Czech Technical University in Prague
Faculty of Electrical Engineering
Department of Cybernetics

Phishing Email Detection in Czech Language for Email.cz

Bc. Vít Listík

A thesis submitted to
the Faculty of Electrical Engineering, Czech Technical University in Prague

Programme: Computer Software Engineering

Prague, September 2015

Czech Technical University in Prague
Faculty of Electrical Engineering

Department of Computer Graphics and Interaction

DIPLOMA THESIS ASSIGNMENT

Student: **Bc. Vít Listík**

Study programme: Open Informatics
Specialisation: Software Engineering

Title of Diploma Thesis: **Phishing Email Detection in Czech Language**

Guidelines:

Phishing is a technique which uses means of electronic communication to acquire sensitive personal information (passwords, credit card details) by masquerading as a trustworthy entity. The goal of this thesis is to find, implement and test an algorithm for detecting phishing emails. The final algorithm should make use state-of-the-art methods of NLP (Natural Language Processing) or other machine learning techniques.

- Review state of the art techniques in the field of fishing detection
- Review methods for evaluating quality of fishing detection algorithms
- Gather the training and testing data from large corpus of bulk emails
- Select a method for detection and implement it as a baseline
- Measure the performance of the baseline algorithm
- Modify the selected methods to get the best results
- Evaluate the experiments and summarize the results

Bibliography/Sources:

[1] ALMOMANI, Ammar, et al. A survey of phishing email filtering techniques. Communications Surveys & Tutorials, IEEE, 2013, 15.4: 2070-2090.
[2] KHONJI, Mahmoud; IRAQI, Youssef; JONES, Andrew. Phishing detection: a literature survey. Communications Surveys & Tutorials, IEEE, 2013, 15.4: 2091-2121.
[3] RAMANATHAN, Venkatesh; WECHSLER, Harry. Phishing detection and impersonated entity discovery using Conditional Random Field and Latent Dirichlet Allocation. Computers & Security, 2013, 34: 123-139.
[4] VERMA, Rakesh; HOSSAIN, Nabil. Semantic Feature Selection for Text with Application to Phishing Email Detection. In: Information Security and Cryptology--ICISC 2013. Springer International Publishing, 2014. p. 455-468.

Diploma Thesis Supervisor: Ing. Jan Šedivý, CSc.

Valid until the end of the summer semester of academic year 2015/2016

prof. Ing. Jiří Žára, CSc.
Head of Department

prof. Ing. Pavel Ripka, CSc.
Dean

Prague, March 24, 2015

Thesis Supervisor:
Ing. Jan Šedivý Ph.D.
Department of Cybernetics
Faculty of Electrical Engineering
Czech Technical University in Prague
Karlovo namesti 13
121 35 Praha 2
Czech Republic

Abstract

Phishing emails detection methods which are used nowadays are often based on links blacklisting. Goal of this work is to detect these emails automatically. State of the art techniques were evaluated and decision tree classifier based on 25 features was trained on public phishing data set. Promising results of this approach reached with testing data set, were not confirmed in live traffic.

Used data set is not representative most probably because it contains old emails. New solution using configurable scales was designed. This solution is based on two phases. First phase is prefiltering and second is phishing detection itself. Prefiltering phase is used to reduce heavy computations and consists of two steps. First step is based on 30 traffic statistics features which directly modifies metric called *phishing-score* because traffic statistics for phishing emails are not available for training. Second phase uses decision tree classifier, which is based on 25 content features, for binary classification (phishing, non-phishing).

Second phase is also divided into two steps and is conditioned by high score from prefiltering phase. At first it detects sender domain by domain specific keywords, commonly used image sources, plain links to domains and header from. Secondly it detects most suspicious link and decides whether domain extracted from links is commonly targeted by detected domain. This step decides whether email is phishing or not by adding *phishing-score*.

Whole system is based on phishing-score and administrator is noticed when some email reaches given *phishing-score* threshold. This threshold was set via ROC evaluation, which was built on manually classified emails with high *phishing-score*. In current setup this system is capable of detecting 98% of phishing attacks with 26% of misclassifications.

Keywords:

Phishing, email, machine learning, natural language processing, Czech language.

Acknowledgement

Fist of all, I would like to thank to my thesis supervisor Ing. Jan Šedivý Ph.D for helpful advices and guidance throughout this work. My thanks also goes to Ing. Tomáš Gogár who helped me a lot with data processing and gave me very important advices with machine learning techniques. I would also like to thank Bc. Tibor Schmidt who gave me valuable advices about statistics of email traffic. Then I want to thank Ing. Michal Bukovský and his whole Email.cz team in Seznam.cz for the opportunity to work with them and with their data. J. Džubák from Hoax.cz also deserve my thanks for the willingness to provide me their data. Last but not least I want to thank my parents and the closest ones for the support throughout my whole studies.

Declaration

I hereby declare that I have completed this thesis independently and that I have listed all the literature and publications used.

I have no objection to usage of this work in compliance with the act §60 Zákon č. 121/200Sb. (copyright law), and with the rights connected with the copyright act including the changes in the act.

In Prague on May 11, 2015 .....

Contents

List of Figures x

List of Tables xiii

Abbreviations xiv

1 Introduction 1
- 1.1 Motivation . . . 1
- 1.2 Problem Statement - Phishing . . . 1
  - 1.2.1 Email . . . 2
  - 1.2.2 Email Protocols . . . 2
  - 1.2.3 Message Structure . . . 3
  - 1.2.4 Mail Delivery Process . . . 4
  - 1.2.5 Types of Messages (HAM, SPAM) . . . 4
- 1.3 Related Work/Previous Results . . . 5
- 1.4 Contributions of the Thesis . . . 5

2 Background and State-of-the-Art 7
- 2.1 Theoretical Background . . . 7
- 2.2 Forms of Phishing . . . 8
- 2.3 Previous Results and Related Work . . . 8
  - 2.3.1 Content Based . . . 8
  - 2.3.2 Meta Information . . . 9
  - 2.3.3 Public Data Set . . . 10
  - 2.3.4 List of Used Features . . . 10
- 2.4 Evaluation of Existing Methods . . . 21
  - 2.4.1 Testing . . . 22
  - 2.4.2 Results . . . 23
  - 2.4.3 Production Environment . . . 25
  - 2.4.4 Discussion . . . 25

3 Overview of Our Approach 27
- 3.1 Traffic Statistics . . . 29

3.1.1 Domain Reputation . . . 32
3.1.2 DKIM . . . 32
3.1.3 SPF . . . 34
3.1.4 Phishing Keywords . . . 35
3.1.5 Topic Categorization Model . . . 35
3.2 Phisheable . . . 35
3.3 Email Content Model . . . 38
3.3.1 Phishing Data Sets . . . 38
3.3.2 Non-phishing Data Set . . . 39
3.3.3 Used Features . . . 39
3.4 Classification . . . 42
3.5 Attacked Domain Detection . . . 43
3.6 Domain Specific Keywords Recognizer . . . 44
3.6.1 Dataset . . . 44
3.6.2 Cacheable . . . 45
3.6.3 Mf-idf . . . 45
3.6.4 Computation . . . 45
3.6.5 Classes . . . 46
3.6.6 Phisheable . . . 46
3.6.7 Vectorizer . . . 46
3.6.8 Number of Emails . . . 47
3.6.9 Model Testing . . . 48
3.6.10 Classifier . . . 48
3.7 Domain Image Sources . . . 48
3.7.1 Rank Classifier . . . 49
3.8 Claimed Domains . . . 49
3.9 Suspicious Link Detection . . . 50
3.10 Link Statistics . . . 51
3.10.1 Common Links . . . 51
3.10.2 Global Links . . . 51
3.10.3 Malicious Link Detection . . . 52
3.11 Final Decision . . . 52

4 Main Results **53**
4.1 Traffic Statistics . . . 53
4.1.1 Text Forms . . . 54
4.1.2 Phishing Keywords . . . 54
4.1.3 Phisheable . . . 54
4.1.4 Results . . . 55
4.2 Email Content Model . . . 56
4.3 Claimed Domain Detection . . . 57
4.3.1 Overall Performance . . . 57
4.4 Suspicious Link Detection . . . 57

4.5 Whole System Evaluation 57
4.5.1 Phishing Score Distribution 57
4.5.2 System Evaluation 58

5 Conclusions **61**
5.1 Summary 61
5.2 Contributions of the Thesis 62
5.3 Future Work 62

Bibliography **63**

A Content of attached DVD **69**

B Statistical Evaluation of Content Features **71**

List of Figures

1.1 Email delivery process. Source [70] 5
1.2 Example of phishing alert based on link blacklisting shown in email client (MUA) Mozilla Thunderbird 5

2.1 Precision. 23
2.2 Recall. 23
2.3 Accuracy. 23
2.4 F1-score. 24
2.5 Graph shows count of phishing and non-phishing emails based on classifier result. Time is shown on X axis 25

3.1 Phishing detection proces 28
3.2 Phish score calculation process 33
3.3 DKIM verification process. Source [3] 34
3.4 DKIM score calculation process 36
3.5 SPF verification process. Source [3] 37
3.6 Where TF is term frequency (How many times word appeared in one document) and DF is document frequency (In how many distinct documents word appeared) and DC is number of documents in dataset 38
3.7 Min leaf size evaluation 44
3.8 MF-IDF where MF is mail frequency - in how many emails from domain word appeared, DC domain count - how many domains are in dataset, DF - domain frequency - in how many domains word appeared 45
3.9 TF-MF-IDF where TF is term frequency - how many times word appeared in domain mails, MF is mail frequency - in how many emails from domain word appeared, DC domain count - how many domains are in dataset, DF - domain frequency - in how many domains word appeared 46
3.10 Testing vectorizers for domain keywords recognizer 47
3.11 Testing how number of emails affects classifier performance. 47
3.12 Main part of rank classifier. 49
3.13 Where max is maximum count a min minimal count, 0.2 is percentage limit. 51
3.14 Where max is maximum count a min minimal count, 0.1 is percentage limit. 51

4.1 Phishing score based on traffic stats. Green is inbox and yellow is spam. X axis shows phishing-score, Y axis count of occurrences 53
4.2 Phishing score based on traffic stats more than zero. Green is inbox and yellow is spam. X axis shows phishing-score, Y axis count of occurrences . 53
4.3 Text form regular expression. 54
4.4 Phishing form regular expression. 54
4.5 Graph shows count of phishing and non-phishing emails based on classifier result. Time is shown in X axis. 56
4.6 ROC of phishing email classification for unique emails, based on phishing-score threshold setup. 59
4.7 ROC of phishing email classification for real count of emails delivered (not unique), based on phishing-score threshold setup. 60

List of Tables

2.1 HTML features. 11
2.2 Link features. 15
2.3 Image features. 15
2.4 JS features. 16
2.5 Spamfilter features. 17
2.6 Content features. 18
2.7 Wordlist features. 19
2.8 Subject features. 20
2.9 Header features. 21
2.10 Used features. 22
2.11 Learned decision tree test results. 23
2.12 Used features usage count. 24

3.1 Traffic statistic features. 32
3.2 SPF operators. 35
3.3 Phishing datasets. 39
3.4 Content features. 41
3.5 Used content features evaluation. 43
3.6 Test results for various models used for domain keywords recognition. . . . 48
3.7 Features used for suspicious link detection. Where + means positive (more suspicious) and - means negative (less suspicious) 50

4.1 Number of occurrences of features used in traffic statistics filter. 56
4.2 Email content model evaluation. 56
4.3 Phishing-score distribution. Statistics for 4 days. 58
4.4 Phishing emails distribution based on phishing-score. 58
4.5 Classifier statistics based on threshold setup. 60

Abbreviations

List of Abbreviations

A	DNS record in form of IP address
DF	Document frequency
DNS	Domain Name System
FP	False positive
FPR	False positive rate
GB	Gigabyte
HTML	Hypertext markup language
IPv4	Internet Protocol version 4
IPv6	Internet Protocol version 6
JS	Javascript
MD5	Message-digest algorithm used for hashing
MDA	Mail delivery agent
MTA	Mail transfer agent
MUA	Mail user agent
MX	Mail exchanger DNS record
NLP	Natural language processing
POP	Post Office Protocol
ROC	Receiver operating characteristic
SMTP	Simple Mail Transfer Protocol
TF	Term frequency
TP	True positive
TPR	True positive rate
URL	Uniform resource locator

Commonly used terms

Envelope from	Defines sender on the SMTP server
Ham	Wanted email messages, opposite to spam
Header from	Sender defined property in email message headers
spam	Unwanted email massages, opposite to ham

Chapter 1

Introduction

Phishing is kind of electronic identity theft which uses social engineering techniques. Phishing is often used to steal personal information like important online accounts e.q. bank or email, or to steal credit card numbers. Attackers use these information for stealing money directly or for other kinds of fraudulent activities like selling stolen email accounts to botnets.

Email message containing text luring to click on link leading to phishing website is essential for phishing success. Phishing attacks are successful because sent messages and target websites are often looking very trustworthy. Legitimate sources are copied in many attacks. In text of sent email messages attackers often urge to take quick action e.q. to change password. To maximize phishing success emails are often sent to many recipients.

The goal of this thesis is to find, implement and test an algorithm for detecting phishing emails. Detecting algorithm will probably be based on NLP (Natural Language Processing) or other machine learning techniques.

1.1 Motivation

The biggest Czech freemail provider which delivers millions of emails a day and also some phishing emails among them. It is desired to design and implement custom phishing detection solution, because custom spamfilter is used in Seznam.cz.

1.2 Problem Statement - Phishing

In last 10 years worldwide phishing attacks raised from hundreds to tens of thousands [22, 23]. Last year (2014) there were most phishing attacks ever measured [20]. In latest measured month (September 2014) 53 661 unique phishing emails was sent [22]. This number is very slightly decreasing in last years [21].

These high numbers of attacks made in year 2007 about 3.6 billion US dollars loss and in September 2014 Czech Republic was in top 10 countries hosting malicious websites [18, 22].

Phishing is evolving, it developed some variations worth mentioning.

- Spear-phishing is targeted phishing which gathers information about victim which are publicly available and than the attack may be personalized [50].

- Pharming is often also referenced as type of phishing. Pharming is based on DNS spoofing which redirects victim from legit page to fraudulent one. Huge problem is that URL address is correct and the victim does not suspect anything [53].

More than three quarters of attacks are targeted on financial institutions [21]. It shows that phishers are trying to get personal data mostly to steal money. Other intention is the identity steal itself. When attacker gets account he may use legit user permissions or sell the account to somebody who wants to use it.

Social engineering is discipline when attacker claims that he is someone else and often he also claims that he or victim will have problems when they doesn't do what he is saying. Phishing is very often based on exactly the same technique. Sometimes attackers are asking for data directly, to by sent in reply. But most common scenario is that attacker use spoofing and claims to be some known company e.g. bank. In text is often used urgency like: "You have to re-activate your account in 24 hours" or some non-standard situation: "Your account was hacked change your password". Message also contains link to fraudulent website on which attacker creates form which is asking for personal information, mainly passwords or credit card numbers. Template of legit company is often copied, so victim may not spot any differences on first sight.

Recent research results showed, that good phishing websites fooled 90% of participants [30].

Phishing is of course illegal not only in Czech republic. It is classified as type of fraud. Phishing causes problems not only to individuals but also to targeted companies and email providers, which are loosing their good reputation.

Most common first contact with phishing is in the email message, which is described further.

1.2.1 Email

Electronic mail commonly known as email or e-mail is electronic version of traditional letters delivered by post office. Email was one of the main reasons why internet was created. Email main purpose is to deliver information from one person (sender) to one or more other persons (recipients). Email got its structure in 1973 by RFC 561 [25], than it evolved many times to RFC 5322 [39] which is used nowadays.

1.2.2 Email Protocols

SMTP - Simple Mail Transfer Protocol (defined by RFC 821 [67] in 1982) defines how are email messages sent. SMTP is text-based protocol using TCP port 25 or 587 for

communication. It defines set of possible commands for client (sender) and format of transmitted data. It also defines how server (receiver) should react on these commands.

SMTP server has DNS name in its configuration. This name is used in envelope from (RFC 5321) [48].

Sender locates target SMTP server by MX record. Sender extracts receiver's domain from his address (part after @), after that he resolves MX record stored in DNS.

Relay servers are used for sending messages from external clients. When using relay you have to be authenticated, otherwise it would be really simple to send messages from your account.

Most actual version was established in 2008 and is called RFC 5321 [48]. SMTP also has secured alternative which is using SSL and is called SMTPS.

Most common protocols for receiving email messages are described further.

- **POP** - Post Office Protocol is used for message retrieval. Actual version in POP3 described in RFC 1939 (1996) [45]. POP is used to download and delete messages from mailserver, but it can be configured to leave messages on the server.

- **IMAP** - Internet Message Access Protocol is also used for message retrieval but is more complex than POP. IMAP fourth actual version is defined in RFC 3501 [38]. IMAP typically uses port 143 and 993 for SSL version IMAPS. In contrast with POP IMAP leaves messages on server. IMAP is also able to work with user defined folders.

1.2.3 Message Structure

Message has to be in defined format. Email consists of these parts:

- header,

- body,

which are described below.

Header consists of many fields. Filelds start with field name 7-bit ASCII string followed by ":", space or tab and the value itself.

Mandatory fields are From and Date. Commonly used are: To (Bcc, Cc) and Subject.

Email body use MIME format. Mime stands for Multi-Purpose Internet Mail Extensions. Thanks to MIME we can send text messages in other than ASCII character sets, send attachments and multipart messages. MIME originated in RFC 821 as part of SMTP [67]. Because of problems with encoding language specific characters Unicode (UTF-8) is often used. Email text message is often base64 encoded. MIME is defined in RFC 2045 through RFC 2049 [43, 44, 40, 42, 41].

Email body may be in plain text or on HTML format. HTML (Hypertext markup language) [73] is based on tags which defines appearance and behavior.

1.2.4 Mail Delivery Process

Email delivery process is described with three main components.

- **MUA** - mail user agent also known as Email client. Email may be read in desktop or webmail clients. I case of desktop clients emails are downloaded to computer, so there is much bigger danger, because email can trigger some action on the computer after opening and spread some malicious software. How desktop clients fight with phishing attacks is shown on Fig. 1.2.
- **MTA** - Mail transfer agent is server which stores emails and sends them to other MTAs.
- **MDA** - Mail delivery agent is server which downloads email from MTA to users MUA.

Email delivery process may proceed as follows:

1. User compose message in his Mail User Agent (MUA).
2. MUA sends an email via SMTP to local or relay SMTP server - MTA.
3. MTA finds destination of MTA server based DNS MX records.
4. MTA delivers email to desired location via SMTP.
5. Receiving MTA stores email on MDA
6. Receivers MUA downloads email from MDA via POP or IMAP.

Whole process is shown on Fig. 1.1

1.2.5 Types of Messages (HAM, SPAM)

- **SPAM** - undesired, unsolicited bulk email also referred as junk email. Spam if often sent from botnets. Spam often contains malicious attachment or links. Sending spam is illegal in Czech Republic and many other countries on the World.
- **SCAM** - misleading fraudulent messages. For example Nigerian messages alluring money.
- **SPOOFING** - sent email is pretending to be from someone else (forged sender address in header)
- **BOMBING** - sending huge amount of messages to one address, which often causes that target mailbox is unusable.
- **Other malicious formats** - email is often used for spreading viruses or worms.
- **HAM** - opposite of spam.

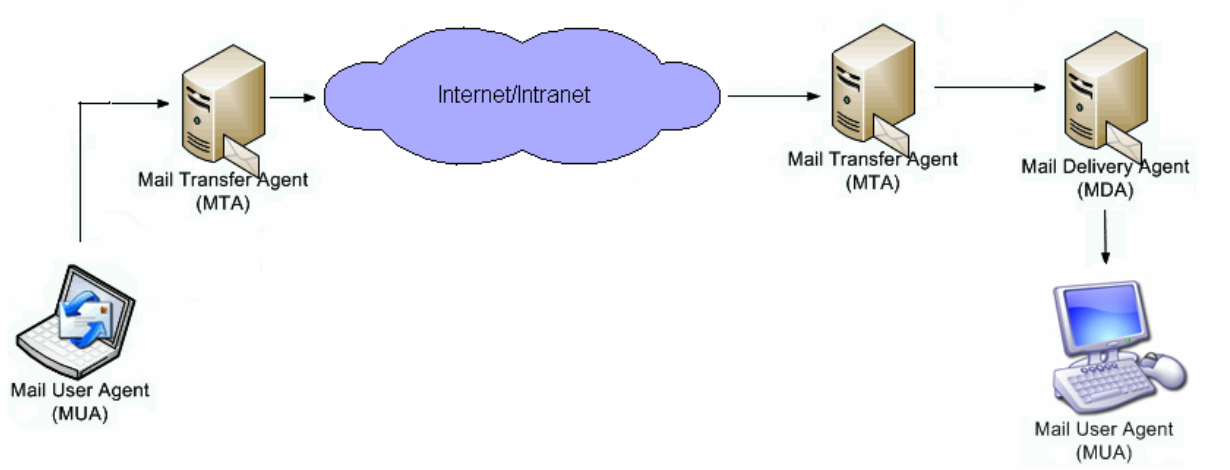

Figure 1.1: Email delivery process. Source [70]

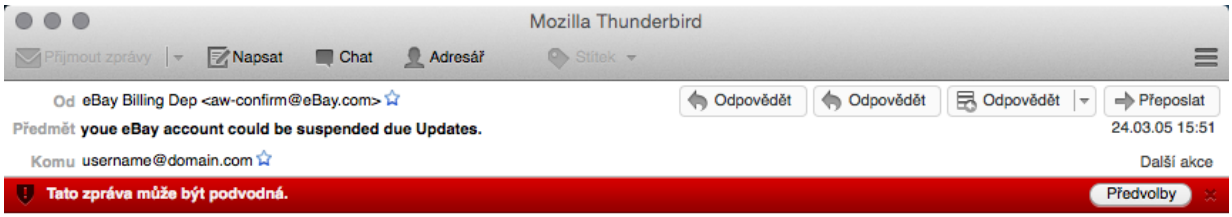

Figure 1.2: Example of phishing alert based on link blacklisting shown in email client (MUA) Mozilla Thunderbird

1.3 Related Work/Previous Results

At the end of 2013 [54] was published. It shows that there are many people working on this topic. Many teams achieved very good results on public dataset. Huge companies usually have custom solutions, but open source solutions are also available. ScamNailer uses lists of addresses from which it generates rules for SpamAssassin [11]. PhishTag is based on link blacklisting and it rewrites malicious links. PhishTag is also connected with SpamAssassin [7].

1.4 Contributions of the Thesis

In this work we want to design and implement phishing detection system. For that purpose we had to review state of the art techniques in this field. Than gather data for training and testing baseline solution based on selected methods from state of the art techniques. Than performance of baseline solution should be measured on testing dataset and in live traffic. Finally we have to implement solution for phishing email detection based on test

results of baseline solution and evaluate this method.

Chapter 2

Background and State-of-the-Art

Email is very old technology, so frauds based on email platform are very common. Phishing and its detection is almost as old as email itself, but attacks are always one step further before detection mechanisms. In following chapters phishing attacks and ways how they are prevented will be introduced.

2.1 Theoretical Background

Phishing is real problem. Fighting against it is hard, because its success is based on victims confusion and it is not technically difficult to look like phished company. Also it is not clear who should provide defense mechanisms, if it should be hosting companies, freemails or phished companies itself.

Attackers create form which requests data which they want. Then they put this form on free webhosting or hacked website. Problem of defending against phishing is that time for securing hacked website or blocking new webhosting accounts is too long and attackers can easily find other webhostings or security holes in other websites.

Defensive mechanisms are implemented on client or server side. Client side refers to web browser, email client and anti-virus software. Client security often consists of updated blacklist and shows warning to end user. Problem of this solution is, that client software often has only a few information about the message so it has to rely on blacklists. Server side may also check blacklists, but it has many more information about the traffic. Problem is that phishing cannot be often seen in statistics because of huge spam and legit traffic.

Phishing is based on identity theft so defensive mechanisms are often based on identity and spoofing detection.

Other totally different approach is to educate users. But there is also problem that employees may be educated, but how to educate normal users? Sometimes companies try to educate their users (e.g. Bank put security videos on their website), but many users stays uninformed and the fact that attackers are inventing new types of attacks makes education harder.

2.2 Forms of Phishing

Phishing may be divided to several categories based on the ways of personal information retrieval.

First category is link based. This category is most common and could be divided to two subcategories.

- One link - Email contains only one link pointing to phishing website. in this situation it is simple to detect which link is malicious, but it may be harder to detect which domain is attacked.
- More links - Phishing link is hidden between other links. In this case phishing link has to be detected. Advantage is that other links often point to attacked domain.

Second category is text based. This category is falling on popularity. It relies only on text in which attacker lure password. We think that many of these attacks should be catched with phishing keywords and text forms detection which are both described further.

Third category is image based. This category makes detection very hard because it contains image with text and link to phishing website and anything else. This category is not very often and we are not going to detect it.

Fourth category is attachment based. This category contain malicious software like viruses or worms, HTML files or PDF forms in attachment. Attachments are checked by anti-virus software so this type of attacks should be covered and its detection is beyond the scope of this work.

2.3 Previous Results and Related Work

Many scientists tried to find solution for this problem. Many different approaches are used for detecting phishing. We wanted to discover most successful methods used nowadays so we did research of what methods are used in most cited articles cited in [18, 54]. Most successful are methods based on information contained in the messages itself or message meta information. Information used for phishing detection are described further.

2.3.1 Content Based

Content based detection methods are very popular, because they may be tested offline, are reproducible and some content has to be present in all successful phishing emails.

Almost every phishing contains URL to malicious website. These URLs tries not to be suspicious. Some techniques listed below are used for this purpose.

- Href mismatch - most common technique of obfuscation. HTML tag for link has following syntax:

  ```
  <a href="real_link">Visible link</a>
  ```

. Attackers may use this feature to host their website wherever they want and put authentic link to "visible link" field. Often words like "click here" or "activate" are used in this field. Problem is that this technique used also by legit companies.

- Domain similarity - some people control what actual link they are going to visit, so attackers buys domains which looks very similar on first sight e.g. paypal.com - paypol.com

- Phishing hosted on hacked websites and freehostings - attackers are often using legit company name in the URL path or subdomain.

Blacklists are often used to prevent damage of phishing attacks and is often based on blocking targets website URL or senders IP address. With blacklists, there is huge problem with speed. Phishing messages are sent, somebody recognizes that email is phishing reports it. Phishing is included on the blacklist. Blacklist has to be updated and after that it starts working. This time lag causes that many users was cheated after the report. Other problem is that attacker may very easily change message content and URL leading to fraudulent page, which results in totally new phishing for the blacklists. Commonly known blacklist is phishtank.com which accepts reports from users, evaluates them and if link is considered as phishing they publish them to their public database. Phishtank offers online API or database download [8]. Other heavily used blacklist is Google Safe Browsing API which is used in Google Chrome and offers only online API [32].

Some of the solutions relies on that phishing will probably look similar to standard company emails. Their solution is based on image recognition. Content of message is rendered and its snapshot is taken. With snapshot is evaluated similarity to legit templates, rendering errors or company logos.

Attackers also sometimes try to put malicious scripts to phishing email messages, which should be launched on webmail clients. These scripts may use browser vulnerabilities like backdoors and steal user data or change browser functionality as attacker wants. Scripts may be easily detected and this type of attack will be ineffective in most webmail clients because javascript is not evaluated in emails.

Phishing messages often contains some common keywords like password, account, user etc. Some methods rely on bag of words model to find phishing keywords dataset and than detect them.

2.3.2 Meta Information

Traffic based information and other external information sometimes called meta information, because data origin is not email itself but delivery statistics and other data from the delivery process, are commonly used. These data may be compared with email content like from mismatch or entirely different sources like search engines.

One of the key traffic information is from mismatch mentioned before. Email has from address which was defined at SMTP commonly known as envelope from and than it has

from address in message headers. Problem of this method is that almost every mail sent from hosting company has address of hosting company in its envelope from.

If phishing message reaches user inbox, users often mark these messages as spam. These spam clics are analyzed and phishing is than delivered to spam folder or checked by administrator.

External sources may also be used for phishing recognition. When checking email links search engine may be used to search for history of the domain extracted from link, and if the domain is not known for the search engine, it may be phishing.

Some detection methods use spamfilter result as part of phishing evaluation. Idea of this method is based on that phishing messages will be sent in doses like spam and they may contain same mistakes like incorrect signs or blacklisted sender IP addresses.

2.3.3 Public Data Set

Almost every solution was tested on public phishing data set[65]. Spam and ham public data set was often used as negative samples [13]. Merged phishing data set [65] consists of 4450 emails sent from 2004 to 2007. It is separated to four parts by deliver time. All parts are in mbox format described in RFC 4155 [33]. SpamAssasin dataset [13] is divided to ham and spam and was collected from 2002 to 2005. It consists of 6047 messages in eml format compressed with bzip2 and divided to these groups:

- **spam**: 500 spam messages, all received from non-spam-trap[1] sources.
- **easy_ham**: 2500 non-spam messages. These are typically quite easy to differentiate from spam, since they frequently do not contain any spammish signatures (like HTML etc).
- **hard_ham**: 250 non-spam messages which are closer in many respects to typical spam: use of HTML, unusual HTML markup, coloured text, "spammish-sounding" phrases etc.
- **easy_ham_2**: 1400 non-spam messages. A more recent addition to the set.
- **spam_2**: 1397 spam messages. Again, more recent. [14]

2.3.4 List of Used Features

Feature selection is very important for phishing detection as far as for any other problem solved via machine learning. We gathered features from recent approaches and grouped them by common marks. For every feature we aggregated its appearances in recent articles.

HTML features are most commonly used. HTML markup [73] is essential for these features. Detection is based on tags presence and their properties. These features are used

[1]Spamtraps are email addresses published on web pages. When any message is delivered to spamtrap mailbox it is considered as spam because nobody approved delivering messages to this address

because attackers may take advantage of HTML capabilities. Tables, forms, tags, colors and HTML format itself is used.List of HTML features is shown in Tab. 2.1.

Name	Type	Description	Justification	Used
HTML email	binary	Is email in HTML format	When email is in HTML format, it may use more efficient phishing methods	[31, 71, 66, 77, 17, 63, 72, 52, 24]
Plaintext	binary	Is email in plain text	Opposite of HTML	[77]
Forms	binary	HTML contains form	Forms are used to gain information from the user	[17, 62, 77, 63]
Table	binary	HTML contains table tag	Tables are often used for formatting	[52, 63]
Tables	count	Number of tables contained		[28, 77]
Comment tags	count	Number of HTML comment tags		[52]
Tags	count	Number of HTML tags		[28]
White text	binary	Text color was set to white	It is assumed that white text is invisible	[52]
Colors	count	Number of color element (both CSS and HTML format)	With more colors invisible text may be achieved	[52]
CSS	binary	CSS was used		[52]
Fake tags	binary	HTML tags not in W3C specification		[63]

Table 2.1: HTML features.

Link features are key features for phishing detection as may be seen from high count of unique features used in related articles. Because links may lead victims to fraudulent pages. Links may be obfuscated in many ways. Link is defined as [74]:

description

So from its definition is legitimate to hide real target. Phishers use this to make victims believe that they are going to visit known page. It hard to determine when link points to unexpected target. Mainly IP address detection, link keywords, link mismatch and uncommon symbols are detected. List of link features is shown in Tab. 2.2.

Name	Type	Description	Justification	Used
Links	count	Number of <a> tags	Links may lead to phishing websites	[31, 72, 62, 77, 24, 28, 71, 63]
More than 3 links	binary			[17]
Visible links	count	Number of links which are visible		[77, 63]
Domains	count	Number of domains in links		[31, 72]
URLs to IPs	count	Links which leads to IP address not to domains	Attackers sometimes don't have DNS records	[31, 26, 27, 71, 66, 49, 17, 72, 24]
Age of linked domains	value	When was domain registered in WHOIS	Young domains may be used for phishing	[31, 63]
Link mismatch	count	legit_domain	Attackers often put something else to visible part of the link and to actual href	[31, 26, 27, 62, 28, 66, 17, 24]
Sender domain differ from url domain	count	Header from is different from linked domain	Attackers claim in header from that they are from legit domain but they send phishing links	[66, 17]
"Here" links to non-common domain	count	"click here" point elsewhere than other links (privacy policy)	Click here link will be probably pointing to phishing website	[31]
Links contain words	binary	Similar to click here, but with words (click, here, login, update)		[17, 72]

Diff href and visible link	binary	Similar to click here, but with no specific words		[66]
"Here" links to common domain	binary			[72]
More than 3 domains	binary			[17]
Image IP origin	count	Image src is IP address		[66]
Domains	count	Number of domains in URLs		[66]
Subdomains	count	Number of subdomains in URL		[66]
Image links	count	Links with image as their visible part	Link may point to elsewhere than it claims	[49, 17, 52, 72]
Number of dots	count	Number of dots in link (other version of detecting IP address)		[31, 49]
Number of dots or slashes	count	IP address or long url		[66]
more than 3 dots on domain	count	IP address		[17]
Max number of dots in link	count	IP address		[17, 72, 71]
Internal and external links	count	Number internal and external links (sender domain)		[24]
Internal links	count	Links pointing to domain claimed in header from		[72]
External links	count	Links pointing elsewhere than claimed in header from		[72]

Number of not 80 port	count	Links pointing to non 80 ports	Web pages runs on port 80, every other port is suspicious	[72, 17]
Port	count	Number of times port was specified	Port 80 is often not specified, when any port is used is suspicious	[72]
Invisible links	count	Number of invisible links	Invisible links may be clicked by accident	[62]
Hacked	count	Number of linked domains which were marked as hacked in WHOIS	Hacked website may host phishing	[77]
Hyperlinks	count	Number of non-blank hyperlinks		[52]
Link cotains at (@)	count	Number of links contaning @		[72]
Symbols	binary	Links contain symbols like % or & in url		[49]
Symbols and digits	binary	Links containing numbers or “&”, “%”, or “@’		[52]
Encoding tricks	binary	Links with encoding tricks <a href=”http://034%02E%0333%34%2E%311%39%355%2E%o340o31:%34%39%30%33/%6C/%69%6E%64%65%78%2E%68%74%6D”>	Link is unreadeble so it may look legit	[27]
Hexadecimal chars	binary	Links contains hexadecimal characters		[17]
Fake https	binary	Claimed HTTPS pointing to HTTP		[17]
Domain is similar to claimed	binary	Actual domain is edited version of claimed domain	Optically similar domain name	[27]

Domain age	value	Domain is unknown in search engine	Newly registered domains may be phish	[55]

Table 2.2: Link features.

Images may be used for harder text recognition or for determination of image origin. Because attackers often use images from attacked domain. But real companies often take advantage of CDN systems [2] which results to same outcome. Images are defined as[75]:

```
<img src="path_to_image" alt="description" />
```

. List of image features is shown in Tab. 2.3.

Name	Type	Description	Justification	Used
Images from external domain different from links	count	image src points to different domain than links	Attackers often use images from legit sources	[66]
Images	count	Number of images (img tag)		[28, 63]
Image	binary	Contains image		[77]

Table 2.3: Image features.

JavaScript abbr. JS defined in ECMA-262 standard.[5] With javascript small scripts may be sent in emails. These scripts may catch click actions or somehow change default browser behavior and confuse user or send his actions to attacker. Javascript is included to HTML document by:

```
<script type="application/javascript" src="path_to_script"></script>
```

or by:

```
<script type="application/javascript">
code
</script>
```

Not only presence and origin of script is checked, but also some concrete functions like *onclick* or *popups* (alert) are checked too. List of JS features is listed in Tab. 2.4.

Name	Type	Description	Justification	Used
JS	binary	Contains Javascript - script tag	Attackers may use browser backdoors or change link destinations	[31, 17, 24, 72, 63, 52, 71, 62]
Scripts	count	Number of scripts		[28]
External scripts	binary	Scripts loaded from external sources <script src=" external_domain" >		[72, 77]
script based	binary	Url features		[72]
Status change	binary	Rewrite staus bar		[72]
Popup	binary	Is alert in JS code		[72]
Onclick	count	Number of onclick events	Onclick events may change link destination	[72]

Table 2.4: JS features.

Spamfilter like [1] result is sometimes used for phishing detection because phishing attacks may have common signs with spam. These signs may be:

- unknown senders,
- huge amount of messages,
- hitting spamtraps (recipients from stolen or crawled [2] database),

and many others. List of spamfilter features is shown in Tab. 2.5

Name	Type	Description	Justification	Used

[2]Crawling is method of gathering data from websites used by search engines by also by spammers which are collecting email addresses

Spam filter output	value	spamscore (spam assasin score)	Phishing may have some signs of spam	[31, 17, 24]
Spam / not spam	binary	spamfilter result		[24]

Table 2.5: Spamfilter features.

Content is second most important property after links. In the content attackers tries to provoke false positive of urgent feeling in the user. Size and vocabulary features are often used. List of content features is shown in Tab. 2.7.

Name	Type	Description	Justification	Used
Email size	value	Message size in KB	Phishing messages may have uncommon size	[63]
Big message	binary	Message larger than 25 KB	Phishing message will probably would be big	[17]
Word count	count	Count of words	Phishing messages may have common word count	[26, 17]
Character count	count	Number of characters in total	Same as words count	[26, 72]
Unique words count	count	Number of unique words		[26, 17]
Vocabulary richness	value	Unique words count / word count		[26]
Has content	binary	Are there any words in the message (Html tags are not counted)		[63]
No characters	count	Are any characters in the message	Same as *Has content*	[17]
Body Richness	vaue	Is message multipart	More opportunities to cheat	[17]
Number of body parts	count	Number of parts in multipart message		[24]
Body parts	count	Number of discrete and composite body parts		[24]

Alternative body parts	count	Number of alternative body parts		[24]
Long words	count	Number of words with more than 15 chars		[52]
Strange words	count	Number of words with at least two of letters J, K, Q, X, Z		[52]
No vovel words	count	Number of words with no vovels		[52]
Starnge words 2	count	Number of words with speacial chars and punctuation, digits at the beginning or in the middle		[52]
Uppercase words	count	Number of uppercase words		[52]
No vovel proportion	count	Proportion of alphabetic words with no vowels and at least 7 characters		[52]
Long words proportion	count	Proportion of alphabetic words at least 15 characters long		[52]
Strange words 3	binary	Word with 3 or more repeated characters in a row		[52]
Structure of greeting	value	Greeting is considered as first line		[26]
Greeting	binary	Does message contains greeting		[63]
Generalization in addressing recipients	binary	Phish mails do not contain personalised data		[26]
Signature	binary	Is signature present at the end		[63]
Attachments	count	Nuber of attachments		[28]

Table 2.6: Content features.

Another part of content checking is blacklisting. Word blacklists are often created from sample dataset with statistics or by hand. List of blacklist features is shown in Tab. 2.7.

Name	Type	Description	Justification	Used
Blacklist	binary	Presence of at least one of listed words (account, update, confirm, verify, secur, notif, log, click, inconvenien)		[24]
Blacklist count	count	Count of present blacklisted words		[62, 26, 17]
Suspension	binary	Presence of word suspension		[17]
Verify account	binary	Presence of verify your account		[17]
Keywords distribution	value	Are keywords close to each other		[26]
Domain specific keywords	binary	Positive blacklist - mail is about domain		[62]

Table 2.7: Wordlist features.

Subject is defined in message headers. Subject is the first thing which user sees, so attacker have to make subject so interesting that it provokes user to open the message. Keywords, re or fwd which refers to reply and froward and subject length are detected in subject. List of subject features is shown in Tab. 2.8.

Name	Type	Description	Justification	Used
Structure of subject	value	text feature		[26]
Is reply	binary	is RE in subject		[72]
Is forward	binary	is FWD in subject		[72]
Number of Words	count	Number of words in subject		[72]

Subject length	count	Number of characters in subject		[72]
Subject richness	value	Unique words count / word count		[72]
Verify	binary	Does subject contains verify		[72]
Debit	binary	Does subject contains debit		[72]
Bank	binary	Does subject contains bank		[72]
subject BL words	binary	Blacklisted words		[62, 49]

Table 2.8: Subject features.

Subject is not the only header field commonly used. Advantage of header fields is that they can be easily parsed because of defined structure. In headers is present **from** field, in which sender claims who he is [46]. This is very important feature for phishing detection. Besides from, priority header is also used. List of header features is shown in Tab. 2.9.

Name	Type	Description	Justification	Used
High priority	binary	Priority not se to normal or medium	High priority is used to invoke stress in user	[52]
To and From	binary	Feature indicating whether the strings "From:" and "To:" were both present		[52]
Priority and content type headers	binary	Are priority and content type present		[52]
Sender length	count	Number of words in sender		[72]
Sender reply diff	binary	Difference between sender and reply to domain		[72]

Sender differs from claimed domain	binary	Difference between sender and modal domain		[72]
Domain sender	binary	Similarity of domain name and message ID		[49]
Unique sender	binary	One sender from one domain (behavioural)		[49]
Unique domain	binary	Domain is commonly used in other domains		[49]

Table 2.9: Header features.

2.4 Evaluation of Existing Methods

Because we chose language unspecific features which are described in Tab. 2.10 we may use public phishing dataset for phishing detection [65] training and testing. We combined this public dataset with Seznam.cz dataset which consists of 228 mostly czech phishing emails which were reported to helpdesk system. For non-phishing email we did not use spamassasin corpus, but randomly picked calibrated [3] data from production classified as "transactional" topic. In total we used 4678 phishing emails and 7954 ham emails. We labeled the data and send them to the classifier.

We decided to use Decision tree classifier [68] because it was used in many successful solutions [18].

We chose features that were used in many solutions. We preferred quantifying features over binary ones, because decision trees have no problems with not normalized feature values and these features may bring more information. We may use many features because if feature is not determining, decision tree should decide to ignore it. We added link protocol features because we believe that attackers do not use secure connection (HTTPS) [37]. And text forms (e.g your password: _____) which are common in plaintext emails. Following features shown in Tab. 2.10 were implemented:

Feature name

[3] Data from production is chosen by defined criteria, anonymized and sent to calibrating system. People (calibrators) read these emails and chose best fitting category from given set for every email. Email is marked as calibrated when 3 calibrators agree on one topic

HTML email
Forms
Tables
Links
URLs to IPs
Link mismatch
Sender domain differ from url domain
Number of dots
Number of dots or slashes
Number of not 80 port
Images
JS
Word count
Character count
Unique words count
Vocabulary richness
Attachments
Is reply
Is forward
Number of subject words
Subject length (chars)
Http links
Https links
Other links (ftp)
Text forms

Table 2.10: Used features.

2.4.1 Testing

For model testing are commonly used precision, accuracy and f1-score metrics. For these metrics we need to define some terms.

- TP - True positive - Positive examples which were classified as positive (correctly).
- FP - False positive - Negative examples which were classified as negative (correctly).

- FN - False negative - Positive examples which were classified as negative (misclassified).
- TN - True negative - Negative examples which were classified as positive (misclassified).

Precision is defined as how many of positive classifications was actually positive. Higher is better. Equation is shown on Fig. 2.1.

$$\frac{TP}{TP + FP} \tag{2.1}$$

Figure 2.1: Precision.

Recall is defined as how many positive samples was classified as positive. Higher is better. Equation is shown on Fig. 2.2.

$$\frac{TP}{TP + FN} \tag{2.2}$$

Figure 2.2: Recall.

Accuracy is defined as how many positive and negative samples was classified correctly. Higher is better. Equation is shown on Fig. 2.3.

$$\frac{TP + TN}{P + N} \tag{2.3}$$

Figure 2.3: Accuracy.

F1-score is defined in Fig. 2.4. Range from 0 zero to 1. Higher is better.

2.4.2 Results

We used 2527 email as test samples and got promising results as shown in Tab. 2.11), where 1 is phishing and 0 is non-phishing:

Precision	0.941558441558			
Accuracy:	0.9525128611			
class	precision	recall	f1-score	support
0	0.96	0.97	0.96	1591
1	0.94	0.93	0.94	936
avg / total	0.95	0.95	0.95	2527

Table 2.11: Learned decision tree test results.

$$\frac{2TP}{2TP + FP + FN} \tag{2.4}$$

Figure 2.4: F1-score.

These results denotes that phishing should be correctly recognized with for 95% of samples with this model.

We were curious which features were important and which weren't used at all. In Tab. 2.12 it is shown which features were used in decision tree and how many times. Importance of feature is not determined by usage count but it gives us picture how the model looks.

Feature name	Occurrences
Vocabulary richness	102
Character count	95
Unique words count	89
Subject length (chars)	72
Word count	58
Number of subject words	54
Text forms	16
Number of dots or slashes	8
Images	7
Links	5
Number of dots	5
Https links	4
Http links	3
Other links (ftp)	3
Forms	2
Number of not 80 port	2
HTML email	1
Tables	1
Sender domain differ from url domain	1
Is reply	1

Table 2.12: Used features usage count.

There are only a few level-1 and level-2 nodes in our decision tree which are based on Number of dots, Vocabulary richness, Subject length and Unique words count. This confirms result from table Tab. 2.12.

We tested this model in production environment described in 2.4.3. Fig. 2.5 shows that model decides about 58% of traffic that it is phishing. Statistically that is of course not true, model suffers with low precision .

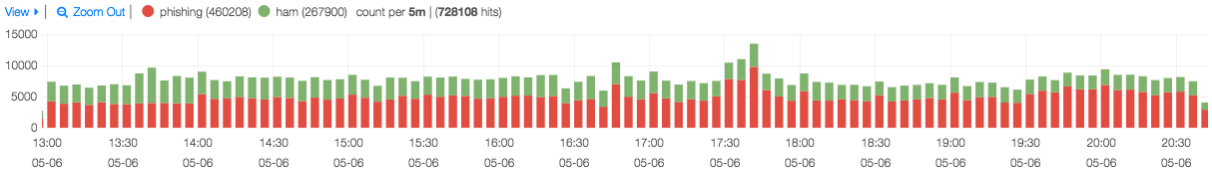

Figure 2.5: Graph shows count of phishing and non-phishing emails based on classifier result. Time is shown on X axis

2.4.3 Production Environment

System was tested in production environment, that means it was used on component which evaluates live traffic at Email.cz. This component process about 50 million emails every day. Our tested system was evaluating emails from representative smaller part of live traffic.

2.4.4 Discussion

Model was promising with testing dataset but it does not work well on live data. What could be the reason of classifying so many messages as phishing? We assume that common traffic email messages evolved and so phishing does. We presume that commonly used features are too unspecific for phishing detection and trained model overfitted them. Other problem is that phishing attacks from 2004 are not representative learning corpus. It may be seen that model is strongly relaying on very unspecific features which may change in time.

We decided not to continue working on this solution and tried to design different solution which will be more suitable for Seznam.cz needs. Our solution is described in next chapters.

Chapter 3

Overview of Our Approach

Because baseline solution was not working as expected we presume that our dataset is not corresponding with reality. It is very hard to get more data which will be representative, so we cannot use machine learning techniques. We decided to split phishing detection into two phases and use manually adjustable scales.

Our approach is based on 2 main steps - **Prefiltering** and **Phishing-detection**. First step filters emails based on traffic statistics and on email content. Second step detects sender domain, and checks links destination. Overall approach is shown in Fig. 3.1. If final score exceeds given limit emails is claimed as phishing. Each step is conditioned by adjustable limit because more precise checks are consuming more performance.

Prefiltering step should reduce amount of heavy computations in subsequent phases because it splits the traffic to suspicious and normal. It consists of two sub-steps. First part analyzes traffic statistics described in 3.1 while the second part is based on message content and is described in 3.3. Each step adds phishing-score to an email. If phishing-score which email got in prefiltering phase is higher than threshold, email is further analyzed in phishing-detection phase.

Phishing is based on identity theft, so we need to detect which entity send the email. Identity detection is described in 3.5. This decision is based on 4 features:

- Domain claimed in header from
- Domain specific keywords
- Common image sources
- Domains mentioned in text

After entity is detected, we need to detect that links contained in email is not common for this entity. Firstly most suspicous link is found, this part is described in 3.9. For all chosen domains most suspicious link is checked. If checked link is commonly used by one of the recognized domains email is marked as non-phishing by significantly lowering phishing-score, if link does not match email is claimed as phishing and overall score is increased. How are common links for domain verified is described in 3.10.

If overall score reaches limit email is claimed phishing. Whole detection process is described in following chapters.

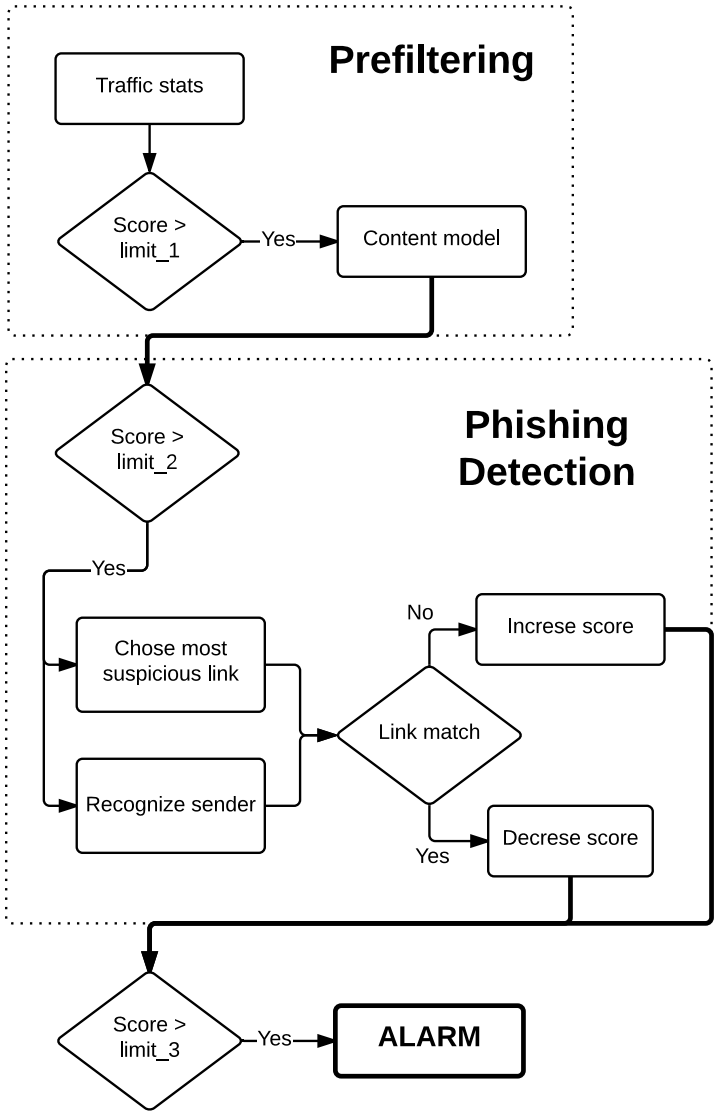

Figure 3.1: Phishing detection proces

3.1 Traffic Statistics

We divided traffic to certainly legit and possible phishing. This division should minimize mistakes and reduce heavy computations to only smaller part of traffic. For this module we decided to use configurable scores for each found property. Features may add or subtract score. This approach works well for many other components used in Email.cz and SpamAssassin works on the same principle. [1] The Fig. 3.2 shows process of email checks described further.

Following checks are mostly based on income statistics, sender domain statistics and defined standards. This approach has to be used because we have no traffic statistics for phishing emails. Domain statistics and standards are reasonable because standards may be enforced and statistics are used because we want to detect non-standard behavior. Complete list of used features is shown in Tab. 3.1.

Feature name	Feature description
Mail - is unique	If mail is unique it may be phishing, but it probably will be very personalised phishing and is very hard and not effective to detect it. This check is based on duplication detection system.
Domain - has reputation	Weather domain have some communication statistics described in 3.1.1.
Domain is young	Determines weather date when we first saw domain is less than a week.
Unseen domain	If domain has no domain reputation described in 3.1.1 is is probable it never send any email to Email.cz users. Domain Reputation is computed every night, so this should apply only for really new domains.
Young domain	This can be phishing feature because phishers commonly register new domain for the attacks. This feature is similar to *Unseen domain* but detected domains already have domain reputation (3.1.1) and are younger than a week.
DKIM	DKIM is described in 3.1.1. We do not consider DKIM signatures for unknown and young domains, because when attacker registers new domain and sign his emails with his valid key we should not trust him.
SPF	SPF is described in 3.1.3. We established same conditions for SPF as we do for DKIM.

Unusual IP	This feature is described in 3.1.1. For each IP field we check if IP from which was email sent is in the common IP list and how many IP are common for this domain. If IP count is low and IP is not in the list we consider this as more anomaly behavior.
Country	Country is based on 3.1.1. For each country field we check if country from which was email sent is in the common countries list and how many countries are in the list. If country count is low and country is not in the list we consider this as more anomaly behavior.
New big sender	This feature is little bit similar to young domain. But it determines that domain started to send many emails in last month. So it may be assumed that is now or it has been hacked.
Old sender	This feature checks if domain has sent more emails than given limit and if these emails were sent in longer period, approximately 2 months. This feature decrease the score.
Old sender spf	This feature is based on *old sender* and it adds SPF check to it, if SPF matches it is considered as stronger version of *old sender*.
Transac sender	This feature determines based on domain statistics whether it sends mainly transactional emails.
From mismatch	From mismatch is conditioned by different envelope and header from. This also occurs often for domains which are sending emails from webhosting companies so it is not strong feature. Attacker also can change his envelope from is he is sending from server with full access.
Phisheable	Determines if header from is in phisheable list described in 3.2.
Is similar	Is conditioned by young domain or unseen domain. Similar domains are calculated from Levehenstein distance [76] of header from domain and all phisheable domains (described in3.2). This feature should determine if attacker is trying to confuse user with domain similar to that which user knows.
Email topic	Topic is described in 3.1.5. We picked topics which are probable to be attacked and topics which are not.
Abroad	Is determined by the origin of IP. If IP is not from Czech Republic we are more careful about it.

Txt form	Some phishing attacks use simple plain text forms to avoid necessity of links usage. This feature determines whether email contains this form. This form is commonly used in plain-text emails and looks like paper form. It may look like: "password: _____". We assume that it uses field name some, separator for example ":" and than line showing where should content be written highlighted with dots, dashes or underscores. We used regular expression for finding these forms. To consider form as present we have to find at least two occurrences.
Adscore	When email has signs of ad we count adscore similarly as phish-score. Phishing may have some signs of ad (mainly is sent in many copies). We add adscore multiplied by coefficient to phish-score.
Keywords	Keywords is described in 3.1.4. When any keyword from given set appears we consider email as suspicious.
Subject keywords	Same as keywords but used on subject.
Attachments	When email has some attachments it should be sent for further examination.
Unsubscribe	We use existing custom made unsubscribe links detection. If this link is detected we believe that email if probably not phishing.
Phish form	We discovered from the dataset that attackers use also other kind of forms which is not so easily discoverable. They do not fill space for the content with anything. We decided to detect these forms based on keywords followed by separator. To consider this type of form as present we again have to find at least two occurrences.
Href camouflage	Occurs when domain in *href* attribute is different from visible domain.
References	This feature is based on email headers, concretely on references field which determines if email is reply. References field is list of message IDs which were replied to and grows with each reply. This feature decrease score and was added because some replied and forwarded emails were misclassified.
In reply to	This feature is based on email headers, concretely on in-reply-to field which determines if email is reply. This feature decrease score and was added because some replied and forwarded emails were misclassified.

Long mail	This feature determines if email is longer than one standard page because it may be presumed that phishing will be short.
Many emails	This feature detects plain email addresses in text. This feature supports *references* and *in reply to* features.

Table 3.1: Traffic statistic features.

Some more complex features are described in following chapters 3.1.1, 3.1.2, 3.1.3, 3.1.4 and 3.1.5 in more detail .

3.1.1 Domain Reputation

Domain reputation is statistics based on how domain behaves. Domain is extracted from envelope from and stripped to its publicsuffix. [64] For this work we take advantage of domain common IP, country and time when was domain firstly seen. For common IP we observe common IP for all emails, common IP for transactional emails (more info in 3.1.5) and common IP for DKIM signed (more info in 3.1.2) emails. For common countries we observe common country for all emails and common country for DKIM signed emails. Time when domain was firstly seen is determined by time of first email delivered from that domain. Domain reputation is calculated every night. Each feature is checked only if it reasonable, so DKIM IP is checked only if email is DKIM signed and transactional IP is checked only if email is classified as transactional.

3.1.2 DKIM

DKIM was presented in 2009 in rfc5585 [35]. Whole DKIM verification process is shown on Fig. 3.3. Sender creates an email with DKIM signature in format defined in rfc6376 [36]. Message is received and after headers are parsed DKIM signature is verified against TXT record on DNS server for domain specified in signature. Message may be signed with more than one signature. DKIM only ensures that sender has key for signing domain that he climes in DKIM header but it may have have nothing in common with domain provided in from.

On Fig. 3.4 on page 36 is shown how we used DKIM signatures. If domain is not signed at all we add score, but more importantly when it is not signed and domain often sends emails signed we may add high score. If DKIM signature fails to verify we add score, but when it often has valid signature we may again add higher score. When DKIM signature is valid we check is it matches with envelope or header from. If it matches we may subtract high score. If it doesn't match we again check if this situation is often and add appropriate score.

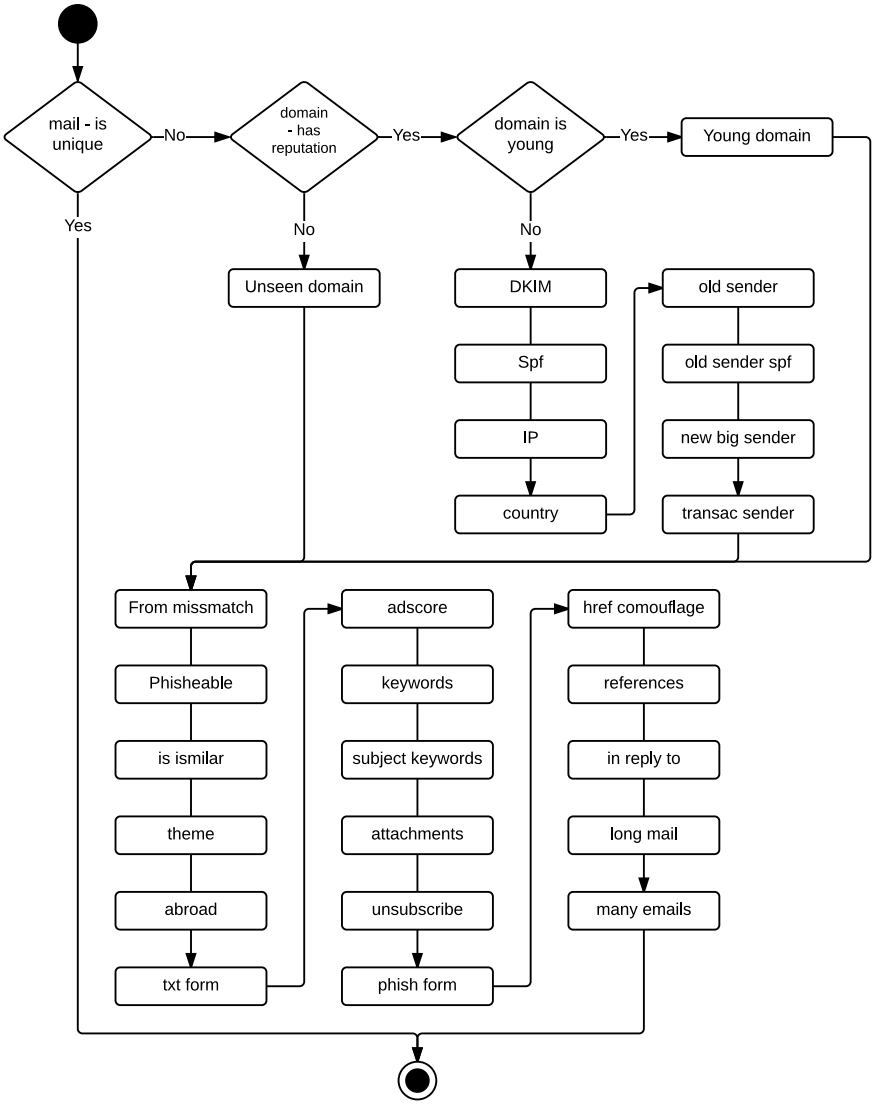

Figure 3.2: Phish score calculation process

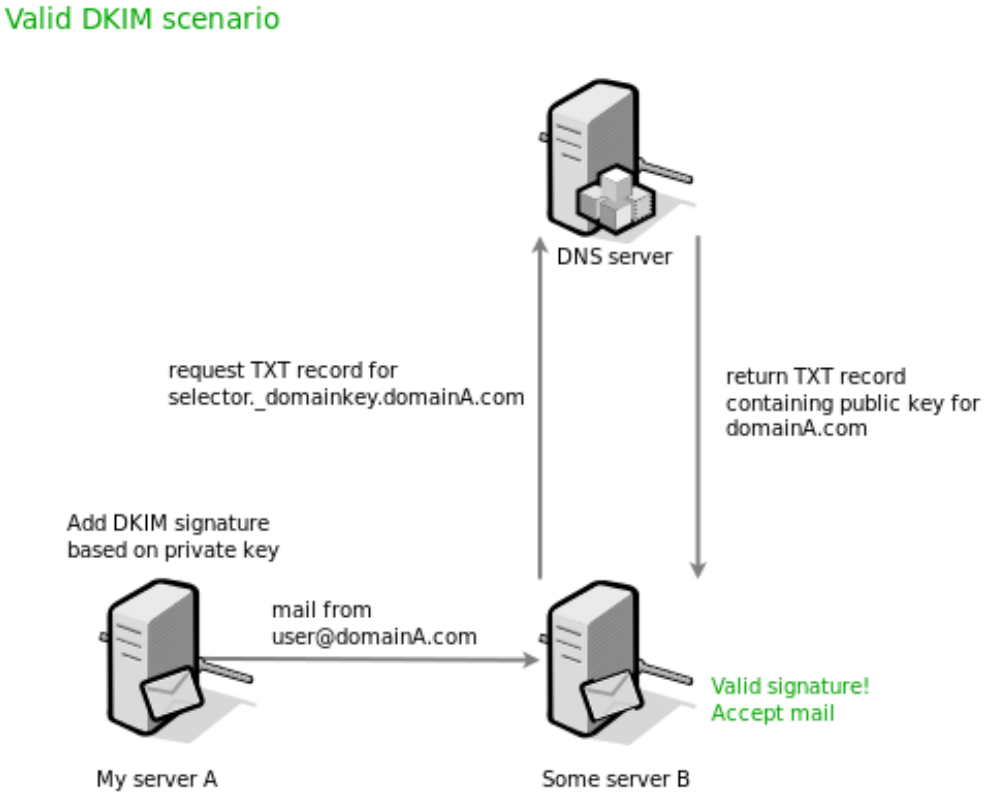

Figure 3.3: DKIM verification process.
Source [3]

3.1.3 SPF

SPF was presented in 2006 in rfc4408 [47] and updated by rfc7208 [51]. SPF is based on TXT record on DNS server, which may look as follows:

$*.\mathrm{EXAMPLE.COM.} \quad \mathrm{TXT} \quad \text{"v=spf1 a:A.EXAMPLE.COM -all"}$

Process of checking SPF record is depicted on Fig. 3.5. This process is similar to checking DKIM described in 3.1.2 but in the message is no signature, only senders IP is checked. Each domain owner can set up SPF record for his domain, in this record is provided list of whitelisted servers specified by IPv4, IPv6, A, MX or other methods. It also specifies how to behave to others not in the list. There are four possibilities described in Tab. 3.2. [15]

Sign	Name	Description
+	pass	Allow to all
-	fail	Deny to all, suitable for banks
~	softfail	Allow and mark

?	neutral	Allow to all, used by freemails

Table 3.2: SPF operators.

SPF domain in taken from envelope from. Email with SPF status fail is thrown away. Other statuses cannot determine phishing. We only decrease score if SPF is present and status is pass.

3.1.4 Phishing Keywords

As told before phishing is based on social engineering. Social engineering is based on feelings, mainly stress, which may be provoked by messages calling for immediate action. These messages may be recognized by usage of certain keywords. We wanted to check for presence of these keywords in prefiltering phase so we have to find phishing specific keywords. We used TF-IDF for finding these keywords. [69, 16] Generally we used similar method as we do in 3.6 with all phishing emails as one domain. For that we used domain dataset described in 3.6.1 merged with each part of phishing dataset described in 3.3.1. We also tried to merge all phishing datasets to one dataset. We computed TF-IDF using equation (3.6). We assumed that phishing specific words should have bigger TF-IDF than commonly used words.

For public dataset there were mostly emails in English language. So we downloaded english Wikipedia text content and used it for better commonly used english words distribution, because in our dataset contained only a few english emails.

As phishing keywords were classified for example these Czech: obnovení, účet, potvrdit, heslo, aktivace, pozastavit and přístup and these English words: login, secure, account, upgrade, verify, validate and password.

3.1.5 Topic Categorization Model

We are using topic categorization model in production. This model is based on keywords for each topic. Data for model training is gathered from calibrated emails. Multinomial linear SVM is used as model for this problem. More detailed description of this problem is over the scope of this work.

3.2 Phisheable

It is reasonable to attack only on big companies where people have their important personal accounts like banks, government, e-shops and other well known companies. This is true only when people are following basic security rules and doesn't have same password for many services. But how to find these companies which are having interesting data for the

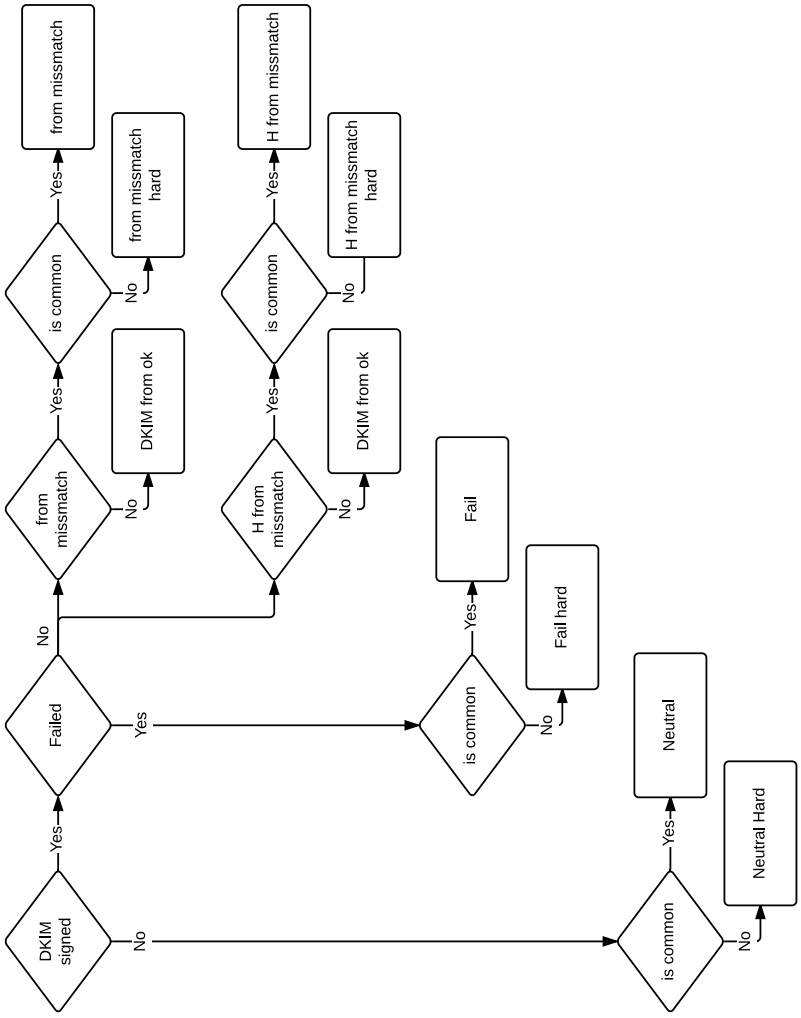

Figure 3.4: DKIM score calculation process

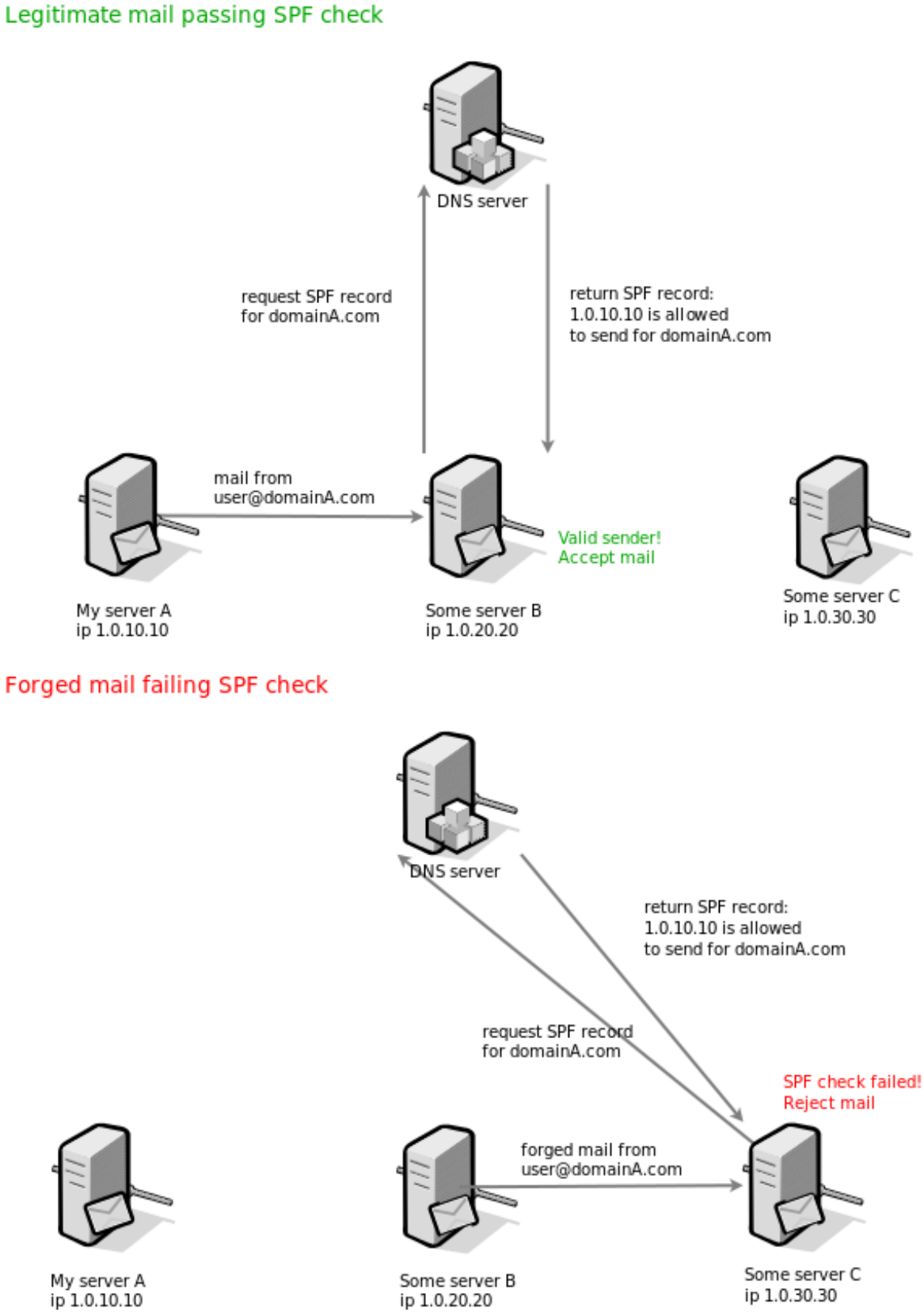

Figure 3.5: SPF verification process.
Source [3]

$$TF \times \frac{DC}{DF} \tag{3.1}$$

Figure 3.6: Where TF is term frequency (How many times word appeared in one document) and DF is document frequency (In how many distinct documents word appeared) and DC is number of documents in dataset

attackers? We picked a set of domains which we labeled "phisheable" based on two rules. For both rules we computed data on last two months.

1. Domains which are sending huge amount of emails with topic *transactional* or *account statement* sorted by number of incoming emails.

2. Communicative domains are defined as $IN > 1000$ and $IN < 5 * OUT$ where IN is amount of emails we received from that domain and OUT is amount of emails we sent to that domain. We assume that people are sometimes communicating with phisheable domains. If domain fulfills these conditions it is added to communicative list, which is sorted by number of incoming emails.

We took 2500 domains from each category and got 4527 phisheable unique domains.

3.3 Email Content Model

We believe that model described in 2.4 overfitted our dataset. We decided to clean up our dataset, gather more examples and pick better features. These steps are described in following chapters.

3.3.1 Phishing Data Sets

For training and testing email content model we used data sets described in Tab. 3.3.

Name	Description

Custom (Seznam.cz) dataset	We hand picked our data set for phishing emails only. We also added some actual phishing emails. And we contacted server Hoax.cz which gathers hoax (Hoax is email containing false information which is attacking to feelings or threatening users and encourages to be shared with others. This activity is gathering new email addresses for attackers.) and phishing emails. [4] They sent us their whole phishing database (about 49 emails). As data were examined some duplicate emails were found, so we did a script which calculates MD5 hash based on the file content and saved file with this hash as a name to new location. This datata set size growed to 115 emails.
Public data set	Public datat set is described in 2.3.3 on page 10.
Phishtank data set	We wanted more data which are also more actual so we downloaded known phishing links form database called phishtank. [8] We got 26918 links. Than we went trough 2 years of historical data and downloaded every unique email which contains one of these links. Which resulted in 1130 emails.

Table 3.3: Phishing datasets.

3.3.2 Non-phishing Data Set

For training model we used emails classified to "transactional" topic by human calibrators. This datset consist of 10128 emails in czech language and should talk about registrations or things connected with accounts. We chose this dataset because these emails are similar to emails picked by traffic statistics.

3.3.3 Used Features

We examined each used feature described in 2.4 and how much does it appear in positive and negative data set shown in B. Than we thought about new features, based on previous experience gained from the results. We immediately tested each invented feature which is also shown in B. For training the model we chose only that features which differs in phishing data set and non-phishing data set. When choosing new features we preferred features which have something in common with links, because links are one of the key features for phishing success.

New features are described in Tab. 3.4. We needed new features because some of features used in baseline model were differentiating phishing and legit emails in dataset

but nowadays are not commonly used for phishing attacks, based on [22, 21]. For example IP addresses were used only in and 0.86% non 80 port in 0.56% of detected phishing emails.

Some features were evaluated with new features again because, we fundamentally changed the detection method. For example *Number of links*, or *text based forms*.

Global links	This feature shows how many links are pointing to global domains 3.10.2. We presume that non-phishing emails will be pointing to global domain more than phishing emails.
Num of params	We wanted to know if non-phishing emails has more link parameters than non-phishing ones.
Num of phish subdomain	We noticed that attackers put original domains to subdomains of phishing links to fool victims. We search for detected domains in subdomains.
Num of phish path	This feature is similar to *numOfPhishSubdomain*, but it is checking for detected domains in path.
Num of domain phish	Some attackers create domains which contains phished domain in its name. This feature is similar to *numOfPhishSubdomain*, but it is checking for detected domains in domain itself.
Num of param phish	This feature is similar to *numOfPhishSubdomain*, but it is checking for detected domains in link parameters.
Num of black sheeps	Some phishing emails are pointing to attacked domains too. We presume that non-phishing emails will be pointing to only a few domains, most commonly only to itself. This feature shows how many unique domains is targeted in email.
Num of sheep link global	This feature is similar to *numOfBlackSheeps* but it does not count links pointing on domains listed in global links list described in 3.10.2. Link pointing to global domains are omitted because many other domains point to these domains, so it may be presumed that they are not hosting phishing.
Num of black sheeps phish	This feature is similar to *numOFSheepLinkGlobal* but it counts only links not pointing to domains detected in claimed domains 3.8.
Phish keywords	This feature is detecting how many phishing keywords is present in email content.
Subject phish_keywords	This feature is detecting how many phishing keywords is present in email subject.
Txt lin	This feature shows how many times text links were used. Text links are extracted from text which contains no HTML tags.

Text link with path	Sometimes attackers use links only in text for to hide before spamfilters. We are detecting this behavior with text links detection and than checking which links have non-empty path.
Phish keywords ratio	This feature shows ratio of phishing keywords to all words.
Subject phish keywords ratio	This feature shows ratio of phishing keywords in subject to all words.
Txt and href	We presume that non-phishing emails admit their link destination, which is detected, by link destination is also present in text.
Short	Phishing link may be hidden with url shortener, we have list of url shorteners and we are detecting if domain is found in this list.
Link miss close	This feature is based on idea that link target may be camouflaged and camouflaged domain may be similar to visible domain.
Fake https	This feature is similar to *link_miss_close* but it detects if link visible part is pointing to secure HTTP protocol, but real target is not secured.
Freehost	Attackers commonly use freehosting companies to host phishing websites. We used a list of freehosting domains to detect this behavior. The list of freehosting domains is not part of this work.
Tld in path	This feature is inspired from *numOfPhishPath* but it searches only for TLD presence. This search is based on TLD list originated in Seznam.
Www in path	This feature is very similar to *tld_in_path* but it checks for "www" in path.
Num of supicious IP links	This feature determines how many suspicious links (described in 3.9) are pointing to IP address instead of domain name.
Txt form	Updated feature
Phish form	Updated feature
Num of links	Updated feature - this feature counts all links in email based on internal system.

Table 3.4: Content features.

3.4 Classification

We decided to use binary classification similar to classification used in baseline solution described in 2.4. We used decision trees classifier again. For preventing problems encountered with first classifier, we tried to grow positive dataset with more actual data described in 3.3.1. We added new features which were based only on email content 3.3.3. For each feature we tested its appearance in both positive and negative dataset. We chose only features which differed significantly in between the datasets. Each feature was also tested alone and by feature evaluation methods.

Because our dataset is not large we decided to use 10-fold cross validation [56]. This method splits dataset to 10 parts and and runs 10 times. In each iteration algorithm separates one part, which was not yet tested, for testing and rest of the dataset for training. Advantage of this method is that we can use whole dataset for training because we have tested the model in cross validation. Another advantage of cross validation is that we can measure how stable the classifier is, because it is trained and tested on different data every time. Unstable classifier will more likely have different accuracy in each test.

We tested which minimal leaf size is best for used decision trees model. Result of test is shown on Fig. 3.7. We chose minimal leaf size 2 because it was more stable than slightly better size 1. We also tested higher values (10, 20 and 100) which were always worse than 2 and not significantly more stable.

List of used features is shown in Tab. 3.5, for each feature we also tested its accuracy without other features.

Feature name	Accuracy	Variance
Contains form	0.6414	(+/- 0.0039)
Num of scripts	0.6465	(+/- 0.0107)
Num of phish forms	0.6451	(+/- 0.0055)
Num of phish keywords	0.7827	(+/- 0.0170)
Subject contains phish keywords	0.6499	(+/- 0.0051)
Freehost	0.6464	(+/- 0.0061)
Short	0.6399	(+/- 0.0026)
Num of supicious ip links	0.7257	(+/- 0.0099)
Num of non 80 port link	0.6534	(+/- 0.0061)
Num of subdmain phish	0.6695	(+/- 0.0145)
Num of path phish	0.6706	(+/- 0.0493)
Num of domain phish	0.6439	(+/- 0.0055)
Num of param phish	0.6412	(+/- 0.0083)
Tld in path	0.6414	(+/- 0.0156)
Www in path	0.6542	(+/- 0.0162)

Num of dots link	0.7741	(+/- 0.0218)
Num of slashes link	0.7023	(+/- 0.0436)
Num of black sheep link	0.6861	(+/- 0.0138)
Num of black sheep link phish	0.6673	(+/- 0.0141)
Num of black sheep link global	0.7448	(+/- 0.0547)
Num of sender diff url	0.7387	(+/- 0.0106)
Next link with path	0.6589	(+/- 0.0061)
Txt and href	0.6372	(+/- 0.0024)
Fake https	0.6972	(+/- 0.0132)
Camouflage	0.7502	(+/- 0.0314)
Num of https link	0.6730	(+/- 0.0214)

Table 3.5: Used content features evaluation.

We also did feature evaluation method called Leave-p-out cross-validation. In this method 1 to p features is omitted and classifier performance is measured. If feature was not good classifier performance should be higher. We did this evaluation for p = 1 and p = 2 and we did not discovered any features which should be omitted. Results of these test are accessible in Appendix A.

Final classifier used for email content model has accuracy 0.9525 with variance +/- 0.0203.

3.5 Attacked Domain Detection

Key for phishing is to make victim think that phishing mail was sent from legit company. So for phishing detection we need to detect that somebody is pretending he is someone he isn't. Firstly we need to to detect that email look like it has been sent from from known company. We created a list of domains which are reasonable to attack described in 3.2 which we want to be able to recognize.

We developed four methods for recognition which domain is talked about. Fist method is based on keywords. Second method is based on source attribute of image tags. Third method is claimed domain in header from and fourth is domain mentioned in text. All methods are described further.

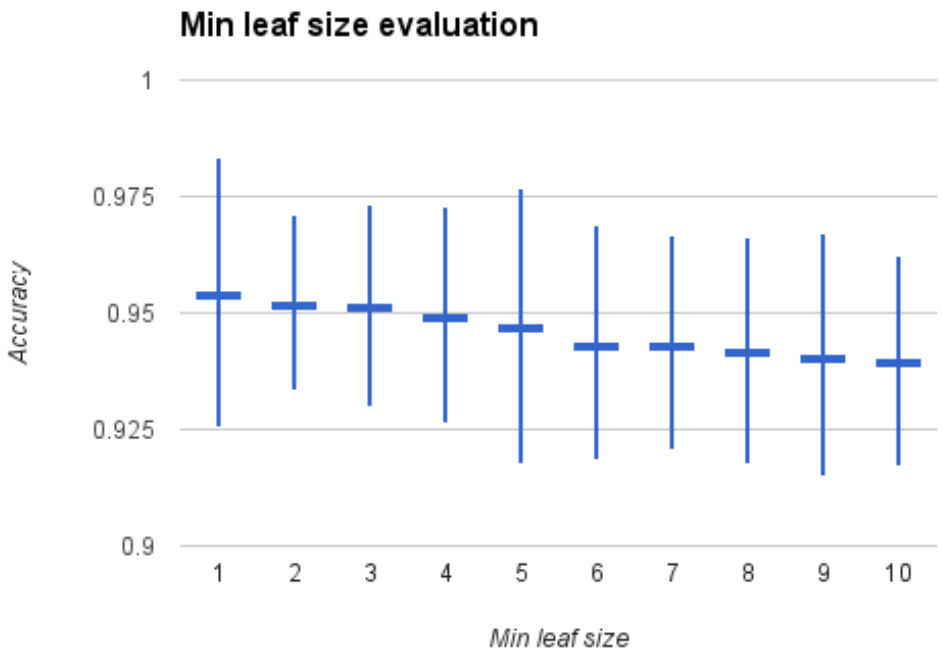

Figure 3.7: Min leaf size evaluation

3.6 Domain Specific Keywords Recognizer

Our approach is based on assumption that each entity determined by domain have some commonly used words like company name and company slogan. We found these words with NLP technique based on slightly modified TF-IDF described in 3.6.3 and than we trained model for recognizing which domain is email talking about.

3.6.1 Dataset

We picked email examples from one month with given conditions. These conditions were that we consider only unique emails (based on custom hashing), time difference between two emails from one domain has to be at least 30 minutes and domain has to sent at lest 100 emails.

Dataset statistics:

- 44 000 domains
- 166 GB
- $\sim$ 10 000 000 emails

Process of downloading these picked emails took almost one week. We keep these emails in following structure: domains as folders and emails as eml files kept in appropriate folders.

We forged these eml files to one file because of performance containing one line for each email. On each line is tabulator separated domain and all words contained in email. We extracted words from message body by that we splitted the text with all non-character and non-digit words

We also preserved

3.6.2 Cacheable

As part of this work *Cacheable decorator* for python was developed. [9, 10] This decorator is based on pickle, it stores result of function call and if function is called with same parameters it loads newest result and returns it. This behavior is useable for functions which loads big amount of data and does heavy computations. Code of this function was open-sourced and licensed under MIT license published on https://github.com/tivvit/python-cached-decorator.

We used cacheable decorator for functions in pipeline for testing various models, because it consists of more parts which are used many times.

3.6.3 Mf-idf

For computing domain representative words we used slightly modified TF-IDF. [69, 16]

Our approach is based on that domain specific keywords should be present in many emails from domain but should not be present in other domains. We called this metric MF-IDF. For each domain we sorted all words by MF-IDF and chopped the tail because of data size reduction. Computation of MF-IDF is shown on Fig. 3.8.

$$MF \times \log \frac{DC}{DF} \tag{3.2}$$

Figure 3.8: MF-IDF where MF is mail frequency - in how many emails from domain word appeared, DC domain count - how many domains are in dataset, DF - domain frequency - in how many domains word appeared

we also tried TF-MF-IDF which does not work so well. Computation of TF-MF-IDF is shown on Fig. 3.8.

3.6.4 Computation

Because data is huge, we computed MF-IDF on Hadoop. [19] Hadoop is distributed system for processing large data sets based on map reduce model. [29]

Mapreduce is based on parallelization. Data is splitted to smaller chunks which are processed on cluster nodes. Each node is doing map phase in which is data formed to

$$TF \times MF \times \log \frac{DC}{DF} \tag{3.3}$$

Figure 3.9: TF-MF-IDF where TF is term frequency - how many times word appeared in domain mails, MF is mail frequency - in how many emails from domain word appeared, DC domain count - how many domains are in dataset, DF - domain frequency - in how many domains word appeared

key-value structure. After each map phase is optional aggregate phase which is preparing data for reducers. In reduce phase is data sorted and combined together by key.

3.6.5 Classes

For this task we are classifying to many classes. Firstly we thought that we may train the model to classify every domain in the data set but it cannot be trained on machine with more than 100 GB of RAM because model training is consuming huge amount of memory and trained model probably wouldn't be precise. This was main reason to create phisheable list described in 3.6.6.

For correct classification we have to gave the model opportunity to say this domain is unknown. For recognition of this situation we took random emails from not phisheable data set represented by of one third of amount of all phisheable emails and labeled them as *others*.

3.6.6 Phisheable

As described in 3.6.5 we needed to pick domains which are reasonable to attack on. We figured out that some domains in the phisheable dataset have less than 10 words in word data set, so we decided we will not be using them any further, so we ended up with 3600 phisheable domains.

3.6.7 Vectorizer

For NLP are commonly used separated words. Separating of these words is referred as vectorizing. [57] We had words for each domain prepared from MF-IDF method described in 3.6.3. For our purpose may be used *Count vectorizer*and *TF-IDF vectorizer*. Both vectorizers are capable to learn the dictionary from training data set. We prepared the dictionary in MF-IDF step, so we provided this dictionary to vectorizer directly.

We tested both vectorizers in standard conditions described in 3.6.9. As variable we chose number of words for each domain because it directly affects which words will be chosen and how effective will be the system overall. On Fig. 3.10 is shown that both methods works similarly for one word. TF-IDF raises a little bit to 4 words and than starts to fall, count vectorizer falls immediately.

Is is reasonable that only 1 word works best for count vectorizer because that word is in most times company name.

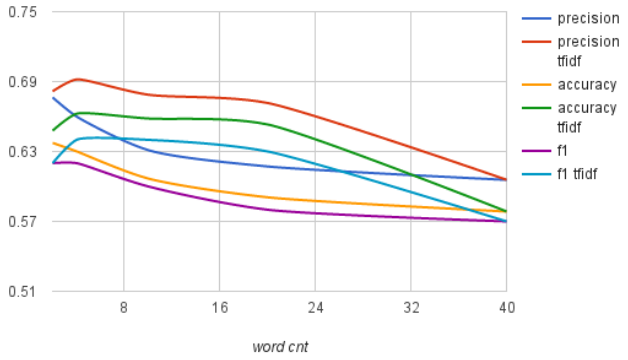

Figure 3.10: Testing vectorizers for domain keywords recognizer

3.6.8 Number of Emails

We tested how number of emails affects performance of the classifier in standard conditions described in 3.6.9. We have many emails so why not use them all? Because the speed of all components raises rapidly. Training classifier on 100 emails took many hours. On the figure Fig. 3.11 is also shown that with number of emails performance grows approximately logarithmically so it will be not reasonable to use much bigger amount of emails. We set the step to 5 emails.

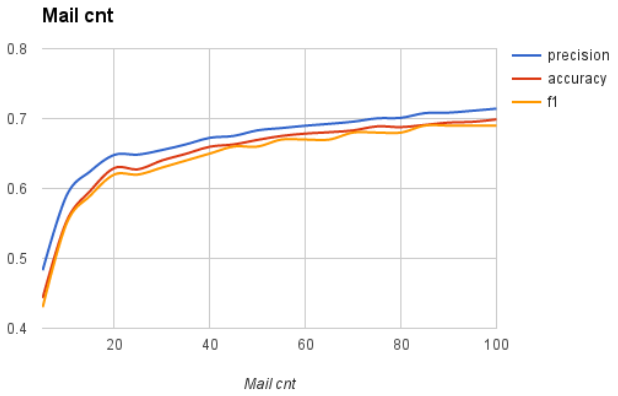

Figure 3.11: Testing how number of emails affects classifier performance.

3.6.9 Model Testing

We wanted to use python-based freeware implementation of mathematical models, so we chose SciPy because it is very popular and it is installed on production machines. [12]

Firstly we set standard conditions. We used 15 emails (because of speed), 10 words and count vectorizer. When we are not testing the models we were using Multinomial SVM. We splitted data set to 80% for training and 20% for testing.

When testing which model is best for our purpose we tested Multi-class SVM [58], Multinomial Naive Bayes [59], SDG [61] and random forests [60]. Results are shown in table Tab. 3.6

Model name	F1-score
Multi-class SVM	0.59
Multinomial Naive Bayes	0.41
SDG	0.50
Random forests	0.56

Table 3.6: Test results for various models used for domain keywords recognition.

Best overall score that we achieved was with multi-class SVM and 100 emails. F1-score for this test was 0.71.

3.6.10 Classifier

We wanted to train multi-class classifier for this purpose, but for every tested classifier it was problem to handle this amount of classes. Because of not good results in test and more importantly performance issues even on smaller amount of classes, we decided to use slightly different approach. Our approach is based only on MF-IDF described in 3.6.3. We built inverse dictionary which holds for each word top 10 domains and their MF-IDF score for that word. With this approach we may take advantage of statistics data computed on huge data set. This dictionary serve as data source for Ranking classifier described in 3.7.1. This data set contains 143610 words from 32646 domains.

3.7 Domain Image Sources

Many phishing attacks use images hosted on attacked website. We wanted to take advantage from this behaviour, so we used method known as link revert. We calculated for each image source attribute on which domain does it point and paired it with domain determined in header from. We came up with structure where key is domain where image is

hosted and it contains structure of domains which used image from that source sorted by how many times image was linked from that domain. We normalized count of occurrences between 1 and 0 and use that as data source for rank classifier described in 3.7.1.

In this step we noticed that for some domains like *paypal* or *amazon* we have too precise data. Too precise in the way that they appear in many countries and they use many TLDs. [34] But we don't care about from which country was email sent but which company sent it. Based on this assumption we throw away TLD in all link based metrics.

3.7.1 Rank Classifier

We needed classifier which can work with precomputed data and classifies to many categories based on occurrences and their score in the data set because we had suitable data in logs processable on hadoop, but it would be difficult to get raw data for learning classic classifier. This classifier works on very simple statistics. Each entity has score between 0 and 1 and each sample may hit one or more entities, because entity score is normalized each occurrence may be added to entity final score. At the end entities are sorted based on score from the highest. Algorithm main part is shown in 3.12

```
result = {}
for i in occurrences:
  for entity, score in statistical_data[i]:
    if entity in result:
        result[entity] += score
    else:
        result[entity] = score

sorted_result = sorted(result.items(), key=lambda x: x[1],
   reverse=True)
return sorted_result
```

Figure 3.12: Main part of rank classifier.

3.8 Claimed Domains

In header from may be anything what sender wants. So when attacker wants make victim believe that he is somebody known for the victim it is reasonable to introduce in header from with its name.

Second type of claimed domains is similar to keywords method. It is based on text links without path used in email. It is common to use domain name in greeting or in footer. We detect text links with regular expression and than check if TLD exists.

3.9 Suspicious Link Detection

For phishing detection is key to detect which link is link is pointing to phishing website. For this detection we used features which were also used in Email content model described in 3.3.3 and were related to links. For detecting which link is most suspicious to lead to phishing website we simply counted how many positive and negative phishing features each link has. For further evaluation we chose link with highest number of positive features lowered by number of negative features. All used features are shown in Tab. 3.7.

Feature name	Positive / negative
Global links	-
Number of parameters	-
Is HTTPS	-
Text and href	-
Number of Phish Subdomain	+
Number of Phish Path	+
Number of domain phish	+
Number of Phish param	+
Number of black sheep links	+
Number of black sheep links phisheable	+
Number of black sheep links global	+
Text link with path	+
Shortener	+
Fake HTTPS	+
Freehost	+
Camouflage	+
TLD in path	+
Www in path	+
IP link	+
Non 80 port	+
Contains keyword	+

Table 3.7: Features used for suspicious link detection. Where + means positive (more suspicious) and - means negative (less suspicious)

3.10 Link Statistics

Second most important thing for phishing success after text content is to get link pointing to phishing website to email. We computed which domains are commonly linked form domains and than we can check if send link is common.

3.10.1 Common Links

We gather links into three categories. Links mentioned in text, links contained in href part of a tags and links contained in src part of img tag. In further text we refer these types as txt, href and src links respectively. For each type we computed per each domain referred in header from most linked domains. Than we sorted each category for each domain by most linked domains and took only top 20% shown in Fig. 3.13.

$$max - min \times 0.2 \tag{3.4}$$

Figure 3.13: Where max is maximum count a min minimal count, 0.2 is percentage limit.

We assumed that we would need these stats for all 3 types of links but we discovered that src links are much more suitable for detection who sent the email described in 3.7. And also for txt links which may also show which domain is attacked and they also may indicate suspicious behaviour described in 3.3.3. Because of that we ended with only href links in common links data set.

3.10.2 Global Links

We wanted to know which domains are commonly linked in emails, because we may ignore these links as probably not phisheable. Firstly we run over all emails delivered in last month and gathered 1136374 domains and how many times they were linked. Than we sorted these domains by how many times it was linked and took top 10% (shown on Fig. 3.14) from the top of the list, which resulted to 1497 domains.

$$max - min \times 0.1 \tag{3.5}$$

Figure 3.14: Where max is maximum count a min minimal count, 0.1 is percentage limit.

We run trough this data set and discovered that it is not representative for our purpose. We again go trough all the emails from the last four months and for each linked domain we calculated from how many domains with low spamscore it was linked. We again took 1500 most linked domains and mark them as global links.

3.10.3 Malicious Link Detection

Our main goal is to detect whether most suspicious link is pointing to domain which claimed domain commonly use or if it points to unknown location. Global links are taken as legit. If data for domain are present and domain matches one of commonly linked domains, link is marked as legit. If we have no data for the domain, which means that domain is not in phisheable list, we check if domain matches from domain and whether is not younger than one week and is not freehosting (described in 3.1). Last check is based on matching link domain and header from domain, when this domain is classified as old sender (also described in 3.1). If link does not match any of these conditions, email is classified as phishing.

3.11 Final Decision

Final decision is based on overall score which was generated by modules described before. Phishing email has to get trough traffic statistics filter, which checks for suspicious behavior like non matching signatures and domain statistics. Than it has to get trough content model, which decides if email is phishing or not based on HTML and text features. Last step for positive phishing mark is that it has to fail test for claimed domain common links. Claimed domain is based on keywords, text links, header from and image sources. Domain common links are based on statistics which domains are commonly linked in emails from checked domain. Last check is done on most suspicious link, based on set of rules. Link check based on domain common links is the strongest feature because known companies, are not changing their behavior quickly and because of the pre-filtering phase which should separate legit traffic already.

We set the system to rather have bigger recall than smaller precision. We decided for this this because we want to detect as many phishing messages as possible. We wanted high recall because of creation of phishing emails data set which will be used for improving this detection system. Another reason was that human is checking the result of this system and may take appropriate actions to prevent detected messages to be delivered to user inbox. After long term evaluation of correct classifications made by this module it will be used to inform users directly, or revoke to deliver the messages.

Chapter 4

Main Results

4.1 Traffic Statistics

We implemented traffic statistics because we wanted to split traffic to normal and suspicious. As may be seen on Fig. 4.1 , which is screenshot from production Kibana used for visualization of logs, this module separated approximately 90% of traffic as normal (phishing-score zero and less, which means email is not phishing with high probability). [6] On the image we could also see that spamicity does not correlate with phishing score and that most messages haven't been hit by any rule which is all right because many messages are personal messages.

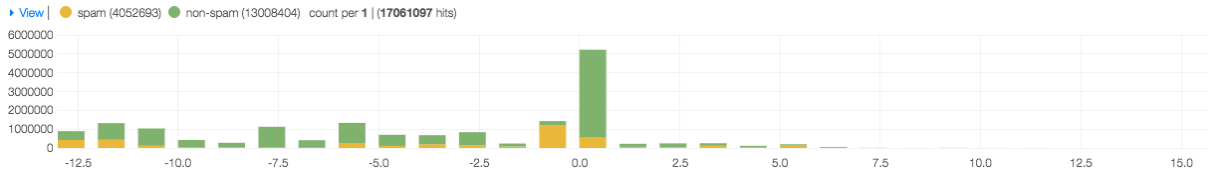

Figure 4.1: Phishing score based on traffic stats. Green is inbox and yellow is spam. X axis shows phishing-score, Y axis count of occurrences

On second image Fig. 4.2 it is shown that only very few messages has very high score.

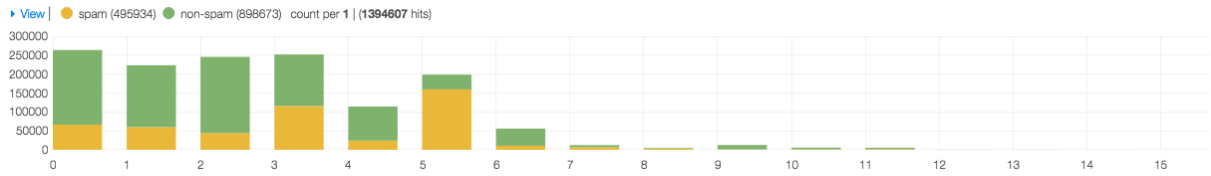

Figure 4.2: Phishing score based on traffic stats more than zero. Green is inbox and yellow is spam. X axis shows phishing-score, Y axis count of occurrences

Some features which were not working as supposed were discovered on live data, these features are described in 4.1.1, 4.1.2 and 4.1.3.

4.1.1 Text Forms

First version of text forms detector was checking for at least four subsequent dots, dashes or underscores. This check has low precision, so we came with more precise detection. First check finding text forms is shown on Fig. 4.3.

```
r"\w{3,20}[ :=\(\);]{1,3}[\.\-_]{4,50}"
```

Figure 4.3: Text form regular expression.

Second check is even more precise and checks for keyword presence. Keywords were picked based on testing dataset which is in A .

```
ur"((?i)login|pass|jmen|name|hesl|uziv|id|user|mail|add?res)+\w
    {0,16}\W{0,4}[\n\-_\.]+"
```

Figure 4.4: Phishing form regular expression.

Problems with these two methods are that text form also matches some separators and the phishing form finds also successful registration summaries.

4.1.2 Phishing Keywords

Phishing keywords data set contained also very specific words like company names. We go through all keywords and picked the final data set by hand.

We are checking for keywords not only in body but also in subject. For each type we have separate configuration which is based on rectangular pulse function which gives us ability to react differently when more matches are found.

4.1.3 Phisheable

In production was soon discovered that phisheable data set contains also freemail domains. These domains were discarded from the list because other freemails won't probably be attacked on email.cz servers and freemails has no specific signs from which they can be recognized, described further in 3.6.6. Freemails were discarded based on freemail list which creation is beyond the scope of this work.

4.1.4 Results

In table Tab. 4.1 are shown sorted features from the most occurred. Feature "zahranici" is most common because testing was done only on email sent from abroad. This statistics is measured on four days.

Feature name	Number of occurrences
ps-zahranici	24458650
ps-spf_ok	19957230
ps-not_theme	16773506
ps-dkim_from_ok	15577703
ps-dkim_header_from_ok	15319983
ps-unsub_link	15024078
ps-phisheable	14686288
ps-from_mismatch	10092760
ps-phish_keyword	6677077
ps-theme	5785920
ps-dkim_header_from_mismatch	4299764
ps-dkim_from_mismatch	4030460
ps-dkim_neutral	2369137
ps-phish_form	2260616
ps-unseen_domain	2252887
ps-attachments	1464354
ps-uncommon_ip	977961
ps-uncommon_ip_hard	833247
ps-uncommon_dkim_ip_hard	721877
ps-dkim_fail	571859
ps-uncommon_dkim_ip	400838
ps-uncommon_transactional_ip	353236
ps-subject_phish_keyword	335875
ps-uncommon_country_hard	286642
ps-uncommon_dkim_country_hard	258123
ps-uncommon_country	151806
ps-young_domain	147179
ps-uncommon_dkim_country	144947
ps-uncommon_transactional_ip_hard	117050

ps-txt_form	87743
ps-claimed_link_mismatch	53384
ps-similar_domain	941

Table 4.1: Number of occurrences of features used in traffic statistics filter.

4.2 Email Content Model

Email content model result is shown on Fig. 4.5. We may see that now it classifies about 25% of checked emails as phishing. That is much better result than baseline solution described in 2.4.

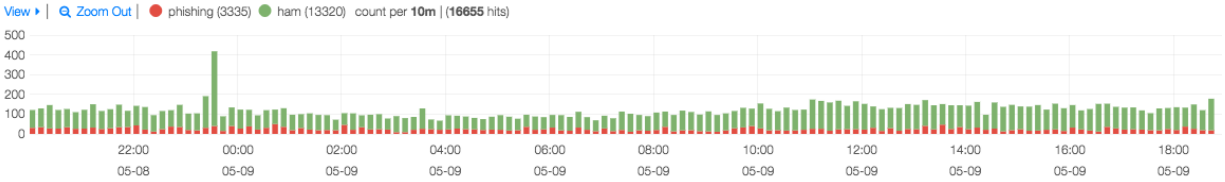

Figure 4.5: Graph shows count of phishing and non-phishing emails based on classifier result. Time is shown in X axis.

We also manually classified 100 random unique emails which were classified by this model and had high phishing-score. In Tab. 4.2 are shown results of this evaluation. This evaluation is of course not complete, but it gives representative results. In column *Unique emails* are shown results based on unique emails count and in column *Total emails* are results based on how many emails were delivered. Results show that classifier suffers with many false positives, on the other hand we may presume that it have only very few false negatives based in our small test in which it has none.

Metric	Unique emails	Total emails
Precision	15.00%	22.31%
Recall	100.00%	100.00%
Accuracy	57.50%	76.69%
F1-score	26.09%	36.48%

Table 4.2: Email content model evaluation.

4.3 Claimed Domain Detection

This method was tested on non-phishing data set described in 3.3.2. From each email was extracted header from taken as correct result. We may presume that transactional emails will admit their correct domain in header from. Results were validated only for phisheable domains because we want to classify mainly phisheable domains.

We had not testedheader from detection because it was taken as correct result. We may do this because other methods were trained on data where header from was also used as correct result.

Domain keyword recognizer correctly recognizes domain for **77.15%** of tested emails. Domain image sources correctly recognizes domain for **82.19%** of tested emails. Text links correctly recognizes domain for **55.81%** of tested domains.

4.3.1 Overall Performance

With results merged from all three methods we were able to recognize correctly **100%** of domains. When 25% of domains from the data set are unknown for all methods. All three methods agreed on same solution for 23.22% test samples.

4.4 Suspicious Link Detection

This method was tested on manually classified phishing emails picked from live traffic. We were able to find 15 totally unique phishing emails. In these emails we manually found link pointing to phishing website. In some copies of picked emails were distinct links. This method classified **100%** of these links as most suspicious links. Four of these links were yet unknown for Google safe browsing API [32], which shows that commonly used blacklisting of phishing links may be slow.

4.5 Whole System Evaluation

Our system is based on rating emails with metrics called phishing-score. We tested this system on part of email traffic, which consists of approximately 50 million emails per day, delivered to Seznam.cz. This system classifies approximately 1 million emails per day.

4.5.1 Phishing Score Distribution

Used metric (phishing-score) should determine whether email is phishing or not. This score could be negative, which means email is very probably legit. Phishing score is distributed

from -35 to 30 in current setup. In Tab. 4.3 may be seen that majority of emails has very low score. Emails with higher score will be further examined in next chapter 4.5.2.

Phishing-score	Email count
less than 0	4896283
between 0 and 10	182267
higher than 10	12102

Table 4.3: Phishing-score distribution. Statistics for 4 days.

4.5.2 System Evaluation

In this chapter we will evaluate phishing-detection classifier. As shown in chapter 4.5.1 about 5 million emails in short amount of time got score lover than 0. We cannot verify all these emails by hand, and we suppose that portion of phishing emails will be very small. Because of that we targeted emails with highest phishing-score and manually classified 3517 of them. Distribution of classified emails based on phishing-score is shown in Tab. 4.4. In table is shown that for phishing-score lower than 8 phishing email count is not rising so we suppose that in lower scores there are no phishing messages.

Phishing-score	Phishing email count	Legit email count
more than 24	7	0
more than 20	20	7
more than 16	35	48
more than 12	43	249
more than 8	47	559
less than 6	47	1004

Table 4.4: Phishing emails distribution based on phishing-score.

Our classifier decision is based on threshold, true positive rate (TPR) and false positive rate (FPR) are dependent on threshold setup. How TPF and FPR change based on phishing-score threshold setup is shown on Fig. 4.6. On figure may be seen that we can classify 40% of phishing emails almost without any false positives. But we want to detect more than 40% of emails, because human will be verifying classifier result, so we may have FP.

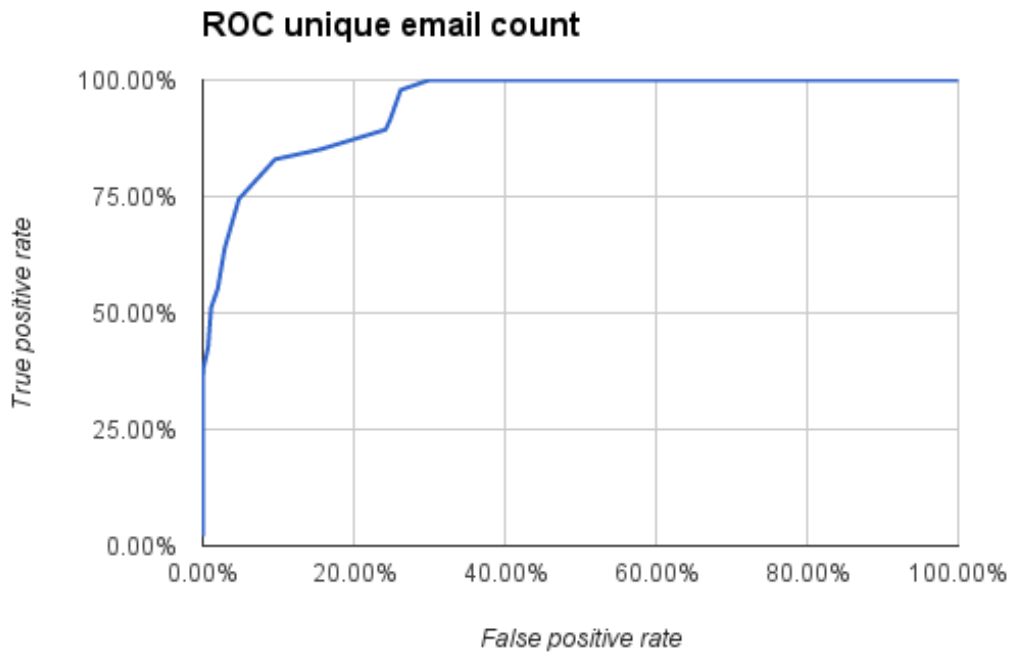

Figure 4.6: ROC of phishing email classification for unique emails, based on phishing-score threshold setup.

On Fig. 4.6 was shown ROC for unique emails. On Fig. 4.7 are shown email counts.These results are different because if only unique emails are counted there are only very few phishing emails, but when absolute counts are taken some emails may be delivered in huge doses and their correct classification is than more significant. Following results are probably caused by that phishing emails are sent in big doses and misclassified emails may be some testing messages, or some system failures, so these were not so often. From this figure it may seem reasonable to choose phishing score 19 because it classifies 87.99% of phishing messages with 4.02% of misclassifications. Or to chose phishing score 12 with 98.68% TPR and 31.44% FPR. But we think that this result is not so representative as result shown in Fig. 4.6.

Interesting phishing score threshold setups are shown in Tab. 4.5. If the threshold will be set up to 21 the classifier will have absolutely minimal FPR, which will be usable for alerting users, but it would not detect more than 40% of threads. If we set threshold to 10 we will detect all threads but we will have almost 30% FPR. Will decided to use threshold 11 which detects 97.87% of attacks with 26.20% FPR. This FPR Means that approximately two hundred emails will be misclassified each day, is is relatively high number but phishing and non-phishing emails may be in most times differentiated very quickly.

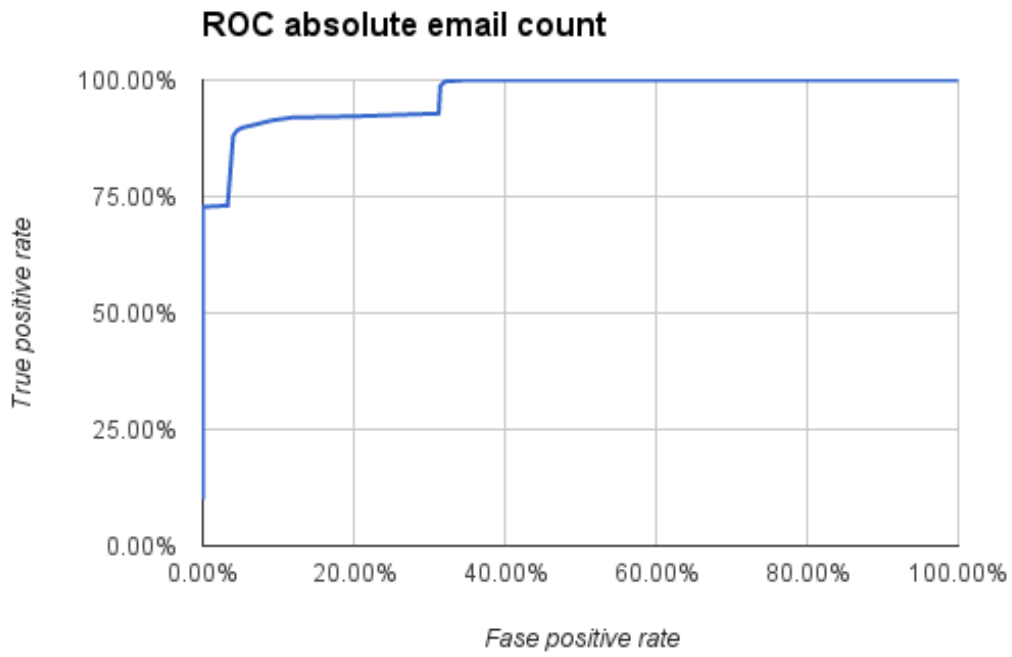

Figure 4.7: ROC of phishing email classification for real count of emails delivered (not unique), based on phishing-score threshold setup.

Phishing score	TPR	FPR
21	0.10%	38.30%
17	2.89%	63.83%
16	4.78%	74.47%
11	26.20%	97.87%
10	29.98%	100.00%

Table 4.5: Classifier statistics based on threshold setup.

Chapter 5

Conclusions

5.1 Summary

Our main goal is to detect phishing emails. For that purpose we wanted to find which features are commonly used for this purpose. We did literature survey of recent articles and find out most commonly used features for phishing detection.

We used most common features which we got from literature survey for baseline solution. For this solution was used decision trees model trained on data gathered from public data sets mixed with data gathered in Seznam.cz.

Baseline solution showed promising results in test, but in production environment it suffers with huge amount of misclassifications. We think that bad results of the classifier were caused by outdated training data set and usage of broad features.

We tried to gather better data set, preferably in czech language but unique phishing messages are relatively rare especially in czech, and no other publicly available data sets was found. We decided to use own implementation with manually configurable parameters because we cannot train correct parameters from the data set. This implementation consists of two steps. First one does prefiltering based on traffic statistics and email content. Second step is phishing detection itself based on sender recognition and link evaluation.

Prefiltering is based on 30 features which are mainly based on traffic statistics and 25 content based features. Traffic features are used directly to adjust phishing-score, whereas content features are trained by decision tree classifier.

Phishing detection system is based on statistics for sender domain. Firstly sender has to be recognized which is done by domain specific keywords, commonly used image sources, plain links to domains and header from, which was able of detection 100% of domains in testing. Than most suspicious link is selected, this part exhibits 100% performance on validated live data. Last part is that most suspicious link is checked if its domain is commonly linked from detected domain.

Parameter scale was initially set by expert estimation. The the correctness of the scale was evaluated on classified data. Whole classified traffic cannot be manually labeled

because it is huge and phishing emails are rare. We iteratively set the scales and than check emails with highest score manually classify them and adjust scales based on the results.

Last and most important scale is phishing-score threshold, which decides whether email is phishing or not. We evaluated possible thresholds by ROC and decided for setup which is able to detect 98% of phishing attacks with 26% of misclassifications.

5.2 Contributions of the Thesis

Goal of this thesis was to find method for detecting phishing emails, implement it and test it on live data. Our implementation is capable of detecting 98% of phishing emails delivered. Nowadays these detections are used for informing anti-spam administrator, but it future it is planned that this module will be informing end users directly via alert stripe like in Mozilla Thunderbird shown on Fig. 1.2.

5.3 Future Work

We would love to be able to detect phishing emails only with machine learning techniques. For that we would need to gather more data for which will be used system described in this thesis.

As fist step we will apply anomaly detection algorithm instead of manually setup parameters in prefiltering phase called traffic statistics described in 3.1.

It should be considered to recalculate periodically each statistics used in this work, e.q. domain link statistics, domain keywords and domain common image sources.

Bibliography

[1] Apache Spamassassin. http://spamassassin.apache.org/. [Online; accessed 2015-02-01].

[2] CDN. http://en.wikipedia.org/wiki/Content_delivery_network. [Online; accessed 2015-02-01].

[3] DKIM verify. http://old.blog.phusion.nl/category/unix/. [Online; accessed 2015-02-01].

[4] hoax.cz. http://www.hoax.cz/cze/. [Online; accessed 2015-02-01].

[5] JavaScript. http://www.ecma-international.org/ecma-262/5.1/. [Online; accessed 2015-02-01].

[6] Kibana. https://www.elastic.co/products/kibana. [Online; accessed 2015-02-01].

[7] PhishTag. http://search.cpan.org/dist/Mail-SpamAssassin/lib/Mail/SpamAssassin/Plugin/PhishTag.pm. [Online; accessed 2015-02-01].

[8] phishtank.com. https://www.phishtank.com/. [Online; accessed 2015-02-01].

[9] Python. https://www.python.org/. [Online; accessed 2015-02-01].

[10] Python decorators. https://wiki.python.org/moin/PythonDecorators. [Online; accessed 2015-02-01].

[11] ScamNailer. http://www.scamnailer.info/documentation.html. [Online; accessed 2015-02-01].

[12] SciPy. http://www.scipy.org/. [Online; accessed 2015-02-01].

[13] Spamassassin corpus. http://spamassassin.apache.org/publiccorpus/. [Online; accessed 2015-02-01].

[14] Spamassassin corpus desctiption. http://spamassassin.apache.org/publiccorpus/readme.html. [Online; accessed 2015-02-01].

[15] SPF Record Syntax. http://www.openspf.org/SPF_Record_Syntax. [Online; accessed 2015-02-01].

[16] Tf-idf. http://www.tfidf.com/. [Online; accessed 2015-02-01].

[17] Ammar Ali Deeb Al-Mo, Tat-Chee Wan, Karim Al-Saedi, Altyeb Altaher, Sureswaran Ramadass, Ahmad Manasrah, Loai Bani Melhiml, and Mohammad Anbar. An online model on evolving phishing e-mail detection and classification method. *Journal of Applied Sciences*, 11(18):3301–3307, dec 2011.

[18] Ammar Almomani, BB Gupta, Samer Atawneh, A Meulenberg, and Eman Almomani. A survey of phishing email filtering techniques. *Communications Surveys & Tutorials, IEEE*, 15(4):2070–2090, 2013.

[19] Apache. Hadoop. https://hadoop.apache.org/. [Online; accessed 2015-02-01].

[20] APWG. Phishing Activity Trends Report 1st Quarter 2014. http://docs.apwg.org/reports/apwg_trends_report_q1_2014.pdf. [Online; accessed 2015-02-01].

[21] APWG. Phishing Activity Trends Report 3rd Quarter 2013. http://docs.apwg.org/reports/apwg_trends_report_q3_2013.pdf. [Online; accessed 2015-02-01].

[22] APWG. Phishing Activity Trends Report 3rd Quarter 2014. http://docs.apwg.org/reports/apwg_trends_report_q3_2014.pdf. [Online; accessed 2015-02-01].

[23] APWG. Phishing Attacks Trends Report January 2004. http://docs.apwg.org/reports/APWG.Phishing.Attack.Report.Jan2004.pdf. [Online; accessed 2015-02-01].

[24] Andre Bergholz, Jeong Ho Chang, Gerhard Paaß, Frank Reichartz, and Siehyun Strobel. Improved phishing detection using model-based features. In *CEAS*, 2008.

[25] Abhay Bhushan. Standardizing Network Mail Headers. https://tools.ietf.org/html/rfc561. [Online; accessed 2015-02-01].

[26] Madhusudhanan Chandrasekaran, Krishnan Narayanan, and Shambhu Upadhyaya. Phishing email detection based on structural properties. In *NYS Cyber Security Conference*, pages 1–7, 2006.

[27] Juan Chen and Chuanxiong Guo. Online detection and prevention of phishing attacks. In *2006 First International Conference on Communications and Networking in China*. Institute of Electrical & Electronics Engineers (IEEE), oct 2006.

[28] Richard Dazeley, John L. Yearwood, Byeong H. Kang, and Andrei V. Kelarev. Consensus clustering and supervised classification for profiling phishing emails in internet commerce security. In *Knowledge Management and Acquisition for Smart Systems and Services*, pages 235–246. Springer Science, Business Media, 2010.

[29] Jeffrey Dean and Sanjay Ghemawat. Mapreduce: simplified data processing on large clusters. *Communications of the ACM*, 51(1):107–113, 2008.

[30] Rachna Dhamija, J Doug Tygar, and Marti Hearst. Why phishing works. In *Proceedings of the SIGCHI conference on Human Factors in computing systems*, pages 581–590. ACM, 2006.

[31] Ian Fette, Norman Sadeh, and Anthony Tomasic. Learning to detect phishing emails. In *Proceedings of the 16th international conference on World Wide Web*, pages 649–656. ACM, 2007.

[32] Google. Safe browsing API. `https://developers.google.com/safe-browsing/`. [Online; accessed 2015-02-01].

[33] E. Hall Network Working Group. RFC 4155. `http://tools.ietf.org/html/rfc4155`. [Online; accessed 2015-02-01].

[34] Network Working Group. Domain Name System Structure and Delegation. `https://tools.ietf.org/html/rfc1591`. [Online; accessed 2015-02-01].

[35] Network Working Group. DomainKeys Identified Mail (DKIM) Service Overview. `https://tools.ietf.org/html/rfc5585`. [Online; accessed 2015-02-01].

[36] Network Working Group. DomainKeys Identified Mail (DKIM) Signatures. `https://tools.ietf.org/html/rfc6376`. [Online; accessed 2015-02-01].

[37] Network Working Group. HTTP Over TLS. `http://tools.ietf.org/html/rfc2818`. [Online; accessed 2015-02-01].

[38] Network Working Group. INTERNET MESSAGE ACCESS PROTOCOL - VERSION 4rev1. `https://tools.ietf.org/html/rfc3501`. [Online; accessed 2015-02-01].

[39] Network Working Group. Internet Message Format. `https://tools.ietf.org/html/rfc5322`. [Online; accessed 2015-02-01].

[40] Network Working Group. MIME (Multipurpose Internet Mail Extensions) Part Three: Message Header Extensions for Non-ASCII Text. `https://tools.ietf.org/html/rfc2047`. [Online; accessed 2015-02-01].

[41] Network Working Group. Multipurpose Internet Mail Extensions (MIME) Part Five: Conformance Criteria and Examples. `https://tools.ietf.org/html/rfc2049`. [Online; accessed 2015-02-01].

[42] Network Working Group. Multipurpose Internet Mail Extensions (MIME) Part Four: Registration Procedures. `https://tools.ietf.org/html/rfc2048`. [Online; accessed 2015-02-01].

[43] Network Working Group. Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies. `https://tools.ietf.org/html/rfc2045`. [Online; accessed 2015-02-01].

[44] Network Working Group. Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types. `https://tools.ietf.org/html/rfc2046`. [Online; accessed 2015-02-01].

[45] Network Working Group. Post Office Protocol - Version 3. `https://tools.ietf.org/html/rfc1939`. [Online; accessed 2015-02-01].

[46] Network Working Group. Registration of Mail and MIME Header Fields. `https://tools.ietf.org/html/rfc4021`. [Online; accessed 2015-02-01].

[47] Network Working Group. Sender Policy Framework (SPF) for Authorizing Use of Domains in E-Mail, Version 1. `https://tools.ietf.org/html/rfc4408`. [Online; accessed 2015-02-01].

[48] Network Working Group. Simple Mail Transfer Protocol. `https://tools.ietf.org/html/rfc5321`. [Online; accessed 2015-02-01].

[49] Isredza Rahmi A. Hamid and Jemal Abawajy. Hybrid feature selection for phishing email detection. In *Algorithms and Architectures for Parallel Processing*, pages 266–275. Springer Science, Business Media, 2011.

[50] Jason Hong. The state of phishing attacks. *Communications of the ACM*, 55(1):74–81, 2012.

[51] Internet Engineering Task Force (IETF). Sender Policy Framework (SPF) for Authorizing Use of Domains in E-Mail, Version 1. `https://tools.ietf.org/html/rfc7208`. [Online; accessed 2015-02-01].

[52] Rafiqul Islam and Jemal Abawajy. A multi-tier phishing detection and filtering approach. *Journal of Network and Computer Applications*, 36(1):324–335, jan 2013.

[53] Chris Karlof, Umesh Shankar, J Doug Tygar, and David Wagner. Dynamic pharming attacks and locked same-origin policies for web browsers. In *Proceedings of the 14th ACM conference on Computer and communications security*, pages 58–71. ACM, 2007.

[54] Mahmoud Khonji, Youssef Iraqi, and Andrew Jones. Phishing detection: a literature survey. *Communications Surveys & Tutorials, IEEE*, 15(4):2091–2121, 2013.

[55] Mahmoud Khonji, Andrew Jones, and Youssef Iraqi. A novel phishing classification based on URL features. In *2011 IEEE GCC Conference and Exhibition (GCC)*. Institute of Electrical & Electronics Engineers (IEEE), feb 2011.

[56] Scikit learn. Cross-validation: evaluating estimator performance. `http://scikit-learn.org/stable/modules/cross_validation.html`. [Online; accessed 2015-02-01].

[57] Scikit learn. Feature extraction. `http://scikit-learn.org/stable/modules/feature_extraction.html`. [Online; accessed 2015-02-01].

[58] Scikit learn. Multi-class SVM. `http://scikit-learn.org/stable/modules/svm.html#multi-class-classification`. [Online; accessed 2015-02-01].

[59] Scikit learn. MultinomialNB. `http://scikit-learn.org/stable/modules/generated/sklearn.naive_bayes.MultinomialNB.html#sklearn-naive-bayes-multinomialnb`. [Online; accessed 2015-02-01].

[60] Scikit learn. RandomForestClassifier. `http://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html`. [Online; accessed 2015-02-01].

[61] Scikit learn. SDG Classifier. `http://scikit-learn.org/stable/modules/generated/sklearn.linear_model.SGDClassifier.html#sklearn.linear_model.SGDClassifier`. [Online; accessed 2015-02-01].

[62] Liping Ma, Bahadorrezda Ofoghi, Paul Watters, and Simon Brown. Detecting phishing emails using hybrid features. In *2009 Symposia and Workshops on Ubiquitous, Autonomic and Trusted Computing*. Institute of Electrical & Electronics Engineers (IEEE), 2009.

[63] Liping Ma, John Yearwood, and Paul Watters. Establishing phishing provenance using orthographic features. In *2009 eCrime Researchers Summit*. Institute of Electrical & Electronics Engineers (IEEE), oct 2009.

[64] Mozilla. Public Suffix List. `https://publicsuffix.org/`. [Online; accessed 2015-02-01].

[65] J. Nazario. Phishing corpus. `http://monkey.org/~jose/wiki/doku.php?id=phishingcorpus`. [Online; accessed 2015-02-01].

[66] Cleber K. Olivo, Altair O. Santin, and Luiz S. Oliveira. Obtaining the threat model for e-mail phishing. *Applied Soft Computing*, 13(12):4841–4848, dec 2013.

[67] Jonathan B. Postel. SIMPLE MAIL TRANSFER PROTOCOL. `https://tools.ietf.org/html/rfc821`. [Online; accessed 2015-02-01].

[68] scikit learn. Decision tree. `http://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html#sklearn.tree.DecisionTreeClassifier`. [Online; accessed 2015-02-01].

[69] Stanford. Tf-idf weighting. http://nlp.stanford.edu/IR-book/html/htmledition/tf-idf-weighting-1.html. [Online; accessed 2015-02-01].

[70] The technical stuff. E-Mail working process. http://www.thetechnicalstuff.com/email-working-process/. [Online; accessed 2015-02-01].

[71] Fergus Toolan and Joe Carthy. Phishing detection using classifier ensembles. In *eCrime Researchers Summit, 2009. eCRIME09.*, pages 1–9. IEEE, 2009.

[72] Fergus Toolan and Joe Carthy. Feature selection for spam and phishing detection. In *2010 eCrime Researchers Summit*. Institute of Electrical & Electronics Engineers (IEEE), 2010.

[73] W3C. HTML 5. http://www.w3.org/TR/html5/. [Online; accessed 2015-02-01].

[74] W3C. HTML 5 - a element. http://www.w3.org/TR/html5/text-level-semantics.html#the-a-element. [Online; accessed 2015-02-01].

[75] W3C. HTML 5 - img element. https://www.w3.org/wiki/HTML/Elements/img. [Online; accessed 2015-02-01].

[76] Wikipedia. Levenshtein distance. http://en.wikipedia.org/wiki/Levenshtein_distance. [Online; accessed 2015-02-01].

[77] John Yearwood, Musa Mammadov, and Arunava Banerjee. Profiling phishing emails based on hyperlink information. In *2010 International Conference on Advances in Social Networks Analysis and Mining*. Institute of Electrical & Electronics Engineers (IEEE), aug 2010.

Appendix A

Content of attached DVD

- Thesis
  - **Phishing_Email_Detection_in_Czech_Language.pdf**
    This work in electronic version.
  - **Thesis.zip**
    Source codes of this work.
- Source-codes
  - **mbox_extractor.py**
    Python implementation of mbox to eml converter.
  - **phish_pipeline**
    Pipeline for training and testing email content model.
  - **deduplicate.sh**
    Shell script for copying files to new location with name based on file content.
  - **RankingClassifier.py**
    Python implementation of ranking classifier.
  - **Cached.py**
    Python cached implementation.
  - **Phish-score.py**
    Module for detection phishing emails, usable in Seznam.cz internal system.
  - **Domain_specific_keyword_classifier**
    Pipeline for training and testing many versions of domain specific classifier.
  - **Phishing_feature_extractor**
    System for testing features used in email content model.
- Data

- **Text_forms_data**
  Text forms used in phishing emails.
- **Model_features_cross_validation_stats**
  Results of feature cross-validation tests.
- **Feature_stats**
  Complete feature statistics.

Appendix B

Statistical Evaluation of Content Features

In all following figures red means phishing data set and green means non-phishing data set.

Is HTML

Dataset	Mean	Median	Std	Min	Max	Zero	Zero [%]	Non-zero	Non-zero [%]
PHISH	0.96	1.00	0.21	0.00	1.00	255	4.40%	5536	95.60%
NON-PHISH	0.74	1.00	0.44	0.00	1.00	2702	26.43%	7523	73.57%

Table B.1

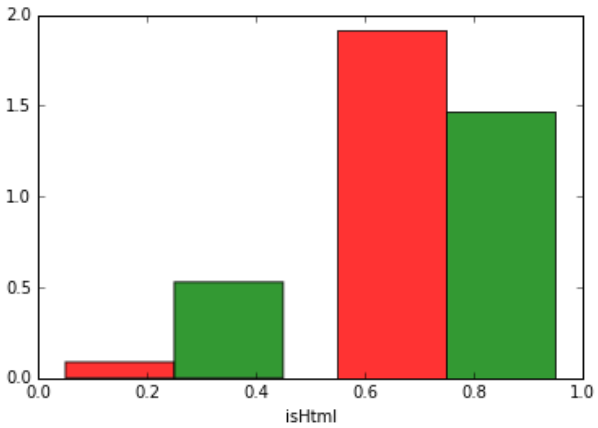

Figure B.1

Contains Form

Dataset	Mean	Median	Std	Min	Max	Zero	Zero [%]	Non-zero	Non-zero [%]
PHISH	0.02	0.00	0.13	0.00	1.00	5692	98.29%	99	1.71%
NON-PHISH	0.00	0.00	0.04	0.00	1.00	10210	99.85%	15	0.15%

Table B.2

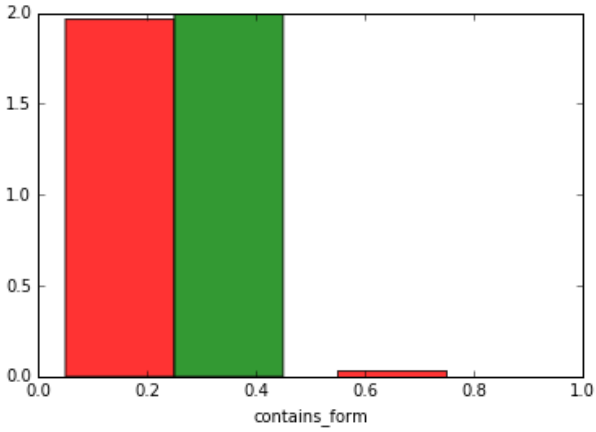

Figure B.2

Contains Table

Dataset	Mean	Median	Std	Min	Max	Zero	Zero [%]	Non-zero	Non-zero [%]
PHISH	0.40	0.00	0.49	0.00	1.00	3500	60.44%	2291	39.56%
NON-PHISH	0.46	0.00	0.50	0.00	1.00	5479	53.58%	4746	46.42%

Table B.3

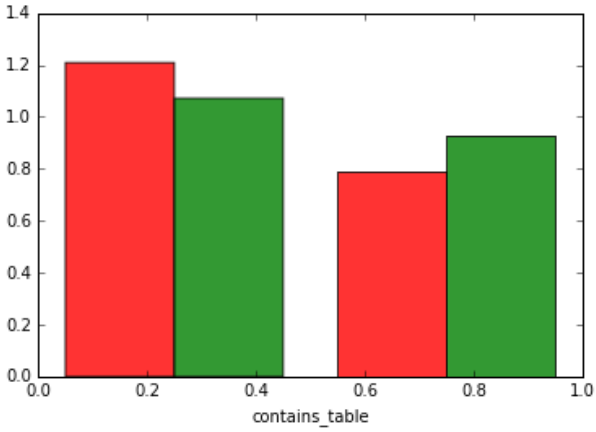

Figure B.3

Number of Urls

Dataset	Mean	Median	Std	Min	Max	Zero	Zero [%]	Non-zero	Non-zero [%]
PHISH	4.68	2.00	13.73	0.00	736.00	1375	23.74%	4416	76.26%
NON-PHISH	6.83	4.00	11.19	0.00	276.00	3885	38.00%	6340	62.00%

Table B.4

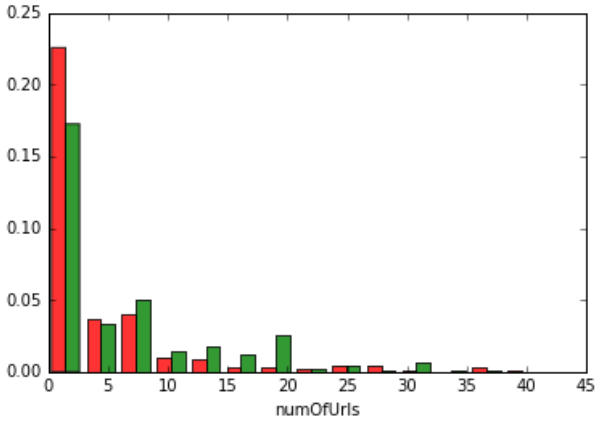

Figure B.4

Number of IP links

Dataset	Mean	Median	Std	Min	Max	Zero	Zero [%]	Non-zero	Non-zero [%]
PHISH	0.08	0.00	0.18	0.00	1.00	4683	80.87%	1108	19.13%
NON-PHISH	0.00	0.00	0.01	0.00	0.50	10223	99.98%	2	0.02%

Table B.5

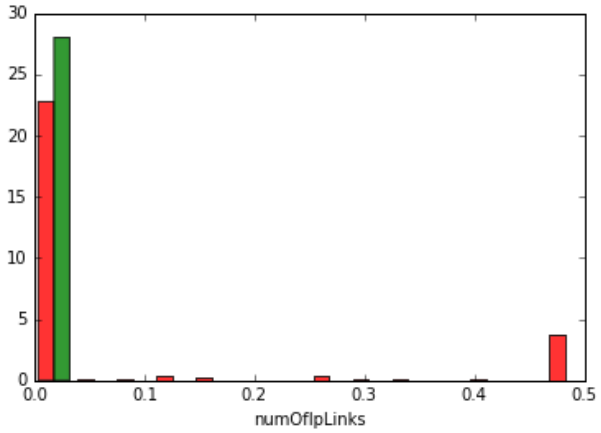

Figure B.5

Number of Sender Differ Url

Dataset	Mean	Median	Std	Min	Max	Zero	Zero [%]	Non-zero	Non-zero [%]
PHISH	0.35	0.50	0.22	0.00	1.00	1521	26.26%	4270	73.74%
NON-PHISH	0.26	0.33	0.23	0.00	1.00	4157	40.66%	6068	59.34%

Table B.6

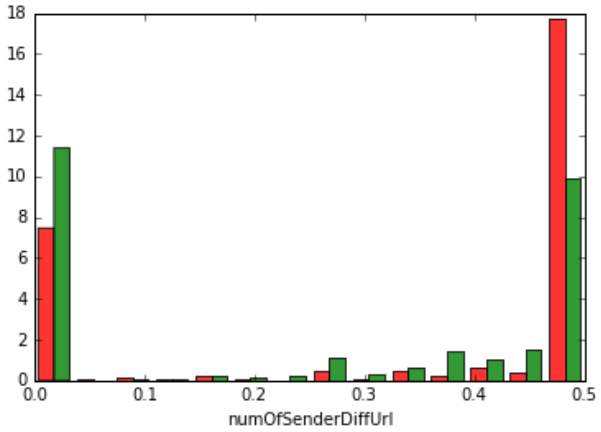

Figure B.6

Maximal Number of Dots in URL

Dataset	Mean	Median	Std	Min	Max	Zero	Zero [%]	Non-zero	Non-zero [%]
PHISH	3.19	3.00	2.58	0.00	28.00	1482	25.59%	4309	74.41%
NON-PHISH	1.70	2.00	1.75	0.00	15.00	3926	38.40%	6299	61.60%

Table B.7

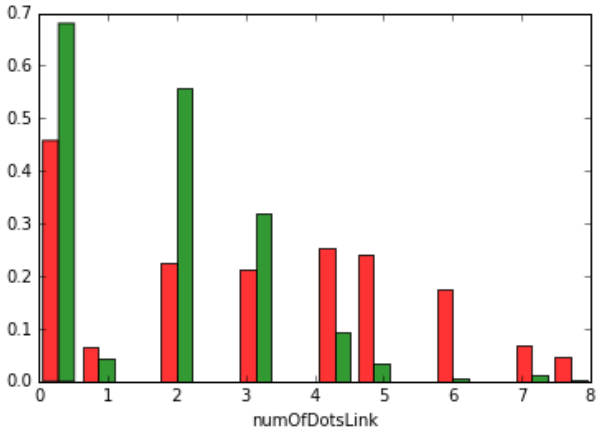

Figure B.7

Maximal Number of Slashes in URL

Dataset	Mean	Median	Std	Min	Max	Zero	Zero [%]	Non-zero	Non-zero [%]
PHISH	2.58	2.00	2.55	0.00	23.00	1568	27.08%	4223	72.92%
NON-PHISH	1.40	1.00	1.67	0.00	11.00	4504	44.05%	5721	55.95%

Table B.8

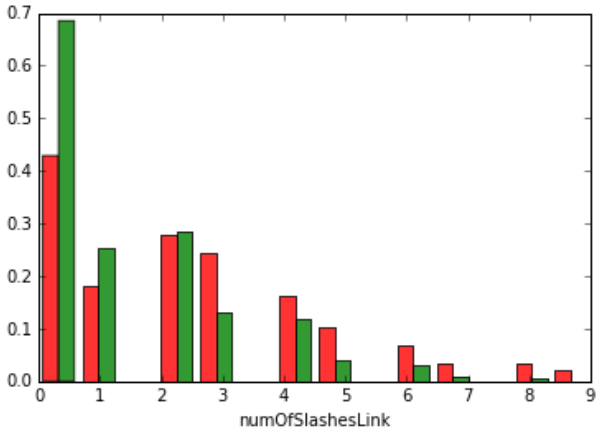

Figure B.8

Number of Non 80 Port

Dataset	Mean	Median	Std	Min	Max	Zero	Zero [%]	Non-zero	Non-zero [%]
PHISH	0.07	0.00	0.47	0.00	15.00	5581	96.37%	210	3.63%
NON-PHISH	0.00	0.00	0.00	0.00	0.00	10225	100.00%	0	0.00%

Table B.9

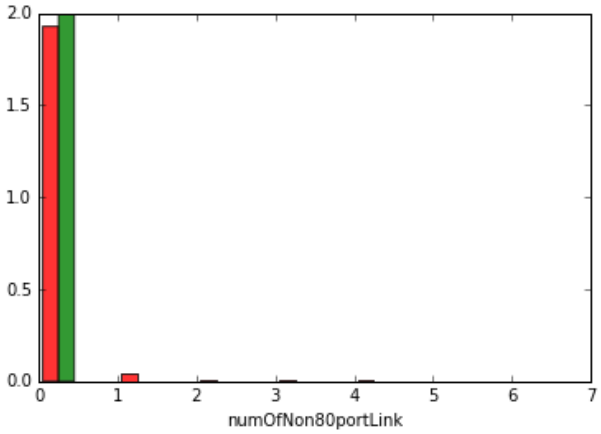

Figure B.9

Number of Images

Dataset	Mean	Median	Std	Min	Max	Zero	Zero [%]	Non-zero	Non-zero [%]
PHISH	3.39	1.00	9.39	0.00	168.00	2801	48.37%	2990	51.63%
NON-PHISH	2.46	0.00	11.06	0.00	257.00	6066	59.33%	4159	40.67%

Table B.10

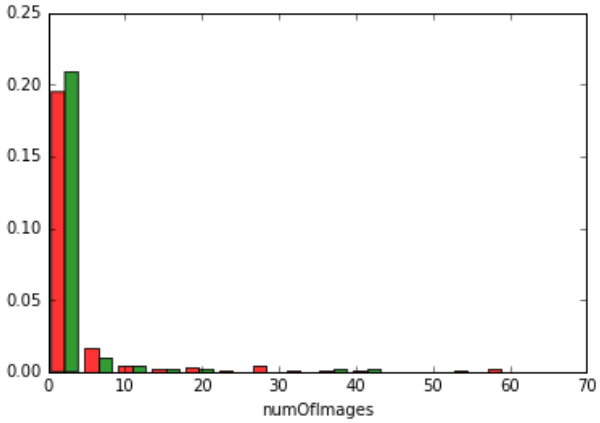

Figure B.10

Number of Scripts

Dataset	Mean	Median	Std	Min	Max	Zero	Zero [%]	Non-zero	Non-zero [%]
PHISH	0.19	0.00	1.56	0.00	30.00	5603	96.75%	188	3.25%
NON-PHISH	0.08	0.00	1.11	0.00	48.00	10161	99.37%	64	0.63%

Table B.11

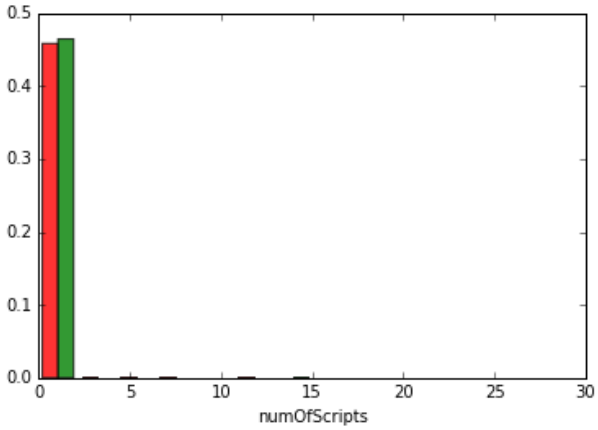

Figure B.11

Number of Words

Dataset	Mean	Median	Std	Min	Max	Zero	Zero [%]	Non-zero	Non-zero [%]
PHISH	275.78	224.00	270.70	2.00	6357.00	0	0.00%	5791	100.00%
NON-PHISH	173.72	98.00	243.01	2.00	6297.00	0	0.00%	10225	100.00%

Table B.12

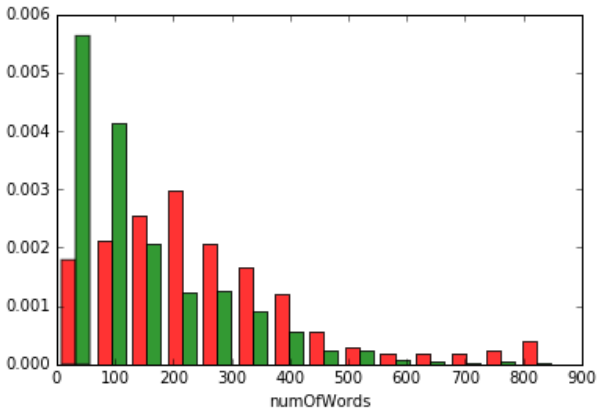

Figure B.12

Number of Charcters

Dataset	Mean	Median	Std	Min	Max	Zero	Zero [%]	Non-zero	Non-zero
PHISH	1427.35	1268.00	1140.27	1.00	17534.00	0	0.00%	5791	100.00%
NON-PHISH	1003.14	662.00	3110.48	2.00	283658.00	0	0.00%	10225	100.00%

Table B.13

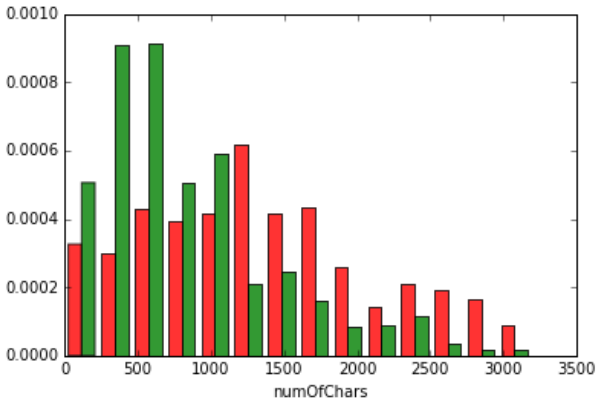

Figure B.13

Number of Unique words

Dataset	Mean	Median	Std	Min	Max	Zero	Zero [%]	Non-zero	Non-zero [%]
PHISH	128.17	119.00	80.27	1.00	1178.00	0	0.00%	5791	100.00%
NON-PHISH	84.39	62.00	90.66	2.00	2079.00	0	0.00%	10225	100.00%

Table B.14

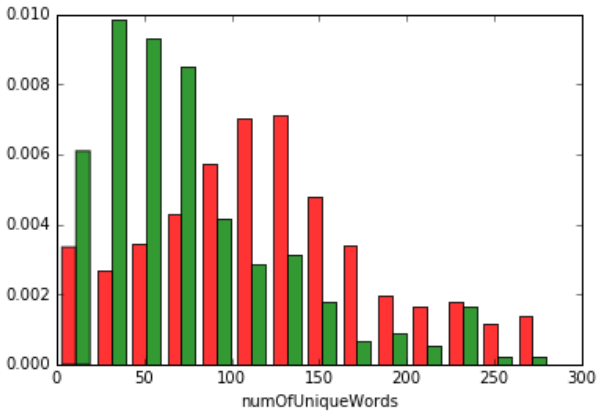

Figure B.14

Vocabulary Richness

Dataset	Mean	Median	Std	Min	Max	Zero	Zero [%]	Non-zero	Non-zero [%]
PHISH	0.55	0.54	0.18	0.03	1.00	0	0.00%	5791	100.00%
NON-PHISH	0.66	0.67	0.24	0.01	1.00	0	0.00%	10225	100.00%

Table B.15

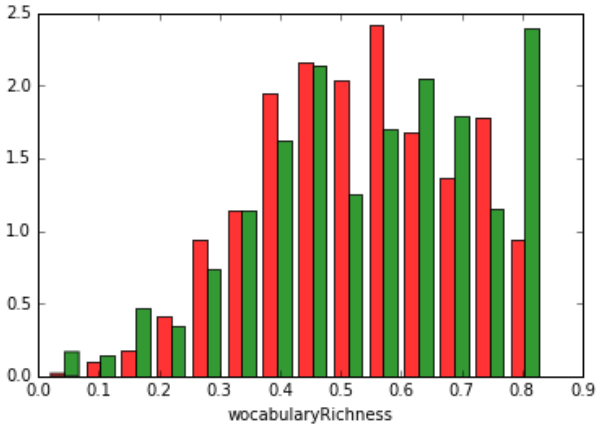

Figure B.15

Is Reply

Dataset	Mean	Median	Std	Min	Max	Zero	Zero [%]	Non-zero	Non-zero [%]
PHISH	0.32	0.00	0.47	0.00	1.00	3912	67.55%	1879	32.45%
NON-PHISH	0.38	0.00	0.48	0.00	1.00	6358	62.18%	3867	37.82%

Table B.16

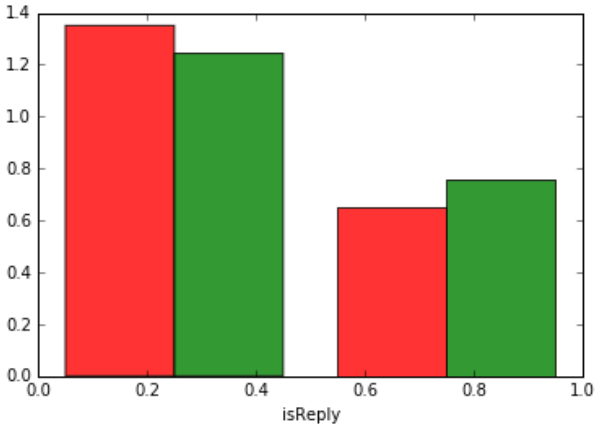

Figure B.16

Is Forward

Dataset	Mean	Median	Std	Min	Max	Zero	Zero [%]	Non-zero	Non-zero [%]
PHISH	0.00	0.00	0.07	0.00	1.00	5764	99.53%	27	0.47%
NON-PHISH	0.00	0.00	0.07	0.00	1.00	10176	99.52%	49	0.48%

Table B.17

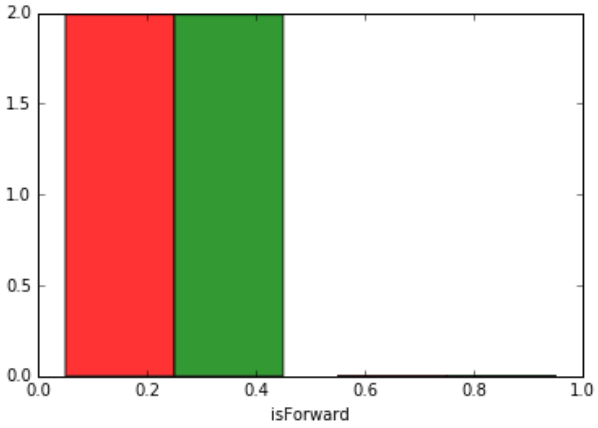

Figure B.17

Number of Subject Words

Dataset	Mean	Median	Std	Min	Max	Zero	Zero [%]	Non-zero	Non-zero [%]
PHISH	5.88	5.00	4.68	1.00	202.00	0	0.00%	5791	100.00%
NON-PHISH	5.05	5.00	2.41	1.00	23.00	0	0.00%	10225	100.00%

Table B.18

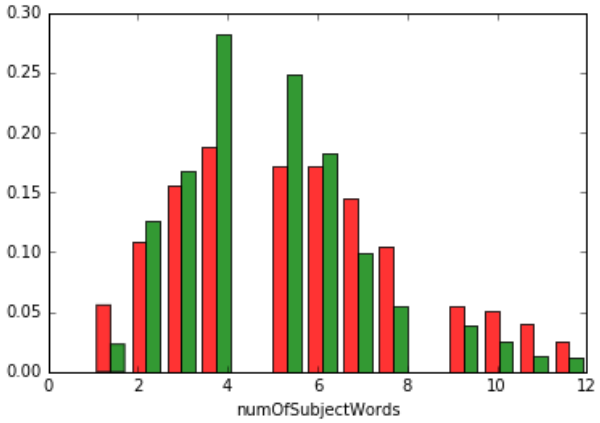

Figure B.18

Number of Subject Characters

Dataset	Mean	Median	Std	Min	Max	Zero	Zero [%]	Non-zero	Non-zero [%]
PHISH	41.21	37.00	35.98	0.00	1604.00	60	1.04%	5731	98.96%
NON-PHISH	38.34	36.00	17.74	0.00	194.00	22	0.22%	10203	99.78%

Table B.19

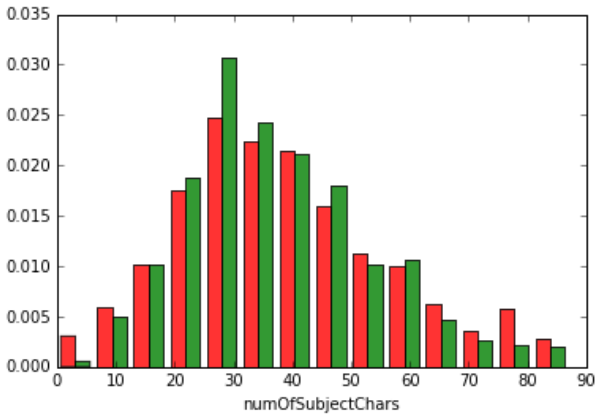

Figure B.19

Number of HTTP links

Dataset	Mean	Median	Std	Min	Max	Zero	Zero [%]	Non-zero	Non-zero [%]
PHISH	1.96	1.00	6.58	0.00	366.00	1564	27.01%	4227	72.99%
NON-PHISH	2.53	1.00	4.71	0.00	106.00	4715	46.11%	5510	53.89%

Table B.20

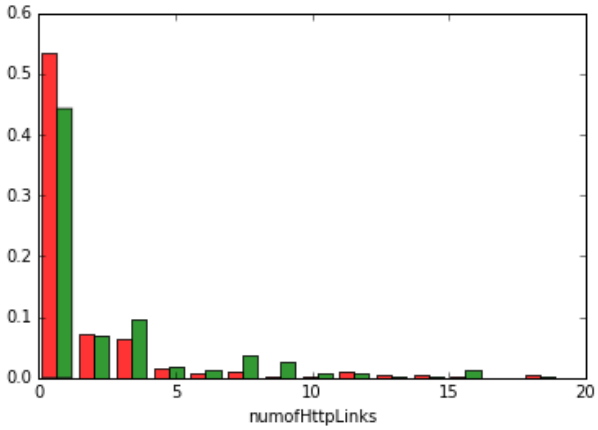

Figure B.20

Number of HTTPS links

Dataset	Mean	Median	Std	Min	Max	Zero	Zero [%]	Non-zero	Non-zero [%]
PHISH	0.13	0.00	0.98	0.00	35.00	5472	94.49%	319	5.51%
NON-PHISH	0.36	0.00	1.27	0.00	28.00	8767	85.74%	1458	14.26%

Table B.21

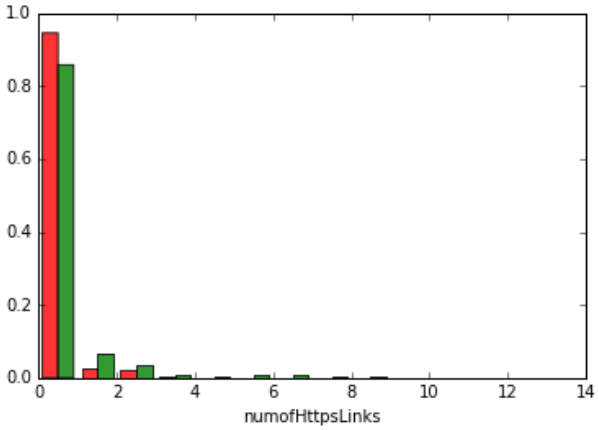

Figure B.21