Project Milestone 2

ISQS 5342 Big Data Security

Texas Tech University

04/14/2024

Emily Spector

James Parker

Jonathan Busch

Contents

[Part 1: Dataset Identification and Justification 2](#_Toc1967872770)

[Part 2: Collection and Parsing Strategies 4](#_Toc1134854811)

[Part 3: Data Summary 5](#_Toc1772298742)

[Phishing URL Detection 5](#_Toc1889020733)

[Spam Email Detection 7](#_Toc1354846433)

[Part 4: Group Roles and Signatures 8](#_Toc735629325)

[Appendix A 8](#_Toc226640950)

# Part 1: Dataset Identification and Justification

For this project's purposes, two datasets were collected. Both datasets were retrieved from Kaggle.com which acts as a repository for datasets for the purposes of data science projects. The first dataset includes various emails classified as either spam or legitimate. Spam is a constant issue for Education with bad actors usually targeting educational institution through the vector of email spam. Successfully identified spam will allow these emails to be automatically quarantined, increasing the overall security of the educational institution. This data is generated by the email server as it stores emails that have been sent to it. The data we retrieved includes 5826 emails. This data source was selected due to its generous length allowing for classification to be done effectively, the ability to perform text analysis which will expand our skillset, and the cleanliness of the data. Other organizations within the education industry are also likely to analyze their emails to create more accurate anti-spam models to protect themselves against spam, and the attacks that can be launched from spam. To summarize, this data was chosen due to its ease of use, relevance to educational institutions, and for its inclusion of text allowing us to perform text analysis.

The second data set as mentioned above was also found on Kaggle.com. This data set includes various features of links contained within emails. Each row is also labeled as either a phishing link or a legitimate one. This dataset will allow us to create a model that can classify whether a given link is a phishing link or not. This sample of phishing links was generated by extracting links from 5000 phishing webpages and 5000 legitimate webpages and extracting the links features. Datasets allowing for the classification of phishing links are valuable to the industry of Education as these models can be utilized to automatically block emails containing dangerous phishing links. This data is generated by website hosters who set their URLs which are then included in emails to be sent as either legitimate links or harmful phishing links. This data includes 10,000 records and 49 features. This data was selected due to relevance to the education industry due to the large number of phishing attacks attempted towards .edu emails. Other organizations, various colleges and schools such as Texas A&M, will also be interested in analyzing any data from emails including phishing links to quarantine away dangerous emails. This will improve the overall security of the organization as the likelihood of phishing links being clicked decreases as more malicious emails are removed. The data set includes counts of various characters within the URL, the length of the url, characteristics of the website the url links to, and the classification of the link as phishing or not.

# Part 2: Collection and Parsing Strategies

Since these datasets were both obtained through Kaggle, it was essential to do further research to understand exactly how the data was initially collected.

The dataset relating to spam mail detection is the SpamAssassindataset, which is a popular collection of emails labeled as either spam or ham (non-spam). There were several methods used in collecting this data.

User Contributions: Individuals voluntarily shared their personal email data, sending in email contents of both spam and non-spam messages.

Web Scraping: Publicly available email data was scraped from online repositories and forums, including mailing list archives and discussion groups.

The raw SpamAssassin data had already been processed and parsed into a CSV file prior to our collection; this was originally done by extracting text files from a database of emails for their contents. Using the email packages email.parser and email.policy in Python, the text data of these emails was able to be extracted and parsed into one column of a data frame, with another column for the email’s label as either spam or ham, and ultimately finalized as a CSV file.

The data collection process for the phishing links data was quite similar and involved gathering URLS from multiple sources. Webpages labeled as a phishing webpage came from URLS found on PhishTank and OpenPhish, two collaborative platforms that collect and distribute data related to phishing attacks. Like SpamAssassin, these platforms rely on manual user submissions and automated web scrapers (automated by GNU Wget, a program that retrieves HTTP and HTTPS files). The legitimate webpages were collected from Alexa and the Common Crawl archive. This data is parsed into a CSV as well, with columns denoting different features of the webpage and the page’s phishing classification.

Both of our datasets contain data where its collection is highly dependent on manual submissions and automated scraping of publicized data. The methods used to obtain these features provide comprehensive coverage within the dataset. Access to verified and diverse datasets involving phishing URLs and spam messages are crucial for researchers to do their own studies model building. The methodology of obtaining data this way, and these specific resources, are widely accepted in cybersecurity for research, individual users, and companies that provide threat and spam detection services for their customers.

# Part 3: Data Summary

## Phishing URL Detection

Below is the “data dictionary” for the phishing URL detection dataset which includes all the variables sorted into “bins.” These bins group the variables based on their functionalities, such as measuring URL structure, character counts and symbols, security checks, content analysis, and metrics derived from applying rules and thresholds. Each bin serves a specific purpose in analyzing and categorizing webpage URLs. A more detailed data dictionary can be found in Appendix A, which includes a description of every individual variable, the null count, distinct count, minimum, maximum, average, and standard deviation.

|  |  |
| --- | --- |
| Name | Description |
| id | ID (1-10,000) |
| NumDots, SubdomainLevel, PathLevel, UrlLength, HostnameLength, PathLength, QueryLength, DoubleSlashInPath, NumNumericChars | These metrics focus on the structural aspects of the URL, such as the number of dots in the URL, the level of subdomains, the depth of the path, and the overall length of the URL components (hostname, path, query). These metrics provide insights into the complexity and organization of the URL itself. |
| NumDash, NumDashInHostname, AtSymbol, TildeSymbol, NumUnderscore, NumPercent, NumQueryComponents, NumAmpersand, NumHash | This bin includes metrics that count specific characters like dashes, underscores, percentages, query components, ampersands, and hashtags within the URL. It also checks for the presence of symbols like "@" and "∼." These metrics help identify patterns or unusual characters in the URL. |
| NoHttps, HttpsInHostname, IpAddress, DomainInSubdomains, DomainInPaths, InsecureForms, RelativeFormAction, ExtFormAction, AbnormalFormAction | These metrics focus on security-related aspects and protocol checks within the URL. They check for HTTPS usage, presence of an IP address, domain structure in subdomains and paths, as well as the security status of form actions (HTTPS, relative URLs, external domains, abnormal actions). |
| NumSensitiveWords, EmbeddedBrandName, PctExtHyperlinks, PctExtResourceUrls, ExtFavicon, PctNullSelfRedirectHyperlinks, FrequentDomainNameMismatch, FakeLinkInStatusBar, RightClickDisabled, PopUpWindow, SubmitInfoToEmail, IframeOrFrame, MissingTitle, ImagesOnlyInForm | This bin covers metrics related to content analysis and link behavior. It includes counts of sensitive words in the URL, presence of brand names, percentages of external hyperlinks and resource URLs, favicon sourcing, abnormal hyperlinks (self-redirects, null values), domain name mismatches, and various HTML-related checks (fake links, right-click disable, pop-ups, mailto forms, iframes). |
| SubdomainLevelRT, UrlLengthRT, PctExtResourceUrlsRT, AbnormalExtFormActionR, ExtMetaScriptLinkRT, PctExtNullSelfRedirectHyperlinksRT | These metrics are derived from applying rules and thresholds to specific aspects of the URL or HTML content. They include rule-based thresholds for subdomain levels, URL length, percentage of external resource URLs, abnormal form actions, external meta/script/link tags, and null/self-redirect hyperlinks. These metrics help identify deviations from expected norms or thresholds. |
| CLASS\_LABEL | Classification of the URL |

Table I – Data Dictionary for Phishing URL Detection

To give examples of what the data looks like, we have included 5 examples of row data. Each of the examples shows a different category of variables, corresponding to the bins above. Please note that each of the examples below are just “slices” of the entire data point, and every id contains data for every variable.

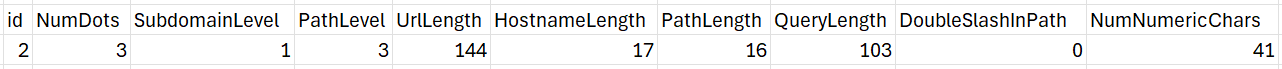


Figure 1 – URL Structure Metrics

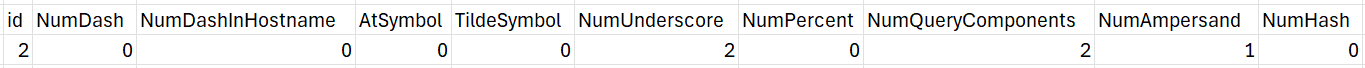


Figure 2 – Character Counts and Symbols

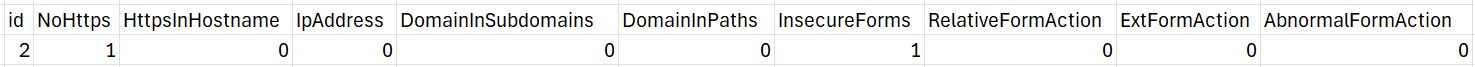


Figure 3 – Security Checks



Figure 4 – Content Analysis

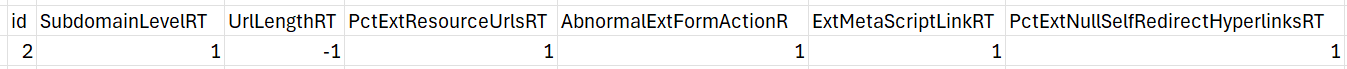


Figure 5 – Rule-Threshold Metrics

## Spam Email Detection

Table II shows the data dictionary for the Spam Email Detection dataset. This dataset has only 2 variables, the text variable, and the target variable. The text variable includes the actual text from the emails, and the target variable is a binary variable that denotes whether the email is a spam email.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Description | Null Count | Distinct Count | Min | Max |
| text | Contains the components of the email such as sender information, recipient information, email metadata, email content, signature, footer, etc. | 0 | 5826 | N/A | N/A |
| target | 1 if spam email  0 if ham email | 0 | 2 | 0 | 1 |

Table II – Data Dictionary for Spam Email Detection

Additionally, we have included 3 examples of data instances below.

A screenshot of a computer

Description automatically generated

Figure 6 – Data Instances of Spam Email Dataset

# Part 4: Group Roles and Signatures

Jonathan Busch-Part 1-33%

Emily Spector-Part 2-33%

James Parker-Part 3-33%

# Appendix A

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Variable Name | Description | Null Count | Distinct | MIN | MAX | AVG | STD |
| id | ID (1-10,000) | 0 | 10000 | 1 | 10000 | N/A | N/A |
| NumDots | Counts the number of dots in webpage URL | 0 | 17 | 1 | 21 | 2.45 | 1.35 |
| SubdomainLevel | Counts the level of subdomain in webpage URL | 0 | 13 | 0 | 14 | 0.59 | 0.75 |
| PathLevel | Counts the depth of the path in webpage URL | 0 | 18 | 0 | 18 | 3.30 | 1.86 |
| UrlLength | Counts the total characters in the webpage URL | 0 | 226 | 12 | 253 | 70.26 | 33.37 |
| NumDash | Counts the number of “-” in webpage URL | 0 | 25 | 0 | 55 | 1.82 | 3.11 |
| NumDashInHostname | Counts the number of “-” in hostname part of webpage URL | 0 | 10 | 0 | 9 | 0.14 | 0.55 |
| AtSymbol | Checks if “@” symbol exist in webpage URL | 0 | 2 | 0 | 1 | N/A | N/A |
| TildeSymbol | Checks if “ ∼ ” symbol exist in webpage URL | 0 | 2 | 0 | 1 | N/A | N/A |
| NumUnderscore | Counts the number of “\_” in webpage URL | 0 | 16 | 0 | 18 | 0.32 | 1.11 |
| NumPercent | Counts the number of “%” in webpage URL | 0 | 14 | 0 | 19 | 0.07 | 0.62 |
| NumQueryComponents | Counts the number of query parts in webpage URL | 0 | 19 | 0 | 23 | 0.46 | 1.34 |
| NumAmpersand | Counts the number of “&” in webpage URL | 0 | 16 | 0 | 22 | 0.28 | 1.12 |
| NumHash | Counts the number of “#” in webpage URL | 0 | 2 | 0 | 1 | N/A | N/A |
| NumNumericChars | Counts the number of numeric characters in the webpage URL | 0 | 82 | 0 | 111 | 5.81 | 9.62 |
| NoHttps | Checks if HTTPS exist in webpage URL | 0 | 2 | 0 | 1 | N/A | N/A |
| RandomString | Checks if random strings exist in webpage URL | 0 | 2 | 0 | 1 | N/A | N/A |
| IpAddress | Checks if IP address is used in hostname part of webpage URL | 0 | 2 | 0 | 1 | N/A | N/A |
| DomainInSubdomains | Checks if TLD or ccTLD is used as part of subdomain in webpage URL | 0 | 2 | 0 | 1 | N/A | N/A |
| DomainInPaths | Checks if TLD or ccTLD is used in the path of webpage URL | 0 | 2 | 0 | 1 | N/A | N/A |
| HttpsInHostname | Checks if HTTPS in obfuscated in hostname part of webpage URL | 0 | 1 | 0 | 0 | N/A | N/A |
| HostnameLength | Counts the total characters in hostname part of webpage URL | 0 | 74 | 4 | 137 | 18.82 | 8.12 |
| PathLength | Counts the total characters in path of webpage URL | 0 | 147 | 0 | 161 | 35.56 | 24.59 |
| QueryLength | Counts the total characters in query part of webpage URL | 0 | 164 | 0 | 188 | 8.61 | 24.31 |
| DoubleSlashInPath | Checks if “//” exist in the path of webpage URL | 0 | 2 | 0 | 1 | N/A | N/A |
| NumSensitiveWords | Counts the number of sensitive words (i.e., “secure”, “account”, “webscr”, “login”, “ebayisapi”, “signin”, “banking”, “confirm”) in webpage URL | 0 | 4 | 0 | 3 | 0.11 | 0.37 |
| EmbeddedBrandName | [Checks if brand name appears in subdomains and path of webpage URL [30] (Brand name here is assumed as the most frequent domain name in the webpage HTML content)](https://www.sciencedirect.com/science/article/pii/S0020025519300763?via%3Dihub" \l "bib0030) | 0 | 2 | 0 | 1 | N/A | N/A |
| PctExtHyperlinks | Counts the percentage of external hyperlinks in webpage HTML source code | 0 | 2510 | 0 | 1 | 0.24 | 0.34 |
| PctExtResourceUrls | Counts the percentage of external resource URLs in webpage HTML source code | 0 | 1489 | 0 | 1 | 0.39 | 0.39 |
| ExtFavicon | Checks if the favicon is loaded from a domain name that is different from the webpage URL domain name | 0 | 2 | 0 | 1 | N/A | N/A |
| InsecureForms | Checks if the form action attribute contains a URL without HTTPS protocol | 0 | 2 | 0 | 1 | N/A | N/A |
| RelativeFormAction | Checks if the form action attribute contains a relative URL | 0 | 2 | 0 | 1 | N/A | N/A |
| ExtFormAction | Checks if the form action attribute contains a URL from an external domain | 0 | 2 | 0 | 1 | N/A | N/A |
| AbnormalFormAction | Check if the form action attribute contains a “#”, “about:blank”, an empty string, or “javascript:true” | 0 | 2 | 0 | 1 | N/A | N/A |
| PctNullSelfRedirectHyperlinks | Counts the percentage of hyperlinks fields containing empty value, self-redirect value such as “#”, the URL of current webpage, or some abnormal value such as “file://E:/” | 0 | 1311 | 0 | 1 | 0.14 | 0.31 |
| FrequentDomainNameMismatch | Checks if the most frequent domain name in HTML source code does not match the webpage URL domain name | 0 | 2 | 0 | 1 | N/A | N/A |
| FakeLinkInStatusBar | Checks if HTML source code contains JavaScript command onMouseOver to display a fake URL in the status bar | 0 | 2 | 0 | 1 | N/A | N/A |
| RightClickDisabled | Checks if HTML source code contains JavaScript command to disable right click function | 0 | 2 | 0 | 1 | N/A | N/A |
| PopUpWindow | Checks if HTML source code contains JavaScript command to launch pop-ups | 0 | 2 | 0 | 1 | N/A | N/A |
| SubmitInfoToEmail | Check if HTML source code contains the HTML “mailto” function | 0 | 2 | 0 | 1 | N/A | N/A |
| IframeOrFrame | Checks if iframe or frame is used in HTML source code | 0 | 2 | 0 | 1 | N/A | N/A |
| MissingTitle | Checks if the title tag is empty in HTML source code | 0 | 2 | 0 | 1 | N/A | N/A |
| ImagesOnlyInForm | Checks if the form scope in HTML source code contains no text at all but images only | 0 | 2 | 0 | 1 | N/A | N/A |
| SubdomainLevelRT | Counts the number of dots in hostname part of webpage URL (Apply rules and thresholds to generate value) | 0 | 3 | -1 | 1 | N/A | N/A |
| UrlLengthRT | Counts the total characters in the webpage URL (Apply rules and thresholds to generate value) | 0 | 3 | -1 | 1 | N/A | N/A |
| PctExtResourceUrlsRT | Counts the percentage of external resource URLs in webpage HTML source code (Apply rules and thresholds to generate value) | 0 | 3 | -1 | 1 | N/A | N/A |
| AbnormalExtFormActionR | Check if the form action attribute contains a foreign domain, “about:blank” or an empty string (Apply rules to generate value) | 0 | 3 | -1 | 1 | N/A | N/A |
| ExtMetaScriptLinkRT | Counts percentage of meta, script and link tags containing external URL in the attributes (Apply rules and thresholds to generate value) | 0 | 3 | -1 | 1 | N/A | N/A |
| PctExtNullSelfRedirectHyperlinksRT | Counts the percentage of hyperlinks in HTML source code that uses different domain names, starts with “#”, or using “JavaScript ::void(0)” (Apply rules and thresholds to generate value) | 0 | 3 | -1 | 1 | N/A | N/A |
| CLASS\_LABEL | Classification of the URL | 0 | 2 | 0 | 1 | N/A | N/A |