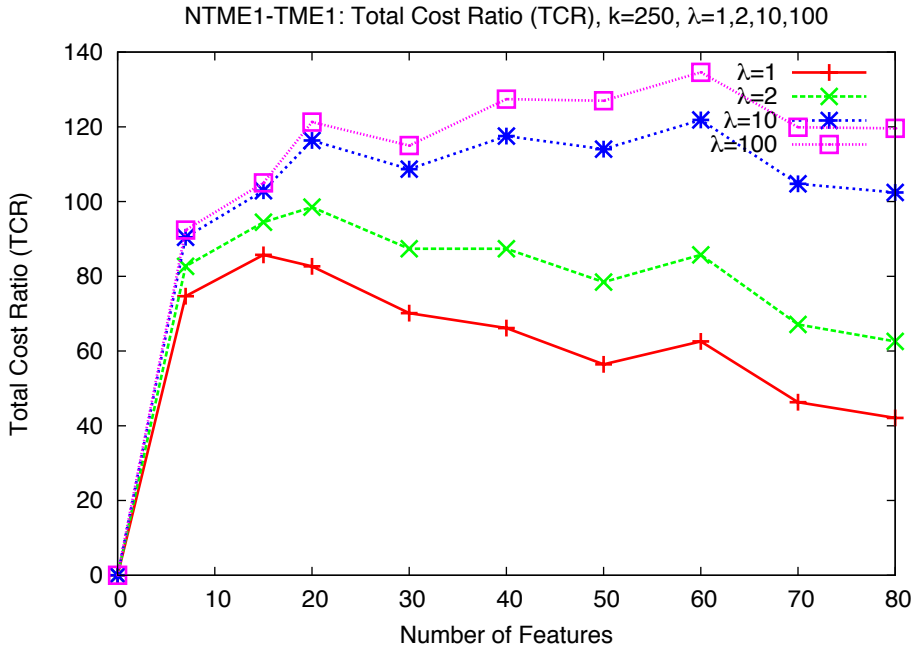


(c) TCR at k=50, λ=1,2,10,100

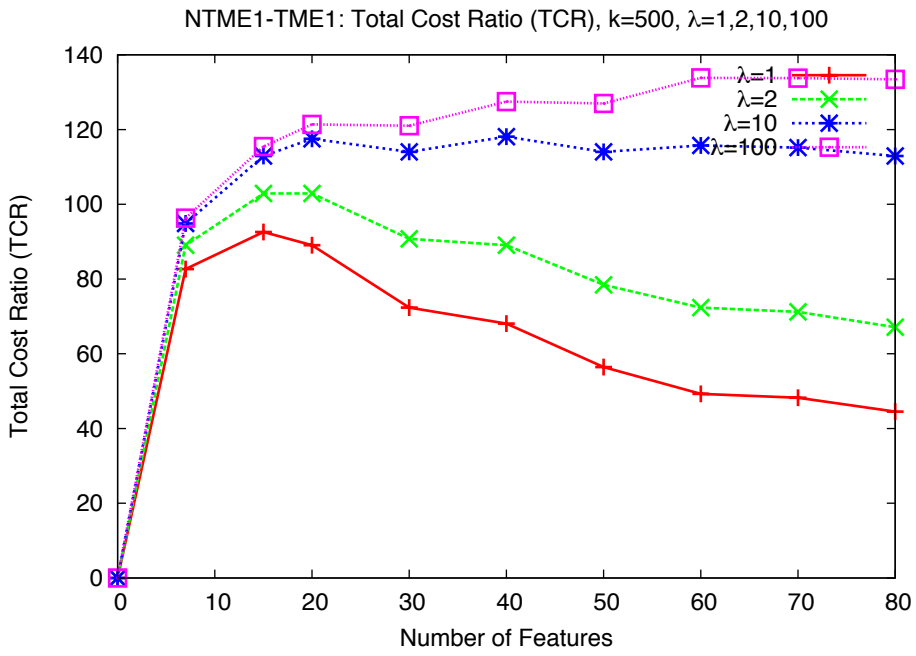


(d) TCR at k=100, λ=1,2,10,100

Figure 5.2: Total cost ratio optimization for random forest using the NTME1-TME1 data set (continued...)



(e) TCR at k=250, λ=1,2,10,100



(f) TCR at k=500, λ=1,2,10,100

Figure 5.2: Total cost ratio optimization for random forest using the NTME1-TME1 data set

analyzing the *NTME1-TME1* data set. The full data for random forest parameter optimization for the *NTME1-TME1* data set is available in Appendix C.1.

5.2.3 Cost sensitive learning

In Section 4.5.3 cost sensitive learning and classification techniques were outlined that account for the difference in cost between a false positive and false negative. Table 5.9 shows the results of processing the *NTME1-TME1* data set through a random forest classifier using the parameters optimized in the previous section ($k = 50$, $m = 30$). Several executions are done, each time with a different false negative, false positive cost ratio (λ) for the learning process. Evaluation is still done considering the cost difference between false negatives and false positives. The full data for cost sensitive learning for the *NTME1-TME1* data set is available in Appendix C.2. Making false

Table 5.9: Summary of cost sensitive learning for the *NTME1-TME1* data set with $k = 50$, $m = 30$

cost ratio	**TCR,** $\lambda = 1$	**TCR,** $\lambda = 2$	**TCR,** $\lambda = 10$	**TCR,** $\lambda = 100$
$\lambda = 1$	79.83	105.23	141.16	152.91
$\lambda = 2$	66.14	90.78	129.33	142.99
$\lambda = 10$	48.23	77.17	148.40	187.30
$\lambda = 100$	8.61	16.96	75.90	348.12

negatives twice the cost of false positives in the learning process actually decreased the overall total cost ratio (TCR). An additional false negative and 5 more false positives were generated. However, when increasing λ, the TCR improved. With false negatives costing one-hundred times false positives, only 4 false negatives were generated. However, this came at a cost of 265 false positives (1.27%). Organizations would need to evaluate whether the increased true positive detection strength is worth the increased false positives. Increasing false positives too high can lead to analyst desensitization.

5.2.4 Feature reduction

In this section the number of features available for classification is successively reduced to show how the false negative performance degrades. The full data is available in

Appendix C.3. Figure 5.3 shows how the false negative rate degrades as successively fewer features, by order of decreasing importance, are included in the random forest classifier. More than the top 20 features have to be removed from the random forest classifier before the false negative performance is equal to SpamAssassin+ClamAV. Figure 5.4 shows how the false negative rate degrades as successively fewer features, by

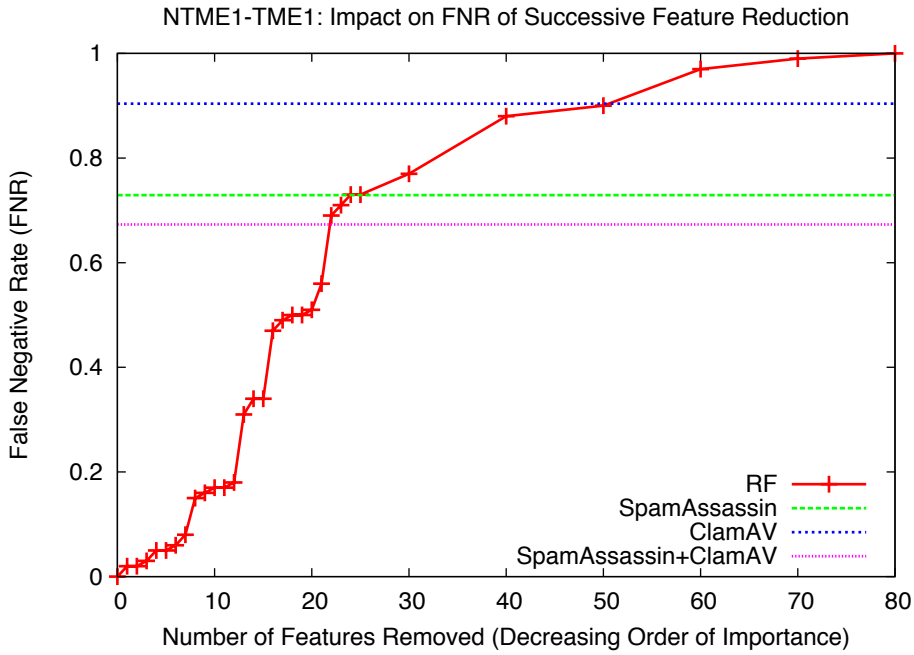


Figure 5.3: Feature reduction for the *NTME1-TME1* data set - Removing features in order of decreasing importance

order of increasing importance, are included in the random forest classifier. Even with just one feature available for classification, the random forest classifier outperforms SpamAssassin, ClamAV and SpamAssassin+ClamAV. The full data is available in Appendix C.4

5.2.5 Comparing false negative rates between two detection methods

Section 4.3.3 described how to compare if two detection methods differ significantly in ability to detect targeted malicious email. Table 5.10 shows the contingency table for the random forest based classifer developed in this study against SpamAssassin. This table summarizes the TME detection ability difference between the two methods.

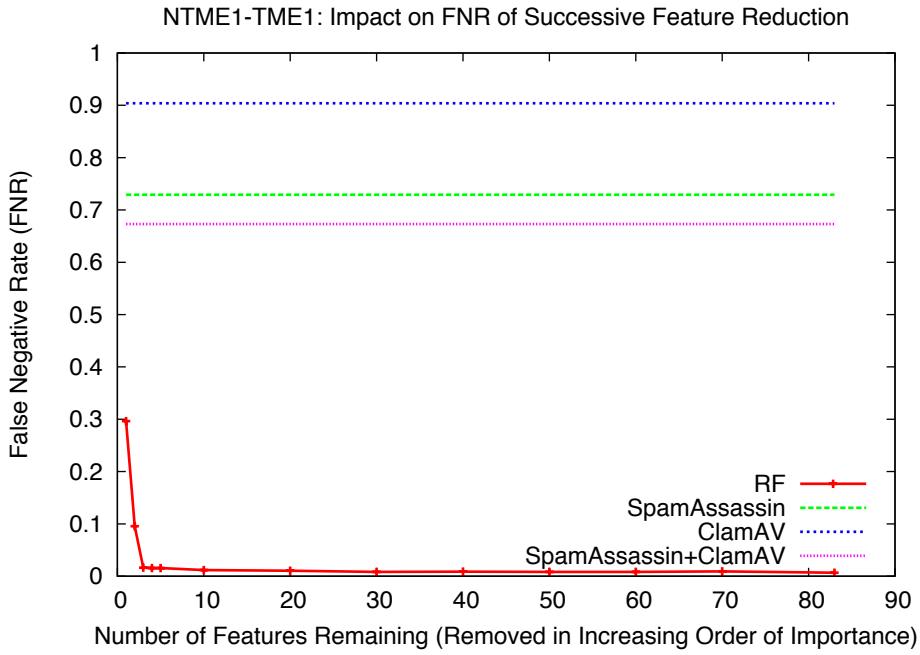


Figure 5.4: Feature reduction for the *NTME1-TME1* data set - Removing features in order of increasing importance

Using Equation 4.17 the χ^2 test statistic is 1,662.1 which is greater than the critical

Table 5.10: *NTME1-TME1*: Contingency Table for TME detection between Random Forest and SpamAssassin

	RF-Correct	RF-Error
SpamAssassin-Correct	621	5
SpamAssassin-Error	1,679	10

value of 6.635 at the $\alpha = 0.01$ level of significance. This means the null hypothesis that the two detection methods are the same in the ability to detect targeted malicious email is rejected. Table 5.11 shows the contingency table for Random Forest vs. ClamAV. The χ^2 test statistic is 2,073.00 which is greater than the critical value of 6.635 at the $\alpha = 0.01$ level of significance. This means the null hypothesis that the two detection methods are the same in the ability to detect targeted malicious email is rejected. Table 5.12 shows the contingency table for Random Forest vs. SpamAssassin+ClamAV. The χ^2 test statistic is 1,541.1 which is greater than the critical value of 6.635 at the $\alpha = 0.01$ level of significance. This means the null hypothesis that the two detection

Table 5.11: *NTME1-TME1*: Contingency Table for TME detection between Random Forest and ClamAV

	RF-Correct	RF-Error
ClamAV-Correct	222	1
ClamAV-Error	2,078	14

Table 5.12: *NTME1-TME1*: Contingency Table for TME detection between Random Forest and SpamAssassin+ClamAV

	RF-Correct	RF-Error
SpamAssassin+ClamAV-Correct	742	5
SpamAssassin+ClamAV-Error	1,558	10

methods are the same in the ability to detect targeted malicious email is rejected.

In summary, the random forest classifier with persistent threat and recipient oriented features outperformed conventional techniques such as SpamAssassin and ClamAV. The random forest classifier was able to achieve a false negative rate of 0.6% with only a 0.1% false positive rate.

5.3 Random forest classifier against the *TS1* data set

This section presents the results of processing the *TS1* data set using an optimized random forest classifier. The classifier is trained using the *NTME1-TME1* data set and then tested using the *TS1* data set. This is important since the *TS1* and *NTME1-TME1* are independent: there are no emails from the *TS1* data set in the *NTME1-TME1* data set. This helps establish the classifier's ability to detect TME completely outside of the training data set.

First, conventional techniques will be assessed. Second, the random forest parameters will be optimized. Third, a cost sensitive random forest classifier will be assessed.

5.3.1 Conventional email filtering techniques

SpamAssassin

Table 5.13 contains the results of executing SpamAssassin against the targeted malicious emails in the *TS1* data set (reference Table 4.6). The false negative rate is calculated using Equation 4.22. Again, since SpamAssassin positives are not TME false positives, a false positive count of 0 is conservatively used. Table 5.14 shows the TCR for SpamAssassin against the *TS1* data set. The TCR does not change for increasing λ since there are no false positives. With a false positive count of 0, 1.13 is the highest TCR SpamAssassin can achieve given its high false negative rate.

Table 5.13: Results of running SpamAssassin against the TME in the *TS1* data set

Outcome	# Emails
True Positives (TP)	5
False Negatives (FN)	39
Total TME	44
False Negative Rate	0.89

Table 5.14: SpamAssassin Total Cost Ratio for *TS1* data set

Outcome	# Emails
True Positives (TP)	5
False Negatives (FN)	39
False Positives (FP)	0
True Negatives (TN)	1,457,685
Total Cost Ratio for λ=1,2,10,100	1.13

ClamAV

Table 5.15 contains the results of executing ClamAV against the targeted malicious emails in the *TS1* data set (reference Table 4.6). The false negative rate is calculated using Equation 4.22. Again, since ClamAV true positives are not TME false positives,

Table 5.15: Results of running ClamAV against the TME in the *TS1* data set

Outcome	# Emails
True Positives (TP)	7
False Negatives (FN)	37
Total TME	44
Total Emails with Attachment	44
False Negative Rate	0.84

a false positive count of 0 is conservatively used. Table 5.16 shows the TCR for ClamAV against the *TS1* data set. The TCR does not change for increasing λ since there are no false positives. With a false positive count of 0, 1.19 is the highest TCR ClamAV can achieve given its high false negative rate.

Table 5.16: ClamAV Total Cost Ratio for *TS1* data set

Outcome	# Emails
True Positives (TP)	7
False Negatives (FN)	37
False Positives (FP)	0
True Negatives (TN)	1,457,685
Total Cost Ratio for λ=1,2,10,100	1.19

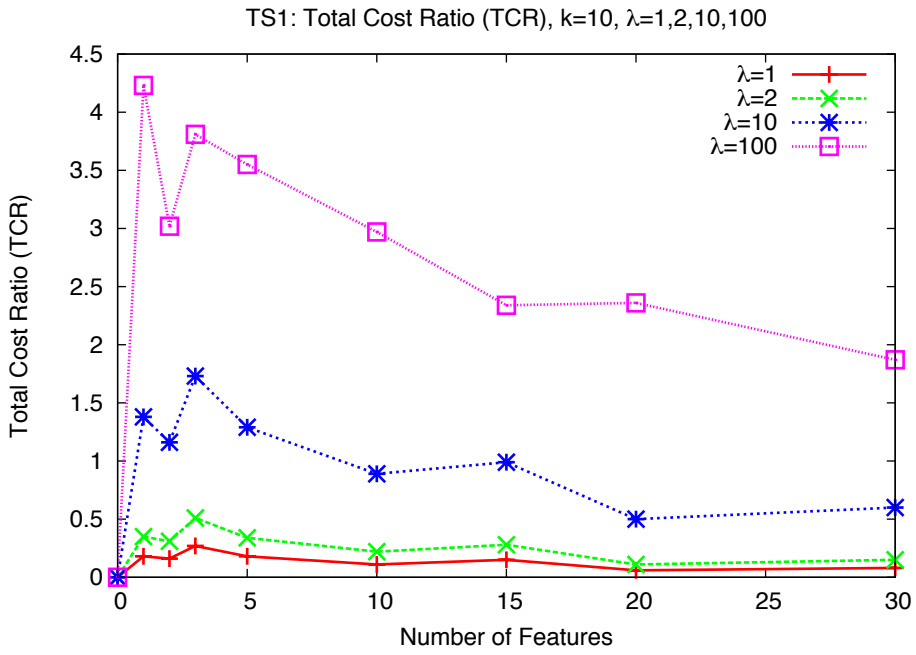
SpamAssassin+ClamAV

ClamAV detected the same TME as SpamAssassin plus 2 additional TME. Therefore, the joint SpamAssassin+ClamAV results are the same as the ClamAV results detailed in the previous section.

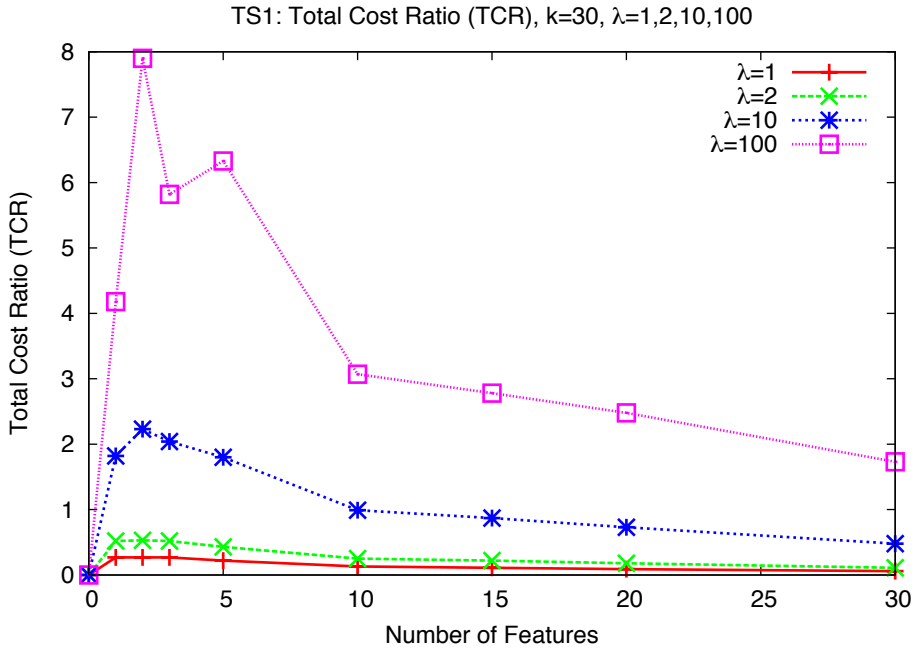
5.3.2 Random forest parameter optimization

Figure 5.5 shows Total Cost Ratio (TCR) graphs for varying number of trees when using increasing subsets of features for node splitting.

A λ=1 means the cost of a false positive and false negative is the same. However, as described with the cost sensitive evaluation in Sections 4.5.3 and 4.6.3, λ can be

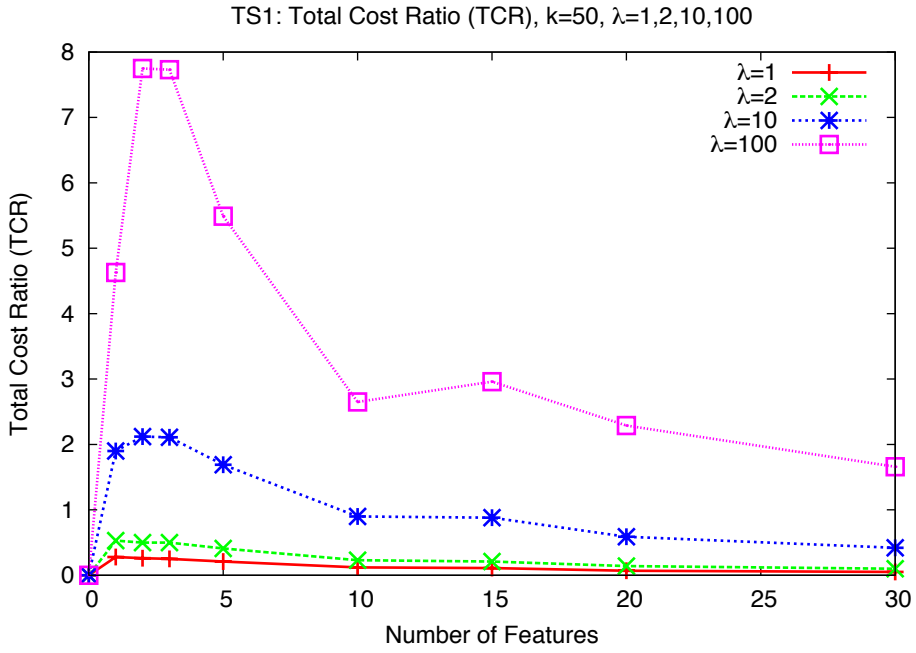


(a) TCR at k=10, λ=1,2,10,100

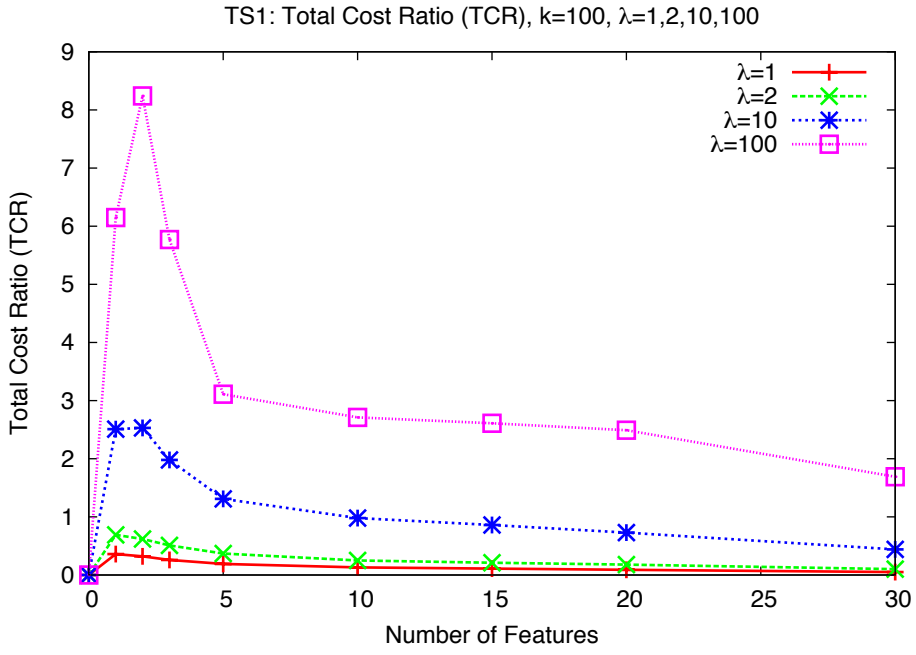


(b) TCR at k=30, λ=1,2,10,100

Figure 5.5: Total cost ratio optimization for random forest using the TS1 data set (continued...)

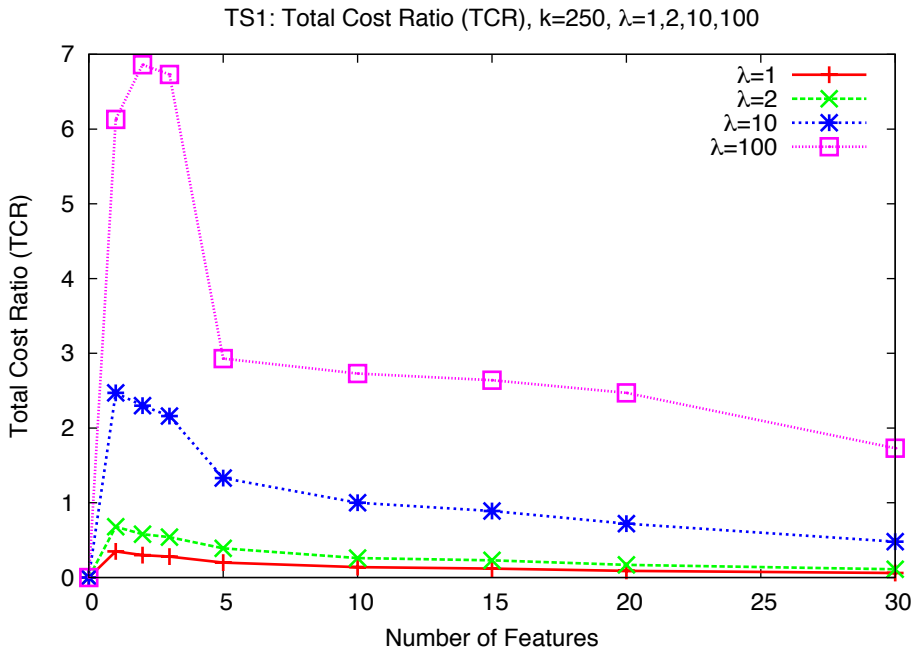


(c) TCR at k=50, λ=1,2,10,100

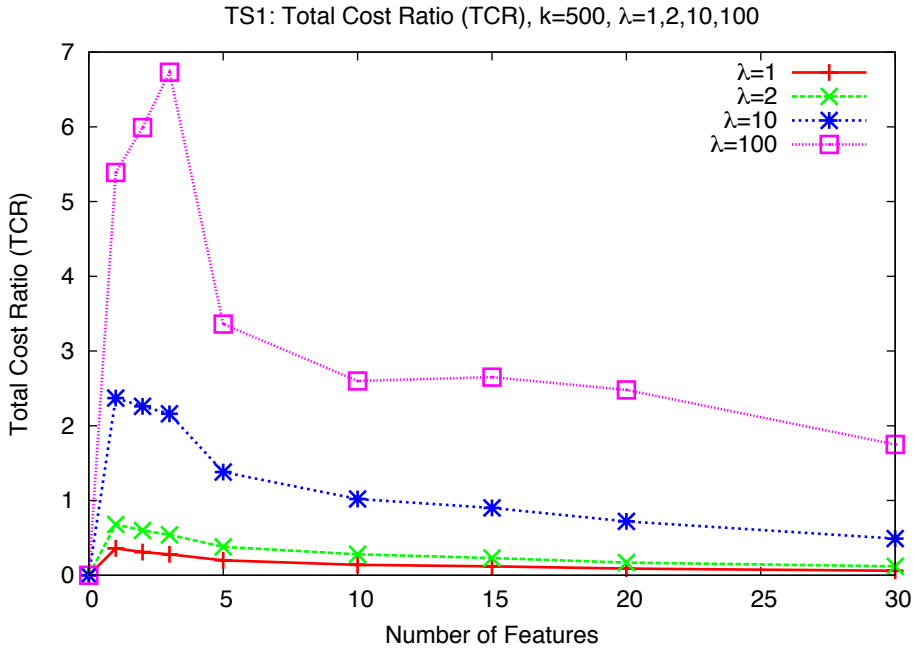


(d) TCR at k=100, λ=1,2,10,100

Figure 5.5: Total cost ratio optimization for random forest using the TS1 data set (continued...)



(e) TCR at k=250, λ=1,2,10,100



(f) TCR at k=500, λ=1,2,10,100

Figure 5.5: Total cost ratio optimization for random forest using the TS1 data set

increased to change the cost ratio between false negatives and false positives. Assuming false negative and false positive misclassification costs are equal, the Random Forest has the highest TCR of 0.36 with $k = 100$ and $m = 1$. Assuming false negatives cost twice as much as false positives, the Random Forest has the highest TCR of 0.69 with $k = 100$ and $m = 1$. Assuming false negatives cost ten or one-hundred times as much as false positives, the Random Forest has the highest TCR of 2.53 and 8.24, respectively, with $k = 100$ and $m = 2$. The false negative rate was 0.09. Unless organizations place a much higher cost on false negatives than false positives then the random forest classifier at $\lambda = 1$ or $\lambda = 2$ does not outperform the baseline (e.g. with no filter present). A random forest with parameters $k = 100$ and $m = 2$ will be used for comparison purposes when analyzing the *TS1* data set. The full data for random forest parameter optimization for the *TS1* data set is available in Appendix D.1.

5.3.3 Cost sensitive learning

Table 5.17 shows the results of processing the *TS1* data set through a random forest classifier using the parameters optimized in the previous section ($k = 100$, $m = 2$). Several executions are done, each time with a different false negative, false positive cost ratio (λ) for the learning process. Evaluation is still done considering the cost difference between false negatives and false positives. The full data for cost sensitive learning for the *TS1* data set is available in Appendix D.2. Summary data is shown in Table 5.17. Making false negatives twice the cost of false positives in the learning process did not

Table 5.17: Summary of cost sensitive learning for the *TS1* data set with $k = 100$, $m = 2$

cost ratio	**TCR,** $\lambda = 1$	**TCR,** $\lambda = 2$	**TCR,** $\lambda = 10$	**TCR,** $\lambda = 100$
$\lambda = 1$	0.32	0.62	2.53	8.24
$\lambda = 2$	0.32	0.62	2.53	8.24
$\lambda = 10$	0.13	0.26	1.18	6.01
$\lambda = 100$	0.00	0.00	0.02	0.22

change the overall total cost ratio (TCR) and no additional false negatives or false positives were generated. Increasing the false negative to false positive cost ratio to $\lambda = 10$ generated more false positives without reducing false negatives. This resulted

in a lower TCR. Finally, with false negatives costing one-hundred times false positives, only 2 false negatives were generated (half of the false negatives of the previous three cost ratios). However, this came at a cost of 19,419 false positives (1.3%). Keeping the cost of false negatives and false positives the same yielded the most acceptable results. While a $\lambda = 100$ only generates 2 false negatives, there are a high number of false positives that could lead to analyst desensitization.

5.3.4 Comparing false negative rates between two detection methods

Section 4.3.3 described how to compare if two detection methods differ significantly in ability to detect targeted malicious email. Table 5.18 shows the contingency table for the random forest based classifer developed in this study against SpamAssassin. This table summarizes the TME detection ability difference between the two methods. Using Equation 4.17 the χ^2 test statistic is 33.03 which is greater than the critical

Table 5.18: *TS1*: Contingency Table for TME detection between Random Forest and SpamAssassin

	RF-Correct	RF-Error
SpamAssassin-Correct	5	0
SpamAssassin-Error	35	4

value of 6.635 at the $\alpha = 0.01$ level of significance. This means the null hypothesis that the two detection methods are the same in the ability to detect targeted malicious email is rejected. Table 5.19 shows the contingency table for Random Forest vs. ClamAV (which is the same as Random Forest vs. SpamAssassin+ClamAV). The χ^2

Table 5.19: *TS1*: Contingency Table for TME detection between Random Forest and ClamAV/SpamAssassin+ClamAV

	RF-Correct	RF-Error
ClamAV-Correct	7	0
ClamAV-Error	33	4

test statistic is 31.03 which is greater than the critical value of 6.635 at the $\alpha = 0.01$

level of significance. This means the null hypothesis that the two detection methods are the same in the ability to detect targeted malicious email is rejected.

In summary, the random forest classifier with persistent threat and recipient oriented features outperformed conventional techniques such as SpamAssassin and ClamAV. The random forest classifier was able to achieve a false negative rate of 9.1% with a negligible 0.009% false positive rate.

Chapter 6: Summary

The purpose of this study was to develop classification methods, using persistent threat and recipient oriented features, designed to detect targeted malicious email (TME). Additionally, the study aimed to demonstrate that incorporating these features results in a detection capability that is superior to conventional email filtering techniques.

The background provided context to the problem of targeted malicious email. First, the nature and caliber of threat actors behind targeted malicious email was characterized. Examples were provided from other research studies and also congressional testimony. Second, numerous examples of targeted malicious email documented in open-source material were reviewed. These examples came from journalists, government security reporting, reports to the United States Congress, and security vendors. This review characterized TME as: capable of evading conventional detection techniques; specifically crafted with a high degree of recipient relevance; low in volume; laced with a malicious attachment or link; and, used for acquisition of sensitive information.

A email primer was then provided containing foundational background on the inner-workings of email. A threat kill chain was then decomposed to demonstrate the importance of persistent threat and recipient oriented features to detecting the tactics, techniques, procedures and infrastructure of individual and institutional threat actors. A comprehensive literature review was conducted to examine the current state of email filtering research. The vast majority of email filtering research today is focused on the detection of spam. The current filtering techniques were categorized into five classes: authentication, contextual, characterization, reputation and resource consumption. The weaknesses of these approaches were also assessed.

Next, the goals and hypotheses of this study were outlined. A summary of the data used in this study was provided. This data included emails as well as other data sources used for feature generation. Next, the statistical techniques used to analyze the data and results were described. These techniques included inference for proportions, inference based on two independent samples and correlation analysis

techniques. All formulas for test statistic and confidence interval calculation were provided. Next, a thorough review of threat specific and recipient oriented features was provided. For each category of features, comparisons were made between targeted malicious email and non-targeted malicious email. For attachments, TME showed a much higher density than NTME. More specifically, .doc, .pdf, and .ppt file types were more prevalent in TME than NTME. TME had significantly more empty *Cc* headers and significantly more *Cc* addresses not addressed to the recipient's company. Base64, Big5 and GB2312 character encodings were more prevalent in TME than NTME. Date header analysis showed a higher proportion of the "+0200", "-0700", "+0800" and "+0900" timezones in TME and a higher proportion of the "-0500", "-0600", and "-0800" timezones in NTME. DomainKeys Identified Mail (DKIM) is still a relatively new Internet standard but showed a higher proportion in NTME than TME. On average, the size of TME was greater than the size of NTME, presumably due to the higher density of attachments in TME which contribute to the increased email size. Analysis of the envelope recipients revealed several interesting trends. First, there was a positive correlation between the number of Google search hits for a particular email address and the average amount of TME received by that email address. Second, comparing the distribution of TME, NTME and spam across the job titles in the company revealed those job titles which receive the most amount of TME. The difference in proportions was shown to be significant with Business Development, Program Management and Communication; these were the job titles which received the highest proportion of TME when compared to NTME. Third, an analysis of TME across business areas revealed four business areas that had a significantly greater average number of recipients than as compared to NTME. Finally, TME was shown to have a greater number of valid envelope recipients, invalid envelope recipients, total envelope recipients and average job level (where a higher job level equates to greater seniority in the company) than NTME. *From* headers showed a greater proportion of gmail, .gov, and yahoo domains in TME than NTME. There was a greater proportion of aol, hotmail and .mil domains in NTME than TME. TME also showed a greater proportion of ".gov", ".mil", and the company's domain name in the *From* header

phrase than NTME. Not surprisingly, NTME had a greater proportion of emails appearing to be from email list servers than TME. Analyzing hyperlinks showed a greater proportion of links to .exe and .htm files in NTME than TME and a greater proportion of links to .zip files in TME than NTME. The *Message-ID* field showed a greater proportion of the string "[t: redacted]" in TME than NTME and the MIME boundary beginning with "2rfk" was more prevalent in TME than NTME. Analyzing received lines also revealed the strings "[s: redacted]" and "[v: redacted]" to be more present in TME than NTME. The *Reply-To* header was more prevalent in NTME than TME but a *Reply-To* address to a Gmail, Hotmail or Yahoo! address was more prevalent in TME than NTME. In TME, the *To* header was proportionally more empty than in NTME. The *X-Forwarded-To* header was more prevalent in NTME than TME. Finally, analysis of the *X-Mailer* header showed a greater proportion of X-Mailer values of "aspnet", "blat", "dreammail", "extreme mail","foxmail", "ghostmail", and "outlook express" in TME than NTME.

Next, a description of the software created and used in this study was provided. The Random Forest algorithm was introduced as the classifier used in the study and a cost sensitive model was developed. Unlike spam where false positives usually have a greater cost than false negatives, with TME, false negatives have a much greater cost than false positives. False positives were modeled as more acceptable given the high impact of TME. Comparisons were drawn between detection-only and blocking scenarios. Next, feature importance measures were shown that can be used to determine the most important features for classification. A set of measures for classifier performance was outlined which formed the basis of the comparison between conventional techniques and the new techniques in this study for detecting TME.

To evaluate the performance of the newly developed persistent threat and recipient aware classifier, first the most important features were enumerated. Calculation of feature importance was done using the mean decrease in Gini index. Next a joint data set of TME and NTME (data set *NTME1-TME1*) was analyzed using conventional email techniques, SpamAssassin and ClamAV. SpamAssassin had a false negative rate of 0.73 and ClamAV had a false negative rate of 0.79. A joint SpamAssassin+ClamAV

detection method yielded a 0.67 false negative rate. Optimizing the random forest parameters yielded optimum performance using a forest with 50 trees selecting 30 features randomly for splitting at each node in the tree. This yielded a false negative rate of 0.006. Implementing cost sensitivity in the learning phase resulted in a reduction of false negatives but also an increase in false positives. Organizations would have to decide how many false positives they are willing to tolerate to increase false negative detection performance. Next, the random forest classifier was executed with successively fewer features, by importance, to determine its false negative degradation characteristics. More than 20 features had to be removed from the random forest classifier before the false negative performance was equal to SpamAssassin+ClamAV. This increased performance was confirmed using a McNemar test that demonstrated that with a $\alpha = 0.01$ level of significance, the difference between a random forest classifier and the conventional email filtering techniques was significant. As another measure to characterize the differences between the techniques developed in this study and conventional email filtering techniques, a separate test set was used to evaluate the classifiers. Data set *TS1* was not presented to the classifier in any form (cross-validation or otherwise) and was only used for testing after a classifier trained on the joint *NTME1-TME1* data set. On the *TS1* data set, SpamAssassin had a false negative rate of 0.89 while ClamAV had a false negative rate of 0.84. Optimizing the random forest classifier yielded the best performance using a forest of 100 trees selecting 2 features randomly for splitting at each node in the tree. This yielded a false negative rate of 0.091. Incorporating cost sensitivity in the learning process did reduce false negatives but generated a high number of false positives that could result in analyst desensitization. Finally, a McNemar test showed that random forest classifier had a superior false negative rate as compared to SpamAssassin, ClamAV or a joint SpamAssassin+ClamAV.

For future research, feature extraction can be extended to further phases of the kill chain such as file attachment metadata. In the weaponization stage of an attack (see Figure 2.3), threat actors may inadvertently leave remnants of information such as file paths on the system used to create an exploit, locale information such as time zone

or even author name. The Adobe Portable Document Format (PDF) has metadata fields for author, date with time zone and even the file path of where the file resides. All of these features might associate multiple targeted malicious emails into a related campaign. From a recipient standpoint, features characterizing the types and amounts of email received by a particular email address can be developed. For example, for each recipient, the number of emails and attachments received over a fixed time period might help uncover emails which fall outside the recipient's normal email receiving patterns. Recipient oriented features can also be extended to include other facets of an individual's behavior: countries visited, conferences attended, or even military status. Finally, for emails with links, features could be developed to indicate whether the domain of the link has ever been visited. Domain creation related information such as age can be extracted and incorporated as features. Aside from extending the features available for use in classification, a multi-class model can be developed. There may be many valid campaigns of attack email each with different characteristics. If different threat actors are behind different campaigns, then features can be mapped to these different threat actors and aligned to a different class outcome for the purposes of classification.

Section 3.2 outlined three hypotheses for this study:

H1 Targeted malicious email demonstrates association to persistent threat features of email such as locale and tools as compared to non-targeted malicious email that does not show an association to persistent threat features.

H2 Targeted malicious email demonstrates association to recipient oriented features such as role, reputation, relationships and access as compared to non-targeted malicious email that does not show an association to recipient oriented features.

H3 Detection of targeted malicious email using persistent threat and recipient oriented features results in fewer false negatives than detection of targeted malicious email using conventional email filtering techniques.

All three of these hypotheses were confirmed in this study.

In summary, this research established targeted malicious email (TME) as a separate class of email that was not previously researched in the academic literature. New detection methods were created based on persistent threat and recipient oriented features. It was shown that persistent threat features are necessary for detecting targeted malicious email. Further, recipient oriented features such as the average number of TME received, the average Google search count and the average job level were in the top twenty features relevant to separating TME from NTME.

Targeted Malicious Email (TME) presents a great risk for those organizations plagued by it. The impact of sensitive data loss can be severe not only to a company but also to a country. The techniques developed in this study can be used to increase the ability of organizations to detect TME over conventional techniques.

References

Abaca Technology. Groundbreaking Technology Redefines Spam Prevention, October 2007. URL http://www.abaca.com/downloads/Abaca%20Groundbreaking%20Technology_WP.pdf.

E. Allman, J. Callas, M. Delaney, M. Libbey, J. Fenton, and M. Thomas. RFC4871 - DomainKeys Identified Mail (DKIM) Signatures, May 2007. URL http://www.ietf.org/rfc/rfc4871.txt.

Ion Androutsopoulos, John Koutsias, Konstantinos V. Chandrinos, and Constantine D. Spyropoulos. An experimental comparison of naive Bayesian and keyword-based anti-spam filtering with personal e-mail messages. In *Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 160–167, Athens, Greece, 2000a. ACM. doi: http://doi.acm.org/10.1145/345508.345569. URL http://www.iit.demokritos.gr/skel/i-config/downloads/.

Ion Androutsopoulos, Georgios Paliouras, Vangelis Karkaletsis, Georgios Sakkis, Constantine D. Spyropoulos, and Panagiotis Stamatopoulos. Learning to filter spam e-mail: A comparison of a naive bayesian and a memory-based approach. In *Proceedings of the Workshop on Machine Learning and Textual Information Access, 4th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD 2000)*, pages 1–13, 2000b.

Adam Back. Hashcash - A Denial of Service Counter-Measure, August 2002. URL http://www.cypherspace.org/adam/Hashcash/Hashcash.pdf.

Julian E. Barnes. Cyber-attack on Defense Department computers raises concerns, November 2008. URL http://www.latimes.com/news/nationworld/nation/la-na-cyberattack28-2008nov28,0,711902,print.story.

Andre Bergholz, Gerhard Paab, Frank Reichartz, Siehyun Strobel, and Jeong-Ho Chang. Improved Phishing Detection using Model-Based Features. In *Conference on Email and Anti-Spam*, 2008.

Robert Beverly and Karen Sollins. Exploiting Transport-Level Characteristics of Spam. In *Conference on Email and Anti-Spam*, 2008. URL http://hdl.handle.net/1721.1/40287.

Manasi Bhattacharyya, Shlomo Hershkop, and Eleazar Eskin. MET: An experimental system for Malicious Email Tracking. In *Proceedings of the 2002 workshop on New security paradigms*, pages 3–10. ACM, 2002. doi: http://doi.acm.org/10.1145/844102.844104.

P.O. Boykin and V. Roychowdhury. Personal Email Networks: An Effective Anti-Spam Tool, February 2004. URL http://www.arxiv.org/abs/cond-mat/0402143.

Leo Breiman. Random Forests. *Machine Learning*, 45(1):5–32, October 2001.

Leo Breiman, Jerome H. Friedman, Richard A. Olshen, and Charles J. Stone. *Classification and Regression Trees*. Wadsworth, Belmont, California, 1984.

Pedro H. Calais, Douglas E. V. Pires, Dorgival Olavo Guedes, Wagner Meira Jr., Cristine Hoepers, and Klaus Steding-Jessen. A Campaign-based Characterization of Spamming Strategies. In *In proceedings of CEAS 2008: Conference on Email and Anti-Spam*, 2008. URL http://www.ceas.cc/2008/papers/ceas2008-paper-45.pdf.

Madhusudhanan Chandrasekaran, Ramkumar Chinchani, and Shambhu Upadhyaya. PHONEY: Mimicking User Response to Detect Phishing Attacks. In *WOWMOM '06: Proceedings of the 2006 International Symposium on on World of Wireless, Mobile and Multimedia Networks*, pages 668–672. IEEE Computer Society, 2006. doi: http://dx.doi.org/10.1109/WOWMOM.2006.87.

Bin Chen, Shoubin Dong, and Weidong Fang. Introduction of Fingerprint Vector based Bayesian Method for Spam Filtering. In *Conference on Email and Anti-Spam*, August 2007.

Chao Chen, Andy Liaw, and Leo Breiman. Using Random Forest to Learn Imbalanced Data, July 2004. URL http://stat-www.berkeley.edu/tech-reports/666.pdf.

Paul Alexandru Chirita, J Org Diederich, Wolfgang Nejdl, Deutscher Pavillon, Deutscher Pavillon, and Deutscher Pavillon. Mailrank: Using ranking for spam detection. In *In Proc. of the 14th Intl. CIKM Conf. on Information and Knowledge Management*, pages 373–380. ACM, 2005.

Thomas Claburn. Pro-Tibet Groups Targeted in Cyberspace. InformationWeek, March 2008. URL http://www.informationweek.com/story/showArticle.jhtml?articleID=206905235.

James Clark, Irena Koprinska, and Josiah Poon. A neural network based approach to automated e-mail classification. In *Proceedings of IEEE/WIC International Conference on Web Intelligence, 2003*, pages 702–705, October 2003.

D. Crocker, J. Leslie, and D. Otis. Certified Server Validation (CSV), February 2005. URL http://tools.ietf.org/html/draft-ietf-marid-csv-intro-02.

Ernesto Damiani, Sabrina De, Capitani Vimercati, Stefano Paraboschi, and Pierangela Samarati. P2P-based collaborative spam detection and filtering. In *In 4th IEEE Conference on P2P*, pages 176–183, 2004. URL http://femto.org/p2p2004/papers/damiani.pdf.

M. Delaney. RFC4870 - Domain-Based Email Authentication Using Public Keys Advertised in the DNS (DomainKeys), May 2007. URL http://www.ietf.org/rfc/rfc4870.txt.

Sarah Jane Delany and Derek Bridge. Feature-based and feature-free textual cbr: A

comparison in spam filtering. In *Proceedings of the 17th Irish Conference on Artificial Intelligence and Cognitive Science (AICS06)*, pages 244–253, 2006.

Sarah Jane Delany, Padraig Cunningham, and Lorcan Coyle. An Assessment of Case-Based Reasoning for Spam Filtering. *Artificial Intelligence Review*, 24(3-4): 359–378, 2005. URL http://portal.acm.org/citation.cfm?id=1107375.

Jay L. Devore. *Probability and Statistics*. Thomson Brooks/Cole, 6th edition edition, 2004.

Thomas G. Dietterich. Approximate Statistical Tests for Comparing Supervised Classification Learning Algorithms. *Neural Computation*, 10:1895–1923, 1998.

Pedro Domingos. MetaCost: a general method for making classifiers cost-sensitive. In *Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 155–164, New York, NY, USA, 1999. ACM.

Harris Drucker, Donghui Wu, and V.N. Vapnik. Support vector machines for spam categorization. *IEEE Transactions on Neural Networks*, 10(5):1048–1054, September 1999.

Zhenhai Duan, Yingfei Dong, and Kartik Gopalan. DiffMail: A Differentiated Message Delivery Architecture to Control Spam, October 2004. URL http://www.cs.fsu.edu/~duan/publications/diffmail_tr.pdf.

Richard O. Duda, Peter E. Hart, and David G. Stork. *Pattern Classification (2nd Edition)*. Wiley-Interscience, 2000. URL http://www.wiley.com/WileyCDA/WileyTitle/productCd-0471056693.html.

Cynthia Dwork, Andrew Goldberg, and Moni Naor. On Memory-Bound Functions for Fighting Spam. In *Proceedings of the 23rd Annual International Cryptology Conference (CRYPTO 2003)*, pages 426–444, Santa Barbara, CA, 2003. URL http://research.microsoft.com/apps/pubs/default.aspx?id=65154.

Keith Epstein and Ben Elgin. Network Security Breaches Plague NASA, November 2008. URL http://www.businessweek.com/print/magazine/content/08_48/b4110072404167.htm.

David Erickson, Martin Casado, and Nick McKeown. The Effectiveness of Whitelisting: a User-Study. In *Conference on Email and Anti-Spam*, 2008.

B. Everitt. *The Analysis of Contingency Tables*. Chapman and Hall, 1977.

F-Secure. Targeted Malware Attacks Against Pro-Tibet Groups, March 2008. URL http://www.f-secure.com/weblog/archives/00001406.html.

Chris Fleizach, Geoffrey M. Voelker, and Stefan Savage. Slicing Spam with Occams Razor. In *In proceedings of CEAS 2007 - Fourth conference on email and anti-spam*, August 2007.

Jason Fritz. How China Will Use Cyber Warfare to Leapfrog in Military Competitiveness. *Culture Mandala*, 8(1):28–80, October 2008.

Scott Garriss, Michael Kaminsky, Michael J. Freedman, Brad Karp, David Mazires, and Haifeng Yu. RE: Reliable Email. In *Proceedings of NSDI '06*, pages 297–310, 2006.

Lawrence K. Gershwin. Statement for the Record for the Joint Economic Committee: Cyber Threat Trends and US Network Security, June 2001. URL http://www.house.gov/jec/hearings/gershwin.pdf.

Jennifer Golbeck and James Hendler. Reputation Network Analysis for Email Filtering. In *In Proc. of the Conference on Email and Anti-Spam (CEAS), Mountain View*, 2004.

Luiz Henrique Gomes, Cristiano Cazita, Jussara M. Almeida, Virgilio Almeida, and Jr. Wagner Meira. Characterizing a spam traffic. In *IMC '04: Proceedings of the 4th ACM SIGCOMM conference on Internet measurement*, pages 356–369, New York, NY, USA, 2004. ACM. doi: http://doi.acm.org/10.1145/1028788.1028837.

Paul Graham. A Plan for Spam, 2002. URL http://www.paulgraham.com/spam.html.

Brian Grow, Keith Epstein, and Chi-Chu Tschang. The New E-spionage Threat. *BusinessWeek*, April 2008. URL http://www.businessweek.com/print/magazine/content/08_16/b4080032218430.htm.

Mark Hall, Eibe Frank, Geoffrey Holmes, Bernard Pfahringer, Peter Reutemann, and Ian H. Witten. The WEKA Data Mining Software: An Update. *SIGKDD Explorations*, 11, 2009.

Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Satistical Learning: Data Mining, Inference, and Prediction*. Springer, New York, NY, 2nd edition edition, 2008.

Haibo He and Edwardo A. Garcia. Learning from Imbalanced Data. *IEEE Transactions on Knowledge and Data Engineering*, 21(9):1263–1284, September 2009.

Eric M. Hutchins, Michael J. Cloppert, and Rohan M. Amin. Intelligence-Driven Computer Network Defense Informed by Analysis of Adversary Campaigns and Intrusion Kill Chains. 2010.

iDefense. Targeted Attacks and Their Impact, November 2005. URL http://complianceandprivacy.com/WhitePapers/iDefense_Targeted_Attacks_110405.pdf. Accessed on November 15, 2008.

John Ioannidis. Fighting Spam by Encapsulating Policy in Email Addresses. In *Proceedings of the network and distributed system security symposium, NDSS*, 2003.

Markus Jakobsson. Modeling and Preventing Phishing Attacks, February 2005. URL http://www.informatics.indiana.edu/markus/papers/phishing_jakobsson.pdf.

Jaeyeon Jung and Emil Sit. An empirical study of spam traffic and the use of DNS black lists. In *IMC '04: Proceedings of the 4th ACM SIGCOMM conference on Internet measurement*, pages 370–375, New York, NY, USA, 2004. ACM. doi: http://doi.acm.org/10.1145/1028788.1028838.

Taghi M. Khoshgoftaar, Moiz Golawala, and Jason Van Hulse. An Empirical Study of Learning from Imbalanced Data Using Random Forest. In *19th IEEE International Conference on Tools with Artificial Intelligence*, 2007. doi: DOI10.1109/ICTAI.2007.46.

Oleg Kolesnikov, Wenke Lee, and Richard Lipton. Filtering Spam Using Search Engines, 2003. URL citeseer.ist.psu.edu/kolesnikov03filtering.html.

Irena Koprinska, Josian Poon, James Clark, and Jason Chan. Learning to classify e-mail. *Information Sciences*, 177(10):2167–22187, 2007. doi: http://dx.doi.org/10.1016/j.ins.2006.12.005.

Bryan Krekel. Capability of the People's Republic of China to Conduct Cyber Warfare and Computer Network Exploitation, October 2009. URL http://www.uscc.gov/researchpapers/2009/NorthropGrumman_PRC_Cyber_Paper_FINAL_Approved%20Report_16Oct2009.pdf.

Paul B. Kurtz. Testification before House Permanent Select Committee on Intelligence, September 2008. URL http://intelligence.house.gov/Media/PDFS/Kurtz091808.pdf.

HoYu Lam and DitYan Yeung. A Learning Approach to Spam Detection based on Social Networks. In *In proceedings of CEAS 2007 - Fourth conference on email and anti-spam*, 2007.

Wenke Lee, Wei Fan, Salvatore J. Stolfo, and Matthew Miller. *Machine Learning and Data Mining for Computer Security*, chapter 8, pages 125–136. Springer London, 2006. doi: 10.1007/1-84628-253-5_8.

Barry Leiba and Jim Fenton. DomainKeys Identified Mail (DKIM): Using Digital Signatures for Domain Verification. In *In Proceedings of CEAS 2007: The Third Conference on Email and Anti-Spam*, 2007.

J. Leslie, D. Crocker, and D. Otis. Domain Name Accreditation (DNA), February 2005. URL http://tools.ietf.org/html/draft-ietf-marid-csv-dna-02.

James Andrew Lewis. Holistic Approaches to Cybersecurity to Enable Network Centric Operations, April 2008. URL http://armedservices.house.gov/pdfs/TUTC040108/Lewis_Testimony040108.pdf.

Kang Li, Calton Pu, and Mustaque Ahamad. Resisting Spam Delivery by TCP Damping. In *In Proceedings of FIrst Conference on Email and Anti-Spam (CEAS), July 2004*, July 2004.

Wenyin Liu, Xiaotie Deng, Guanglin Huang, and Anthony Y. Fu. An Antiphishing Strategy Based on Visual Similarity Assessment. *IEEE Internet Computing*, pages 58–65, March 2006.

J. Lyon and M. Wong. RFC4406 - Sender ID: Authenticating E-Mail, April 2006. URL http://www.ietf.org/rfc/rfc4406.txt.

MAAWG. Messaging Anti-Abuse Working Group (MAAWG) Email Metrics Program: Report 9 - Second Quarter 2008, October 2008. URL http://www.maawg.org/about/MAAWG_2008-Q2_Metrics_Report9.pdf.

Dragos D. Margineantu, 2000. URL http://www.aaai.org/Papers/Workshops/2000/WS-00-05/WS00-05-010.pdf.

M.N. Marsono, M.W. El-Kharashi, and F. Gebali. Rejecting Spam during SMTP Sessions. *Communications, Computers and Signal Processing, 2007. PacRim 2007. IEEE Pacific Rim Conference on*, pages 236–239, August 2007. doi: 10.1109/PACRIM.2007.4313219.

Kate McCarthy, Bibi Zabar, and Gary Weiss. Does Cost-Sensitive Learning Beat Sampling for Classifying Rare Classes?, 2005. URL http://storm.cis.fordham.edu/~gweiss/papers/ubdm05-mccarthy.pdf.

Niamh McCombe. Methods of Author Identification, May 2002. URL https://www.cs.tcd.ie/courses/csll/mccombe0102.pdf.

J. Michael McConnell. Annual Threat Assessment of the Director of National Intelligence for the Senate Select Committee on Intelligence, February 2008. URL http://www.au.af.mil/au/awc/awcgate/dni/threat_assessment_5feb08.pdf.

MessageLabs. MessageLabs Intelligence: November 2007, November 2007a. URL http://www.messagelabs.com/mlireport/MLI_Report_November_2007.pdf.

MessageLabs. MessageLabs Intelligence: June 2007, June 2007b. URL http://www.messagelabs.com/mlireport/MessageLabs%20Intelligence%20-%20Jun%20Q2%20Report%20-%20FINAL.pdf.

MessageLabs. MessageLabs Intelligence Special Report: Targeted Attacks March 2007, March 2007c. URL http://www.messagelabs.com/mlireport/messagelabs_intelligence_special_report__targeted_attacks_march_2007_5.pdf.

MessageLabs. MessageLabs Intelligence Special Report on Trojans Targeted Olympic Organizations, July 2008a. URL http://www.messagelabs.com/mlireport/MLISpecialReport_2008_08_OlympicTargeted_Final.pdf.

MessageLabs. MessageLabs Intelligence: 2008 Annual Security Report, December 2008b. URL http://www.messagelabs.com/mlireport/MLIReport_Annual_2008_FINAL.pdf.

MessageLabs. MessageLabs Intelligence: 2009 Annual Security Report, December 2009. URL http://www.messagelabs.com/mlireport/2009MLIAnnualReport_Final_PrintResolution.pdf.

Microsoft. MS02-058: OLEXP: An Unchecked Buffer in Outlook Express S/MIME Parsing May Permit System Compromise, October 2002. URL http://support.microsoft.com/kb/328676.

Mozilla. Mozilla Foundation Security Advisory 2008-12: Heap buffer overflow in external MIME bodies, February 2008. URL http://www.mozilla.org/security/announce/2008/mfsa2008-12.html.

Cormac O'Brien and Carl Vogel. Spam filters: Bayes vs. chi-squared; letters vs. words. In *In ISICT 03: Proceedings of the 1st international symposium on Information and communication technologies*, pages 291–296. Words, 2003.

D. Otis, D. Crocker, and J. Leslie. Client SMTP Authorization (CSA), February 2005. URL http://tools.ietf.org/html/draft-ietf-marid-csv-csa-02.

Patrick Pantel and Dekang Lin. SpamCop: A Spam Classification & Organization Program. In *Proceedings of the AAAI-98 Workshop on Learning for Text Categorization*, Madison, Wisconsin, 1998.

R Development Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2009. URL http://www.R-project.org. ISBN 3-900051-07-0.

Moheeb Abu Rajab, Jay Zarfoss, Fabian Monrose, and Andreas Terzis. A multifaceted Approach to Understanding the Botnet Phenomenon. In *IMC '06: Proceedings of the 6th ACM SIGCOMM conference on Internet measurement*, pages 41–52, New York, NY, USA, 2006. ACM Press. doi: http://doi.acm.org/10.1145/1177080.1177086.

Anirudh Ramachandran, Nick Feamster, and Santosh Vempala. Filtering Spam with Behavioral Blacklisting. In *ACM Conference on Computer and Communications Security*, pages 342–351, 2007. doi: http://doi.acm.org/10.1145/1315245.1315288.

P. Resnick. RFC 5322 - Internet Message Format, October 2008. URL http://tools.ietf.org/html/rfc5322.

Ismael Rivera, Myriam Mencke, Juan Miguel Gomez, Giner Alor-Hernandez, and Angel Garcia-Crespo. Collaborative OpenSocial Network Dataset based Email Ranking and Filtering. In *Third International Conference on Systems*, 2008.

Mehran Sahami, Susan Dumais, David Heckerman, and Eric Horvitz. A Bayesian approach to filtering junk email. In *Learning for Text Categorization: Papers from the 1998 Workshop*, Madison, Wisconsin, 1998.

G. Sakkis, I. Androutsopoulos, G. Paliouras, V. Karkaletsis, C. Spyropoulos, and P. Stamatopoulos. A memory-based approach to anti-spam filtering. In *Tech Report DEMO 2001, National Centre for Scientific Research "Demokritos"*, 2001.

Georgios Sakkis, Ion Androutsopoulos, Georgios Paliouras, Vangelis Karkaletsis, Constantine D. Spyropoulos, and Panagiotis Stamatopoulos. A Memory-Based Approach to Anti-Spam Filtering for Mailing Lists. *Information Retrieval*, 6(1): 49–73, January 2003. doi: 10.1023/A:1022948414856.

Steven L. Salzberg. On Comparing Classifiers: Pitfalls to Avoid and a Recommended Approach. *Data Mining and Knowledge Discovery*, 1(3):317–328, September 1997.

Karl-Michael Schneider. A comparison of event models for nave bayes anti-spam e-mail ltering. In *In Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics*, 2003.

Gregg Schudel and Bradley Wood. Modeling Behavior of the Cyber-Terrorist. In *IEEE Symposium on Security Privacy*, 2008.

Jean-Marc Seigneur, Nathan Dimmock, Ciaran Bryce, and Christian Damsgaard Jensen. Combating Spam with TEA (Trustworthy Email Addresses). In *In proceedings of the 2nd Conference on Privacy, Security and Trust*, Canada, 2004.

Burim Sirisanyalak and Ohm Sornil. Artificial Immunity-Based Feature Extraction for Spam Detection. In *Eighth ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing*, pages 359–364, 2007.

Charles Smutz, Sam Wenck, and Michael Cloppert. Vortex, 2010. URL http://sourceforge.net/projects/vortex-ids/.

Alex Stamos. "Aurora" Response Recommendations, February 2010. URL https://www.isecpartners.com/files/iSEC_Aurora_Response_Recommendations.pdf.

Allen Brian Stone. EBIDS-SENLP: A System to Detect Social Engineering Email Using Natural Language Processing. Master's thesis, University of Maryland, 2007.

Bradley Taylor. Sender Reputation in a Large Webmail Service. In *In proceedings of CEAS - Conference on Email and Anti-Spam 2006*, 2006. URL http://www.ceas.cc/2006/19.pdf.

Minh Tran and Grenville Armitage. Evaluting The Use of Spam-triggered TCP Rate Control To Protect SMTP Servers. In *In proceedings of Australian Telecommunications Networks and Applications Conference 2004*, Sydney, Australia, December 2004.

Peter Turney. Types of cost in inductive concept learning. In *Workshop on Cost-Sensitive Learning at the Seventeenth International Conference on Machine Learning (WCSL at ICML-2000)*, Stanford University, California, 2000.

Grigorios Tzortzis and Aristidis Likas. Deep Belief Networks for Spam Filtering. In *19th IEEE International Conference on Tools with Artificial Intelligence*, 2007.

UK-NISCC. National Infrastructure Security Co-ordination Centre: Targeted Trojan Email Attacks, June 2005. URL https://www.cpni.gov.uk/docs/ttea.pdf.

US-CERT. Technical Cyber Security Alert TA05-189A: Targeted Trojan Email Attacks, July 2005. URL http://www.us-cert.gov/cas/techalerts/TA05-189A.html.

US-CERT. Vulnerability Note VU 191609 - Microsoft Windows animated cursor stack buffer overflow, August 2007. URL http://www.kb.cert.org/vuls/id/191609.

US-CERT. Spear Phishing Campaign Directed at USG Employees, March 2008. URL http://www.businessweek.com/pdfs/2008/0816_spearphishing.pdf.

U.S.-China Economic and Security Review Commission. 2008 Report to Congress of the U.S.-China Economic and Security Review Commission, November 2008. URL http://www.uscc.gov/annual_report/2008/annual_report_full_08.pdf.

U.S.-China Economic and Security Review Commission. 2009 Report to Congress of the U.S.-China Economic and Security Review Commission, November 2009. URL http://www.uscc.gov/annual_report/2009/annual_report_full_09.pdf.

U.S. Department of Defense. Annual Report to Congress: Military Power of the People's Republic of China 2008, 2008. URL http://www.defenselink.mil/pubs/pdfs/China_Military_Report_08.pdf.

U.S. Department of Defense. Annual Report to Congress: Military Power of the People's Republic of China 2009, 2009. URL http://www.defenselink.mil/pubs/pdfs/China_Military_Power_Report_2009.pdf.

U.S. Government Accountability Office. GAO-05-434: Critical Infrastructure Protection: Department of Homeland Security Faces Challenges in Fulfilling Cybersecurity Responsibilities, May 2005. URL http://www.gao.gov/products/GAO-05-434.

Nart Villeneuve and Greg Walton. Targeted Malware Attack on Foreign Correspondents based in China, September 2009. URL http://www.infowar-monitor.net/2009/09/targeted-malware-attack-on-foreign-correspondents-based-in-china/.

Shaun Waterman. Chinese Cyberattacks Target US Think Tanks, March 2008. URL http://www.spacewar.com/reports/Chinese_Cyberattacks_Target_US_Think_Tanks_999.html. Accessed November 23, 2008.

Meng Weng Wong and W. Schlitt. RFC4408 - Sender Policy Framework (SPF) for

Authorizing User of Domains in E-Mail, Version 1, 2006. URL http://www.ietf.org/rfc/rfc4408.txt.

Kenichi Yoshida, Fuminori Adachi, Takashi Washio, Hiroshi Motoda, Teruaki Homma, Akihiro Nakashima, Hiromitsu Fujikawa, and Katsuyuki Yamazaki. Density-based spam detector. In *KDD '04: Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 486–493, New York, NY, USA, 2004. ACM. doi: http://doi.acm.org.proxygw.wrlc.org/10.1145/1014052.1014107.

Le Zhang, Jingbo Zhu, and Tianshun Yao. An Evaluation of Statistical Spam Filtering Techniques. *ACM Transactions on Asian Language Information Processing (TALIP)*, 3:243–269, December 2004. doi: 10.1145/1039621.1039625.

Yan Zhou, M. S. Mulekar, and P. Nerellapalli. Adaptive spam filtering using dynamic feature space. In *Proc. 17th IEEE International Conference on Tools with Artificial Intelligence ICTAI 05*, 14–16 Nov. 2005. doi: 10.1109/ICTAI.2005.28.

Appendix A: Google Search Hits

Table A.1 is a breakdown of the number of targeted malicious emails received by email addresses in the company with a certain number of Google search hits.

Table A.1: Detailed List of Extracted Email Features

Google hits	Num Email Addresses	Cum %	Avg TME Received
0	132604	95.290%	0.0074
1	4279	98.365%	0.1687
2	1299	99.299%	0.5050
3	448	99.621%	0.6228
4	157	99.733%	0.7261
5	37	99.760%	0.4595
6	37	99.787%	0.7027
7	33	99.810%	1.0606
8	40	99.839%	0.6500
9	30	99.861%	0.8333
10	11	99.868%	1.5455
11	15	99.879%	0.7333
12	10	99.886%	1.2000
13	7	99.891%	1.1429
14	10	99.899%	1.9000
15	10	99.906%	0.1000
16	6	99.910%	0.8333
17	3	99.912%	1.6667
18	7	99.917%	2.0000
19	9	99.924%	1.0000
20	6	99.928%	0.8333
21	3	99.930%	2.3330
2	2	99.932%	0.5000
23	3	99.934%	2.6667

Continued on next page...

Table A.1 – Continued

Google hits	Num Email Addresses	Cum %	Avg TME Received
24	4	99.937%	1.5000
25	5	99.940%	1.0000
26	4	99.943%	1.5000
27	2	99.945%	0.0000
28	2	99.946%	0.0000
29	4	99.949%	0.5000
30	3	99.951%	0.3000
31	1	99.952%	0.0000
32	3	99.954%	2.0000
33	3	99.956%	0.6667
34	1	99.957%	0.0000
36	3	99.959%	0.3330
37	2	99.960%	0.5000
38	3	99.963%	2.6667
39	1	99.963%	0.0000
40	1	99.964%	1.0000
43	1	99.965%	0.0000
44	1	99.966%	0.0000
46	2	99.967%	1.5000
47	2	99.968%	3.0000
48	4	99.971%	2.0000
49	1	99.972%	2.0000
51	1	99.973%	2.0000
53	1	99.973%	2.0000
55	3	99.976%	0.6667
57	1	99.976%	1.0000
58	1	99.977%	4.0000
61	2	99.978%	2.5000
64	1	99.979%	2.0000

Continued on next page...

Table A.1 – Continued

Google hits	Num Email Addresses	Cum %	Avg TME Received
66	1	99.980%	2.0000
69	2	99.981%	1.5000
70	1	99.982%	1.0000
80	1	99.983%	0.0000
81	1	99.983%	0.0000
84	1	99.984%	7.0000
85	1	99.985%	0.0000
93	2	99.986%	2.0000
95	1	99.987%	2.0000
104	1	99.988%	1.0000
108	1	99.989%	2.0000
113	1	99.989%	5.0000
126	1	99.990%	2.0000
134	1	99.991%	0.0000
135	1	99.991%	0.0000
141	1	99.992%	4.0000
145	1	99.993%	1.0000
149	1	99.994%	1.0000
158	1	99.994%	5.0000
179	1	99.995%	1.0000
190	1	99.996%	0.0000
194	1	99.996%	4.0000
205	1	99.997%	0.0000
250	1	99.998%	5.0000
311	1	99.999%	1.0000
510	1	99.999%	2.0000
672	1	100.000%	1.0000

Appendix B: Random Forest Details

The Random Forest algorithm (Breiman, 2001; Hastie et al., 2008) is as follows:

1. Parameters: k = number of trees to create; m = number of random features to select for node splitting, d = maximum depth of the trees (in this study, trees are grown to maximum size).

2. Select k vectors from the training data such that vector θ_k is chosen independent of $\theta_1, ..., \theta_{k-1}$. This is known as bootstrap sampling.

3. For each of the bootstrap samples grow a tree, T_k, where each node is split using the best split from m randomly selected features. The result is multiple tree classifiers $T_k : h(\mathbf{x}, \theta_k)$ where $\mathbf{x}$ is an input vector of unknown classification.

4. To classify $\mathbf{x}$, process that feature vector down each tree in the forest. Each tree will output a classification, also known as a vote. If $C_k(\mathbf{x})$ represents the classification of the kth tree in the forest, then the aggregate classification of the forest, $C_{forest}(\mathbf{x}) = majority\ vote\ \{C_k(\mathbf{x})\}_1^k$.

Appendix C: Evaluation Data for the *NTME1-TME1* data set

C.1 Random forest parameter optimization for the *NTME1-TME1* data set

Table C.1 shows detailed results for executing the random forest classifier against the *NTME1-TME1* data set. The best performing random forest configurations for various false negative, false positive cost ratios (λ) are highlighted.

Table C.1: Random forest parameter optimization for the *NTME1-TME1* data set

k	m	TP	TN	FP	FN	TPR	TNR	FPR	FNR	Wacc	Werr	TCR, $\lambda = 1$	TCR, $\lambda = 2$	TCR, $\lambda = 10$	TCR, $\lambda = 100$
10	7	2280	20884	10	35	0.985	1.000	0.000	0.015	0.998	0.002	51.44	57.88	64.31	65.95
10	15	2287	20884	10	28	0.988	1.000	0.000	0.012	0.998	0.002	60.92	70.15	79.83	82.38
10	20	2291	20887	7	24	0.990	1.000	0.000	0.010	0.999	0.001	74.68	84.18	93.72	96.18
10	30	2288	20879	15	27	0.988	0.999	0.001	0.012	0.998	0.002	55.12	67.10	81.23	85.27
10	40	2295	20877	17	20	0.991	0.999	0.001	0.009	0.998	0.002	62.57	81.23	106.68	114.77
10	50	2291	20871	23	24	0.990	0.999	0.001	0.010	0.998	0.002	49.26	65.21	88.02	95.54
10	60	2293	20865	29	22	0.990	0.999	0.001	0.010	0.998	0.002	45.39	63.42	92.97	103.86
10	70	2296	20861	33	19	0.992	0.998	0.002	0.008	0.998	0.002	44.52	65.21	103.81	119.76
10	80	2293	20861	33	22	0.990	0.998	0.002	0.010	0.998	0.002	42.09	60.13	91.50	103.67
30	7	2292	20887	7	23	0.990	1.000	0.000	0.010	0.999	0.001	77.17	87.36	97.68	100.35
30	15	2294	20886	8	21	0.991	1.000	0.000	0.009	0.999	0.001	79.83	92.60	106.19	109.82
30	20	2298	20882	12	17	0.993	0.999	0.001	0.007	0.999	0.001	79.83	100.65	127.20	135.22
30	30	2295	20877	17	20	0.991	0.999	0.001	0.009	0.998	0.002	62.57	81.23	106.68	114.77
30	40	2297	20876	18	18	0.992	0.999	0.001	0.008	0.998	0.002	64.31	85.74	116.92	127.34
30	50	2294	20868	26	21	0.991	0.999	0.001	0.009	0.998	0.002	49.26	68.09	98.09	108.89
30	60	2294	20867	27	21	0.991	0.999	0.001	0.009	0.998	0.002	48.23	67.10	97.68	108.84
30	70	2295	20863	31	20	0.991	0.999	0.001	0.009	0.998	0.002	45.39	65.21	100.22	113.98
30	80	2295	20862	32	20	0.991	0.998	0.002	0.009	0.998	0.002	44.52	64.31	99.78	113.93

Continued on next page...

Table C.1 – Continued

k	m	TP	TN	FP	FN	TPR	TNR	FPR	FNR	Wacc	Werr	TCR, $\lambda = 1$	TCR, $\lambda = 2$	TCR, $\lambda = 10$	TCR, $\lambda = 100$
50	7	2293	20889	5	22	0.990	1.000	0.000	0.010	0.999	0.001	85.74	94.49	102.89	104.99
50	15	2294	20885	9	21	0.991	1.000	0.000	0.009	0.999	0.001	77.17	90.78	105.71	109.77
50	20	2298	20884	10	17	0.993	1.000	0.000	0.007	0.999	0.001	85.74	105.23	128.61	135.38
50	30	2300	20880	14	15	0.994	0.999	0.001	0.006	0.999	0.001	79.83	105.23	141.16	152.91
50	40	2296	20877	17	19	0.992	0.999	0.001	0.008	0.998	0.002	64.31	84.18	111.84	120.76
50	50	2296	20872	22	19	0.992	0.999	0.001	0.008	0.998	0.002	56.46	77.17	109.20	120.45
50	60	2296	20867	27	19	0.992	0.999	0.001	0.008	0.998	0.002	50.33	71.23	106.68	120.13
50	70	2294	20862	32	21	0.991	0.998	0.002	0.009	0.998	0.002	43.68	62.57	95.66	108.58
50	80	2297	20862	32	18	0.992	0.998	0.002	0.008	0.998	0.002	46.30	68.09	109.20	126.36
100	7	2289	20887	7	26	0.989	1.000	0.000	0.011	0.999	0.001	70.15	78.47	86.70	88.80
100	15	2295	20886	8	20	0.991	1.000	0.000	0.009	0.999	0.001	82.68	96.46	111.30	115.29
100	20	2297	20883	11	18	0.992	0.999	0.001	0.008	0.999	0.001	79.83	98.51	121.20	127.83
100	30	2297	20878	16	18	0.992	0.999	0.001	0.008	0.999	0.001	68.09	89.04	118.11	127.48
100	40	2297	20880	14	18	0.992	0.999	0.001	0.008	0.999	0.001	72.34	92.60	119.33	127.62
100	50	2296	20873	21	19	0.992	0.999	0.001	0.008	0.998	0.002	57.88	78.47	109.72	120.51
100	60	2297	20866	28	18	0.992	0.999	0.001	0.008	0.998	0.002	50.33	72.34	111.30	126.64
100	70	2295	20864	30	20	0.991	0.999	0.001	0.009	0.998	0.002	46.30	66.14	100.65	114.04
100	80	2297	20862	32	18	0.992	0.998	0.002	0.008	0.998	0.002	46.30	68.09	109.20	126.36
250	7	2290	20888	6	25	0.989	1.000	0.000	0.011	0.999	0.001	74.68	82.68	90.43	92.38
250	15	2293	20889	5	22	0.990	1.000	0.000	0.010	0.999	0.001	85.74	94.49	102.89	104.99
250	20	2296	20885	9	19	0.992	1.000	0.000	0.008	0.999	0.001	82.68	98.51	116.33	121.27
250	30	2295	20881	13	20	0.991	0.999	0.001	0.009	0.999	0.001	70.15	87.36	108.69	115.00
250	40	2297	20877	17	18	0.992	0.999	0.001	0.008	0.998	0.002	66.14	87.36	117.51	127.41
250	50	2297	20871	23	18	0.992	0.999	0.001	0.008	0.998	0.002	56.46	78.47	114.04	126.99
250	60	2298	20864	20	17	0.993	0.999	0.001	0.007	0.998	0.002	62.57	85.74	121.84	134.59
250	70	2296	20863	31	19	0.992	0.999	0.001	0.008	0.998	0.002	46.30	67.10	104.75	119.89
250	80	2296	20858	36	19	0.992	0.998	0.002	0.008	0.998	0.002	42.09	62.57	102.43	119.58

Continued on next page. . .

Table C.1 – Continued

k	m	TP	TN	FP	FN	TPR	TNR	FPR	FNR	Wacc	Werr	TCR, $\lambda = 1$	TCR, $\lambda = 2$	TCR, $\lambda = 10$	TCR, $\lambda = 100$
500	7	2291	20890	4	24	0.990	1.000	0.000	0.010	0.999	0.001	82.68	89.04	94.88	96.30
500	15	2295	20889	5	20	0.991	1.000	0.000	0.009	0.999	0.001	92.60	102.89	112.93	115.46
500	20	2296	20887	7	19	0.992	1.000	0.000	0.008	0.999	0.001	89.04	102.89	117.51	121.39
500	30	2296	20881	13	19	0.992	0.999	0.001	0.008	0.999	0.001	72.34	90.78	114.04	121.01
500	40	2297	20878	16	18	0.992	0.999	0.001	0.008	0.999	0.001	68.09	89.04	118.11	127.48
500	50	2297	20871	23	18	0.992	0.999	0.001	0.008	0.998	0.002	56.46	78.47	114.04	126.99
500	60	2298	20864	30	17	0.993	0.999	0.001	0.007	0.998	0.002	49.26	72.34	115.75	133.82
500	70	2298	20863	31	17	0.993	0.999	0.001	0.007	0.998	0.002	48.23	71.23	115.17	133.74
500	80	2298	20859	35	17	0.993	0.998	0.002	0.007	0.998	0.002	44.52	67.10	112.93	133.43

C.2 Cost sensitive learning for the *NTME1-TME1* data set

Table C.2 shows detailed results for executing the random forest classifier against the *NTME1-TME1* data set with various values for λ.

Table C.2: Cost sensitive learning for the *NTME1-TME1* data set with $k = 50$, $m = 30$

cost	TP	TN	FP	FN	TPR	TNR	FPR	FNR	Wacc	Werr	TCR, $\lambda = 1$	TCR, $\lambda = 2$	TCR, $\lambda = 10$	TCR, $\lambda = 100$
$\lambda = 1$	2300	20880	14	15	0.9935	0.9993	0.0007	0.0065	0.998750	0.001250	79.83	105.23	141.16	152.91
$\lambda = 2$	2299	20875	19	16	0.9931	0.9991	0.0009	0.0069	0.998492	0.001508	66.14	90.78	129.33	142.99
$\lambda = 10$	2303	20858	36	12	0.9948	0.9983	0.0017	0.0052	0.997932	0.002068	48.23	77.17	148.40	187.30
$\lambda = 100$	2311	20629	265	4	0.9983	0.9873	0.0127	0.0017	0.988410	0.011590	8.61	16.96	75.90	348.12

C.3 Feature reduction for the *NTME1-TME1* data set - Removing Most Important Features

Table C.3 shows detailed results for executing the random forest classifier against the *NTME1-TME1* data set with successively fewer features (removing the most important features first).

C.4 Feature reduction for the *NTME1-TME1* data set - Removing Least Important Features

Table C.4 shows detailed results for executing the random forest classifier against the *NTME1-TME1* data set with successively fewer features (removing the least important features first).

Table C.3: Feature reduction for the *NTME1-TME1* data set with $k = 50$, $m = 30$

F	**TP**	**TN**	**FP**	**FN**	**TPR**	**TNR**	**FPR**	**FNR**	**Wacc**	**Werr**	**TCR,** $\lambda = 1$	**TCR,** $\lambda = 2$	**TCR,** $\lambda = 10$	**TCR,** $\lambda = 100$
1	2267	20877	17	48	0.9793	0.9992	0.0008	0.0207	0.997199	0.002801	35.62	40.97	46.58	48.06
2	2263	20867	27	52	0.9775	0.9987	0.0013	0.0225	0.996596	0.003404	29.30	35.34	42.32	44.29
3	2256	20877	17	59	0.9745	0.9992	0.0008	0.0255	0.996725	0.003275	30.46	34.30	38.14	39.12
4	2205	20811	83	110	0.9525	0.9960	0.0040	0.0475	0.991684	0.008316	11.99	15.28	19.57	20.89
5	2200	20815	79	115	0.9503	0.9962	0.0038	0.0497	0.991641	0.008359	11.93	14.98	18.84	19.99
6	2173	20787	107	142	0.9387	0.9949	0.0051	0.0613	0.989271	0.010729	9.30	11.84	15.16	16.18
7	2130	20730	164	185	0.9201	0.9922	0.0078	0.0799	0.984963	0.015037	6.63	8.67	11.49	12.40
8	1963	20695	199	352	0.8479	0.9905	0.0095	0.1521	0.976259	0.023741	4.20	5.13	6.22	6.54
9	1952	20691	203	363	0.8432	0.9903	0.0097	0.1568	0.975613	0.024387	4.09	4.98	6.04	6.34
10	1926	20675	219	389	0.8320	0.9895	0.0105	0.1680	0.973803	0.026197	3.81	4.64	5.63	5.92
11	1923	20668	226	392	0.8307	0.9892	0.0108	0.1693	0.973372	0.026628	3.75	4.58	5.58	5.87
12	1904	20651	243	411	0.8225	0.9884	0.0116	0.1775	0.971821	0.028179	3.54	4.35	5.32	5.60
13	1607	20578	316	708	0.6942	0.9849	0.0151	0.3058	0.955879	0.044121	2.26	2.67	3.13	3.26
14	1527	20588	306	788	0.6596	0.9854	0.0146	0.3404	0.952863	0.047137	2.12	2.46	2.83	2.93
15	1532	20582	312	783	0.6618	0.9851	0.0149	0.3382	0.952820	0.047180	2.11	2.47	2.84	2.94
16	1229	20696	198	1086	0.5309	0.9905	0.0095	0.4691	0.944677	0.055323	1.80	1.95	2.09	2.13
17	1187	20675	219	1128	0.5127	0.9895	0.0105	0.4873	0.941962	0.058038	1.72	1.87	2.01	2.05
18	1169	20678	216	1146	0.5050	0.9897	0.0103	0.4950	0.941316	0.058684	1.70	1.85	1.98	2.02
19	1151	20673	221	1164	0.4972	0.9894	0.0106	0.5028	0.940325	0.059675	1.67	1.82	1.95	1.99
20	1140	20699	195	1175	0.4924	0.9907	0.0093	0.5076	0.940971	0.059029	1.69	1.82	1.94	1.97
21	1020	20696	198	1295	0.4406	0.9905	0.0095	0.5594	0.935672	0.064328	1.55	1.66	1.76	1.78
22	712	20721	173	1603	0.3076	0.9917	0.0083	0.6924	0.923478	0.076522	1.30	1.37	1.43	1.44
23	663	20746	148	1652	0.2864	0.9929	0.0071	0.7136	0.922444	0.077556	1.29	1.34	1.39	1.40
24	629	20780	114	1686	0.2717	0.9945	0.0055	0.7283	0.922444	0.077556	1.29	1.33	1.36	1.37
25	616	20783	111	1699	0.2661	0.9947	0.0053	0.7339	0.922013	0.077987	1.28	1.32	1.35	1.36
30	522	20773	121	1793	0.2255	0.9942	0.0058	0.7745	0.917532	0.082468	1.21	1.25	1.28	1.29
40	287	20847	47	2028	0.1240	0.9978	0.0022	0.8760	0.910595	0.089405	1.12	1.13	1.14	1.14
50	225	20854	40	2090	0.0972	0.9981	0.0019	0.9028	0.908225	0.091775	1.09	1.10	1.11	1.11
60	67	20884	10	2248	0.0289	0.9995	0.0005	0.9711	0.902710	0.097290	1.03	1.03	1.03	1.03
70	20	20894	0	2295	0.0086	1.0000	0.0000	0.9914	0.901116	0.098884	1.01	1.01	1.01	1.01
80	0	20894	0	2315	0.0000	1.0000	0.0000	1.0000	0.900254	0.099746	1.00	1.00	1.00	1.00

Table C.4: Feature reduction for the *NTME1-TME1* data set with $k = 50$, $m = 30$

F	TP	TN	FP	FN	TPR	TNR	FPR	FNR	Wacc	Werr	TCR, $\lambda = 1$	TCR, $\lambda = 2$	TCR, $\lambda = 10$	TCR, $\lambda = 100$
83	2300	20880	14	15	0.9935	0.9993	0.0007	0.0065	0.998750	0.001250	79.83	105.23	141.16	152.91
70	2294	20877	17	21	0.9909	0.9992	0.0008	0.0091	0.998363	0.001637	60.92	78.47	101.98	109.35
60	2296	20876	18	19	0.9918	0.9991	0.0009	0.0082	0.998406	0.001594	62.57	82.68	111.30	120.70
50	2296	20872	22	19	0.9918	0.9989	0.0011	0.0082	0.998233	0.001767	56.46	77.17	109.20	120.45
40	2295	20864	30	20	0.9914	0.9986	0.0014	0.0086	0.997846	0.002154	46.30	66.14	100.65	114.04
30	2296	20862	32	19	0.9918	0.9985	0.0015	0.0082	0.997803	0.002197	45.39	66.14	104.28	119.82
20	2291	20857	37	24	0.9896	0.9982	0.0018	0.0104	0.997372	0.002628	37.95	54.47	83.57	94.99
10	2288	20862	32	27	0.9883	0.9985	0.0015	0.0117	0.997458	0.002542	39.24	53.84	76.66	84.74
5	2279	20849	45	36	0.9844	0.9978	0.0022	0.0156	0.996510	0.003490	28.58	39.57	57.16	63.51
4	2279	20849	45	36	0.9844	0.9978	0.0022	0.0156	0.996510	0.003490	28.58	39.57	57.16	63.51
3	2277	20852	42	38	0.9836	0.9980	0.0020	0.0164	0.996553	0.003447	28.94	39.24	54.86	60.26
2	2094	20788	106	221	0.9045	0.9949	0.0051	0.0955	0.985911	0.014089	7.08	8.45	10.00	10.43
1	1629	20764	130	686	0.7037	0.9938	0.0062	0.2963	0.964841	0.035159	2.84	3.08	3.31	3.37

Appendix D: Evaluation Data for the *TS1* data set

D.1 Random forest parameter optimization for the *TS1* data set

Table D.1 shows detailed results for executing the random forest classifier against the *TS1* data set. The best performing random forest configurations for various false negative, false positive cost ratios (λ) are highlighted.

Table D.1: Random forest parameter optimization for the *TS1* data set

k	m	TP	TN	FP	FN	TPR	TNR	FPR	FNR	Wacc	Werr	TCR, $\lambda = 1$	TCR, $\lambda = 2$	TCR, $\lambda = 10$	TCR, $\lambda = 100$
10	1	36	1457446	239	8	0.818	1.000	0.000	0.182	1.000	0.000	0.18	0.35	1.38	4.23
10	2	32	1457426	259	12	0.727	1.000	0.000	0.273	1.000	0.000	0.16	0.31	1.16	3.02
10	3	34	1457531	154	10	0.773	1.000	0.000	0.227	1.000	0.000	0.27	0.51	1.73	3.81
10	5	34	1457444	241	10	0.773	1.000	0.000	0.227	1.000	0.000	0.18	0.34	1.29	3.55
10	10	33	1457302	383	11	0.750	1.000	0.000	0.250	1.000	0.000	0.11	0.22	0.89	2.97
10	15	28	1457401	284	16	0.636	1.000	0.000	0.364	1.000	0.000	0.15	0.28	0.99	2.34
10	20	33	1456921	764	11	0.750	0.999	0.001	0.250	0.999	0.001	0.06	0.11	0.50	2.36
10	30	26	1457134	551	18	0.591	1.000	0.000	0.409	1.000	0.000	0.08	0.15	0.60	1.87
30	1	35	1457533	152	9	0.795	1.000	0.000	0.205	1.000	0.000	0.27	0.52	1.82	4.18
30	2	40	1457528	157	4	0.909	1.000	0.000	0.091	1.000	0.000	0.27	0.53	2.23	7.90
30	3	38	1457529	156	6	0.864	1.000	0.000	0.136	1.000	0.000	0.27	0.52	2.04	5.82
30	5	39	1457490	195	5	0.886	1.000	0.000	0.114	1.000	0.000	0.22	0.43	1.80	6.33
30	10	33	1457350	335	11	0.750	1.000	0.000	0.250	1.000	0.000	0.13	0.25	0.99	3.07
30	15	32	1457302	383	12	0.727	1.000	0.000	0.273	1.000	0.000	0.11	0.22	0.87	2.78
30	20	31	1457213	472	13	0.705	1.000	0.000	0.295	1.000	0.000	0.09	0.18	0.73	2.48
30	30	26	1456947	738	18	0.591	0.999	0.001	0.409	0.999	0.001	0.06	0.11	0.48	1.73
50	1	36	1457534	151	8	0.818	1.000	0.000	0.182	1.000	0.000	0.28	0.53	1.90	4.63
50	2	40	1457517	168	4	0.909	1.000	0.000	0.091	1.000	0.000	0.26	0.50	2.12	7.75

Continued on next page...

Table D.1 – Continued

k	m	TP	TN	FP	FN	TPR	TNR	FPR	FNR	Wacc	Werr	TCR, $\lambda = 1$	TCR, $\lambda = 2$	TCR, $\lambda = 10$	TCR, $\lambda = 100$
50	3	40	1457516	169	4	0.909	1.000	0.000	0.091	1.000	0.000	0.25	0.50	2.11	7.73
50	5	38	1457484	201	6	0.864	1.000	0.000	0.136	1.000	0.000	0.21	0.41	1.69	5.49
50	10	31	1457324	361	13	0.705	1.000	0.000	0.295	1.000	0.000	0.12	0.23	0.90	2.65
50	15	33	1457297	388	11	0.750	1.000	0.000	0.250	1.000	0.000	0.11	0.21	0.88	2.96
50	20	31	1457065	620	13	0.705	1.000	0.000	0.295	1.000	0.000	0.07	0.14	0.59	2.29
50	30	26	1456827	858	18	0.591	0.999	0.001	0.409	0.999	0.001	0.05	0.10	0.42	1.66
100	1	38	1457570	115	6	0.864	1.000	0.000	0.136	1.000	0.000	0.36	0.69	2.51	6.15
100	2	40	1457551	134	4	0.909	1.000	0.000	0.091	1.000	0.000	0.32	0.62	2.53	8.24
100	3	38	1457523	162	6	0.864	1.000	0.000	0.136	1.000	0.000	0.26	0.51	1.98	5.77
100	5	32	1457469	216	12	0.727	1.000	0.000	0.273	1.000	0.000	0.19	0.37	1.31	3.11
100	10	31	1457364	321	13	0.705	1.000	0.000	0.295	1.000	0.000	0.13	0.25	0.98	2.71
100	15	31	1457301	384	13	0.705	1.000	0.000	0.295	1.000	0.000	0.11	0.21	0.86	2.61
100	20	31	1457215	470	13	0.705	1.000	0.000	0.295	1.000	0.000	0.09	0.18	0.73	2.49
100	30	26	1456876	809	18	0.591	0.999	0.001	0.409	0.999	0.001	0.05	0.10	0.44	1.69
250	1	38	1457567	118	6	0.864	1.000	0.000	0.136	1.000	0.000	0.35	0.68	2.47	6.13
250	2	39	1457544	141	5	0.886	1.000	0.000	0.114	1.000	0.000	0.30	0.58	2.30	6.86
250	3	39	1457531	154	5	0.886	1.000	0.000	0.114	1.000	0.000	0.28	0.54	2.16	6.73
250	5	31	1457483	202	13	0.705	1.000	0.000	0.295	1.000	0.000	0.20	0.39	1.33	2.93
250	10	31	1457373	312	13	0.705	1.000	0.000	0.295	1.000	0.000	0.14	0.26	1.00	2.73
250	15	31	1457321	364	13	0.705	1.000	0.000	0.295	1.000	0.000	0.12	0.23	0.89	2.64
250	20	31	1457201	484	13	0.705	1.000	0.000	0.295	1.000	0.000	0.09	0.17	0.72	2.47
250	30	26	1456943	742	18	0.591	0.999	0.001	0.409	0.999	0.001	0.06	0.11	0.48	1.73
500	1	37	1457569	116	7	0.841	1.000	0.000	0.159	1.000	0.000	0.36	0.68	2.37	5.39
500	2	38	1457550	135	6	0.864	1.000	0.000	0.136	1.000	0.000	0.31	0.60	2.26	5.99
500	3	39	1457531	154	5	0.886	1.000	0.000	0.114	1.000	0.000	0.28	0.54	2.16	6.73
500	5	33	1457476	209	11	0.750	1.000	0.000	0.250	1.000	0.000	0.20	0.38	1.38	3.36
500	10	30	1457393	292	14	0.682	1.000	0.000	0.318	1.000	0.000	0.14	0.28	1.02	2.60

Continued on next page...

Table D.1 – Continued

k	m	TP	TN	FP	FN	TPR	TNR	FPR	FNR	Wacc	Werr	TCR, $\lambda = 1$	TCR, $\lambda = 2$	TCR, $\lambda = 10$	TCR, $\lambda = 100$
500	15	31	1457327	358	13	0.705	1.000	0.000	0.295	1.000	0.000	0.12	0.23	0.90	2.65
500	20	31	1457208	477	13	0.705	1.000	0.000	0.295	1.000	0.000	0.09	0.17	0.72	2.48
500	30	26	1456966	719	18	0.591	1.000	0.000	0.409	0.999	0.001	0.06	0.12	0.49	1.75

D.2 Cost sensitive learning for the *TS1* data set

Table D.2 shows detailed results for executing the random forest classifier against the *TS1* data set with various values for λ.

Table D.2: Cost sensitive learning for the *TS1* data set with $k = 100$, $m = 2$

cost	**TP**	**TN**	**FP**	**FN**	**TPR**	**TNR**	**FPR**	**FNR**	**Wacc**	**Werr**	**TCR,** $\lambda = 1$	**TCR,** $\lambda = 2$	**TCR,** $\lambda = 10$	**TCR,** $\lambda = 100$
$\lambda = 1$	40	1457551	134	4	0.909	1.000	0.000	0.091	1.000	0.000	0.32	0.62	2.53	8.24
$\lambda = 2$	40	1457551	134	4	0.909	1.000	0.000	0.091	1.000	0.000	0.32	0.62	2.53	8.24
$\lambda = 10$	40	1457353	332	4	0.909	1.000	0.000	0.091	1.000	0.000	0.13	0.26	1.18	6.01
$\lambda = 100$	42	1438266	19419	2	0.955	0.987	0.013	0.045	0.987	0.013	0.00	0.00	0.02	0.22