ADVERSARIAL ATTACKS & DEFENCES ON PHISHING DETECTION MODELS

Prepared by : Archana Sreekumar (AM.EN.P2CSN18005)
Mentored by : Dr Krishnashree Achuthan
Collaborator : Mr Gilad Gressel, Georgia Tech, USA

OVERVIEW

- What is phishing?
- Adversarial attack on ML model
- Problem statement
- Dataset used
- Basic architecture
- Method of feature selection
- Training & testing using dataset
- Modification & testing on dataset
- Observations
- Adversarial training
- Roadmap until next review
- Conclusion

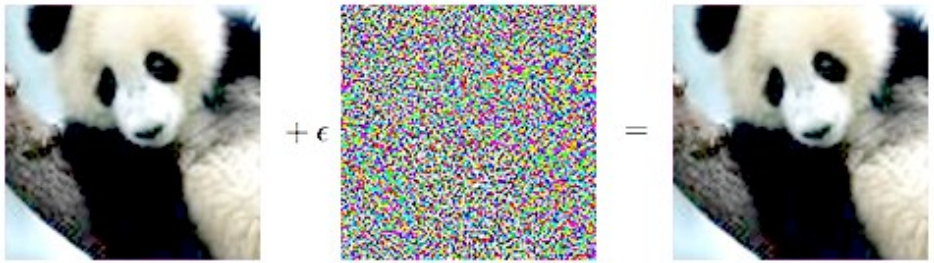
What is Phishing?

“Phishing is the act of tricking people into entering their sensitive information by disguising oneself as a trustworthy entity”

Source: https://whatismyipaddress.com/phishing

Adversarial Attacks on ML Model

“An adversary crafts a malicious input sample with the intention of causing the machine learning model to make an incorrect prediction”



Source: Explaining and Harnessing Adversarial Examples, Goodfellow et al.[5]

Problem Statement

“To learn whether phishing detection models are vulnerable to adversarial attacks and to defend the model against such adversarial attacks.”

- Carefully crafted adversarial examples could evade phishing detection models.
- Millions of user accounts could be compromised through the misclassification of phishing detection models under adversarial attack.

Closely Related Work

Adversarial Sampling Attack Against Phishing Detection

- Demonstrate attack on phishing detection models
- Dataset consist of a number of features for each instance and corresponding label.
- ‘-1’ indicates a legitimate page and ‘1’ indicates phishing page.
- Different combinations of features are considered.
- Original phishing instance and selected features used to create adversarial sample.

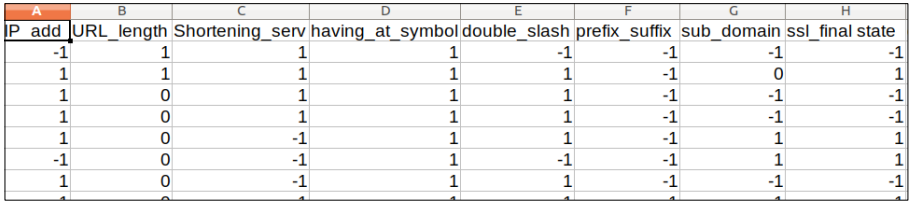
H. Shirazi, B. Bezawada, I. Ray, and C. Anderson. "Adversarial Sampling Attacks Against Phishing Detection." IFIP Annual Conference on Data and Applications Security and Privacy. Springer, Cham, 2019. #citations – 0,Google h5 index - NA

Work Done

Dataset Used

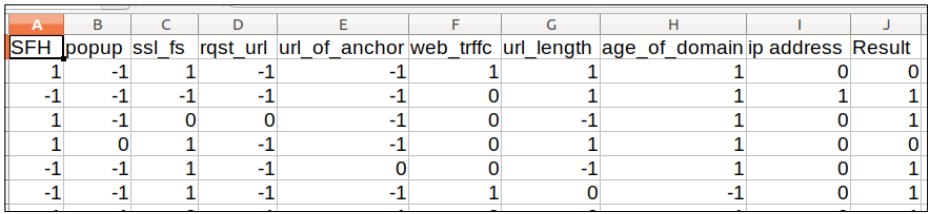
- Phishing dataset from UCI Machine learning repository

  - Dataset 1 [2]
    - Number of attributes: 30
    - Number of instances: 2456
    - Attribute characteristics: Integer



IP_add	URL_length	Shortening_serv	having_at_symbol	double_slash	prefix_suffix	sub_domain	ssl_final state
-1	1	1	1	-1	-1	-1	-1
1	1	1	1	1	-1	0	1
1	0	1	1	1	-1	-1	-1
1	0	1	1	1	-1	-1	-1
1	0	-1	1	1	-1	1	1
-1	0	-1	1	-1	-1	1	1
1	0	-1	1	1	-1	-1	-1

  - Dataset 2 [3]
    - Number of attributes: 10
    - Number of instances: 1353
    - Attribute characteristics: Integer

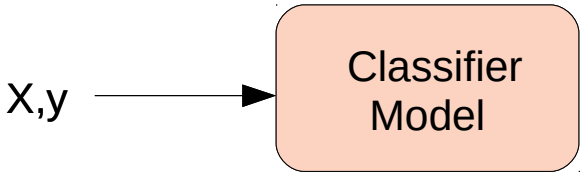


SFH	popup	ssl_fs	rqst_url	url_of_anchor	web_trffc	url_length	age_of_domain	ip address	Result
1	-1	1	-1	-1	1	1	1	0	0
-1	-1	-1	-1	-1	0	1	1	1	1
1	-1	0	0	-1	0	-1	1	0	1
1	0	1	-1	-1	0	1	1	0	0
-1	-1	1	-1	0	0	-1	1	0	1
-1	-1	1	-1	-1	1	0	-1	0	1

Dheeru Dua and Casey Graff. UCI Machine Learning Repository. 2017. url : http://archive.ics.uci.edu/ml

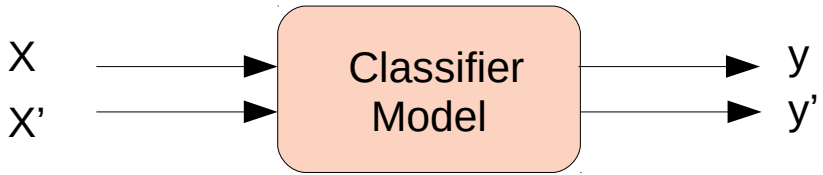
Basic Architecture

Training phase:



X – input data
y – output

Test phase:



X – input data without modification
X’ – input data with modification
y,y’ – prediction

Method of Feature Selection

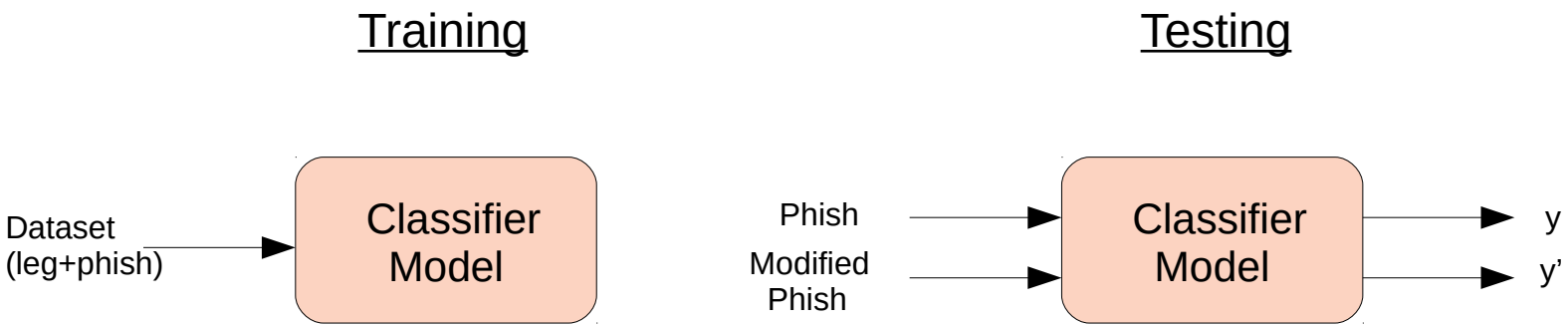
- Label 1 indicates phishing instance & -1 indicates legitimate instance
- Take a set of phishing instances
- Replace all feature values with -1 which indicates a ‘legitimate’ value and check prediction
- Change the value of each feature sequentially to 1 and check the prediction by the model
- Select those features which upon changing to 1 caused a change in prediction to -1.

```
1 data2 = test_data
2 data2 = data2.replace(1,-1)
3 data2 = data2.replace(0,-1)
4 data2.head()
```

	SFH	popup	ssl_fs	rqst_url	url_of_anchor	web_trffc	url_length	age_of_domain	ip address
0	-1	-1	-1	-1	-1	-1	-1	-1	-1
1	-1	-1	-1	-1	-1	-1	-1	-1	-1
2	-1	-1	-1	-1	-1	-1	-1	-1	-1
3	-1	-1	-1	-1	-1	-1	-1	-1	-1
4	-1	-1	-1	-1	-1	-1	-1	-1	-1

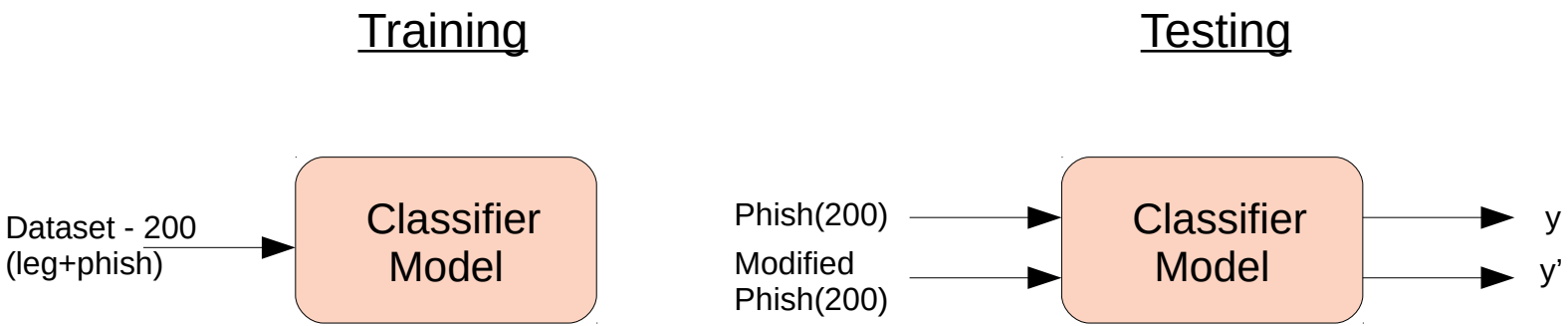
Training & Testing Using Same Data

- Model is trained using phishing and legitimate instances in the dataset
- Testing is done using the same phishing instances in the dataset with and without modification



Training & Testing Using Different Data

- From the dataset 200 phishing instances are separated out
- Model is trained using phishing and legitimate instances in the remaining dataset
- Testing is done using the 200 phishing instances with and without modification



Modification & Testing on Dataset 1

- Result ‘1’ indicate phishing instance & ‘-1’ indicates legitimate instance

Result before feature modification

```
X_test_wom.head()
```

prefix_suffix	sub_domain	domain_reg_len	...	right click	popup window	iframe	url_of_anchor	ssl_final state	Shortening_serv	page rank	having_at_symbol	URL_length	IP_add
-1	1	-1	...	1	-1	1	0	1	-1	-1	1	0	1
-1	1	-1	...	1	1	1	0	1	-1	-1	1	0	-1
-1	1	-1	...	1	1	1	0	1	-1	1	1	0	1
-1	0	1	...	1	1	1	0	1	1	1	1	1	1
1	-1	-1	...	1	1	1	0	1	-1	-1	1	1	1

```
prediction_wo_mod = classifier.predict(X_test_wom)
print(prediction_wo_mod)
```

```
[ 1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1 -1  1  1  1  1  1
  1  1  1  1  1  1  1  1  1 -1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
  1  1  1  1  1  1  1  1  1  1  1  1 -1  1  1  1  1  1  1  1  1  1  1  1
  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
  1 -1  1  1  1  1  1 -1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
  1  1  1  1  1  1 -1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
  1  1  1  1  1  1  1  1  1  1  1  1  1 -1  1  1  1  1  1  1  1  1  1  1
  1  1  1  1  1  1  1  1]
```

Most values are 1

Result after feature modification

```
X_test_wm.head()
```

prefix_suffix	sub_domain	domain_reg_len	...	right click	popup window	iframe	url_of_anchor	ssl_final state	Shortening_serv	page rank	having_at_symbol	URL_length	IP_add
-1	1	-1	...	1	-1	1	-1	-1	-1	-1	1	0	1
-1	1	-1	...	1	1	1	-1	-1	-1	-1	1	0	-1
-1	1	-1	...	1	1	1	-1	-1	-1	1	1	0	1
-1	0	1	...	1	1	1	-1	-1	1	1	1	1	1
1	-1	-1	...	1	1	1	-1	-1	-1	-1	1	1	1

```
prediction_with_mod = classifier.predict(X_test_wm)
print(prediction_with_mod)
```

```
[-1 -1 -1  1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1  1 -1  1  1 -1  1
 -1 -1 -1 -1 -1 -1  1  1 -1 -1 -1  1  1  1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1
 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1  1 -1 -1 -1 -1 -1
 -1 -1  1 -1 -1  1 -1  1  1 -1 -1 -1 -1 -1 -1  1 -1  1 -1 -1  1 -1 -1  1
 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1  1 -1 -1 -1  1 -1 -1 -1 -1 -1  1 -1
 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1  1 -1 -1 -1  1 -1 -1 -1 -1 -1 -1 -1  1 -1
 -1 -1 -1  1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1  1  1 -1 -1 -1 -1 -1 -1 -1 -1
 -1 -1  1 -1  1 -1 -1 -1 -1  1 -1 -1 -1 -1 -1 -1 -1 -1 -1  1 -1 -1 -1  1
 -1 -1 -1 -1 -1  1 -1 -1]
```

Most values are -1

Modification & Testing on Dataset 2

- Feature values 1 for legitimate & -1 for phish

Result before feature modification

```
1 X_test_wom.head()
```

	SFH	popup	ssl_fs	rqst_url	url_of_anchor	web_trffc	url_length	age_of_domain	ip address
0	-1	-1	-1	-1	-1	0	1	1	1
1	1	-1	0	0	-1	0	-1	1	0
2	-1	-1	1	-1	0	0	-1	1	0
3	-1	-1	1	-1	-1	1	0	-1	0
4	-1	0	-1	-1	1	1	0	-1	0

```
1 prediction_wo_mod = classifier.predict(X_test_wom)
2 print(prediction_wo_mod)
```

```
[ 1 -1  1  1  1  1 -1  1  1  1  1 -1  1  1  1  1 -1  1  1  1  1  1  1 -1
 -1  1  1  1  1  1  1  1  1  1  1  1  1 -1  1  1  1  1 -1  1  1  1 -1  1
  1  1  1  1  1  1  1  1  1 -1  0  1  1  1 -1  1  1  1  1  1  1 -1  1  1
  1  1  1  0  1  0  1  1  1  1  1  1  0  1  1  1 -1  1  1 -1  1  1  1  1
  1  1  1  1  1  1  1  1  1  1 -1 -1  1  1  1 -1  1  1 -1 -1  1  1  1  1
  1  1  1  1  1  1  1  1  1  0  1  1  1  1  1  1  1  1  1  1 -1  1  1 -1
  1  1  1  1  1 -1  1 -1  1  1  1  0  1  1  1  1  1 -1 -1  1  1 -1  1 -1
  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1 -1  1  1  1 -1  1  1
  1  1  1  1  1  1  1]
```

Result after feature modification

```
1 X_test_wm.head()
```

	SFH	popup	ssl_fs	rqst_url	url_of_anchor	web_trffc	url_length	age_of_domain	ip address
0	1	1	-1	-1	-1	0	1	1	1
1	1	1	0	0	-1	0	-1	1	0
2	1	1	1	-1	0	0	-1	1	0
3	1	1	1	-1	-1	1	0	-1	0
4	1	1	-1	-1	1	1	0	-1	0

```
1 prediction_with_mod = classifier.predict(X_test_wm)
2 print(prediction_with_mod)
```

```
[-1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1
 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1
 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1  0 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1
 -1 -1 -1 -1 -1  0 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1
 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1
 -1 -1 -1 -1 -1 -1 -1 -1 -1  0 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1
 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1  0 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1
 -1 -1 -1 -1 -1 -1  0 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1
 -1 -1 -1 -1 -1 -1 -1]
```

Observations – Dataset 1

Train Data	Test Data	Accuracy before modification (%)	Accuracy after modification (%)
Complete dataset (phish+leg)	all phish	96	14
Without 200 phish	200 phish	96.5	9.5

Observations – Dataset 2

Train Data	Test Data	Accuracy before modification (%)	Accuracy after modification (%)
Complete dataset (phish+leg)	all phish	92	0
Without 200 phish	200 phish	82	0

Adversarial Training on Dataset-1

- Adding adversarial samples to training data

Dataset used for adversarial training

```
data = pd.read_csv('train-adv.csv')
data.tail(10)
```

	ssl_final state	url_of_anchor	domain_reg_len	...	popup window	iframe	age of domain	dns record	we trffc	page rank	google index	links pointing to page	statistical report	Result
1	-1	-1	-1	...	-1	-1	1	1	1	-1	1	0	1	1
1	-1	-1	-1	...	1	1	1	1	0	-1	1	0	1	1
1	-1	-1	-1	...	1	1	1	-1	0	-1	1	1	1	1
1	-1	-1	-1	...	-1	-1	1	1	1	-1	1	0	1	1
0	-1	-1	-1	...	1	1	1	1	1	-1	1	0	1	1
1	-1	-1	-1	...	1	1	1	-1	0	-1	1	1	1	1
1	-1	-1	-1	...	1	1	1	1	0	-1	1	0	1	1
1	-1	-1	-1	...	1	1	1	1	0	-1	1	1	1	1
1	-1	-1	1	...	1	1	1	1	0	-1	1	0	1	1

Increase in accuracy with modified dataset

```
X_test_wm.head()
```

_slash	prefix_suffix	sub_domain	ssl_final state	domain_reg_len	Favicon	...	right click	popup window	iframe
-1	-1	1	-1	-1	1	...	1	1	1
1	-1	1	-1	-1	-1	...	1	-1	-1
1	-1	0	-1	1	1	...	1	1	1
1	1	1	-1	-1	-1	...	-1	-1	-1
-1	-1	1	-1	-1	1	...	1	1	1

```
uracy with different train set with 2 feature modification is: ",100*(a
```

```
uracy with different train set with 2 feature modification is:  49.5
```

Adversarial Training on Dataset-2

- Adding adversarial samples in training set

Dataset used for adversarial training

```
data = pd.read_csv('train-adv.csv')
data.tail(10)
```

	SFH	popup	ssl_fs	rqst_url	url_of_anchor	web_trffc	url_length	age_of_domain	ip address	Result
1245	1	1	0	1	-1	1	-1	-1	0	1
1246	1	1	-1	0	0	1	0	-1	0	1
1247	1	1	-1	-1	-1	-1	0	1	0	1
1248	1	1	-1	0	1	-1	0	1	0	1
1249	1	1	1	0	1	-1	0	1	0	1
1250	1	1	1	-1	-1	0	-1	-1	0	1
1251	1	1	-1	-1	-1	1	-1	-1	0	1

Increase in accuracy with modified dataset

```
X_test_wm.head()
```

	SFH	popup	ssl_fs	rqst_url	url_of_anchor	web_trffc	url_length	age_of_domain	ip address
0	1	1	-1	-1	-1	0	1	1	1
1	1	1	0	0	-1	0	-1	1	0
2	1	1	1	-1	0	0	-1	1	0
3	1	1	1	-1	-1	1	0	-1	0
4	1	1	-1	-1	1	1	0	-1	0

```
uracy with different train set with 2 feature modification is: ",100*(accuracy_sc
```

```
Accuracy with different train set with 2 feature modification is:  54.08163265306123
```

Roadmap Until Next Review

- Construct a classifier using neural networks.
- Add more real valued features to the dataset.
- Manipulate the features using an optimization method rather than random change to craft an adversarial example.
- Add the manipulated features to the website source code and check the functionality of phishing website.

Conclusion

- Adversarial attacks represent a concrete problem in the field of machine learning.
- The vulnerability of phishing detection models to adversarial attacks could lead to exposure of sensitive information of millions of users.
- There is a need for finding a method to craft an adversarial example which scales inorder to find out the possible vulnerabilities of the model and also to define a robust defence mechanism.

References

[1] H. Shirazi, B. Bezawada, I. Ray, and C. Anderson, “Adversarial Sampling Attacks Against Phishing Detection,” Data and Applications Security and Privacy XXXIII Lecture Notes in Computer Science, pp. 83–101, 2019.

[2] Abdelhamid et al.,(2014a) Phishing Detection based Associative Classification Data Mining. Expert Systems With Applications (ESWA), 41 (2014) 5948â€“5959.

[3] Rami M Mohammad, Fadi Thabtah, and Lee McCluskey. “An assessment of features related to phishing websites using an automated technique”. In: 2012 International Conference for Internet Technology and Secured Transactions. IEEE. 2012, pp. 492–497.

[4] H. Shirazi, B. Bezawada, I. Ray, and C. Anderson, “Adversarial Sampling Attacks Against Phishing Detection,” Data and Applications Security and Privacy XXXIII Lecture Notes in Computer Science, pp. 83–101, 2019.

[5] I. Goodfellow, J. Shlens, and C. Szegedy, “Explaining and Harnessing Adversarial Examples,” Google AI, 01-Jan-1970. [Online]. Available: https://ai.google/research/pubs/pub43405. [Accessed: 21-Sep-2019].

Thank You