Daniel Trevino dtrevi01@syr.edu

Hacking Tweets Classification via Sentiment

Topic:

Social Media (Twitter) API query and Data Analysis using python and Machine Learning Techniques

Purpose:

About once a month, a new computer exploit is released that revolutionizes the computer security industry. In order to gain street credibility and good face with accredited security experts, the group or person releasing the exploit(s) turns to social media to express why they are releasing their exploit. A primary location where the information is released, is Twitter. Shortly after releasing exploits that are damaging, twitter quickly turns to deactivating the account. The most recent account of this happening is from @dookhtegan in the middle of March 2019(1). Quickly identifying these tweets or predicting the tweeted information will help validate if more conspicuous tweets are valid and damaging before twitter removes access.

Problem:

Can sentiment analysis be used to classify tweets as being a real computer exploitation tweet or is it just a fake tweet (looking for original tweets)?

Dataset:

Tweets are collected and stored into MongoDB, then extracted and cleaned. The following snippet is a snapshot of the dataframe after the tweets are cleaned in python with Pandas.

	user.screen_name	user.name	user.statuses_count	text	created_at	user.created_at	lang	user.description	user.favourites_count	user.followers_count	...	user.profile_use_background_image	user_created	user_created_days	tweets_per_day	rule1	rule2a	rule2b	rule3	rule4
0	ghumakkadnitish	Nitish Kaushal IN	277	RT @DimpleAtra: Dear @Twitter @TwitterIndia @T...	Fri Apr 12 04:44:44 +0000 2019	Sun May 07 04:57:03 +0000 2017	en	3 decades old, opinionated, Nationalist	150	22	...	True	2017-05-07 04:57:03	707	0.391796	0	0	0	0	0
1	other95	other95	94635	RT @BulletinAtomic: An elite North Korean hack...	Fri Apr 12 04:44:45 +0000 2019	Fri Jun 05 23:56:06 +0000 2009	en	Anti-Nuke, Academic, Liberal/Progressive, Femi...	19437	695	...	False	2009-06-05 23:56:06	3600	26.287500	0	0	0	1	0
2	DharmFirst	|| धर्मो रक्षति रक्षितः ||	20723	RT @DimpleAtra: Dear @Twitter @TwitterIndia @T...	Fri Apr 12 04:44:46 +0000 2019	Thu Jan 21 14:01:36 +0000 2010	en	Travel, Photography, Football, Politics, Socio...	80397	161	...	True	2010-01-21 14:01:36	3370	6.149258	0	0	0	0	0
3	AngelaDobbins12	Angela Dobbins	16255	RT @JenKirkman: Assange indicted for a hacking...	Fri Apr 12 04:44:47 +0000 2019	Sat Oct 28 17:19:33 +0000 2017	en	Hillary Clinton is the #LegitimatePOTUS #Alway...	15461	379	...	True	2017-10-28 17:19:33	533	30.497186	0	0	0	0	0

As seen above, the data contains the user screen name, username, how many times they have tweeted something (statuses_count), and some further basic information about the user such as, when the account was created and when the tweet capture was created. The user_created_days column is how many days their account has been active since the tweets were collected (14 April 2019) and the tweets_per_day is calculated by dividing the statuses_count / tweets_per_day. The rules will be discussed later this report.

Data Acquisition and Analysis Techniques:

To collect the data for analysis, twitter tweets are used. Over 15k tweets were collected on the following terms:

['Hacking', 'Webshell','F5 Exploit','ZeroDay', '0day', 'WPA3 Exploit', 'router exploit', 'Novidade', 'VPNFilter', 'SMB Exploit', 'cisco exploit','استغلال F5', 'استغلال سيسكو', 'قذيفة الويب','قرصنة الكمبيوتر', 'Компьютерный Эксплоит', 'IoT Exploit']

The tweets collected are taken from the MongoDB collection, then normalized from their JSON like (BSON) structure and analyzed based on several rules developed via two techniques. The first technique is through research papers (see references) and the second technique is parsing through roughly 500 historical tweets from computer security researchers. After developing the rules, the data collected is transformed into testable information. The rules developed are are to classify a tweet as being a bot (not a real user) and not a bot (a person related to the computer security industry. Following are the rules developed:

1. Tweets per day > 50 are classified as a bot && Tweets per day <= 50 as a person (2)
2. user.screen_names and user.name contain more than 4 numbers, binary value returned
3. If profile_use_background_image is False AND user.favourites_count > 2000 OR user.description == None
4. If profile_use_background_image is True AND user.friends_count > 2000 OR user.description == None
5. If Rule(1) or Rule(2) or Rule (3) or Rule(4) are == 1 (True) then Is a bot else ==0

These rules create either an integer or a float64 based on the calculation. After all tweets are classified, initial descriptive statistics are gathered, then sentiment analysis on the twitter text is performed. The sentiment analysis tool used is VADER and the sentiment is compared to the rule classification of a tweet being a bot or not a bot. After the initial sentiment is identified, the data is then analyzed for linearity and as well prediction modeling with Machine Learning techniques via SKLearn.

Development Tasks, Tests, and Outputs:

(1) The first step involved establishing a host-based MongoDB server for logging tweets via a streaming twitter API call. To run the server, MongoDB is downloaded and then the pymongo package is used. The MongoDB logged over 15k tweets over the course of 2 days. The stored tweets (as BSON) allowed for faster access in pulling

```
import pymongo
from pymongo import MongoClient

#creates a new mongodb database
#creates a connection to the mongoDB on local host
client = MongoClient()

#allow access to the MongoServer Scripting Database
#db = client.Scripting

#the following also works
#selects the database
db = client['Hacking_Tweets']
#selected the collection to call
collection = db.Final_Project_Master
#list names in the collection
db.list_collection_names()

['Final_Projectv3', 'Final_Projectv4', 'Final_Projectv2', 'Final_Project']
```

and analyzing the tweets. The collections are seen to the right in the image.

(2) The Twitter API allowed for accessing twitter content via a developer account. Making calls for the content involved using the tweepy python package. Several tests were run to collect the tweets. The first test pulled in specific tweets based on the individual topics listed above. This test failed due to not all twitter fields being collected as part of the data pull. The second test utilized the API to call on specific topics dealing with computer hacking. This test failed and became too cumbersome for processing. The third test resulted in using the Twitter API and tweepy to build a streaming tweet capture that stores the tweets directly into MongoDB. This tweet collection (Projectv3) is the set of tweets used for the analysis of the remaining project.

(3) Testing of the twitter text for sentiment analysis utilized the VADER package. The output of each tweet is classified as neutral, positive, or negative. The sentiment analysis is compared to the tweet being a bot or not a bot. The output of the VADER sentiment comparison is listed and described below.

(4) The final task is to utilize the Sklearn package and assess the tweets based on the rules and sentiment analysis. The models used are linear regression, SVM, and AdaBoost. In order to perform these test, a sample of the original dataset is taken. The output of the models is further discussed below.

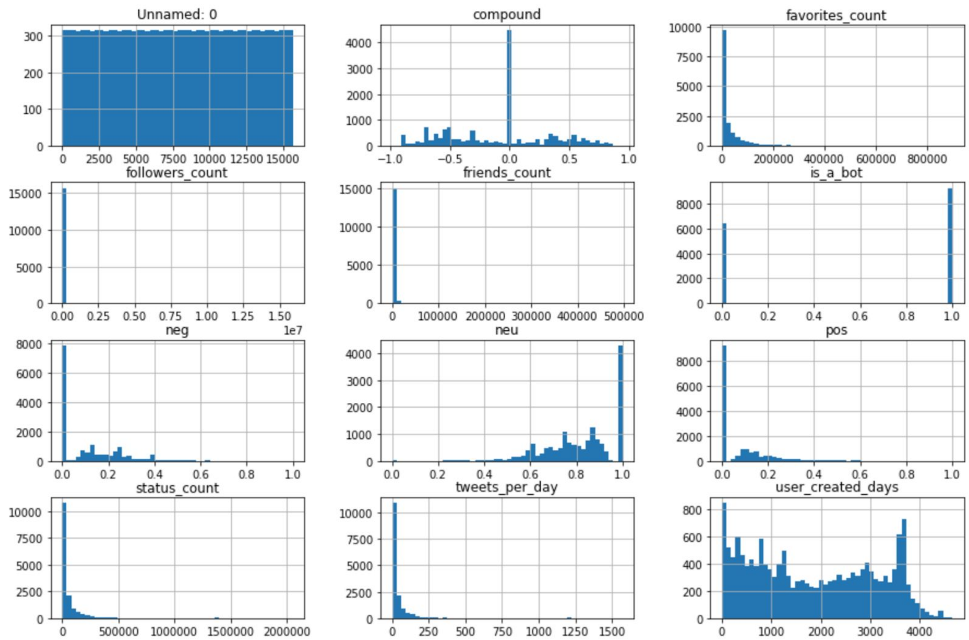
Rules Generation

The rules approach for classifying tweets provided the quickest solution to assessing the 15k tweets. Classifying twitter accounts as a bot or real account took several hours of analysing real cyber security researchers twitter accounts, such as https://twitter.com/thegrugq. The information gathered from the tweets was then coded into rules using python in the Jupyter notebook. Below is an example of rule 2 where regex is used to analyze if a screen name or a username has more than 4 numbers in it. Many of the tweets and retweets identified as not containing real hacking data in the text had 4 or more number in the username or screen name fields. This rule, like the others quickly classified the accounts.

```
#Rule 2 screen.names contains more than 4 numbers
import re

# Function to extract all the numbers from the given string
def getNumbers(str):
    '''Function to grab each number from a string, if >= 4 then returns (1)true, of not, returns (0)false'''
    #find all numbers in the string
    array = re.findall(r'[+-]?\d+(?:\.\d+)?', str)
    #create a list of all the numbers seperated by each number
    array = [list(num) for x in array for num in x]
    #return True if the length of all numbers in the list is 4 or greater or False if less.
    return 1 if len(array) >= 4 else 0
```

After classifying the accounts, there were 6404 real account and 9231 bot accounts. More analysis is needed to further narrow the bot accounts more, as after classifying the accounts several real accounts were identified as being a bot (about 1 in 25). Next are some basic descriptive statistics to help visualize the data and generate further insight into the tweets. The following charts are histograms. They show the data does not follow any relative standard distribution.



Word Cloud Analysis

The intent of the following section is to visualize the text in the tweets, first as a whole, then separated between tweets classified as a real account and tweets classified as a bot. The below word cloud represents the tweets collectively. The majority of tweets collected for this analysis happened during the Julian Assange indictment and some of the words related to the event have been removed.

This word-cloud illustrates the majority of the tweets have almost nothing to do with computer hacking. Re-evaluating the word cloud on tweets classified as a real account lead to the following word cloud below. The below word cloud on the left does not illustrate much difference in the words a tweet contains between the original dataset. Likewise, the word cloud on the right does not illustrate much difference in the tweets. Thus, the assumption, given a set of tweets, that the wording context will be different (based on word clouds) is false.

Vader Sentiment Analysis

The intent behind this analysis is to identify if sentiment may classify a twitter bot or not. To evaluate the sentiment, VADER is used on all translated tweets. Then comparing the sentiment between account classified by the rules as a bot or a real account. The first assessment is to look at all the descriptive statistics for the whole data set. As seen below, all tweets together results in a neg average of 0.10, a neutral score of 0.82, and a positive score of 0.07. Thus, the overall status of all the hacking tweets leads to a neutral sentiment, while also being more negative than positive. Furthermore, looking at the tweets classified only as real hacking accounts results in a less negative tone by .07 as compared to the Bot and All Tweets. Additionally, real accounts have a slightly more positive sentiment towards hacking than tweets classified as bots.

All Tweets

	neg	neu	pos
count	15635.000000	15635.000000	15635.000000
mean	0.102209	0.824223	0.071657
std	0.125847	0.154091	0.107912
min	0.000000	0.000000	0.000000
25%	0.000000	0.726000	0.000000
50%	0.000000	0.842000	0.000000
75%	0.194000	1.000000	0.124000
max	1.000000	1.000000	1.000000

Real Tweets

	neg	neu	pos
count	6404.000000	6404.000000	6404.000000
mean	0.093263	0.828027	0.076685
std	0.124374	0.156769	0.113859
min	0.000000	0.000000	0.000000
25%	0.000000	0.729000	0.000000
50%	0.000000	0.850000	0.000000
75%	0.173000	1.000000	0.136000
max	0.791000	1.000000	1.000000

Bot Tweets

	neg	neu	pos
count	9231.000000	9231.000000	9231.000000
mean	0.108416	0.821584	0.068169
std	0.126495	0.152159	0.103449
min	0.000000	0.000000	0.000000
25%	0.000000	0.725000	0.000000
50%	0.085000	0.842000	0.000000
75%	0.207000	1.000000	0.119000
max	1.000000	1.000000	1.000000

Further Analysis and Predicting Hacking bot Account

To develop a more comprehensive approach to the accounts, we will use the master_df tweets with all 15635 tweets. These tweets are then tested with linear regression to identify if there is any linear relationship in sentiment helping to classify a bot. The resulting r-square output is 0.1331. This linear regression test is the first of the three techniques used in evaluating if sentiment can lead to classifying a false tweet (by a bot). Regardless of the

```
The coefficient for Unnamed: 0 is -1.9048958224724964e-06
The coefficient for status_count is 2.422038792865136e-07
The coefficient for favorites_count is 1.3058294128186548e-06
The coefficient for followers_count is 2.5350803940230513e-08
The coefficient for friends_count is 9.898796806594612e-06
The coefficient for user_created_days is -2.401136789504417e-05
The coefficient for tweets_per_day is 0.000757190906285606
The coefficient for compound is -0.12313403854452368
The coefficient for neg is -0.16258496727806124
The coefficient for neu is -0.0012541173409391711
The coefficient for pos is 0.18505166633405523
```

r-squared score, the coefficient that weighs the most of a classification of accounts is the positive sentiment of a tweet. The statistical significance of the r-square value is not enough to make a valid connection, but looking at the VADER sentiment analysis section above, we also see that the one distinguishing factor in identifying a tweet's validity is how positive the tweet is in comparing real tweets and bot tweets.

The next two classification models are the SVM Model and the AdaBoost model. The data used in these model will be the collection of tweets already classified by the rules and labeled as master_df.

```
In [121]: master_df.shape
Out[121]: (15635, 20)

In [125]: master_df.head(1)
Out[125]:
```

	screen_name	name	status_count	text	created_at	user_created_at	language	description	favorites_count	followers_count
0	ghumakkadnitish	Nitish Kaushal IN	277	RT @DimpleAtra: Dear @Twitter @TwitterIndia @T...	Fri Apr 12 04:44:44 +0000 2019	Sun May 07 04:57:03 +0000 2017	en	3 decades old, opinionated, Nationalist	150	22

Performing the support vector machine test on the master_df data, the SKLearn package is used. For the purpose of developing a better model, the following features are dropped for the dataframe:

filtered_features = ['screen_name', 'name', 'text', 'created_at', 'user_created_at', 'language', 'description','translated_text']

The remaining features will all be used to classify 'is_a_bot'. The new dataframe is shown below and includes only integers and float values. This resulting data frame is used for the Support Vector Machines and Adaptive Boost model for comparison. These two classifiers are selected because the data is not linear and many of the calculated rules are correlated.

	status_count	favorites_count	followers_count	friends_count	user_created_days	tweets_per_day	is_a_bot	compound	neg	neu	pos
0	277	150	22	223	707	0.391796	0	0.3818	0.000	0.874	0.126
1	94635	19437	695	1209	3600	26.287500	1	0.0258	0.000	0.952	0.048
2	20723	80397	161	193	3370	6.149258	0	0.3818	0.000	0.874	0.126
3	16255	15461	379	403	533	30.497186	0	-0.8720	0.382	0.618	0.000
4	389	98	14	32	261	1.490421	0	-0.2960	0.104	0.896	0.000
5	41698	8	524	1713	3029	13.766259	1	-0.3612	0.147	0.785	0.068
6	56906	10754	1173	4999	2419	23.524597	1	0.1779	0.174	0.631	0.196
7	140131	211631	6931	7012	3531	39.685925	1	0.0000	0.000	1.000	0.000
8	2881	4391	2449	2476	3636	0.792354	1	0.2732	0.136	0.679	0.186
9	19567	12450	8943	9779	434	45.085253	1	-0.5267	0.284	0.597	0.119

Support Vector Machines (SVM)

A sample is taken of 4000 of the 15000 values. After two separate sample iterations of running SVM, the best model produced an accurave of 0.655. This SVM model performed

```
Accuracy: 0.655
Precision: 0.6756410256410257
Recall: 0.7659883720930233
```

the best in the test with of classification algorithms, resulting in a ~65% rating of tweets being classified as real and containing legitimate hacking text. Below are the excerpts from the SVM model. This model, of the 3 (linear, SVM, AdaBoost) performed the best at predicting the classification of tweets. Further feature selection and more tweet classifying may make the model more accurate.

```
Accuracy:0.667

Classification report
              precision    recall  f1-score   support

           0       0.61      0.53      0.57      1152
           1       0.70      0.76      0.73      1648

   micro avg       0.67      0.67      0.67      2800
   macro avg       0.65      0.65      0.65      2800
weighted avg       0.66      0.67      0.66      2800

Confusion matrix
[[ 612  540]
 [ 391 1257]]
```

Adaptive Boost (Ada Boost)

Similarly, a sample is taken of 4000 out of the 15000 tweets. After two separate sample iterations of running AdaBoost, the best model produced an accurave of 0.573. This model did not outperform the the SVM prediction model. Additionally, as seen on the right the confusion matrix is confused as there are no true negatives, only false positives and true positives. Thus, the Ada Boost model can only predict tweets as bots roughly 57% of the time but cannot classify the correctly identified text, only text via a bot posting.

```
Accuracy: 0.5733333333333334
Precision: 0.5735785953177257
Recall: 0.997093023255814
```

```
Accuracy:0.589

Classification report
              precision    recall  f1-score   support

           0       0.00      0.00      0.00       572
           1       0.59      1.00      0.74       828

   micro avg       0.59      0.59      0.59      1400
   macro avg       0.30      0.50      0.37      1400
weighted avg       0.35      0.59      0.44      1400

Confusion matrix
[[  0 572]
 [  3 825]]
```

Conclusion

The analysis performed through the experimentation process involved learning how to use python, tweepy, sklearn, and pandas for analyzing tweets. The tweets collected over a 48 hour period covered a range of hacking exploit topics. After the tweets were collected, rules were generated covering several tests to identify if a Twitter account is real or a bot. After the rules were computed, tweets were classified as a bot (1) or not a bot (0). After the rules were calculated, all the tweet text was translated into english via google translate. Then sentiment analysis was performed on the tweets text and several prediction models were performed. The model with the highest prediction is the SVM model. The SVM model returned roughly a 65% accuracy on classifying tweets as a real account tweeting or bot tweeting. The SVM model performed the best due to the many hyperplanes within the data. Additionally, the linear model fails because there is to much correlation within the data based on how the rules were computed.

In conclusion, the model is successful, but there are several further tests that may be performed to enhance the predictability of the model. Additionally, visualizing the network from the real accounts with NetworkX will enhance understanding of the design and lend further proof to the model. As a result based on the tests and models, tweets with more positive text tend to be more valid in identifying if an account is posting information about a hacking exploit. Thus, using this information and further development will help predict if a twitter user is posting about a hacking exploit.

References

1. Greenberg, Andy. “A Mystery Agent Is Doxing Iran's Hackers and Dumping Their Code.” Wired, Conde Nast, 18 Apr. 2019, www.wired.com/story/iran-hackers-oilrig-read-my-lips/

2. Moo-Mena, F., Robles-Sandoval, S., González-Magaña, K., & Rodríguez-Adame, O. (2019). Towards bots detection by analyzing the behavior of user data on twitter. International Journal of Computer Science Issues (IJCSI), 16(1), 21-29. doi:http://dx.doi.org.libezproxy2.syr.edu/10.5281/zenodo.2588241

3. Cook, D. M., Waugh, B., Abdipanah, M., Hashemi, O., & Rahman, S. A. (2014). Twitter deception and influence: Issues of identity, slacktivism, and puppetry.Journal of Information Warfare, 13(1), 58-71,IV. Retrieved from https://search-proquest-com.libezproxy2.syr.edu/docview/1966852043?accountid=14214

4. Paavola, J., Helo, T., Jalonen, H., Sartonen, M., & Huhtinen, A. (2016). Understanding the trolling phenomenon: The automated detection of bots and cyborgs in the social media. Journal of Information Warfare, 15(4), 100-111,III-V. Retrieved from https://search-proquest-com.libezproxy2.syr.edu/docview/1968019702?accountid=14214

5. #BotSpot: Twelve Ways to Spot a Bot. (2017, August 28). Retrieved from https://medium.com/dfrlab/botspot-twelve-ways-to-spot-a-bot-aedc7d9c110c

6. Mjrovai. (n.d.). Mjrovai/Python4DS. Retrieved from https://github.com/Mjrovai/Python4DS/blob/master/Almost_Real_Time_Twitter_Sentiment_Analysis/almost_real_time_twitter_sentiment_analysis_EXT.ipynb

7. S. S. (n.d.). Scraping my Twitter Social Graph with Python and Selenium. Retrieved from https://dev.to/swyx/scraping-my-twitter-social-graph-with-python-and-selenium--hn8

8. KDnuggets. (n.d.). Retrieved from https://www.kdnuggets.com/2017/03/beginners-guide-tweet-analytics-pandas.html

9. Al-Masri, A. (2019, February 13). Creating The Twitter Sentiment Analysis Program in Python with Naive Bayes Classification. Retrieved from https://towardsdatascience.com/creating-the-twitter-sentiment-analysis-program-in-python-with-naive-bayes-classification-672e5589a7ed

10. Sistilli, A. (2017, June 07). Twitter Data Mining: A Guide to Big Data Analytics Using Python. Retrieved from https://www.toptal.com/python/twitter-data-mining-using-python

11. (2015, December 09). Retrieved from https://mran.microsoft.com/snapshot/2016-01-05/web/packages/mongolite/vignettes/intro.html

12. Brown, E. D. (2017, January 19). Collecting / Storing Tweets with Python and MongoDB. Retrieved from https://pythondata.com/collecting-storing-tweets-with-python-and-mongodb/