COMPUTER SCIENCE
& ENGINEERING
TEXAS A&M UNIVERSITY

Textual and Visual Anti-Phishing

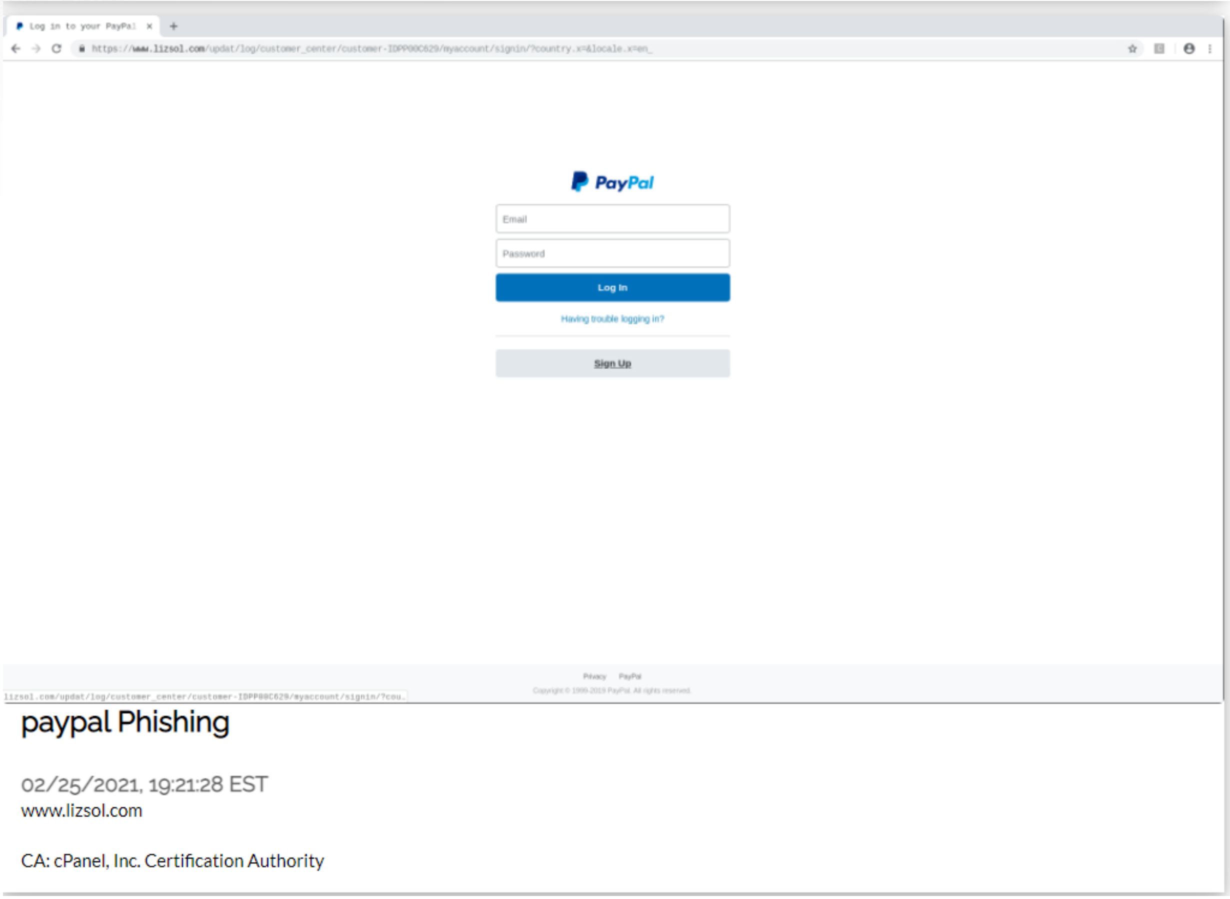
Dr. Martin “Doc” Carlisle

Premise

- Use a Bayesian model on text and visual elements of a webpage to determine if it is a Phishing site

Phishing Websites

- Often used to collect credentials



Example from:

https://phishbank.org/#/

Phishing Activity

https://docs.apwg.org/reports/apwg_trends_report_q4_2020.pdf

Techniques for finding Phish

- Industrial toolbar-based
- User-Interface-based
- Web page content-based

Industrial Toolbar-based

- Examples: SpoofGuard, TrustWatch, Netcraft
  - Wu et al found these ineffective – 20/30 subjects fooled
  - Cranor et al – only one tool of 10 detected more than 60%

User-Interface-based

- E.g. provide custom image per user

Figure 1: The trusted password window uses a background image to prevent spoofing of the window and textboxes.

- Password manager
  - Only provides password to certain domains

Web page content-based

- Use web page info (URL, links, terms, images, forms) to detect phishing
  - CANTINA: compute term frequency-inverse document frequency for terms, then Google a few terms to see if current website is a top result
  - B-APT: Bayesian based on tokens from DOM

Definitions

- Surface level content (not used in this work)
  - URL, hyperlinks
- Textual content
  - Terms or words
    - They “stem” words, e.g. “program”, “programs”, “programming” all go to “program”
- Visual content
  - Color, font size, style, location of images

Zhang et al approach

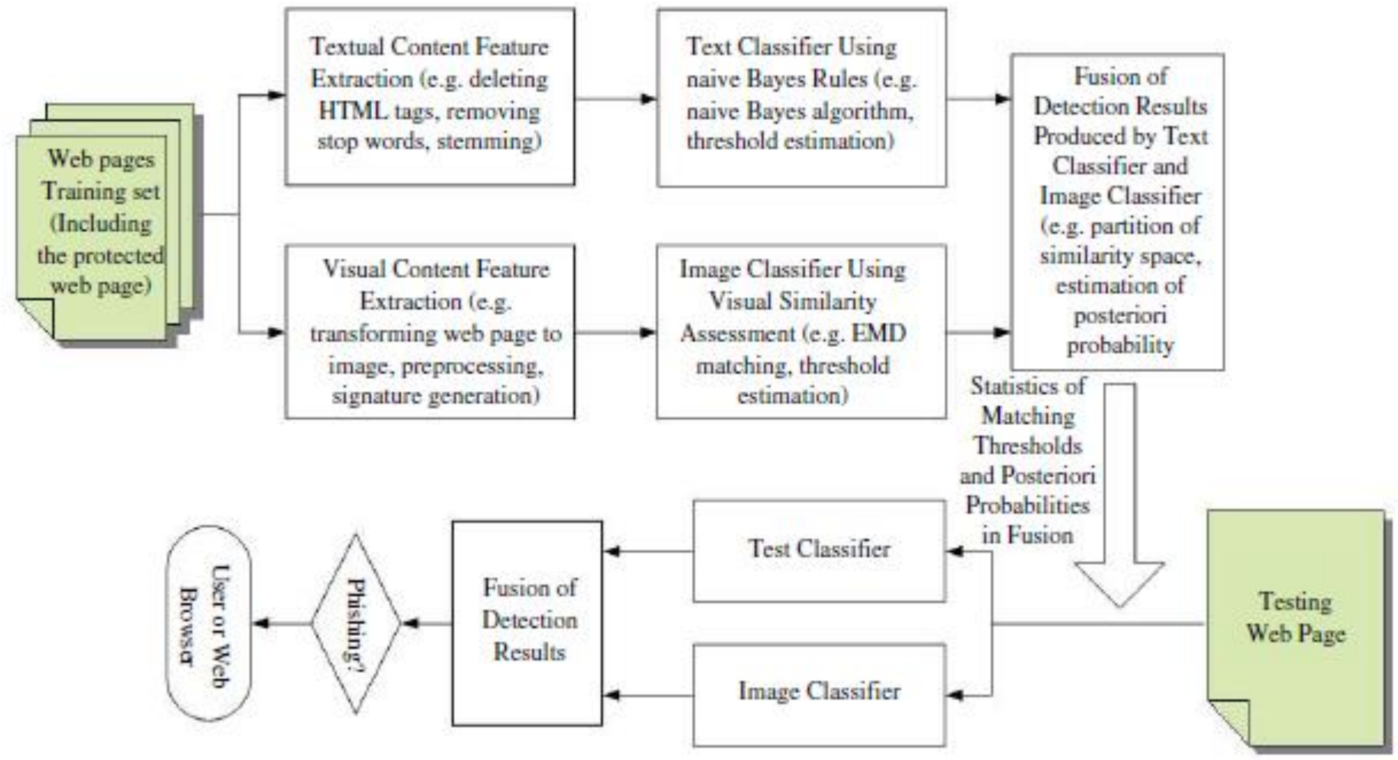


Fig. 1. Overview of the system framework.

Bayes classifier

- Two categories (phish or normal)

$$P(g_j|v_1, v_2, \ldots, v_n) = \frac{P(v_1, v_2, \ldots, v_n|g_j)P(g_j)}{P(v_1, v_2, \ldots, v_n)} \quad (1)$$

- $P(g_j)$ (category j) is computed based on # of training samples belonging to g_j

- Hard to estimate $P(v_1,v_2,v_n|g_j)$

Naïve Bayes

- Assume all components independent

$$P(g_j|v_1, v_2, \ldots, v_n) = \frac{P(g_j)\prod_{i=1}^{n} P(v_i|g_j)}{\sum_{j=1}^{c}\prod_{i=1}^{n} P(v_i|g_j)}.$$

$$P(g_j|v_1, v_2, \ldots, v_n) = \frac{P(v_1, v_2, \ldots, v_n|g_j)P(g_j)}{P(v_1, v_2, \ldots, v_n)} \qquad (1)$$

Text Classifier (I)

- Probability a word is in a phishing or normal page (u_i is a word, g_j is a category, $h_{l,i}$ is from the histogram vector of the l-th web page in the category)

$$P(u_i|g_j) = \frac{1 + \sum_{l=1}^{K_j} h_{l,i}}{\sum_{i=1}^{n} \sum_{l=1}^{K_j} h_{l,i}}$$

Text Classifier (II)

- T is a webpage, u_i is a word, g_j is a category, $h_{i,T}$ is frequency of i^{th} word on web page T and R is the total # of words from the protected web page.

$$P(g_j|T) = \frac{P(g_j)\prod_{i=1}^{n} P(u_i|g_j)^{\frac{h_{i,T}}{R}}}{\sum_{s=1}^{d} P(g_s)\prod_{i=1}^{n} P(u_i|g_s)^{\frac{h_{i,T}}{R}}} \qquad (7)$$

- R enlarges terms to denominator isn't close to 0
- Threshold to determine phish

Image Classifier

- Transform web pages into JPEG images (100x100)
- Features are degraded colors (ARGB) and centroids of those colors (c is coordinate, N is # of pixels of that color)

- Signature

$$\breve{C}_\sigma = \sum_{i=1}^{N_\sigma} (c_{\sigma,i} / N_\sigma)$$

$$S = \{(F_{\sigma_1}, N_{\sigma_1}), (F_{\sigma_2}, N_{\sigma_2}), \ldots, (F_{\sigma_N}, N_{\sigma_N})\}$$

Distance Measurement

- EMD measures dissimiliarity (distance) of two web page images -- d_{ij} is:

$$D_{norm}(F_{\sigma_i}, F_{\sigma_j}) = \mu \cdot ||\sigma_i - \sigma_j|| + \eta \cdot ||C_{\sigma_i} - C_{\sigma_j}|| \quad (9)$$

- Then similarity is 1-EMD

$$EMD(S_a, S_b, D) = \frac{\sum_{i=1}^{m} \sum_{j=1}^{n} f_{ij} \cdot d_{ij}}{\sum_{i=1}^{m} \sum_{j=1}^{n} f_{ij}}. \quad (10)$$

Computation time

- $O(m^3 \log m)$ – 1.43 seconds (too slow!)

Steps

1. Obtain webpage and normalize
2. Compute signature
3. Calculate EMD and similarity between website and protected web page
    1. Presumably they have to do this for every protected site?
4. Classify via threshold

Dataset

TABLE III

WEB PAGE DISTRIBUTION OF CATEGORIES IN SUB-DATASETS

Protected web page	URL	Phishing	Normal	Total
eBay	https://signin.ebay.com	1636	8291	9927
PayPal	https://www.paypal.com/c2	2551	8291	10842
Rapidshare	https://ssl.rapidshare.com/premiumzone.html	489	8291	8780
HSBC	http://www.hsbc.co.uk/1/2/HSBCINTEGRATION/	452	8291	8743
Yahoo	https://login.yahoo.com	204	8291	8495
Alliance-Leicester	https://www.mybank.alliance-leicester.co.uk/index.asp	182	8291	8473
Optus	https://www.optuszoo.com.au/login	101	8291	8392
Steam	https://steamcommunity.com	96	8291	8387

Text Classifier Results

TABLE IV

CLASSIFICATION RESULTS OF TEXT CLASSIFIER WITH DIFFERENT THRESHOLD SETTING STRATEGIES

Protected Web Page	Predefined threshold						Estimated threshold				
	Thr	CCR	F-score	MCC	FNR	FAR	CCR	F-score	MCC	FNR	FAR
eBay	0.20	97.24%	0.9087	0.8977	136/818	1/4145	97.46%	0.9169	0.9060	123/818	3/4145
PayPal	0.25	99.19%	0.9826	0.9774	35/1275	9/4146	98.52%	0.9677	0.9588	76/1275	4/4146
RapidShare	0.10	99.57%	0.9597	0.9581	18/244	1/4146	99.86%	0.9877	0.9869	4/244	2/4146
HSBC	0.10	99.22%	0.9187	0.9180	34/226	0/4145	99.70%	0.9709	0.9694	9/226	4/4145
Yahoo	0.05	98.42%	0.5110	0.5811	67/102	0/4145	99.27%	0.8208	0.8312	31/102	0/4145
Alliance-Leicester	0.05	99.34%	0.8182	0.8293	28/91	0/4145	99.86%	0.9667	0.9660	4/91	2/4145
Optus	0.05	99.57%	0.7805	0.7983	18/50	0/4146	100%	1	1	0/50	0/4146
Steam	0.20	98.86%	0	NaN	48/48	0/4145	99.57%	0.8000	0.7997	12/48	6/4145

Other Text Classifiers

TABLE V
PERFORMANCE OF DIFFERENT TEXT CLASSIFIERS

Protected Web page	KNN			SVM			Bayesian approach		
	CCR	FNR	FAR	CCR	FNR	FAR	CCR	FNR	FAR
eBay	98.73%	10/818	53/4145	99.44%	23/818	5/4145	97.46%	123/818	3/4145
PayPal	99.15%	2/1275	44/4146	99.61%	21/1275	0/4146	98.52%	76/1275	4/4146
RapidShare	98.16%	3/244	78/4146	99.89%	3/244	2/4146	99.86%	4/244	2/4146
HSBC	98.67%	5/226	53/4145	99.84%	6/226	1/4145	99.70%	9/226	4/4145
Yahoo	99.27%	8/102	23/4145	99.69%	13/102	0/4145	99.27%	31/102	0/4145
Alliance-Leicester	97.45%	2/91	106/4145	99.91%	4/91	0/4145	99.86%	4/91	2/4145
Optus	97.81%	1/50	91/4146	99.98%	1/50	0/4146	100%	0/50	0/4146
Steam	96.73%	0/48	137/4145	99.95%	1/48	1/4145	99.57%	12/48	6/4145

Image Classifiers

TABLE VII

CLASSIFICATION RESULTS OF IMAGE CLASSIFIER WITH DIFFERENT THRESHOLD SETTING STRATEGIES

Protected Web page	Predefined threshold						Estimated threshold				
	Thr	CCR	*F*-score	MCC	FNR	FAR	CCR	*F*-score	MCC	FNR	FAR
eBay	0.55	99.50%	0.9845	0.9816	25/818	0/4145	99.54%	0.9857	0.9831	23/818	0/4145
PayPal	0.50	99.80%	0.9957	0.9944	10/1275	1/4146	99.80%	0.9957	0.9944	10/1275	1/4146
RapidShare	0.55	99.41%	0.9437	0.9423	26/244	0/4146	99.38%	0.9417	0.9400	26/244	1/4146
HSBC	0.50	100%	1	1	0/226	0/4145	100%	1	1	0/226	0/4145
Yahoo	0.50	99.95%	0.9901	0.9899	2/102	0/4145	99.95%	0.9901	0.9899	2/102	0/4145
Alliance-Leicester	0.55	100%	1	1	0/91	0/4145	100%	1	1	0/91	0/4145
Optus	0.55	99.38%	0.6487	0.6907	26/50	0/4146	99.59%	0.8000	0.8110	16/50	1/4146
Steam	0.50	99.98%	0.9897	0.9896	0/48	1/4145	99.98%	0.9897	0.9896	0/48	1/4145

Overall framework

1. Train text and image classifier, collect similarity measurements for different classifiers
2. Partition similarity into sub-intervals
3. Estimate probs for text classifier
4. Estimate probs for image classifier
5. Classify each test image
6. If different from two classifiers, calculate decision factor
7. Return final classification

Fusion Algorithm

- Combine text and visual with weights that sum to 1

$$S_{i,W} = \beta \cdot S_{i,T} + (1 - \beta) \cdot S_{i,V} \qquad (23)$$

- Estimate β with Bayesian approach

Fusion Results

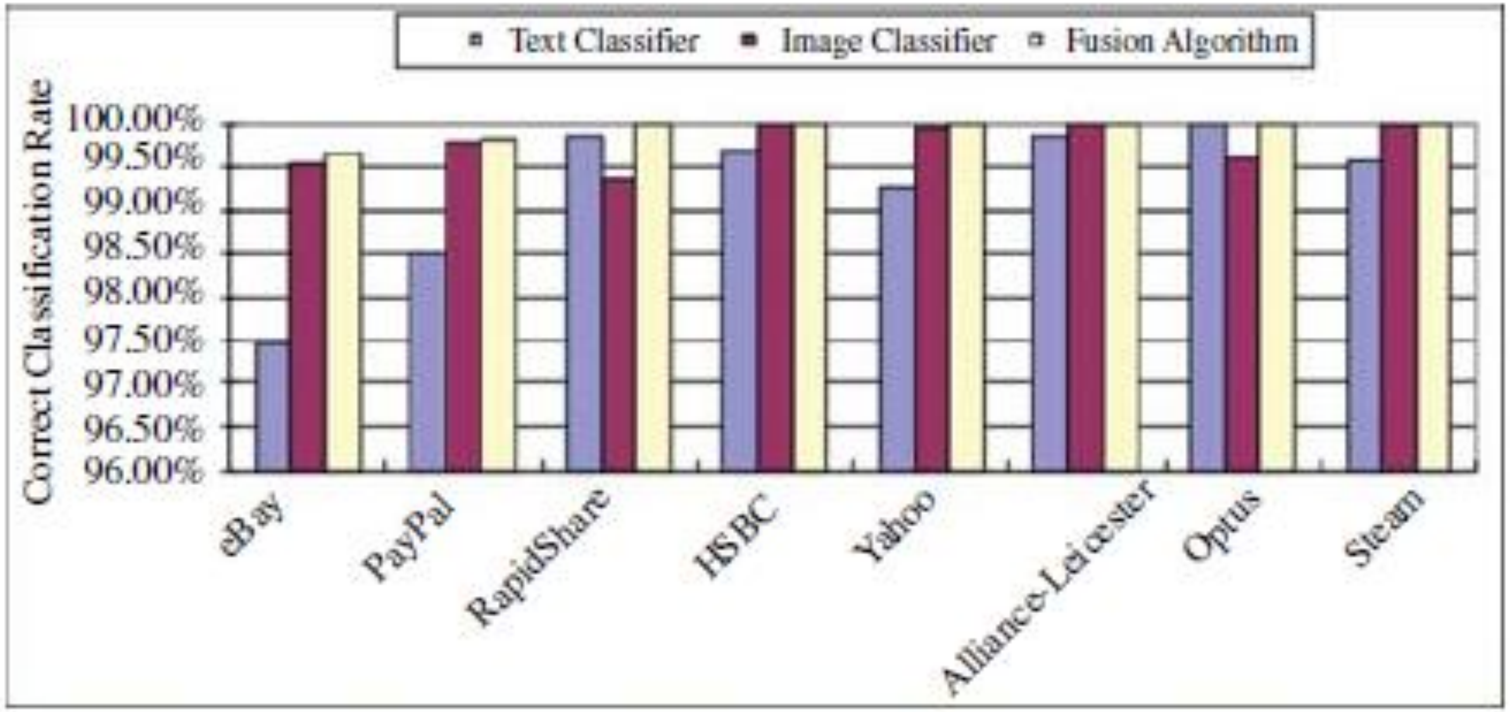


Fig. 5. Overall performances of our proposed schemes.

COMPUTER SCIENCE
& ENGINEERING
TEXAS A&M UNIVERSITY

Diverse Datasets and Phishing Customizable Benchworking Framework

Dr. Martin "Doc" Carlisle

Premise

- Create high-quality common dataset and classifiers to test Phishing models
  - URLs
  - Emails

What makes “high quality”?

- Accessibility
- Completeness
- Consistency
- Integrity
- Validity
- Interpretability
- Timeliness

URL Datasets

- Legit
  - Crawl top 40 website domains (Alexa Sept 5, 2018), three levels of crawling
    - No more than 10 URLs per domain
    - Login dataset (only pages with login form)
  - Phish
    - PhishTank (Sep 5, 2018)
    - Anti-Phish Working Group (APWG, Oct 30, 2018)
    - OpenPhish (Sep 5, 2018)
    - Exclude if URL unavailable, no WHOIS data

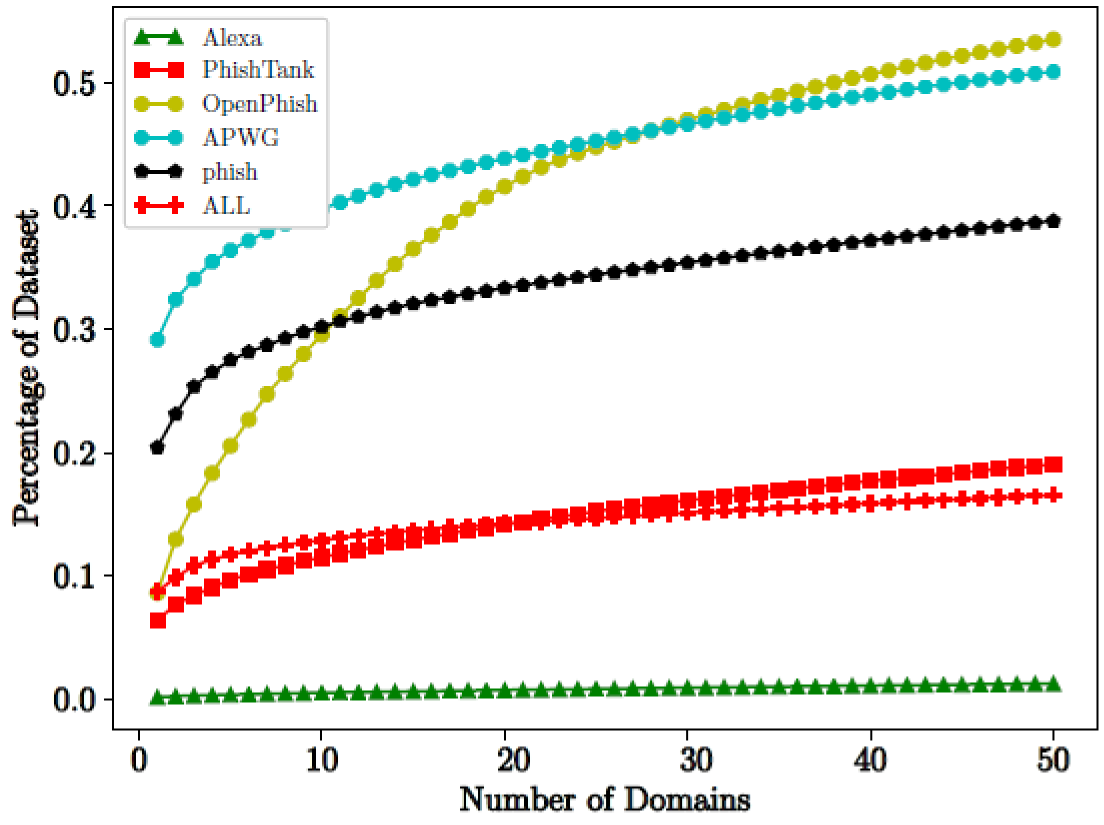
URL Stats

Table 1: Statistics of The URL Benchmark Dataset

Source	URLs	Extracted	Domains	TLDs	Logins
Alexa	31,163	29,173	9,554	285	2,056
Alexa Login	4,370	3,992	1,960	117	3,992
PhishTank	26,346	20,803	10,813	406	4,999
APWG	66,929	45,382	7,760	319	2,812
OpenPhish	2,249	1,336	710	94	326

COMPUTER SCIENCE
& ENGINEERING
TEXAS A&M UNIVERSITY

URL Dataset CDF



Other URL sources

- PhishTank archive
- UCI Phishing
  - Attribute-Relation File Format (ARFF)
  - https://archive.ics.uci.edu/ml/index.php

Email Datasets (I)

- IWSPA-AP
  - Poster on cleaning this (and dataset quality in general): https://dl.acm.org/doi/pdf/10.1145/3319535.3363267

Table 1: Dataset Statistics

	Legitimate	Phishing	Total
Train	5088	612	5700
Test	3825	475	4300

(a) No-header Dataset

	Legitimate	Phishing	Total
Train	4082	501	4583
Test	3699	496	4195

(b) Header Dataset

Email Datasets (II)

- Email Benchmark dataset
  - 10,500 legit + 10,500 phishing
  - Legit sources: wikileaks, Enron, SpamAssassin
  - Phishing sources: Nazario + SpamAssassin
- Bluefin:
  - 300 uncaught phishing emails

Email diversity

- About 85% of emails are less than 10% similar

Table 2: Distribution of cosine similarities for email pairs in the Email Benchmark dataset. FH: Full Header, NH: No Header

Dataset	Ranges of Similarities					
	[0-10]	(10-20]	(20-30]	(30-40]	(40-50]	>50
FH	85.44%	10.47%	2.60%	0.85%	0.29%	0.33%
NH	84.29%	10.74%	3.92%	0.55%	0.18%	0.29%

Lots of classifiers! (I)

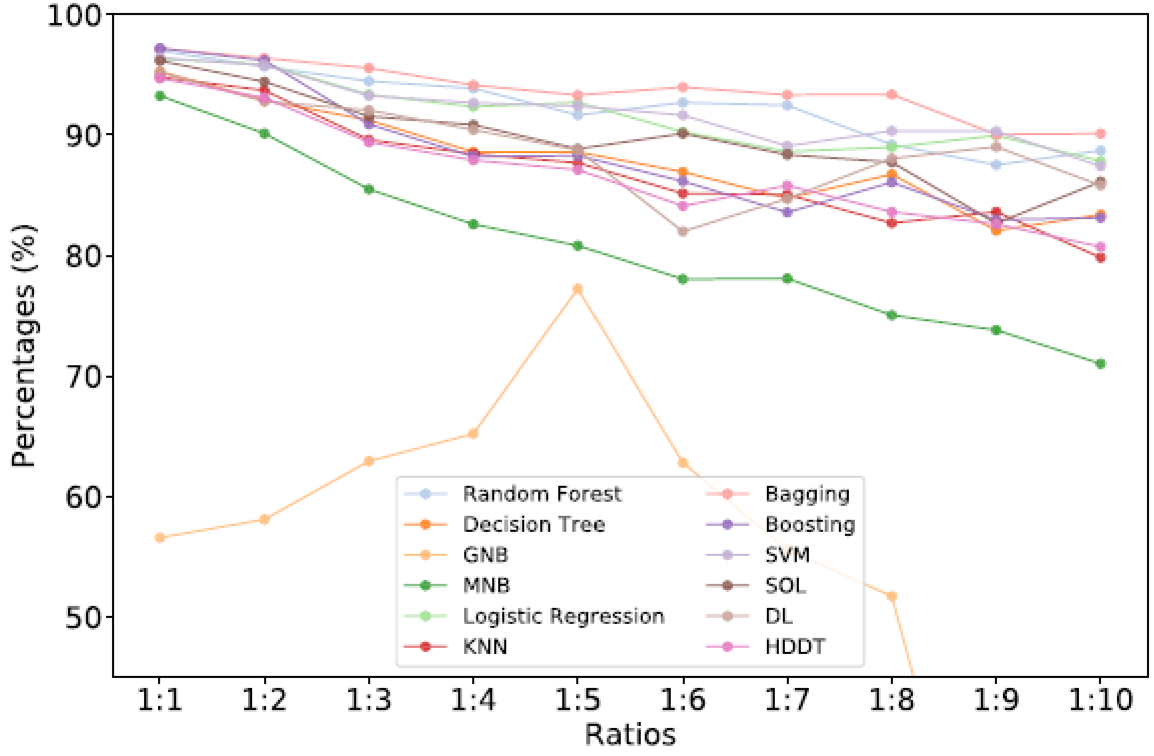
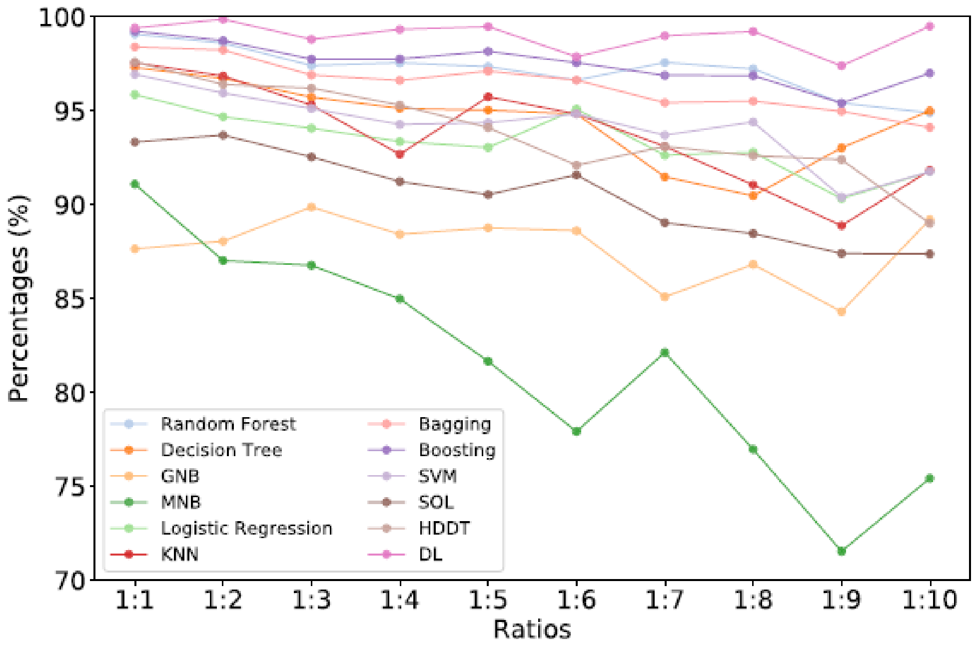


Figure 4: F1-score with varying ratios between phishing and legitimate instances (***a.*** for URL Benchmark Dataset and ***b.*** for email Dataset B). k-NN with $k = 5$ for URLs and $k = 3$ for emails. Bagging and Boosting use Decision Tree as their base classifier SOL: Scalable Online Learning [35], DL: Deep Learning [19], HDDT: Hellinger Distance Decision Tree [21]

Lots of classifiers! (II)

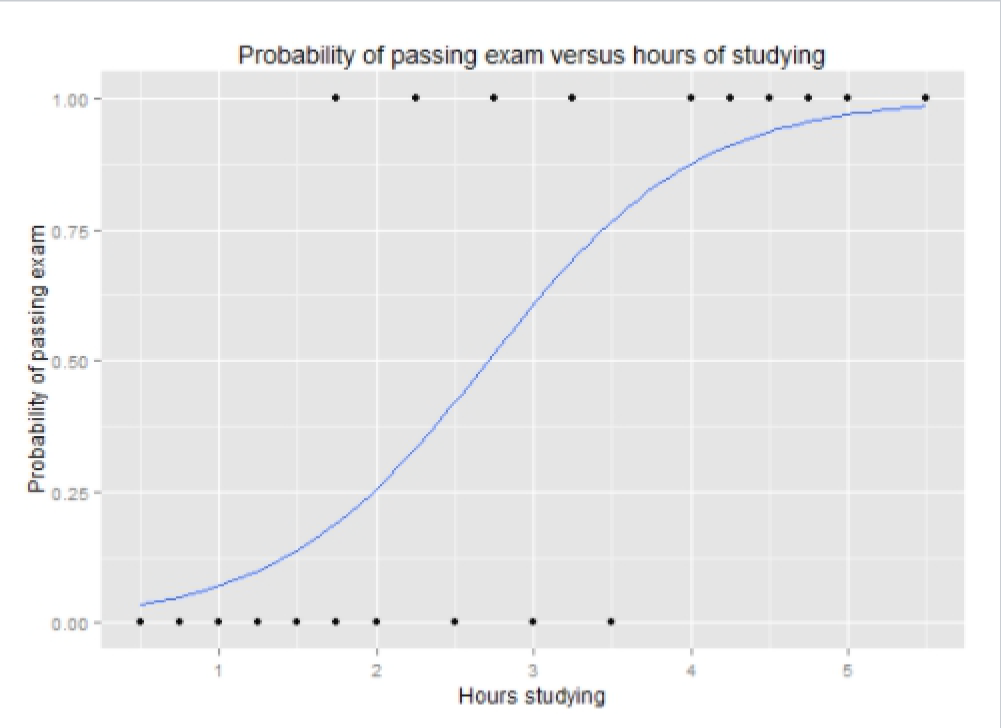


(b)

Figure 4: F1-score with varying ratios between phishing and legitimate instances (*a.* for URL Benchmark Dataset and *b.* for email Dataset B). k-NN with $k = 5$ for URLs and $k = 3$ for emails. Bagging and Boosting use Decision Tree as their base classifier SOL: Scalable Online Learning [35], DL: Deep Learning [19], HDDT: Hellinger Distance Decision Tree [21]

Logistic Regression

- Used to model binary choices



Graph of a logistic regression curve showing probability of passing an exam versus hours studying

Bagging

- Bagging classifier is an ensemble meta-estimator that fits base classifiers each on random subsets of the original dataset and then aggregate their individual predictions (either by voting or by averaging) to form a final prediction

Boosting

- "Can a set of weak learners create a single strong learner?"
- random forests builds each tree independently while gradient boosting builds one tree at a time. This additive model (ensemble) works in a forward stage-wise manner, introducing a weak learner to improve the shortcomings of existing weak learners.

Hellinger-distance Decision Trees

- A proposal to deal with imbalanced data w/o sampling
  - See, e.g. https://www3.nd.edu/~nchawla/papers/DMKD11.pdf

$$d_H(P(Y_+), P(Y_-)) = \sqrt{\sum_{i \in V} \left(\sqrt{P(Y_+|X_i)} - \sqrt{P(Y_-|X_i)} \right)^2}. \qquad (3)$$

Learning to Detect Phishing

Dr. Martin "Doc" Carlisle

Premise

- Create machine learning model to detect phishing emails and websites

Email classification (I)

- IP-based URLs (http://128.168.0.1/paypal.cgi)
- Age of linked domain (< 60 days)
- Nonmatching URLs paypal.com

Email classification (II)

- Here links to “non-modal” domain
  - “here” is linked to domain not referenced most frequently
- HTML email vs plaintext
- # of links, # of domains, # of dots in URL

Email classification (III)

- # of domains
  - www.cs.university.edu
  - www.company.co.jp
- # of dots in URL
  - www.my-bank.update.data.com
  - www.google.com/url?q=http://www.badsite.com

Email classification (IV)

- Contains JavaScript in email
- SpamAssassin guess

Webpage Classification

- Browser history (has user been there b4?)
- Redirected (e.g. tinyURL?)
- Term frequency-inverse document frequency (TF-IDF)
  - Search for key terms and check whether current page is in results

PILFER approach

- Random Forest
  - 10 decision trees
- 10-fold cross validation
  - Each $1/10^{th}$ is tested against other 90% as training data

PILFER Datasets

- Ham corpora from SpamAssassin project
  - (2002 and 2003) – ~6,950 messages
- PhishingCorpus
  - ~860

Data issues

- Old emails meant they only got 505/870 WHOIS information
- Are these representative emails?

PILFER results

- Accuracy: 99.5%
- False positive rate 0.13%
- False negative rate 3.5%

Table 1: Accuracy of classifier compared with baseline spam filter

Classifier	False Positive Rate fp	False Negative Rate fn
PILFER, with S.A. feature	0.0013	0.036
PILFER, without S.A. feature	0.0022	0.085
SpamAssassin (Untrained)	0.0014	0.376
SpamAssassin (Trained)	0.0012	0.130

Features

Table 2: Percentage of emails matching the binary features

Feature	Non-Phishing Matched	Phishing Matched
Has IP link	0.06%	45.04%
Has "fresh" link	0.98%	12.49%
Has "nonmatching" URL	0.14%	50.64%
Has non-modal here link	0.82%	18.20%
Is HTML email	5.55%	93.47%
Contains JavaScript	2.30%	10.15%
SpamAssassin Output	0.12%	87.05%

Discussion

- Phishing is harder than spam?
  – Can’t just look for “V1agra”
- Other technologies may help
  – Sender ID: verify email is from IP address associated with the domain
  – Domain Keys (deprecated) use crypto to sign some parts of header with public key in DNS

COMPUTER SCIENCE & ENGINEERING
TEXAS A&M UNIVERSITY

They tried lots of stuff

Table 4: Average Accuracy of different classifiers on same features over 10 runs, with standard deviations

Classifier	fp	σ_{fp}	fn	σ_{fn}
Random Forest	0.0012	0.0013	0.0380	0.0205
SVM, C = 10	0.0024	0.0019	0.0408	0.0225
RIPPER	0.0025	0.0019	0.0383	0.0204
Decision Table	0.0022	0.0018	0.0555	0.0242
Nearest Neighbor w/ Generalization	0.0017	0.0022	0.0414	0.0265
1R	0.0012	0.0012	0.1295	0.0333
Alternating Decision Tree	0.0020	0.0018	0.0405	0.0229
Decision Stump	0.0012	0.0012	0.1295	0.0333
Pruned C4.5 Tree	0.0019	0.0017	0.0414	0.0235
Hybrid tree w/ Naïve Bayes leaves	0.0022	0.0017	0.0412	0.0209
Random Tree (1 random attribute/node)	0.0016	0.0015	0.0398	0.0200
AdaBoosted C4.5 tree	0.0019	0.0017	0.0414	0.0235
AdaBoosted Decision Stump	0.0016	0.0016	0.0748	0.0355
Voted Perceptron	0.0122	0.0053	0.0942	0.0311
Bayes Net	0.0384	0.0082	0.0689	0.0244
Naïve Bayes	0.0107	0.0030	0.0608	0.0248