Beyond blacklists: Learning to Detect Malicious Web Sites from Suspicious URLs

Justin Ma, Lawrence K. Saul, Stefan Savage, Geoffrey M. Voelker

Claudio Bozzato
Lorenzo Simionato

Problematiche

- Evitare lo spam ed il phishing
- Necessario un metodo automatizzato per rilevare le pagine pericolose
- Le blacklist non permettono di rilevare nuovi attacchi
- Metodo dinamico per identificare siti web maligni
- Quali informazioni utilizzare?
- Come estrarle?

State of the art

- All’attuale stato dell’arte esistono diversi approcci per l’estrazione delle features a partire da un URL
  - URL based
  - Content based
  - Informazioni indirette

URL Based

- Whitelist / blacklist (es: Google Safe Browsing)
- Word based:
  confirm, account, banking, secure, ebayisapi, webscr, login, signin
- Formato dell'URL:
  - Host obfuscation:
    http://www.google.it http://74.125.77.104 http://1249725800
  - Target organization:
    http://www.53.com/icons/small/www.paypal.com/SecureInfo/index.php
  - Numero di "." nell'hostname:
    http://www.53.com.wps.portal.secure.pool34.st
  - Numero di caratteri dopo il nome dell'organizzazione:
    http://ebay.com.clickme.glas11.st
  - Caratteri sospetti ("@","-", ...)
    http://www.paypal.com@1249725800

Content based

- iFRAMEs
  <iframe src=http://attacker/bad.js width=0 height=0></iframe>
- Obfuscated JavaScript
  alert(“hello”)
  var _0xa32a=["\x68\x65\x6C\x6C\x6F"]; alert(_0xa32a[0])
- TF-IDF (Terms Frequency - Inverse Document Frequency)
  Alto TF-IDF = alto TF nel documento e basso IDF tra i documenti
- Robust Hyperlinks (CANTINA)
  - Si trovano i 5 termini con TF-IDF più alto
  - Si cercano questi termini in un motore di ricerca
  - Si controlla che il dominio sia presente tra i primi N URL ritornati
- Immagini conosciute (Loghi)
- Link sospetti (URL based / Content based / Altro)
- Forms pericolosi (richiesta di credenziali)
- DOM walking (Javascript dinamico)

Informazioni indirette

- PageRank: in genere le pagine web di questo tipo non rimangono online per un tempo sufficiente da ottenere un PageRank alto
- PageRank dell'URL e dell'host
- Popolarità in Netcraft
- Data di registrazione del dominio

Classificazione

- Come classifichiamo?
  - Naive Bayes
  - Modello lineare (CANTINA):
    effetto calcolato come differenza tra TP e FP

$$S = f(\sum w_i * h_i) \qquad \text{f(x)} = 1 \text{ if } x>0, \text{f(x)} = -1 \text{ if } x<=0 \qquad w_i = \frac{e_i}{\sum e_i}$$

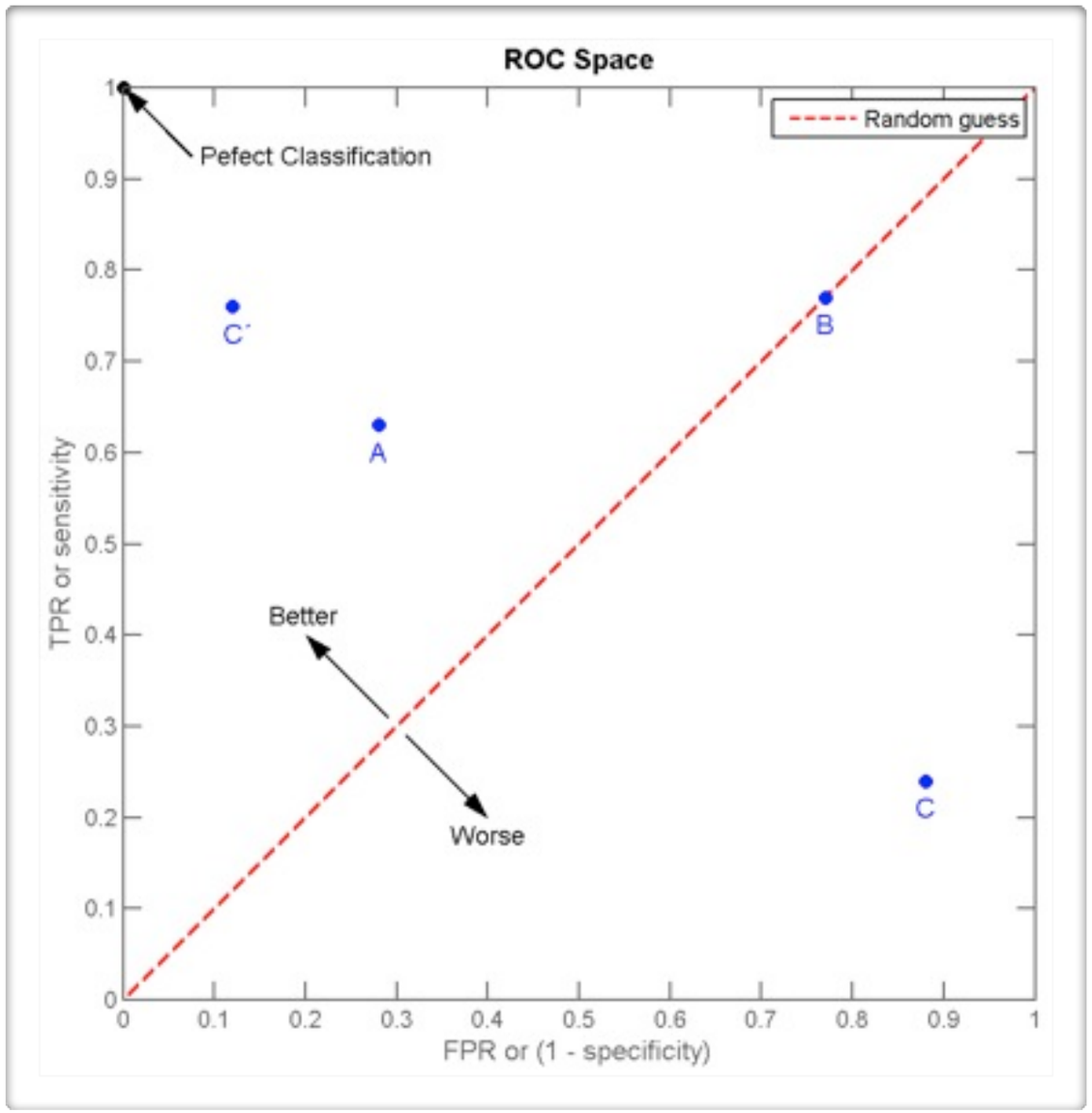
  - LR (Logistic Regression): tecnica ampiamente utilizzata sia per l'accuratezza che per il comportamento in presenza di un grande set di feature (in particolare con molte feature irrilevanti)
  - SVM: tecnica basata sugli iperpiani, che mira a massimizzare la distanza dei punti del training set più vicini.

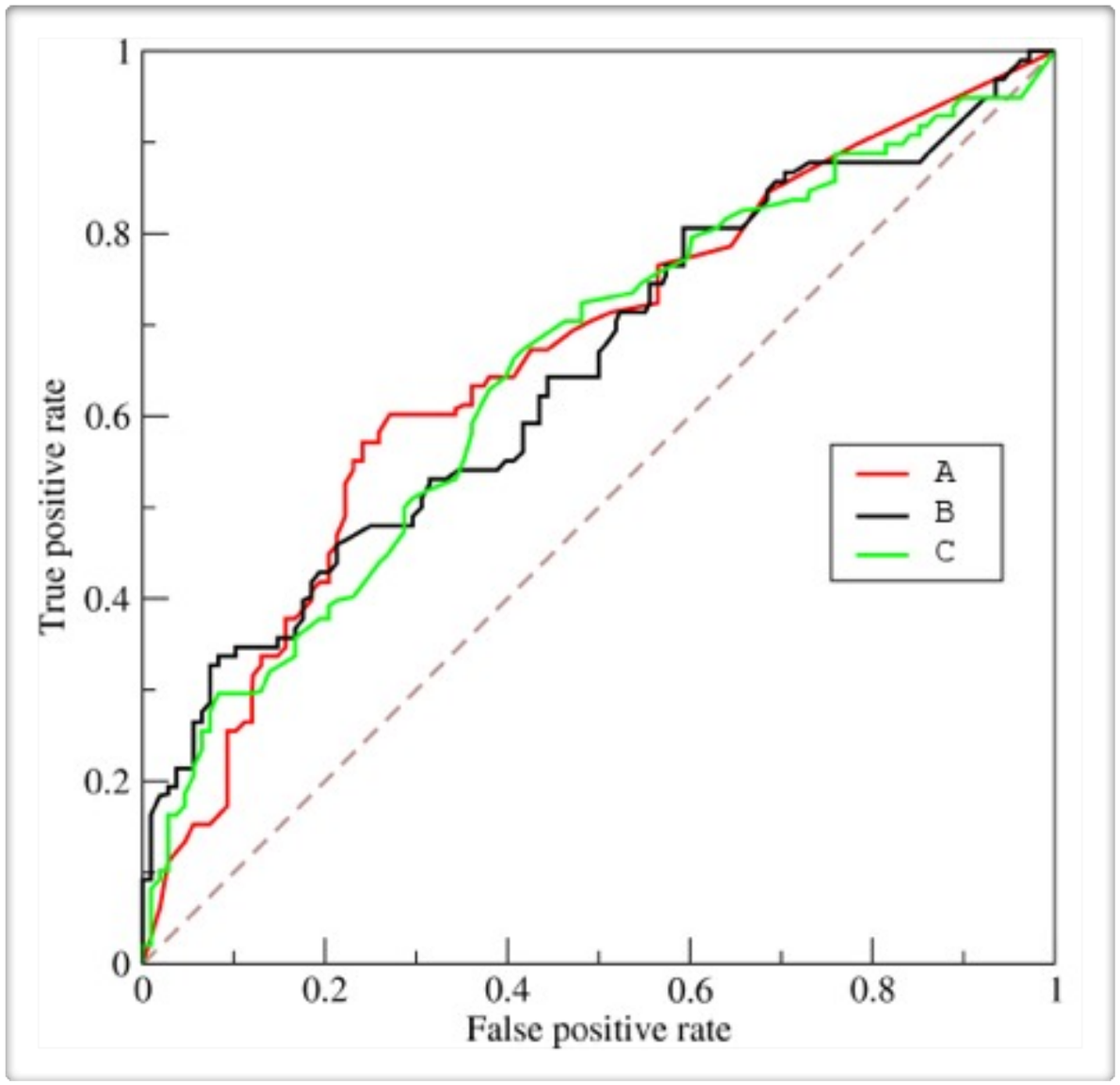
- Cross-validation:
  - Si partizionano i dati in training set e test set
  - Si fa l'analisi sul training set e la validazione sul test set
  - Si ripete per diversi step mediando i risultati

- Bisogna fare attenzione all'overfitting se abbiamo troppe features rispetto al training set

Relative Operating Characteristic

- Bisogna cercare di ridurre al minimo i falsi positivi mantenendo una buona accuratezza
- Scelta del threshold per bilanciare veri positivi e falsi positivi
- Confronto di diversi modelli





Risultati

- Netcraft: da 75% a 96% di accuratezza con 0% di false positives e 84% di true positives
- CANTINA: accuratezza del 90% con 1% di false positives
- “A Framework for Detection and Measurement of Phishing Attacks”: accuratezza di 93.4%, false positive di 0.7% e true positive di 88%
- Google Safe Browsing (Firefox): 53% true positives, 0% false positives
- Gli strumenti attualmente disponibili basati su euristiche ed utilizzabili come toolbar sono soggetti a molti false positives
- SpoofGuard: accuratezza del 91% ma 48% di false positives

L’articolo

- Approccio esclusivamente URL based
- Non si controlla il contenuto delle pagine (rischio per l’utente e overhead)
- Non si considera il contesto in cui si trova il link (pagina web, e-mail, mittente dell’e-mail)
- Raccoglie un grande insieme di features
- E’ il metodo stesso a scegliere le features più importanti
- Accuratezza del 95-99% (sul test set)

Features

- Lexical
  - Hostname
  - Path
  - Lunghezza dell'hostname
  - Lunghezza dell'URL
  - Numero di "."
  - Bag of words (insieme di token)

- Host-Based: permette di sapere dov'è il sito e chi lo possiede
  - IP address: a quale AS appartiene? I record A, NS, MX sono nello stesso AS?
  - WHOIS: data di registrazione, update, scadenza, dettagli nascosti?
    nomi di chi ha registrato il dominio
  - TTL dei record DNS del dominio? (fast-fluxers)
  - Presenza di keywords nell'hostname (server, client)?
  - Locazione geografica dell'host
  - Velocità di connessione

Features

Feature type	DP	YP	DS	YS
DNS NS record	10818	4186	10838	3764
WHOIS info	9070	3959	8702	3525
Hostname	7928	3112	7797	2959
TLD + domain	5787	2186	5609	1980
DNS A record	5627	2470	5207	1862
Geographic	4554	2250	4430	1836
Path tokens	4363	8021	8390	11974
Last token of path	1568	3751	2892	5071
DNS MX record	1323	497	1422	559
TLD	319	284	102	51
Connection speed	31	29	30	30
DNS TTL	14	13	14	12
Blacklists	7	7	7	7
WHOIS dates	6	6	6	6
Spamassassin plugin	5	5	5	5
IP address misc.	4	4	4	4
Lexical misc.	3	3	3	3
All features	51427	30783	55458	33648

Classificazione

	Length URL	N° dots URL	Path token “www”	Hosted in “France”	BlackList I	WhiteList	AS 6185	Registar “GoDaddy”	...
URL1	10	2	0	0	0	0	1	0	...
URL2	15	3	1	0	0	0	0	0	...
URL3	12	3	0	0	0	1	0	1	...
URL4	60	8	1	1	1	0	0	1	...
...	...	....	...	...	...	...	....	...	...

Logistic regression

Date d feature di valore (numerico) x_1, x_2, ... x_d
si calcola z nel seguente modo:

$$z = b + \sum_{i=1}^{d} w_i x_i$$

Dove b e w_i sono dei parametri da calcolare:

- b è detto intercept
- w_i è il peso associato alla feature i

Logistic regression

A partire da z, si calcola:

$$\sigma(z) = \frac{1}{1 + e^{-z}}$$

Se $0 < \sigma(z) < t$:

L'URL viene classificato come benigno

Se $t < \sigma(z) < 1$:

L'URL viene classificato come maligno

t è una threshold da determinare in modo tale da minimizzare gli errori di classificazione

ℓ_1-norm regularization

- L'intercept e i pesi vengono calcolati massimizzando la funzione obiettivo:

$$\mathcal{L}(\mathbf{w}, b) = \sum_{i=1}^{n} \log P(y_i|\mathbf{x}_i) - \gamma \sum_{\alpha=1}^{d} |w_\alpha|.$$

- In questo modo si vogliono:
  - ridurre gli errori di classificazione (primo termine)
  - minimizzare il valore assoluto dei pesi (secondo termine).
    Si favoriscono quindi soluzioni con molti pesi nulli

Data sets

- Per il training ed il testing del modello, sono necessari dei data set con URLs da analizzare
- Datasets con URLs “benigni”:
    - DMOZ (directory, 15000 URLs)
    - Yahoo random URL (url casuali, 15000 URLs)
- Dataset con URLs “maligni”:
    - PhishTank (siti di phishing, 5500 URLs)
    - SpamScatter (link in mail di spam, 15000 URLs)

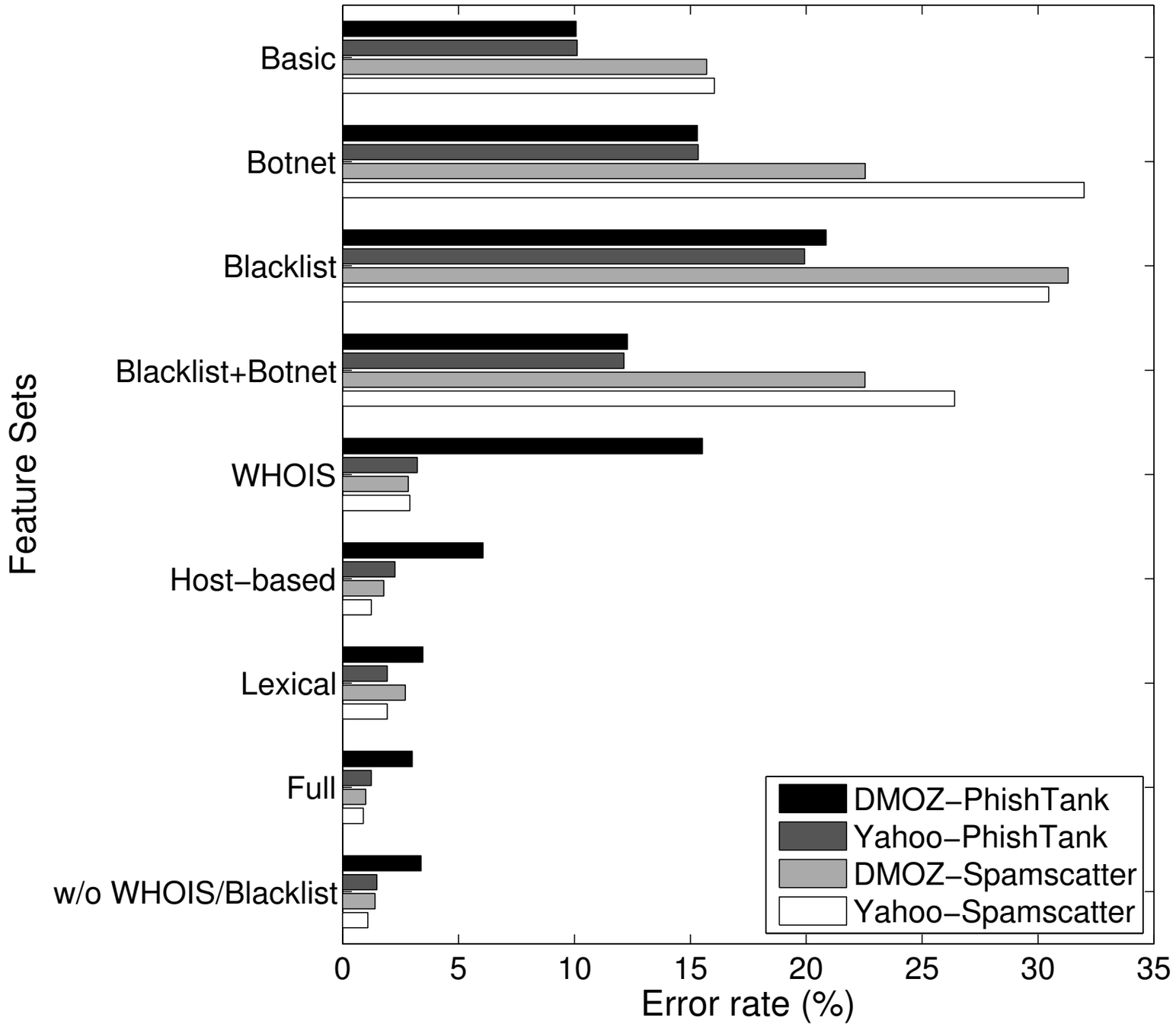
Esperimenti

- Combinazione di dataset benigni e maligni: se ne ottengono quattro: DP, DS, YP, YS
  D: DMOZ, Y: Yahoo, S: Spamscatter, P: PhishTank
- Esperimenti separati per ciascun dataset
- In ciascun esperimento si divide il dataset in 10 parti casuali
- Il 50% di ciascuna parte è usato per il training, l'altro 50% per il testing
- Il risultato della classificazione è calcolato come media dei risultati ottenuti nelle 10 parti

Risultati

- Dai risultati ottenuti, l'utilizzo di un gran numero di feature produce errori molto piccoli (<5%)
- Si ottengono errori simili anche senza l'utilizzo delle feature delle blacklist
- Utilizzando invece solamente feature relative all'URL gli errori sono ben più grandi (>10%)
- Si nota come ci siano differenze, anche significative, variando i dataset. Ad esempio gli URLs raccolti da PhishTank hanno parecchi campi WHOIS non disponibili

Risultati


Basic
Botnet
Blacklist
Blacklist+Botnet
WHOIS
Host−based
Lexical
Full
w/o WHOIS/Blacklist
Feature Sets
0
5
10
15
20
25
30
35
Error rate (%)
DMOZ−PhishTank
Yahoo−PhishTank
DMOZ−Spamscatter
Yahoo−Spamscatter

Analisi

- Con il classificatore LR, siamo in grado di analizzare il "peso" di ciascuna feature, in maniera da trovare quelle più o meno rilevanti
- Se il peso relativo ad una feature è
  - **positivo**: allora essa "penalizza" l'URL
  - **negativo**: allora essa "premia" l'URL
  - **zero**: allora essa non è considerata
- In questo modo possiamo sia **verificare** quello che già sappiamo sugli URLs maligni, sia **imparare** nuove cose

Analisi

- Ad esempio nel dataset YP:
  - **3962 su 30783 feature considerate (peso non zero)**
  - **URL contenenti** com, members **nell'hostname o** banking, paypal, http **nel path sono feature maligne**
  - Date di registrazione del dominio mancanti o molto recenti sono feature maligne
  - IP ranges di Google, Yahoo e AOL sono feature benigne
  - ....

Errori di classificazione

- Maggior causa di false positive:
    - Sito in hosting su IP/AS considerati maligni
- Maggiori cause di false negative:
    - URL con token benigni
    - Siti in hosting su servizi gratuiti (e.g. geocities.com)
    - Siti compromessi
    - ....

Limitazioni

- Conoscendo le feature ed i pesi ad esse associati, un attaccante potrebbe tentare di ingannare il classificatore: URL con poche informazioni (e.g. TinyURL), registrazione su domini benigni (e.g. .org), ecc.

- Problema dei falsi positivi rispetto ad un approccio tradizionale tramite Blacklist (dove i falsi positivi sono praticamente zero).
  Possibile soluzione per ridurli: calcolare i pesi in maniera tale da da minimizzare i falsi positivi, invece degli errori complessivi (naturalmente aumenteranno i falsi negativi).

Limitazioni

- Performance non soddisfacenti.
  Per un dataset (~20000 URLs):
    - 30 min. per la fase di training
    - 5h cross-validation
    - 90s. per il testing (+ query WHOIS/Blacklist/ecc)
- Fase di training troppo lunga nel caso di molti URL: successivo paper con algoritmo di tipo online machine-learning
- Da verificare le perfomance del testing (necessario ad ogni visita di un nuovo indirizzo)?

Limitazioni

- Accuratezza veramente del 95-99%?
- Risultati ottenuti "mischiando" i dataset di training/testing di gran lunga peggiori

Training	**Testing**			
	YS	YP	DS	DP
YS	0.90%	5.66%	9.38%	17.73%
YP	21.33%	1.24%	44.02%	33.54%
DS	13.55%	31.57%	1.00%	13.70%
DP	22.95%	3.06%	22.92%	3.01%
All sources (YDSP)	0.95%	2.40%	1.43%	3.19%

- In una applicazione reale: training con tutti i dataset Risultati? Possibili problemi di overfitting?

Conclusioni

- Approccio interessante per la classificazione degli URLs
- L'utilizzo di LR consente un'analisi delle caratteristiche dei siti maligni, utile anche per scoprire nuovi trend
- Problemi di scalabilità rilevanti
- Da verificare l'accuratezza del modello nei casi reali

Riferimenti

"Beyond Blacklists: Learning to Detect Malicious Web Sites from Suspicious URLs"
Justin Ma, Lawrence K. Saul, Stefan Savage, Geoffrey M. Voelker
In Proc. of the 15th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Paris 2009

"A Framework for Detection and Measurement of Phishing Attacks"
Niels Provos, Sujata Garera, Monica Chew, Aviel D. Rubin
In Proc. of the 2007 ACM workshop on Recurring malcode, Alexandria 2007

"All Your iFRAMEs Point to Us"
Niels Provos, Panayiotis Mavrommatis, Moheeb Abu Rajab, Fabian Monrose
In Proc. of the 17th conference on Security symposium, San Jose 2008

"CANTINA: A Content-Based Approach to Detecting Phishing Web Sites"
Yue Zhang, Jason Hong, Lorrie Cranor
In Proc. of the Sixteenth International World Wide Web Conference (WWW2007), Banff 2007

"The elements of statistical learning: data mining, inference, and prediction"
Trevor Hastie, Robert Tibshirani, Jerome Friedman
Springer, 2003

"Identifying Suspicious URLs: An Application of Large-Scale Online Learning"
Justin Ma, Lawrence K. Sault, Stefan Savage, Geoffrey M. Voelker
In Proc. of the International Conference on Machine Learning (ICML), Montreal 2009