Perception of a new framework for detecting phishing web pages

Youness Mourtaji
Faculty of Sciences and Techniques University, Computer science, systems and telecommunication Laboratory, Tangier, Morocco
mourtaji.y@gmail.com

Pr. Mohammed Bouhorma
Faculty of Sciences and Techniques University, Computer science, systems and telecommunication Laboratory, Tangier, Morocco
mbouhorma@gmail.com

Pr. Alghazzawi
King Abdulaziz University, Information Systems Department, Jeddah, Saudi Arabia
dghazzawi@kau.edu.sa

ABSTRACT

To be able to access a webpage or website we need its URL (Uniform Resource Locator) address, the problem is that not all of those websites are safe. We are interested on phishing web pages, those who try to imitate original ones-generally login or sale webpages -, the dangerousness of this kind of activities is that, it looks as trusting and the principal goal is to steal any user's (normal users, communities, societies, laboratories, etc.) personal information like name, date of birth, e-mail, credentials, login and passwords from e-banking services for example or any other web services. When victim fill his personal informations then it will be to the original one.

Our aim in this paper is to explore four techniques for detecting phishing web pages: Black-list based, Lexical based, Content based and Security and Identity based methods. After that, we construct a model combined with machine learning classifiers to classify if a test URL represents a safe or phishing web page, and we will show our framework that is based on a combination of those methods and the learning way is made by machine learning classifier algorithms, also it is based on rules. The data used to build our model is a gathered from PhishTank[1].

KEYWORDS

Smart systems and Communication, Network security intelligence, malicious URL detection, phishing, machine learning, spark framework.

SCAMS '17, October 25–27, 2017, Tangier, Morocco © 2017 Association for Computing Machinery. ACM ISBN 978-1-4503-5211-6/17/10…$15.00
https://doi.org/10.1145/3175628.3175633

1 INTRODUCTION

Now, Internet become the center of many transactions including money transactions from banking, online services, shopping, etc[2]. This means, it becomes the main focus area for cyber-criminal activities. Actually, criminals try to steal to informations from victims by letting this latter do it unconsciously, by filling some forms in web pages. To do so, generally, they send spoofed mails to victims by telling them that they have some problems in their accounts-banking, facebook, google, etc- and they should click on a url, when done, the victim finds a web page that looks like original one-visually speaking-, but as we know that each web page has its own URL, criminals uses the fact, that victims don't pay attention to the URL. After getting those informations, we can imagine nefarious uses, like selling in black market, hacking accounts, making borrows, accessing private places, with the name of the victim, criminal's goals and techniques become more sophisticated than ever, the design and the use of the malware scenario has changed, it is stealthier and polymorphic than damaging the machines. What makes phishing dangerous, is that, it doesn't depend on victim's knowledge in using computer or internet. This kind of attack becomes one of the main threat on the Internet nowadays [3].

The taxonomy of Phishing come from the fact it uses techniques like real fishing: lure, hook, and catch.

- The lure is the social engineering techniques (messages, visual perfection, …) used to catch the attention of the user, generally its main idea is telling victims to fill their informations in the hook.
- The hook is the fake website or email used.
- The catch is when criminals use gathered informations in nefarious manners

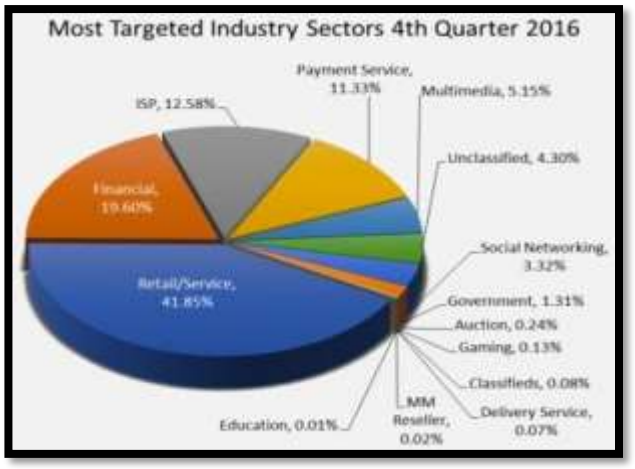

Figure 1: Phishing trends and most affected domains [3].

2 PHISHING DETECTION TECHNIQUES

2.1 Black-list Based Method

Big Companies use their ways to analyze websites either occasionally or by users request, when an URL is detected as phishing URL it is added to a database, generally this technique depends on software or hardware specifications, like browsers (Google safe Browsing, SpoofGuard, NetCraft), when a user tries to access an already black-listed URL in Chrome, it will receive a notification that this URL is nefarious or something like that as alert. Definitely, this technique has a good accuracy [4], because this URL was already black-listed, but what is known today is that, phishers create malicious websites for a short life time to do a special attack, so the blacklisting will offer anything if those websites weren't analyzed yet or the client browser database is out of date or some website was misclassified. This technique is also used to block IP addresses to defend not only phishing but also spam websites and e-mails.

However, this technique can't prevent such kind of attacks, it only capable to detect already black-listed websites. In fact, client's requests to analyze some websites are very recommended because companies can't analyze all existing websites especially new ones. Also, the main disadvantage of this method is that it depends on programs where it is installed like web browser's extensions or e-mail management plugins, so, criminals attack those program's failures and can sometimes easily break their security.

2.2 Lexical Based Method

Lexical based method analyzes the URL from a lexical point of view. The technical definition of a URL (https://www.w3.org/Addressing/URL/url-spec.txt) is as follow: <protocol>://<host>/<path>/<file>?<query>

e.g. https://mail.google.com/mail/#inbox,

- https is the protocol,
- mail.google.com is the hostname,
- mail is the path,
- # is called anchor to indicate to go directly to inbox directory from the path.

Studies show that malicious URL use different techniques to inject abnormal queries to the server where web page is stored (SQL injection for example), sometimes they are very long so normal don't pay attention on what is written exactly, others use an IP address instead of a normal hostname.

According to [5] and [6] URL lexical properties include the length of the hostname, the length of the entire URL, as well as the number of dots in the URL, all of these are real-valued features. the URL is split according to strings delimited by '/', '?', '.', '=', '-' and '_' to extract the binary feature for each token in the hostname (delimited by '.') and in the path URL.

After extracting those features, they are added to other kind of features from other type of analysis like WHOIS, AS, MX numbers -those features contain informations like "where" malicious sites are hosted, "who" own them, and "how" they are managed. The following are properties of the hosts (there could be multiple) that are identified by the hostname as part of the URL. The use of IP address instead of normal domain, so attackers can change it permanently, and the length of URL if it is so long to hide malicious techniques of redirection like using '@' symbol, or to some abnormal queries [5-6]- and that to build a model and then use machine learning to get a result if the current URL is a safe or not. Another observation, is that malicious URLs generally use some sensitive words like 'confirm', 'account', 'banking', 'secure', 'ebayisapi', 'webscr', 'login', 'signin'. All of This is also known as a "bag-of-words."

The main goal of this technique is to generate a lexical profile of the URL, this can also prevent new malicious URL from a lexical point of view not only to detect them, unless the model of this technique should be always updated to detect new abnormal URL forms.

2.3 Content Based Method

This method aims to analyze the content of the web page and tries to look for malicious scripts, like javascripts, shell codes. Javascript can permit to render web page as dynamic, so it can create other content after the web page is loaded or when clicking on button for example. The majority of phishing web pages try to get personal informations from victims, so users should foremost to fill those informations in forms and click a button to send them to hacker server. So, from this point analyzing the content of this web page will try to detect where those informations will be sent. Other feature consists on analyzing if the hacker uses iframes-web page inside a web page that can be specified with visibility to the client or not, e.g.:

<iframe src='http://maliciouswebsites.com/' width='1′ height='1′ style='visibility: hidden;'></iframe>- so malicious web page can be injected in iframes. Also, javascript can prohibit victims to not

do some tasks, like disabling write clicks or not be able to see the source code by encrypting it.

In our method, we will use other specifications which consists on comparing the visually similitude of two pages by analyzing CSS code, and the second is comparing the word appearance similitude using CANTINA+ [7] which is terms based, it is based on frequency of each term in a specific web page and use heuristics to generate a lexical identity for the web page to detect if it exists in search engines like google, if doesn't so it's marked as malicious.

Generally, the technique used in content based methods is calculating the number of each one of those features (Redirect, disabling right mouse click, pop- up windows, iframes, …), and for some threshold it profiles the page as suspicious or it's benign. This method is also used for preventing malicious web pages. But, those days, other problems are found like having the possibility to obfuscate the source code of web pages.

2.4 Security and Identity Based Method

Several phishing attacks leverage links to misdirect users to malicious websites or drive-by downloads. Since URLs bring information to users about what resource one tries to consult, phishers use obfuscation techniques to make phishing URLs look trustworthy for example using https:// for secure http. URLs typically contain DNS information i.e. the domain name and hostname of the server, the path of directories and files indicating the location of the consulted resource on the server. Most users actually read URLs but are not knowledgeable about the information they contain or the DNS hierarchy. Phishers use this lack of knowledge to mix terms of legitimate URLs in order to create phishing URLs and fool users. Except using https:// in URL, other features can be collected from identity of a web page, example: the age of the domain, studies show that phishers use very short age of domain, an average of one year, generally trusty and big companies buy domains for several years [8], also some features can be gathered from content, for example to compare where will the informations filled in forms be sent, is it the same server location of the web page or other location, or is it the same identity at least. Other features will be shown in the next part of our work.
This method is generally used like in criminology field which helps profiling the identity of criminals to detect them or prevent attacks before happening. But it's an abstract concept because profiling identities can't be precise and can't be rule based method. Those informations can be combined with profiling normal user's behaviour while using internet which demands a social engineering, Awareness and sensitization work.

2.5 Security and Identity Based Method

Methods used to detect phishing websites are very various in the proceeding concept, some of them uses heuristics combined with Black-listing, lexicography, content, visual, behavioural or identity based methods, all of them have their pros and cons in detecting false negative and false positive targets, and having accuracy and precision.
In the next section, we will discuss our framework which is based on a mix of black-listing, content, lexical and identity combined with machine learning classifier algorithms, the only main entry we need is the URL. We achieve a good accuracy to detect phishing websites and a low rate for false positives results.

3 OUR METHOD

3.1 Collecting Data

To test the efficiency of our method, we need to buil a model, it's a matrix of binary values which each column represents a specific feature for all rows, and each row is all the features for each URL. We use Phishtank to get phishing web pages and Alexa to get safe websites. After that, we extract features from each row to build our model and we Cross-Validation to bypass overfitting and make it better in to test new incoming URL entries.

In real world, data are gathered from local internet network, our method can be used in batch or in streaming mode, but sometimes to detect the behavioral execution of a suspicious URL, we use honeypots. In general, our framework works as WAF (Web Application Firewall).

3.2 Our Framework

3.2.1 Information extraction

A web page is a file that is stored in a server, we can access it by its unique URL sometimes called web link. Our work is to be able to detect if this web page is safe or phishing by using the URL only, the domain of application is to place our framework as WAF (Web Application Firewall) to not be dependent to browsers or systems specifications. That means, when a user tries to access a phishing website, it will be alerted if his system or browser is hacked. In general, our framework architecture is built as follows:

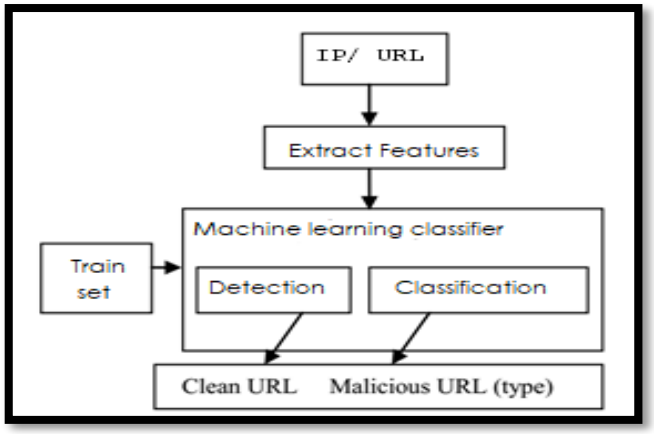

Figure 2: Global life cycle of our framework

Let's take this URL as test case to explain our methodology:

https://172.16.0.12/wordpress/wpincludes/kop/index.html@http://https.axrotoosh.com/modules/blockcustomerprivacy/translations/1amampa/5a5/index_2.html/index_2.html/index_2.html

The hole objective of our method is to transform our problematic to a binary classification problem, that means as output will have only 1 as value if a phishing web page or -1 if it's safe. The main idea is to use URL of web pages as main entry to our framework, then try to extract features from it, by using analyzing methods discussed above, and finally construct a binary row to be compared with our model using machine learning classifier. From a lexical point a view, we ca observe that this URL uses an IP address instead a valid hostname, in the row that we will construct the feature has_ip will have the value, if it wasn't the case it will be -1. Also, This URL is very long, we construct a rule for long_url feature as follow:

- If URL length<54 characters, long_url = -1
- Elif 54<URL length<75, long_url = 0
- Else (URL length>75), long_url = 1

And so on for other features, each one will have {-1, 0, 1} depends on feature analyze result [9][12].

Also, we can observer the presence of HTTPS protocol so normal users will think that is secured website, in our analysis we found that using HTTPS to delude normal users that's a secured website, it is among the most used tricks by cyber criminals in phishing attacks, in our dataset we found that more than 57.33% of phishing URLs use HTTPS protocol. To discover the reliability of SSL certificates of each website, we check if the certificate assigned with HTTPS including the extent of the trust certificate issuer, and the certificate age. Certificate Authorities that are consistently listed among the top trustworthy names include: 'GeoTrust, GoDaddy'.

We transform our problem into a binary classification problem. So, at the end we will have only two results: safe or phishing, we conclude that by applying machine learning algorithms for classifying new incoming URLs, after this, we get a result of new test URL it will be added to the model to be updated to new phishing features description. The next figure shows how we model our framework in real-time need.

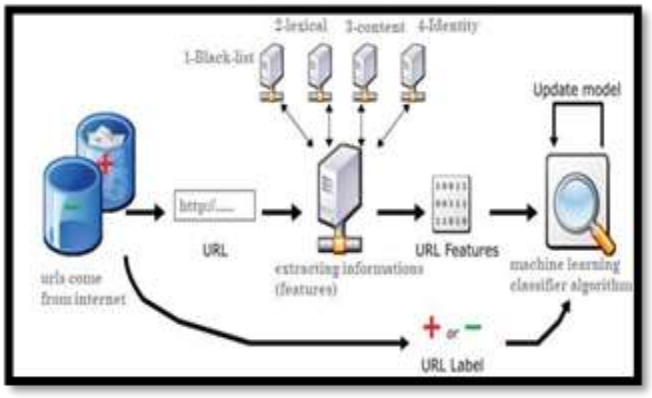

Figure 3: real time phishing detection framework

The table that follows - contains sample of our database overview- explains it well, all the attributes having a binary values space are generally denoting the absence or presence of respective attribute(features). Attributes with three possible values are generally representing its strength

Features Group	Phishing Functional	Values	Technical Feature identifier in the database
Lexical based analytics method	Having IP Address	-1, 1	has_ip
	Having long url	1, 0,-1	long_url
Abnormal Based Feature (Lexical based analytic method)	Request_URL	1,-1	req_url
	Abnormal URL anchor	- 1,0,1	url_of_anchor
	Links_in_tags	1,-1,0	tag_links
Content based analytics method	Redirect	0,1	redirect
	on mouseover	1,-1	mouseover
	Iframe	1,-1	iframe
Security Black-list and	Age of domai	-1, 0,1	domain_age
	DNS Record	-1,1	dns_record
	Web traffic	-1,0,1	traffic

identity based method	Page Rank	-1,1	page_rank
	Google Index	1,-1	google_index
Result	target	1,-1	target

Table 1: Our phishing dataset database, based on real still active phishing URL, it contains a sample of informations that we extracted from an URL

The 'taget' feature is the finql result that will have only two values: 1 if is phishing or -1 if it's a legitimate website. The aim of our heuristic analysis is to transform the problem of analyzing phishing URLs to a binary classification problem to be able to apply machine learning algorithms of binary classification like Logistic Regression or Naïve Bayes. Our dataset contains more than 1K data of real active phishing data gathered from PhishTan

3.2.1 Classification and analysis

We are using Spark Framework [10] which is an open source parallel processing framework for running large-scale data analytics applications across clustered or standalone computers. In this paper, we use all the analysis in one single node, Spark works very well to treat data in single node which very enough to our framework.

Apache Spark can process data from a variety of data sources, including the Hadoop Distributed File System (HDFS), NoSQL databases and relational data stores such as Apache Hive. Spark supports in-memory processing to boost the performance of big data analytics applications, and it can also do conventional disk-based processing when data sets are too large to fit into the available system memory [10]. The main difference between hadoop and Spark is that Spark can perform real time processing while Hadoop cannot, it only performs batch processing, second is that Spark is best the algorithms that uses multiple iterations.

For example, we use Spark Sql Data Frames to be able sql queries directly and we use Python as programing language because Python is a leader coding language for data science, a simple example is to show which features is the most used by phishing in our dataset, so the phishing feature is when it equals to 1 and target is also 1(target=1 refers to phishing result), we wrote a SQL query as follow:

```
frequencies = {}
for feature in header:
    if feature!='target':
        count = spark.sql("SELECT "+ feature +" , count(*) FROM
PhishingTable WHERE "+ feature +"=1 AND target = 1 GROUP BY "+
feature).collect()[0][1]
        frequencies.update({feature:count})
```

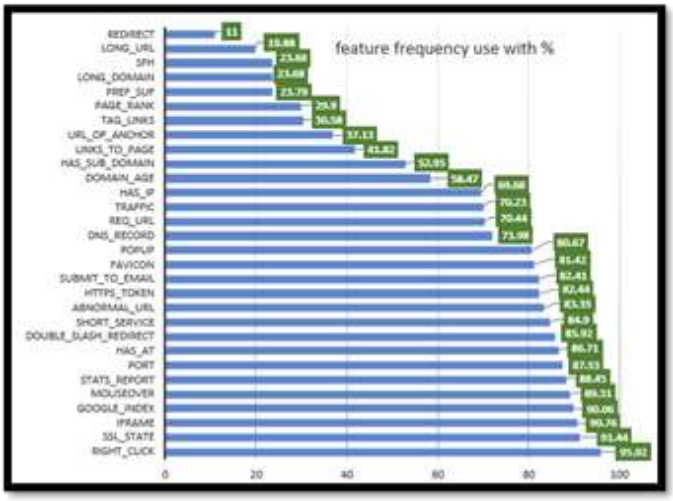

Figure 4 and 5: most used features by phisher

What makes Spark awesome for using, is the simplicity of its syntax and its support for many programming language and multiple data sources. A sample of our database is as follow:

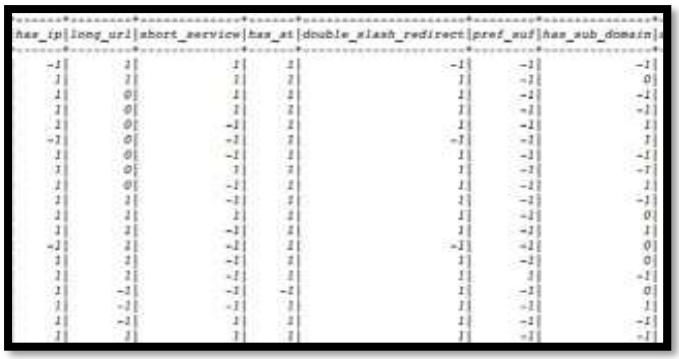

Figure 6: Sample of data used to build the model

After that, we use linear regression to predict new injected data gathered from PhishTank and safe websites from Alexa. Linear regression analysis is the most widely used of all statistical techniques: it is the study of linear, additive relationships between variables [11].

Linear regression is perfect for our work, because the target has only two values 1 to pishing and -1 to safe, we use LinearRegression API from pyspark.ml.regression, and we inject data without target and the goal is to predict the target for a giving row in the test data. And the result was like as follow.

features	prediction
[-1.0,1.0,1.0,1.0...	-0.6540525649055468
[1.0,1.0,1.0,1.0,...	0.3957894432781284
[1.0,0.0,1.0,1.0,...	-0.28057600110381736
[1.0,0.0,1.0,1.0,...	-0.28057600110381736
[1.0,0.0,-1.0,1.0...	0.3957894432781284
[-1.0,0.0,-1.0,1....	0.3957894432781284
[1.0,0.0,-1.0,1.0...	-0.6540525649055468
[1.0,0.0,1.0,1.0,...	-0.28057600110381736
[1.0,0.0,-1.0,1.0...	0.3957894432781284
[1.0,1.0,-1.0,1.0...	0.3957894432781284
[1.0,1.0,1.0,1.0,...	0.3957894432781284
[1.0,1.0,-1.0,1.0...	-0.6540525649055468
[-1.0,1.0,-1.0,1....	-0.31586984271457386
[1.0,1.0,-1.0,1.0...	-0.6540525649055468
[1.0,1.0,-1.0,1.0...	0.3957894432781284
[1.0,-1.0,-1.0,-1...	-0.31586984271457386
[1.0,-1.0,-1.0,1....	0.3957894432781284
[1.0,-1.0,1.0,1.0...	0.057606721087155 51
[1.0,1.0,1.0,1.0,...	0.3957894432781284
[1.0,1.0,1.0,1.0,...	0.3957894432781284

Table 2: Example of predicted target in test data (the idea of test of our framework, each line is the row corresponding of a test url that we already to binary features as in Features in the figure without target feature, and prediction is the predicted value for each row)

So, if the prediction is less than 0 we will say that is safe, else it's phishing url. We get: RMSE (root mean square error): 0.148314, which is fairly well in our test program.

Assume that (1):
- TP: Number of urls that were classified as phishing correctly
- FP: Number of phishing urls that were classified as safe
- FN: Number of safe urls that were classified as phishing
- TN: Number of safe urls that were classified correctly as safe

From (1) we can measure the efficiency our model, by measuring accuracy, precision and recall, by the following relations:
- Precision-phishing = TP/(TP + FP)
- Recall-phishing = TP/(TP + F N)
- Accuracy = (TP + T N)/(TP + FP + T N + F N)

All treatments were made in one - i7-4600U quad cores- single node with SSD disk drive and 16Go of ram, in around 1200 tests on URLs. Our framework represents an accuracy about 95.5%, and 1.4% of false positive.

4 CONCLUSIONS

In this paper, the aim is to show that our framework is combining different perspectives to detect and also prevent phishing websites or web pages using only the URL as main entry and Spark as platform, and show that it will be the next generation of streaming analytic data platform.

Our future work will try to answer those real questions:
- What is the effective minimal set of features that can be utilized to predict a phishing URL?
- How good is rule-based data-mining technique in predicting phishing websites?
- Which rule based classification technique is more accurate in predicting phishing websites?

REFERENCES

[1] https://www.phishtank.com/developer_info.php

[2] Abubakr Sirageldin, Baharum B. Baharudin, Low Tang Jung, *Malicious Web Page Detection:A Machine Learning Approach*, Advances in Computer Science and its Applications Volume 279 of the series Lecture Notes in Electrical Engineering pp 217-224

[3] APWG 2016, Phishing Activity Trends Report 3rd Quarter 2016

[4] Steve Sheng, Brad Wardman, Gary Warner, Chengshan Zhang 2009, *An Empirical Analysis of Phishing Blacklists*, Proceedings of Sixth Conference on Email and Anti-Spam (CEAS)

[5] Myriam Abramson, David W. Aha 2012, *What's in a URL? Genre Classification from URLs*, Workshops at the Twenty-Sixth AAAI Conference on Artificial Intelligence, Intelligent Techniques for Web Personalization and Recommendation

[6] Aaron Blum, Brad Wardman, Thamar Solorio, Gary Warner 2010, *Lexical feature based phishing URL detection using online learning*, Proceeding AISec '10 Proceedings of the 3rd ACM workshop on Artificial intelligence and security Pages 54-60 ACM New York, NY, USA

[7] Guang Xiang, Jason Hong, Carolyn P. Rose, Lorrie Cranor 2011, CANTINA+: A Feature-rich Machine Learning Framework for Detecting Phishing Web Sites, ACM Transactions on Information and System Security (TISSEC): Volume 14 Issue 2, September

[8] Ee Hung Chang, Kang Leng Chiew, San Nah Sz, and Wei King Tiong 2013, *Phishing Detection via Identification of Website Identity*, Conference: 2013 International Conference on IT Convergence and Security (ICITCS)

[9] Sadia Afroz, Rachel Greenstadt 2011, *PhishZoo: Detecting Phishing Websites by Looking at Them, Semantic* Computing (ICSC), Fifth IEEE International Conference

[10] http://hadoop.apache.org/

[11] Sunila Gollapudi 2016, *Practical Machine Learning*, https://www.packtpub.com/big-data-and-business- intelligence/practical-machine-learning, January

[12] Anh Le, Athina Markopoulou, Michalis Faloutsos, *PhishDef: URL Names Say It All*, Cryptography and Security (cs.CR); Learning (cs.LG); Networking and Internet Architecture (cs.NI), arXiv.org