Assignment 1 Project Report

Team members: Zixi Jiang, Peizhen Li, Xiaoyu Wang, Yuchen Zhang, Xiuwen Zhang, Nat Zheng
GitHub Repo: https://github.com/Anthonyive/DSCI-550-Assignment-1

Generating the dataset

After we converted the original text file into separated json files using Tika, we found the dataset a little messy. Within 4029 json files that we created, each file has different numbers and types of attributes. Several files are metadata of attachments of another email, and some files are a list of email addresses without any attributes. Secondly, since each of our team members did different tasks in task 5, the structures of the outputs were slightly different and we did not have a uniform standard for the unknown and missing values. We cleaned the data so that we have a standard format for every missing value and the same structure for all the outputs to be integrated together. Our final TSV file contains 28 features for each email.

Expanding the dataset

In this section, we provide some clarifications on our features generation process. In addition, we illustrate what kind of query or questions could be answered by these additional features.
We mainly used keywords scanning to determine the attack type. A few features are rather straightforward, being able to be extracted from the metadata or text content. Below are attributes that we would love to provide detailed explanation on how we obtained them.

- Attacker title: Since the attackers' titles are always at the beginning of their names, we found the attribute about the sender's name and retrieved the title by first splitting the name and taking the first word. For those senders' names without blank space, we made a list for the titles and searched to classify them.
- Urgency of the email: We used the Python NLTK package to tokenize text and count the frequency of the words "urgent" and "now" in each email, as well as the total frequency in all emails. We found that usage of urgency words are low in most fraudulent emails. The highest frequency of "urgent" and "now" in a single email is 5.
- Attacker location: We tracked the attacker location by extracting the IP address of the email and using an IP API to obtain the latitude and longitude for each IP. For the rest of the emails that do not provide an ip address, we imported the geoparser package to extract locations that are mentioned in the email content. The google geocode API provided us the latitude and longitude of each location.
- Attacker relationship: We used the model from "Deep Learning has (almost) all the answers: Yes/No Question Answering with Transformers", which uses a question-answering dataset called BoolQ from google research that contains 15942 samples of boolean questions. The model has an evaluation accuracy higher than 80%. The goal of using this model is to extract answers from a given boolean question combining with a given email content. We took all the affirmative answers and converted them into human-readable form.
- Attacker email sentiment: We used the method of analyzeSentiment from Google Cloud Natural Language API. For each email, we used the API to get a sentiment score. A negative score indicates Negative, a positive score means Positive and a score equals 0 means neutral.
- Attacker language style: We used Pyspellchecker package to check the email body and generated the number of wrong-spelled words. We detected wrong capitalizations in the text and calculated the ratio of the wrong_cap words to the total number of words.
- Attacker estimated age: We used USC Data Science AgePredictor with our pre-trained model built with Blog Authorship Corpus from kaggle. This dataset contains posts of 19,320 bloggers collected from blogger.com in August 2004. We used the age and text columns, separated the corpus into training, testing, evaluating with correspondingly 80%, 10%, 10%. The highest accuracy of the training set is 53.205% and 56.23% for the evaluation. Both accuracy scores are slightly higher than the model provided by the AgePredictor.

Features generated by joining external datasets

Dataset I: Prevalence of Selected Measures Among Adults Aged 20 and Over

How likely do the people that these attackers pretend to be have some diseases?

The MIME type of this dataset is application/xml. This data represents the ratio of high total cholesterol, hypertension, and obesity in different populations grouped by age from 1999 to 2018. We joined the datasets by the feature keys of date and predicted age. For each email, based on the predicted age of the attacker and the year that email was sent, we get the probability of the email sender having hypertension, obesity, and high total cholesterol respectively as three new features. These could help us see what type of people are the attackers faking to be. If the probability of the attackers having diseases is high, it might relate with those emails that use healthcare related excuses to deceive the victims. We could examine emails that mentioned certain diseases and dig the relationship between health related scams with the disease rate at that year.

- Feature 1: Hypertensions rate (the estimated probability of email sender having hypertension)
- Feature 2: Obesity rate (the estimated probability of email sender having obesity)
- Feature 3: High total cholesterol rate (the estimated probability of email sender having high total cholesterol)

It comes from data.gov as a XML file, published by National Center for Health Statistics, could be retrieved at https://catalog.data.gov/dataset/prevalence-of-selected-measures-among-adults-aged-20-and-over-united-states-1999-2000-2017

Dataset II: S&P 500 Historical Stock Exchanges

What is the stock price and volume on the specific day when the attacker sent the email?

The MIME type of this dataset is text/csv. This dataset contains historical stock data of the 500 largest U.S. publicly traded companies from 1998 to 2017, and we believe this somehow indicates the overall performance of the stock market on the specific day. The common attribute between this dataset and fraud email dataset is the date. We extracted three attributes: daily open price, close price and exchange volume from this dataset, merged them to the fraud email dataset based on the date. These three attributes help us to analyze the relationship between the performance of the stock market and the number of spam emails sent.

- Feature 1: open (stock' open price on the trading day)
- Feature 2: close (stock's close price on the trading day)
- Feature 3: volume (number of stocks has been trade on the trading day)

Reference: https://finance.yahoo.com/quote/%5EGSPC/history

Dataset III: Gender by Name Data Set by UCI Machine Learning Repository

What is the gender of the attacker?

The MIME type of this dataset is text/csv. The dataset combines raw counts for first/given names of male and female babies in those time periods, and then calculates a probability for a name given the aggregate count. We use the spacy library to do entity extraction to extract names from email contents. Then, we use the dataset as a look up table and see whether these corresponding name's genders are male, female or unknown. This dataset allowed us to answer what gender the attacker wanted to pretend when they were attacking the victim and potentially why they wanted to pretend to be like that. If the gender majority is female, maybe that is the majority of attackers wanted to pretend and they wanted to be approaching or obliging.

- Feature 1: male count
- Feature 2: female count
- Feature 3: unknown count (the count of extracted names that could not be identified by the dataset)

Dataset: https://archive.ics.uci.edu/ml/datasets/Gender+by+Name

Tika Similarity Analysis

Jaccard Similarity

x-coordinate	y-coordinate	Similarity_score
sep_by_email/3721.json	sep_by_email/2833.json	0.272727273
sep_by_email/3721.json	sep_by_email/729.json	0.206896552
sep_by_email/3721.json	sep_by_email/3371.json	0.166666667

Cosine Similarity

x-coordinate	y-coordinate	Similarity_score
data/3721.json	data/2833.json	0.910866934
data/3721.json	data/729.json	0.881448098
data/3721.json	data/3371.json	0.889105341

Edit Distance Similarity

x-coordinate	y-coordinate	Similarity_score
../sep_by_email/3721.json	../sep_by_email/2833.json	0.597851171
../sep_by_email/3721.json	../sep_by_email/729.json	0.588767631
../sep_by_email/3721.json	../sep_by_email/3371.json	0.547351754

These three images are snapshots of Jaccard Similarity, Cosine Similarity and Edit Distance Similarity. Jaccard Similarity overall provides the lowest score between two independent json files while Cosine Similarity presents the highest similarity score of two individual json files. Cosine similarity metric is more accurate since the cosine index is used to distinguish between plagiarized files in general. On the other hand, Jaccard index will be more reliable when identifying mirror objects, such as mirror sites, while Edit Distance metric is used to calculate the shortest distance converting one word to the other, which is not quite appropriate in this context. The relative high similarity score produced by Cosine metric indicates that the attributes and features that we extracted from these phishing emails are constructed with similar contents, structures and semantics.

Unintended consequences

As big data aims to improve efficiencies in social control and marketing, it could aid cyber attackers in targeting potential victims and locating individuals as well even if those data might be disparate and low-quality. Through exploring our additional datasets, we found that many similar public-accessible datasets could be easily utilized by attackers to form specific attacks to certains groups of people with generalities. For example, disease related datasets provide demographics of certain diseases, which could help attackers send out more personalized phishing attacks based on receivers' age and gender. Based on stock market fluctuation, attackers could schedule financial scams or spear phishing targeting equity investors or financial organizations. We believe that private data commons and accessibility of demographic data could be made use of by cyber attackers to improve efficiency and perform more specific attacks.

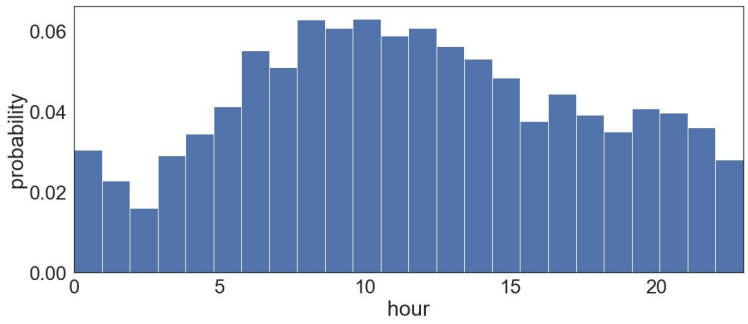
Deep thinking

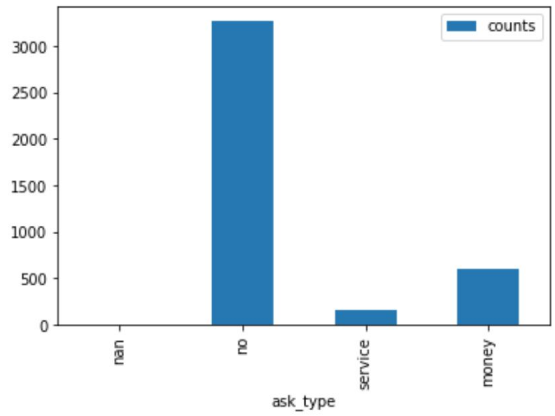
1. *Are there clusters of attackers with similar features that tend to attack victims the same way?*

Emails that have cosine similarity equaling to 0.91 share similar attack stylometrics in terms of the preference of saying now or urgent. They also have similar attacker relationships (mostly online, friend of friend and in person). Male mentionings are much higher than females. Attackers in earlier years have more misspelling and capitalization errors, and less overall vocabulary and grammar errors as years passed by, which might be thanks to improving text generation tools.

2. *Does the time of day of attack matter?*

In order to answer this question, we have drawn a histogram illustrating the time when these attack emails were sent. To make the plot clear, we picked up the nearby hour for each time point. According to the plot at the bottom left, the attackers are more likely to send emails during 8 a.m to 12 p.m in a random day. It is 1.5 times more likely for them to send emails in the morning compared to the afternoon. As a result, the time of day would be a minor matter.



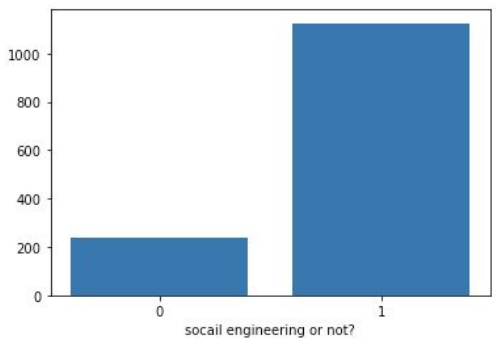


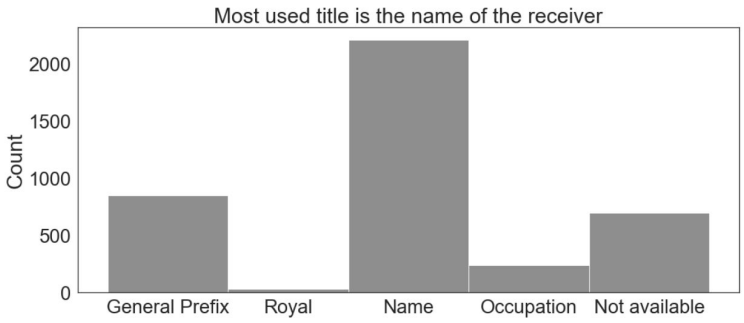
3. *Are specific types of asks more prevalent?*

As shown from the graph at the top right, the ask_type “no” constitutes the majority of the database. Asking for money or providing money is more prevalent than providing service within these scamming emails. After analysing this question, we realized that we should include more keywords that are sensitive to any type of services provided or any kinds of money transaction to improve the accuracy of “ask_type” attributes.

4. *Is there a set of frequently co-occurring features that induce the email to be read?*

Attack type and title combined would induce the email to be read. We found all the emails that were read by the victims and checked if the emails were with social engineering or not. According to the bar plot, among about 1400 emails, over 1000 of them had social engineering. In addition, when the title is the name of the email receiver, probability of the email being read increases. Combined, when the email takes a social engineering approach and uses the receiver’s name as title, it is much more likely that the email will be read.



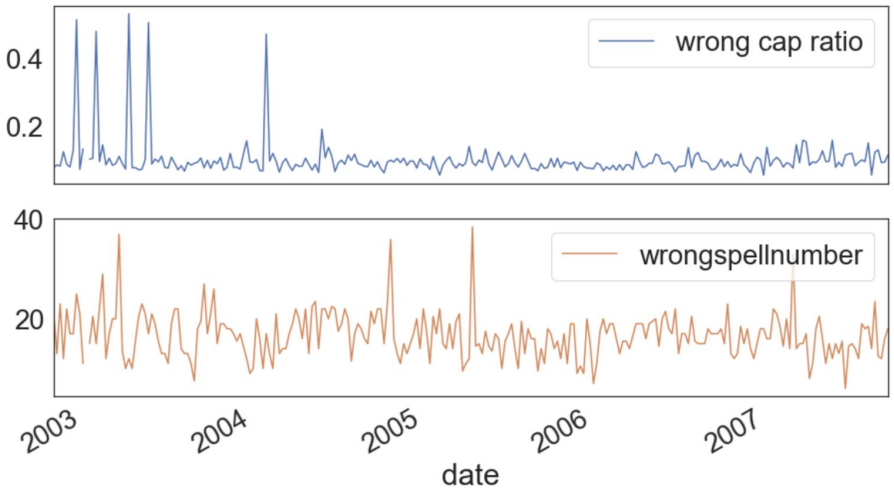


5. *What insights do the “indirect” features you extracted tell us about the data?*

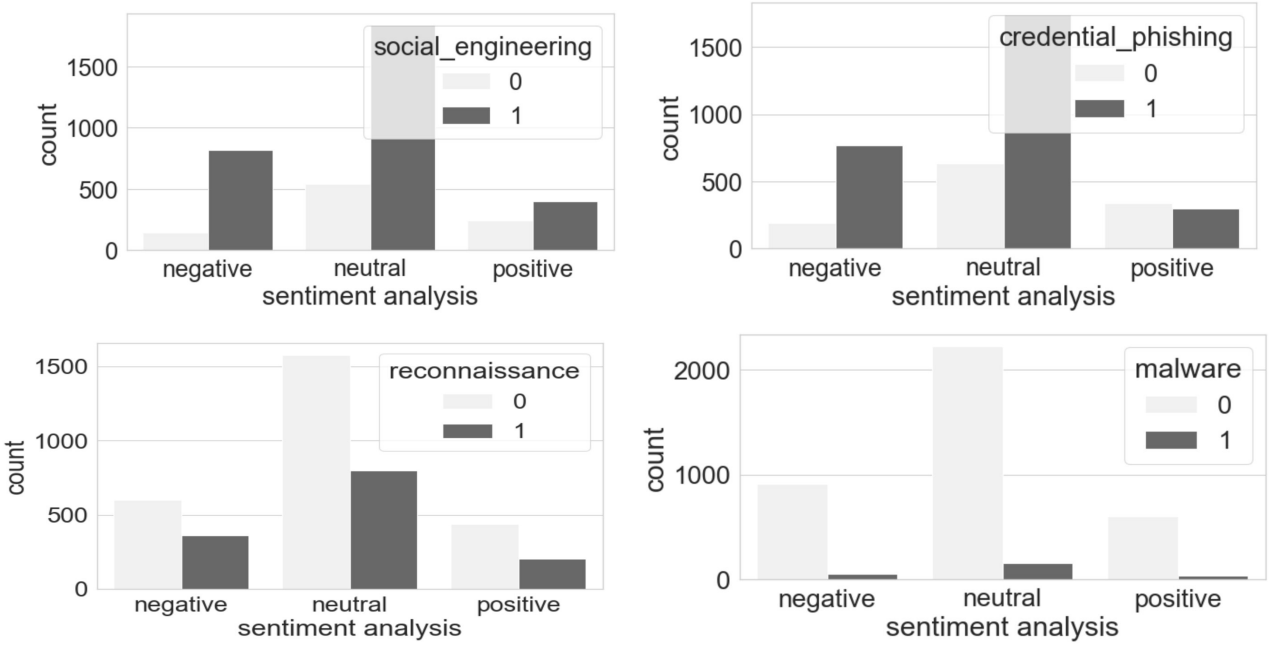
Language Style:

As seen on the following graph, we plotted the time-series data of language errors from 2002 to 2007 in the email content. During 2002 to 2004, several emails displayed significant wrong capitalization ratios, as seen in the top graph represented by the peaks, which indicates potential random capitalizations. As time passes by,

the wrong cap ratio has become stable and thus telling us that the attackers are making less basic language style mistakes. On the other hand, the wrong spell number in each email doesn't seem to decrease significantly over time. One of the potential reasons might be that these emails contain other-than-English content or weird word choices that would be identified as misspelling by the up-to-date spellchecker we used to generate the feature. These features suggest that phishing emails did not demonstrate significant improvement in their language style over the period of 2002 to 2007.



Attack type and sentiment analysis:
We plotted the four types of attack against the content sentiment. While credential phishing and social engineering both dominated the most of the email scams, their sentiment is slightly different. When the content sentiment is identified as positive, it has a slightly higher chance to be social engineering than credential phishing. Overall, most emails, regardless of their attack type, have neutral or negative sentiment, which might indicate that attackers prefer to use either neutral or threatening/upsetting content to deceive the victims.



6. *What clusters of Attacks and Attacker stylometrics made the most sense? Why?*

From the perspective of Attacks, social engineering made the most sense, using "friendship", "intimate" relationship or any types of urgent words can easily attract victims' attention. Reconnaissance is not capable of

explicitly indicating whether victims fell into the trap or not, while malware and credential phishing will quickly alarm victims' cautiousness. Attackers are likely to send out emails during the day time since there are more people to check their emails from morning to noon. The website to check the attacker IP is a credible source and therefore checking the sender's IP is a sustainable and plausible way to detect a phisher.

7. *Also include your thoughts about Apache Tika – what was easy about using it? What wasn't?*

For people like us who are not familiar with Java, Apache Tika has some drawbacks in terms of ease of use. We have some experience with Bash/Zsh, and thus figuring out paths is relatively easy. However, although tika has a GUI, the GUI cannot process large files, which means that Tika is not like the type of programs/applications that just need to be downloaded and "plug-and-play".

In addition, Tika-generated-output sometimes gives us unexpected results. For instance, we fed the entire fraudulent email dataset into Tika, and Tika generated around 20 attributes for each email, which were normal. However, when we concatenated all the outputs generated by Tika, we surprisingly had thousands of attributes. Some of them seem to be wrong keys. On the other hand, Tika is surely powerful in parsing files and extracting metadata. That being said, Tika actually did what it needed to do and was fast to implement.

Contribution

Yuchen Zhang ● Task 4: Tika data generation ● Task 5b vi, ix: Attacker relationship and attacker estimated age ● Task 6, Dataset III (Name gender): Data generation ● Task 7, Tika similarity: generate edit-distance similarities and circle-packing visualization ● Miscellaneous: README.md, package management.	Xiuwen Zhang ● Task 5a: reconnaissance, social engineering, malware, credential phishing ● Task 5b iv, x: what does attacker offer, ip risk score and level ● Task 6 Dataset II: data-cleaning, extract attributes and merge new features into email dataset ● Task 7: Tika similarity: generate cosine similarity ● Attribute analysis
Nat Zheng ● Task 5b v: locate attacker's location by ip and email content ● Task 6 Dataset II : scrape S&P's historical stock data from 1998 to 2017 on Yahoo Finance ● Attribute analysis & visualizations	Zixi Jiang ● Task 5b iii, viii: extract data/time attributes from the metadata; calculate wrong capitalization ratio and grammar errors ● Task 6 Dataset I: find obesity dataset and identify relevant attributes ● Visualizations & attribute analysis & unintended consequences
Peizhen Li ● Task 5b iii: Urgency of the attack email (word strength, "urgent", "now") ● Task 6 Dataset III(Name gender): Dataset finding(We found shooting incident data at first ,but we changed to name and gender data); Extract data ● Attribute analysis & visualizations	Xiaoyu Wang ● Task 5b i,vii,viii(1): attacker title, email sentiment, misspellings ● Task 6 Dataset I: convert json file to usable attributes ● Task 7: Tika similarity: generating jaccard_similarity ● Attribute analysis & visualizations