Applying Data Science to Cybersecurity – A Gentle Introduction

Arun Viswanathan

4/20/2020

CSE351 – Introduction to Data Science, Spring 2020

SUNY Korea

1

Who am I?

- Cybersecurity Researcher @ NASA JPL
- Ph.D. in Computer Science from University of Southern California, 2015
- Experience across industry and academia
- Use AI/ML to solve cybersecurity problems for JPL's space missions

2

Agenda for Today

- Quick overview of cybersecurity
- Cybersecurity data science
- Cybersecurity use cases
- Our tools for today
- Demo: Using clustering to detect malicious activity using Python + ML libraries
- Suggested next steps / class projects

3

What you will hopefully learn from today's presentation?

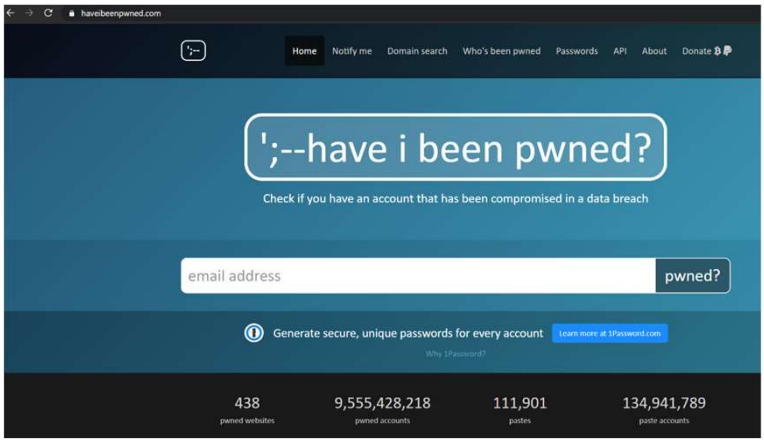
CYBERSECURITY CHALLENGES CONTINUE TO GROW, WITH LOTS OF JOB OPPORTUNITES

DATA SCIENCE IS A KEY TOOL IN OUR ARSENAL TO DEFEND AGAINST CYBER THREATS

LEARN HOW TO APPLY PYTHON + ML LIBRARIES TO A REAL CASE STUDY

4

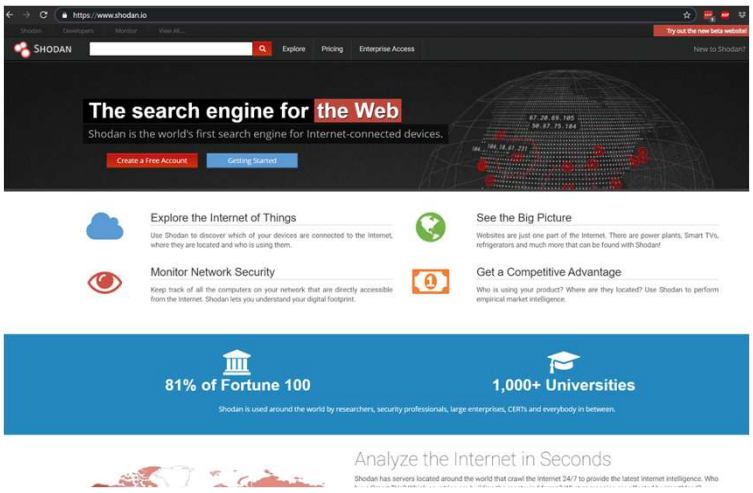
Cybersecurity: A gentle introduction



Go to https://haveibeenpwned.com/ and enter your email address to test if your email address was affected by a data breach!

5

Cybersecurity: A gentle introduction



Go to https://www.shodan.io/ and enter the IP address of a server you know, or search using a string such as “printer”!

6

Current State of Cybersecurity!

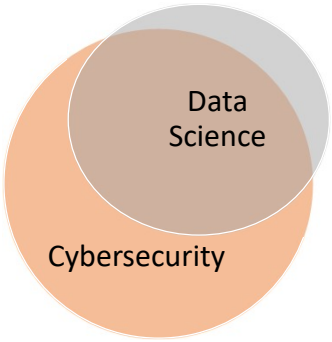
- Data breaches exposed 4.1 billion records in the first half of 2019. (RiskBased)
- 52% of breaches featured hacking, 28% involved malware and 32–33% included phishing or social engineering, respectively. (Verizon)
- Hackers attack every 39 seconds, on average 2,244 times a day. (University of Maryland)
- 53% of companies had over 1,000 sensitive files open to every employee. (Varonis)
- The financial services industry takes in the highest cost from cybercrime at an average of $18.3 million per company surveyed. (Accenture)
- The industry with the highest number of attacks by ransomware is the healthcare industry. Attacks will quadruple by 2020. (CSO Online)

- By 2021, it's projected that there will be 3.5 million unfilled cybersecurity jobs globally. (Cybersecurity Ventures)
- The cybersecurity unemployment rate is 0% and is projected to remain there through 2021. (CSO Online)

Source : https://www.varonis.com/blog/cybersecurity-statistics/

7

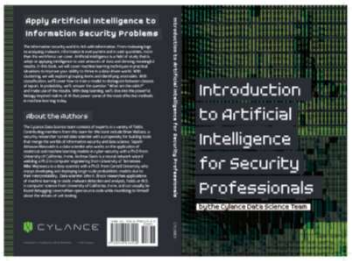
Cybersecurity Data Science

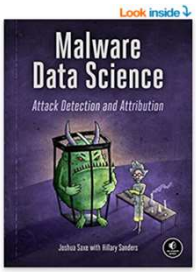


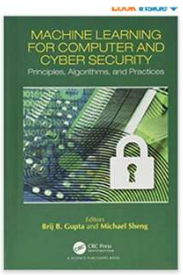
Apply data science techniques to detect, diagnose and remediate cyber security threats.

A good cybersecurity data scientist should possess detailed knowledge of both the cybersecurity and data science domains, be curious and be skeptical!

A couple of books to get started!







8

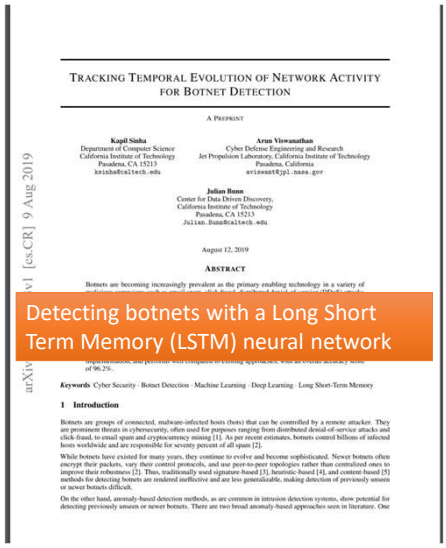
Data Science Use Cases for Cybersecurity

- INTRUSION DETECTION
- ANOMALY DETECTION
- SPAM CLASSIFICATION
- NETWORK TRAFFIC ANALYSIS
- MALWARE DETECTION
- BINARY ANALYSIS
- CYBER THREAT HUNTING
- MALICIOUS DOMAIN DETECTION

9

Some Examples

(with code available on github)



TRACKING TEMPORAL EVOLUTION OF NETWORK ACTIVITY FOR BOTNET DETECTION

A PREPRINT

Kapil Sinha
Department of Computer Science
California Institute of Technology
Pasadena, CA 15213
ksinha@caltech.edu

Arun Viswanathan
Cyber Defense Engineering and Research
Jet Propulsion Laboratory, California Institute of Technology
Pasadena, California
arviswan@jpl.nasa.gov

Julian Bunn
Center for Data Driven Discovery,
California Institute of Technology
Pasadena, CA 15213
Julian.Bunn@caltech.edu

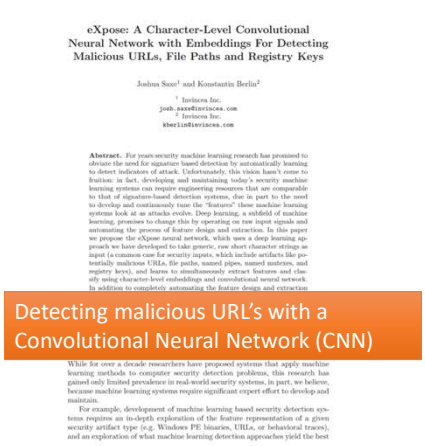
August 12, 2019

ABSTRACT

Botnets are becoming increasingly prevalent as the primary enabling technology in a variety of

arXiv:1908.03511v1 [cs.CR] 9 Aug 2019

Detecting botnets with a Long Short Term Memory (LSTM) neural network



eXpose: A Character-Level Convolutional Neural Network with Embeddings For Detecting Malicious URLs, File Paths and Registry Keys

Joshua Saxe[1] and Konstantin Berlin[2]

[1] Invincea Inc.
josh.saxe@invincea.com
[2] Invincea Inc.
kberlin@invincea.com

Abstract. For years security machine learning research has promised to obviate the need for signature based detection by automatically learning to detect indicators of attack. Unfortunately, this vision hasn't come to fruition: in fact, developing and maintaining today's security machine learning systems can require engineering resources that are comparable to that of signature-based detection systems, due in part to the need to develop and continuously tune the "features" these machine learning systems look at as attacks evolve. Deep learning, a subfield of machine learning, promises to change this by operating on raw input signals and automating the process of feature design and extraction. In this paper we propose the eXpose neural network, which uses a deep learning approach we have developed to take generic, raw short character strings as input (a common case for security inputs, which include artifacts like potentially malicious URLs, file paths, named pipes, named mutexes, and registry keys), and learns to simultaneously extract features and classify using character-level embeddings and convolutional neural network. In addition to completely automating the feature design and extraction

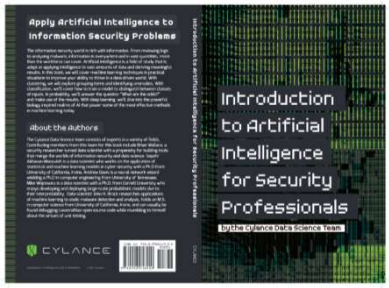
While for over a decade researchers have proposed systems that apply machine learning methods to computer security detection problems, this research has gained only limited prevalence in real-world security systems, in part, we believe, because machine learning systems require significant expert effort to develop and maintain.

For example, development of machine learning based security detection systems requires an in-depth exploration of the feature representation of a given security artifact type (e.g. Windows PE binaries, URLs, or behavioral traces), and an exploration of what machine learning detection approaches yield the best

Detecting malicious URL's with a Convolutional Neural Network (CNN)

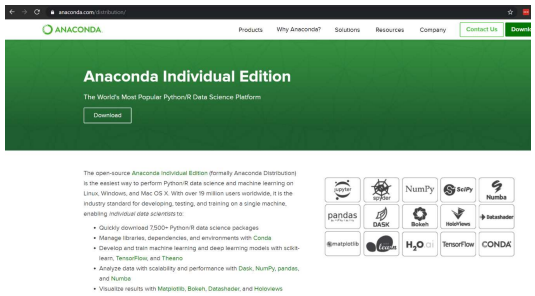
10

Our tools for today



https://github.com/cylance/IntroductionToMachineLearningForSecurityPros

The github link contains free PDF for the book along with source code and data sets.

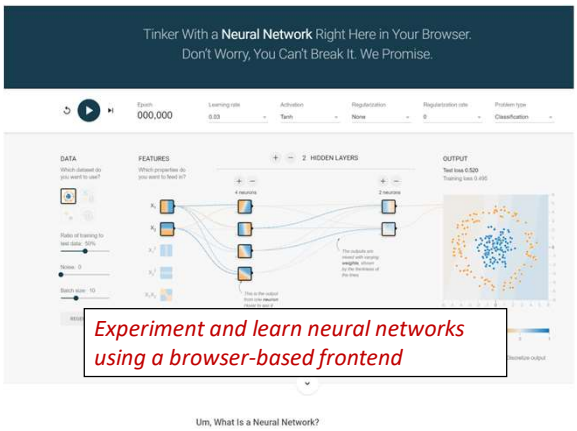


Install anaconda on your Mac, Windows or Linux.

11

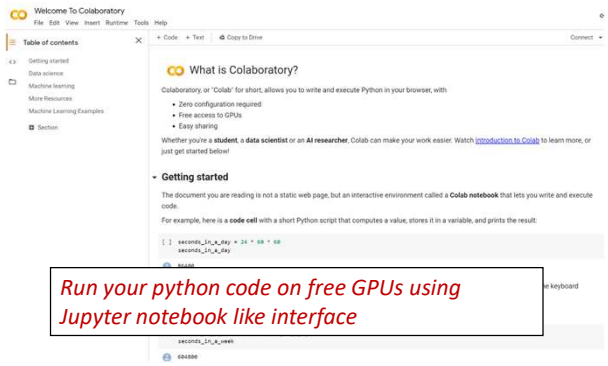
Other Free Tools to Explore

https://playground.tensorflow.org/



Experiment and learn neural networks using a browser-based frontend

https://colab.research.google.com/



Run your python code on free GPUs using Jupyter notebook like interface

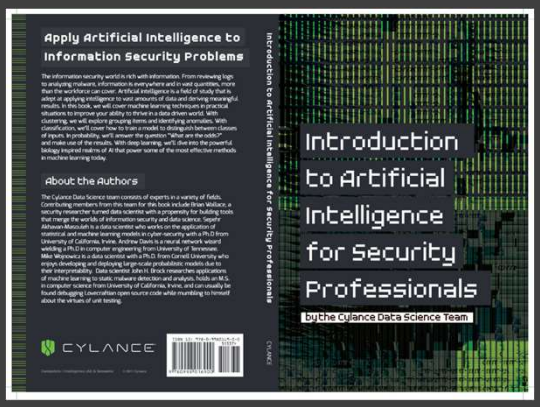
12

Cyber Security Case Study

Objective: Detect potential adversarial activity on a web-server.

Approach: Use clustering algorithm (k-means/DBSCAN) implemented in Python + ML libraries, to identify potential malicious activity.

(Refer Chapter 1 in the book)



13

Real World Motivation

"Panama Papers" Incident

- A hacker (identity unknown) was able to penetrate the webserver, email server and client databases of a law firm (Mossack Fonseca) sometime in late 2015 – early 2016
- Attacker exfiltrated of 11.5 million confidential documents and 2.6 terabytes of client data.
- The confidential documents were leaked to journalists and contained personal financial information about wealthy individuals and public officials from all over the world that had previously been kept private.

Countries with politicians, public officials or close associates implicated in the leak on April 15, 2016 (as of May 19, 2016)

https://en.wikipedia.org/wiki/Panama_Papers

14

How the attack happened (possibly)?

- Attacker possibly targeted known vulnerabilities in Wordpress plugins allowing them to upload malicious PHP scripts to the webserver, and access to wp-config.php
- wp-config.php contains credentials in cleartext to access the database.
- Attacker possibly used this along with other webserver vulnerabilities in apps such as Drupal to gain access.

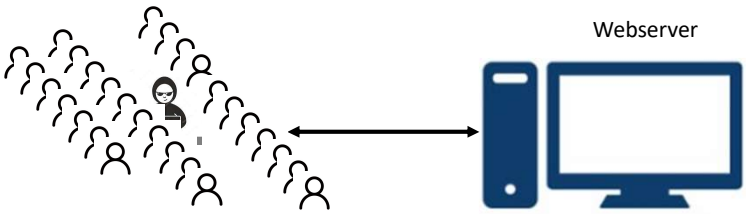
Example Local File Inclusion Attack Against Wordpress

http://victim.com/wp-admin/admin-ajax.php?action=revslider_show_image&img=../wp-config.php

Source: https://blog.sucuri.net/2014/09/slider-revolution-plugin-critical-vulnerability-being-exploited.html

15

Problem Setup



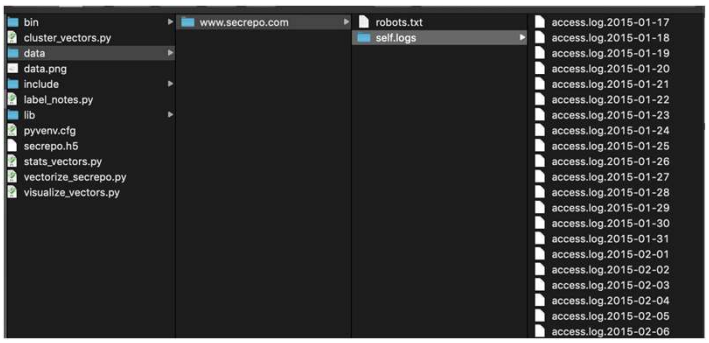
Hundreds of thousands of normal users (IP addresses) accessing the webserver, and there is an adversary in the mix.

Problem: Isolate potentially malicious adversarial traffic to the webserver, and identify the offending IP addresses.

16

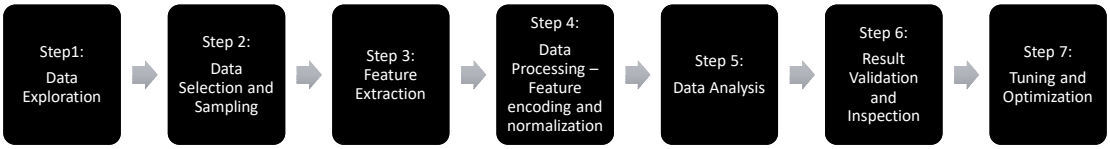
Data

Note: Authors do not use data from the panama incident, but a similar dataset available freely from https://www.secrepo.com (another useful resource for free cybersecurity datasets).



17

Overall Solution Workflow



18

Questions

- What does the data look like?
- How many data points exist?
- What does it mean?
- What features are available?
- What are basic statistics for each feature? Unique values, counts, mean, stdev, variance for each.

Step 1: Data Exploration

IP address of client | Access Time | HTTP Request | HTTP Response Code | Bytes Transferred

64.53.236.187 - - [26/Jan/2015:04:22:02 -0800] "GET / HTTP/1.1" 301 483 "-" "Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/39.0.2171.99 Safari/537.36"

64.53.236.187 - - [26/Jan/2015:04:22:03 -0800] "GET / HTTP/1.1" 200 6498 "-" "Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/39.0.2171.99 Safari/537.36"

64.53.236.187 - - [26/Jan/2015:04:22:03 -0800] "GET /twitter-icon.png HTTP/1.1" 200 27787 "http://www.secrepo.com/" "Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/39.0.2171.99 Safari/537.36"

64.53.236.187 - - [26/Jan/2015:04:22:04 -0800] "GET /bootstrap/img/favicon.ico HTTP/1.1" 200 589 "-" "Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/39.0.2171.99 Safari/537.36"

64.53.236.187 - - [26/Jan/2015:04:22:14 -0800] "GET /zeus/zeus_json.zip HTTP/1.1" 404 340 "http://www.secrepo.com/" "Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/39.0.2171.99 Safari/537.36"
64.53.236.187 - - [26/Jan/2015:04:22:14 -0800] "GET /favicon.ico HTTP/1.1" 200 267 "-" "Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/39.0.2171.99 Safari/537.36"

19

Extracting basic statistics from the dataset

How many total log entries are present in the dataset?

```
$wc -l access.log.*
```

Answer: 812,391

How many unique IP addresses are present?

```
$awk '{print $1}'  access.log.* | sort | uniq | wc -l
```

Answer: 58373

NOTE: you can write another python script to answer these questions, but I chose to use some Unix command line tools, which you can also use if you have a Mac or Linux.

20

Problem Challenges

- Millions of data points (each web server log is a data point).
- Data is mix of categorical data and numerical data, so need to find a suitable form of representation.
- Problem is akin to finding a "needle in a haystack"
- Manual approaches of going over each line of logs is infeasible and error prone.
- Clearly, need an automated approach to assist an analyst in identifying the problem logs.

21

Questions
Using all the data for analysis may require lot of CPU/memory resources and time. What subset of data should we use?

Step 2: Data selection and sampling

Use the first 10,000 IP addresses (instead of all the 58,373 unique ones)

Rationale: This seems to be quite an arbitrary choice, but we can start with 10,000. Another option is to use logs from a random sample of 10,000 IP addresses.

Limit the sample to IP addresses which have at least 5 entries

Rationale: Any IP address with less than 5 entries => sparser activity => low chance of being a serious threat

22

Questions
What features are useful for the specific objective?

Step 3: Feature Extraction

Features Extracted

64.53.236.187 - - [26/Jan/2015:04:22:02 -0800] "GET / HTTP/1.1" 301 483 "-" "Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/39.0.2171.99 Safari/537.36"

- HTTP Verbs in each request
  - A verb could be one of GET, POST, HEAD, OPTIONS, PUT, TRACE
  - 6 features in all
- HTTP Response Codes
  - A response code could be one of 200, 404, 304, 301, 206, 418,416, 403, 405, 503, 500
  - 11 features in all

Total 17 features

Each response code has a meaning in the HTTP protocol. Refer https://developer.mozilla.org/en-US/docs/Web/HTTP/Status for the meaning of each code.

Each data record has many more features, but this analysis just uses the two entries in each record

23

Questions
- How should we encode the features to make them suitable for input to analysis?
- How do we represent an IP address?
- What would happen if we store counts for each feature? Should we scale/normalize our data?

Step 4: Data Processing – Feature Encoding and Normalization

Vectorization

- Store each IP address as a 32-bit integer(e.g. 10.1.1.1 → 0x0A010101 -> 167837953)
- For each IP address, build a vector of length 17 as follows:
  - Count number of occurrences of each HTTP response code and HTTP verb

	GET	POST	HEAD	TRACE	…	…	202	301	404
IP address 1	#count11	#count12	#count13	#count14	…	…	…	…	…
IP address 2	….	…	…	…	…	…	…	…	…
…	…	…	…	…	…	…	…	…	…

Normalization

- Normalize the counts for each vector
  - Scale input vectors individually to unit norm
    - For each row compute, $X = \sqrt{\{x_1^2 + x_2^2 + \ \dots + x_n^2\}}, and$ divide each value by X.
- Question: Why is this step necessary? Why not just use the raw counts?

24

Python code for vectorization and visualization of data

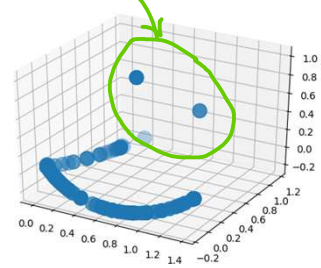
```
$ python vectorize_secrepo.py
```

- Creates a file secrepo.h5 with all the data in vector format.
- View data with hdf5dump if you want to see the contents.

```
$ python visualize_vectors.py -i secrepo.h5
```

Note: I fixed the code to save the generated figure to a .png file, by replacing the call *plt.show()* with *plt.savefig('data.png')*

These look interesting!



25

Questions

- What clustering algorithm should we use? K-means? DBScan?
- What parameters should we set? How should we set them?

Step 5: Data Analysis

Use *k-means* to cluster data, start with a low number of cluster (k=2) perhaps, and analyze the output.

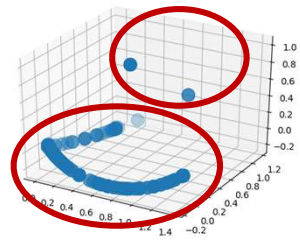
Assumption: You already know what clustering is, and how k-means works

26

Python code for k-means clustering

```
$ python cluster_vectors.py -c kmeans -n 2 -i
secrepo.h5 -o secrepo.h5
```

- -n specifies the number of clusters
- -c is the algorithm to use (k-means or dbscan)
- -I is the input data set
- -o is the dataset to write out the cluster labels (same as the input)



27

Questions

- How good are the clusters? Are they accurate representations of underlying data?

Step 6: Result Validation and Inspection

Use a statistical test to determine how well the samples have been grouped by the clustering algorithm.

Use silhouette coefficient, which computes the average distance between points that lie within a given cluster to the average distance between points assigned to different clusters.

- The lower the coefficient, the better our clustering.

In the code, authors use Silhoutte scoring, which computes a scrore value between -1 and +1.

- The closer the scores are to +1, the better the results.

28

Python code for statistical testing and inspection of results

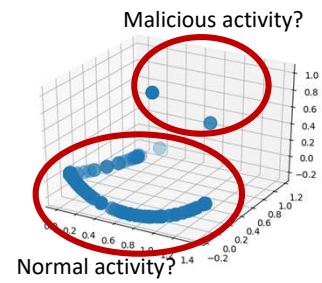
```
$ python stats_vectors.py secrepo.h5
```

- One cluster has more samples, other has less but still many entries than desirable.
- Bigger cluster => Normal activity, Smaller Cluster => Malicious?

```
$ python label_notes.py -i secrepo.h5
```

- Lists the IP addresses belonging to each cluster, we can see the list of potentially malicious IPs in the smaller cluster.
- Pick an IP from each cluster, check the raw logs for activity, investigate the patterns

```
$ grep "<IP address>" data/www.secrepo.com/self.logs/access.log.*
```



29

Questions

- What is a good number of clusters that represent the underlying distribution in the data?

Step 7: Tuning and Optimization

Continue running the k-means algorithm (Step 5) with different number of clusters until the overall Silhouette score is as close to one as possible.

Run Step 6 to check the statistics and the clusters formed.

Authors determined that k=12 was the optimal number for the dataset.

30

Final tuning and results

```
$ python cluster_vectors.py -c kmeans -n 12 -i
secrepo.h5 -o secrepo.h5
```

```
$ python stats_vectors.py -i secrepo.h5
```

```
$ python label_notes.py -i secrepo.h5 -l <label>
```

- This will output the IP addresses in the cluster specified with -l

NOTE: A lower cluster count does not necessarily mean that cluster is malicious! For example, all the malicious entries (the ones trying to hack) are in Cluster 2, which has 47 IPs.

- Label 0 has 9764 samples
- Label 1 has 83 samples
- Label 2 has 47 samples
- Label 3 has 21 samples
- Label 4 has 2 samples
- Label 5 has 18 samples
- Label 6 has 4 samples
- Label 7 has 6 samples
- Label 8 has 11 samples
- Label 9 has 16 samples
- Label 10 has 27 samples
- Label 11 has 1 samples

31

Thoughts on clustering

- Cluster analysis can be applied to virtually any kind of dataset to gather quick insights.
- Clustering is often a first step in data exploration to quickly sift through tons of data looking for outliers and anomalies.
- Clustering algorithms are easy to use, and computationally efficient.
- Clustering algorithms may not be suitable with high dimensional feature spaces.
- Clustering is an unsupervised approach, as such, the results need to carefully validated by a human to assess the quality of results, but often provide quick insights.

32

Next Steps For You (Smaller Projects)

- Download the code from the repo and try out the example.
- Explore clustering with the DBSCAN algorithm and compare results with the k-means algorithm.
- Try to use different features. How about using the HTTP request strings as a feature? How would you encode it? How about using the bytes transferred or the HTTP Agent String as a feature?
  - How does the choice of features affect the accuracy of your algorithm?
- Can you use a different algorithm to detect malicious entries? Have you heard of XGBoost?
- Try out the remaining ML algorithms from the book applied on other scenarios.
  - SMS Spam Classification
  - Botnet detection using Logistic Regression
  - XOR Key length detection using deep learning

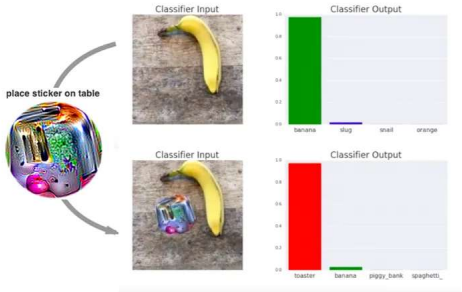
33

Bigger Project Ideas

- Microsoft Malware Prediction Dataset on Kaggle (2019)
  - https://www.kaggle.com/c/microsoft-malware-prediction
- Microsoft Malware Classification Challenge (2015)
  - https://www.kaggle.com/c/malware-classification
- Malicious Intent Detection Challenge (2015)
  - https://www.kaggle.com/c/wallarm-ml-hackathon

34

Finally: Be aware of ML limitations!



https://gizmodo.com/this-simple-sticker-can-trick-neural-networks-into-thin-1821735479

Stickers on stop signs cause NN to misclassify the sign as 45Mph speed limit!

https://spectrum.ieee.org/cars-that-think/transportation/sensors/slight-street-sign-modifications-can-fool-machine-learning-algorithms

35

Some More Reading

Applications of deep learning for traffic identification

- https://www.blackhat.com/docs/us-15/materials/us-15-Wang-The-Applications-Of-Deep-Learning-On-Traffic-Identification.pdf

Deep Neural Networks for Hackers

- https://i.blackhat.com/us-18/Wed-August-8/us-18-saxe-Deep-Learning-For-Hackers-Methods-Applications-and-Open-Source-Tools.pdf

36

To Summarize

- WE TALKED ABOUT CYBERSECURITY CHALLENGES AND THE JOB OPPORTUNITES
- WE DISCUSSED HOW DATA SCIENCE IS A KEY TOOL IN OUR ARSENAL TO DEFEND AGAINST CYBER THREATS
- WE WALKED THROUGH A CYBERSECURITY CASE STUDY WITH PYTHON CODE AND ML LIBRARIES

37

Thank You!

Feel free to get my email from your instructor and contact me if you have any questions or need further information!

38