Scoring Domains

Introduction

This is a model for scoring domains in order to identify suspicious domains based on various factors. Identifying bad domains is important in order to infer likelihood of attack since domain is an important surface for transferring and delivering attack vectors. In order to identify those domains various characteristics are important:

- Age of domain
    - Older domains are more likely to be more established hence more trustworthy
- Lexical structure of domain
    - Creation of spurious domains points to domain generation algorithm (DGA)
    - Creation of variants of well-known domain names to hoodwink users
- TLD of domain
    - Some top-level domains could be hijacked and used for nefarious activities
- DNS properties of domain
    - DNS protocol could be misused to mask the real domains used in certain web transactions
- Historical perspective
    - One assumption that could be used to reduce false alarms in predicting bad domain is whether that domain has been seen before in the logs belonging to a client. Usually, repeated use of a domain could indicate that it is harmless or vice versa. There are no guarantees here but usually anomalies constitute rare events unless we are looking at high frequency attacks such as botnets etc. At the very least a new domain that has not been seen before which also shares other 'bad domain' traits should be highly suspicious and further investigation is required.

Based on these characteristics we create a model which would score domains based on their likelihood being trustworthy. The score is not meant to indicate an attack per se but an indicator that a domain shares hallmarks of known attack artefacts.

Scoring Model – Version 1

The first version of this model relies on – lexical structure of domain and TLD characteristics of the model. Other versions of the model will incorporate other characteristics in an incremental fashion. The first version of this model is based on data that is easily available and uses mathematical techniques that are easy to code up, productionize and audit in a way that is easy to relay the decision-making logic of the model. We use a naïve Multinomial bayes algorithm to score the domains after extracting features from event logs.

Input data

We rely on known good and bad domains to generate lexical features of the model. For good domains, we utilize the Alexa 1 or 10 million to infer good characteristics pf a domain. Obviously, it is possible that a bad domain could be a heavily used Alexa domain but based on experience most good domain will have a longer history of use and hence likely to be heavily trafficked on and the vice versa id also true. We also use bad domain feeds such as PhishTank web feeds to infer lexical characteristics of known bad domains.

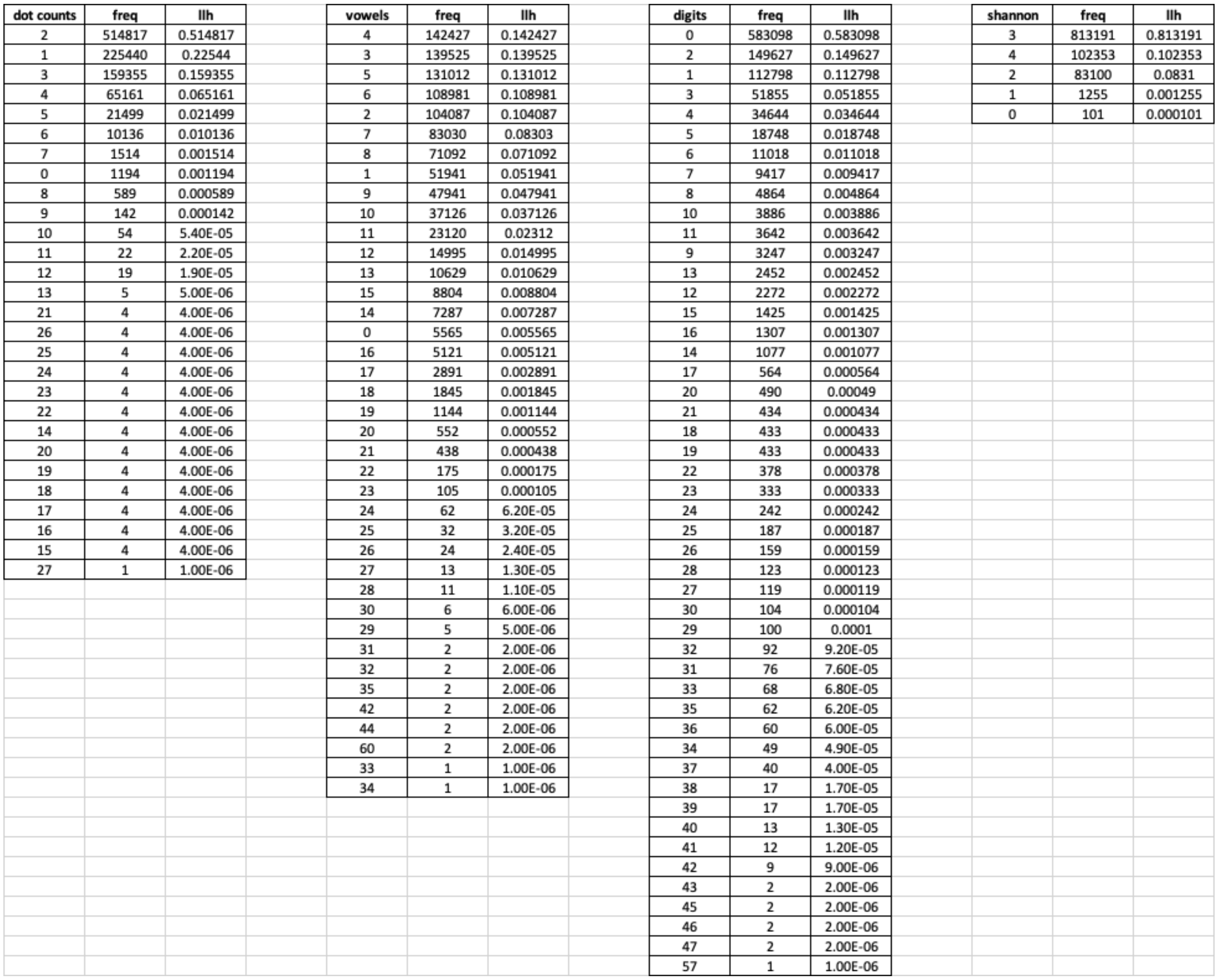
Feature modeling

In order to build our model, we extract the following features from Alexa 1 million events:

- Shannon entropy
    - Captures distribution of characters within a string. Different languages will have most of their words falling between certain entropy ranges and so is the domains especially because most of the original domain names borrowed heavily from known languages. This is not the case anymore hence this feature will continue waning in value.
- Vowels count
    - Most DGA domains will not incorporate language structures and most of the characters will be randomly generated and this feature is useful for detecting such behaviors.
- Count of digits
    - Digits were rare in original domains but are becoming more commonplace now. Many DGA domains are likely to incorporate random digits.
- Count of dots
    - Usually, dots denote subsequence in subdomain structure but can be used to hide bad domains in hijacked TLD or slightly altered domain names.
- Slashes (“/”)
    - Usually capture the path of a URL page. Slashes are not allowed in domain name, but their presence could indicate errors in domain name formatting during data collection. If URLs are being modeled instead of domain names, this feature would be even more useful.
- Count of symbols - ['!','@','#','$','%','^',’&]
    - Usually, domain names will not have symbols other than period(.) and hashes (-). Repeated use of such symbols is rare and could indicate misuse of domain naming. Such symbols could also be used to fake well known domain names.
- Length of domain
    - Original and popular domains will have names that mostly represent their business names and would not be overly large. Newer domains will be longer and more so for bad domains
- 1 X 9 vector of sequences between (digits, alpha, symbols)
    - This captures character subsequence within a string. Ordinary domain names will most likely have alphanumeric subsequence while newer and bad domain names might incorporate sequences that are less common. We generate a nine-feature vector by counting the following subsequence where *“d”* represents digits, “a” for alphanumeric, “s” for symbols ['d_d','d_a','d_s','a_d','a_a','a_s','s_d','s_a','s_s']

Extracting features from good domains

Features are extracted from good domains – in this case Alexa 1 million, by calculating the discrete distributions of the 16 features described above. For instance, the table below shows the distribution of domain lengths from Alexa 1 Million events.



dot counts	freq	llh
2	514817	0.514817
1	225440	0.22544
3	159355	0.159355
4	65161	0.065161
5	21499	0.021499
6	10136	0.010136
7	1514	0.001514
0	1194	0.001194
8	589	0.000589
9	142	0.000142
10	54	5.40E-05
11	22	2.20E-05
12	19	1.90E-05
13	5	5.00E-06
21	4	4.00E-06
26	4	4.00E-06
25	4	4.00E-06
24	4	4.00E-06
23	4	4.00E-06
22	4	4.00E-06
14	4	4.00E-06
20	4	4.00E-06
19	4	4.00E-06
18	4	4.00E-06
17	4	4.00E-06
16	4	4.00E-06
15	4	4.00E-06
27	1	1.00E-06

vowels	freq	llh
4	142427	0.142427
3	139525	0.139525
5	131012	0.131012
6	108981	0.108981
2	104087	0.104087
7	83030	0.08303
8	71092	0.071092
1	51941	0.051941
9	47941	0.047941
10	37126	0.037126
11	23120	0.02312
12	14995	0.014995
13	10629	0.010629
15	8804	0.008804
14	7287	0.007287
0	5565	0.005565
16	5121	0.005121
17	2891	0.002891
18	1845	0.001845
19	1144	0.001144
20	552	0.000552
21	438	0.000438
22	175	0.000175
23	105	0.000105
24	62	6.20E-05
25	32	3.20E-05
26	24	2.40E-05
27	13	1.30E-05
28	11	1.10E-05
30	6	6.00E-06
29	5	5.00E-06
31	2	2.00E-06
32	2	2.00E-06
35	2	2.00E-06
42	2	2.00E-06
44	2	2.00E-06
60	2	2.00E-06
33	1	1.00E-06
34	1	1.00E-06

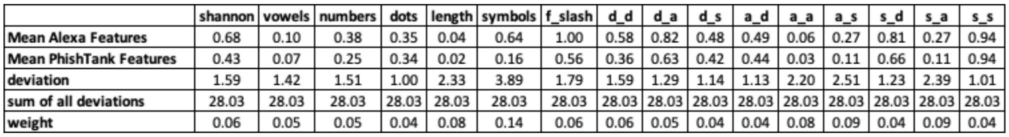
digits	freq	llh
0	583098	0.583098
2	149627	0.149627
1	112798	0.112798
3	51855	0.051855
4	34644	0.034644
5	18748	0.018748
6	11018	0.011018
7	9417	0.009417
8	4864	0.004864
10	3886	0.003886
11	3642	0.003642
9	3247	0.003247
13	2452	0.002452
12	2272	0.002272
15	1425	0.001425
16	1307	0.001307
14	1077	0.001077
17	564	0.000564
20	490	0.00049
21	434	0.000434
18	433	0.000433
19	433	0.000433
22	378	0.000378
23	333	0.000333
24	242	0.000242
25	187	0.000187
26	159	0.000159
28	123	0.000123
27	119	0.000119
30	104	0.000104
29	100	0.0001
32	92	9.20E-05
31	76	7.60E-05
33	68	6.80E-05
35	62	6.20E-05
36	60	6.00E-05
34	49	4.90E-05
37	40	4.00E-05
38	17	1.70E-05
39	17	1.70E-05
40	13	1.30E-05
41	12	1.20E-05
42	9	9.00E-06
43	2	2.00E-06
45	2	2.00E-06
46	2	2.00E-06
47	2	2.00E-06
57	1	1.00E-06

shannon	freq	llh
3	813191	0.813191
4	102353	0.102353
2	83100	0.0831
1	1255	0.001255
0	101	0.000101

The four tables show discrete distributions of dots, vowels, digits and Shannon entropy. Shannon entropy is actually a continuous distribution by we discretized it by rounding down then Shannon entropy values. During the training, which involves computing the discrete values from Alexa 1 million domains, we obtain 16 such tables for all the 16 features. The feature values are the likelihoods of various values witnessed in the Alexa domain events. For instance, the likelihood of seeing 2 dots in a domain name in Alexa 1 million is 0.51 - close to 51%. So, at this juncture we have a trained model of 16 tables which we can use to predict likelihood of future events.

Prediction

For every new domain we extract features the same way then we use 16 training set tables to infer likelihood (from Alexa's perspective). From the 16 predictions (likelihoods) the final prediction (outcome) can be a simple average or to be more precise a weighted outcome. The weighting of the predicted likelihoods can be based on domain knowledge or real differences between distributions of good domains and bad domains. For instance, by comparing the average values of features extracted from good domains and bad domains the weighting can be based on the differences /deviations between them. This will ensure that features that show greatest difference have a bigger effect in the predicted outcomes.



	shannon	vowels	numbers	dots	length	symbols	f_slash	d_d	d_a	d_s	a_d	a_a	a_s	s_d	s_a	s_s
Mean Alexa Features	0.68	0.10	0.38	0.35	0.04	0.64	1.00	0.58	0.82	0.48	0.49	0.06	0.27	0.81	0.27	0.94
Mean PhishTank Features	0.43	0.07	0.25	0.34	0.02	0.16	0.56	0.36	0.63	0.42	0.44	0.03	0.11	0.66	0.11	0.94
deviation	1.59	1.42	1.51	1.00	2.33	3.89	1.79	1.59	1.29	1.14	1.13	2.20	2.51	1.23	2.39	1.01
sum of all deviations	28.03	28.03	28.03	28.03	28.03	28.03	28.03	28.03	28.03	28.03	28.03	28.03	28.03	28.03	28.03	28.03
weight	0.06	0.05	0.05	0.04	0.08	0.14	0.06	0.06	0.05	0.04	0.04	0.08	0.09	0.04	0.09	0.04

Other methods of weighting the predictors can be based on the well-known principal component analysis (PCA) or statistical measures such as Chi-Square distributions but for now we will keep it simple.

The final predicted likelihood of a domain is obtained by performing a dot product of the [16 X 1] feature vector of likelihoods and the [1 X 16] weighting vector.

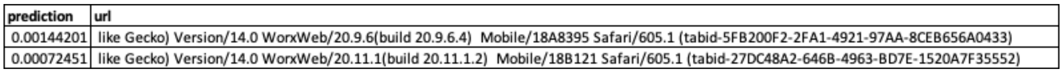
Results

We built a model using Alexa 1 million domains and tested it on two datasets:

- PhishTank data set containing bad domains
- OMF data set containing over 400K domains collected from customer events.

Analysis

The model is able to identify domains that are very different structurally from what normal domains are supposed to look like. Some of the domains identified are malformed probably due to the data collection process. See below for the domains identified from OMF data set with very low prediction indicating how likely they are:

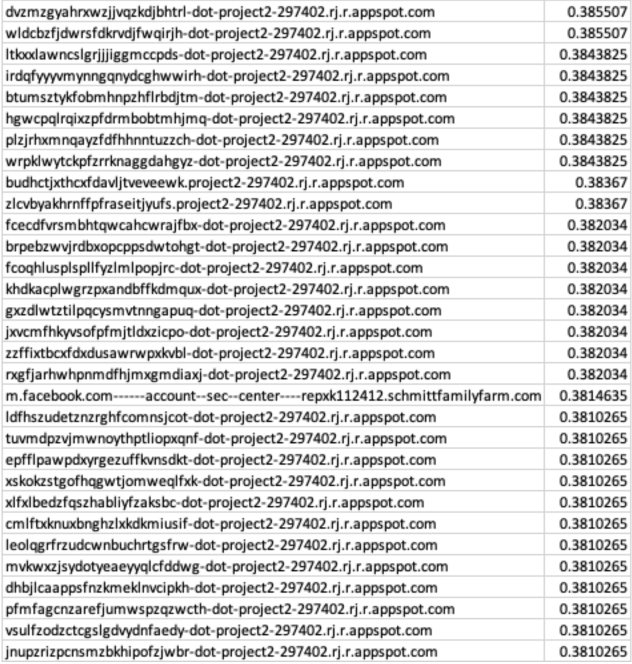


prediction	url
0.00144201	like Gecko) Version/14.0 WorxWeb/20.9.6(build 20.9.6.4) Mobile/18A8395 Safari/605.1 (tabid-5FB200F2-2FA1-4921-97AA-8CEB656A0433)
0.00072451	like Gecko) Version/14.0 WorxWeb/20.11.1(build 20.11.1.2) Mobile/18B121 Safari/605.1 (tabid-27DC48A2-646B-4963-BD7E-1520A7F35552)

It seems that User Agent strings were injected on domain name columns.

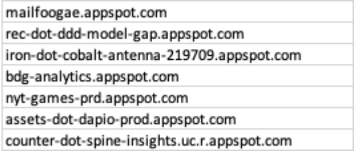
The model is weak when confronted with ‘good-looking’ domain names which conform to expected domain name structures. This is evident from the many events collected from PhishTank (known bad domains) that are able to attain high probability values. It is expected that a model that depends heavily on lexical features would have such weaknesses. To remedy this, we need to include more non-lexical features.

The current model is very good at sorting similar domain names even without considering the characters in the string themselves. We can use this behavior to our advantage to isolate really bad domains. The events below are bad domain events sorted by their likelihood which ensures they are all gathered consecutively. A security analyst looking at them would notice the commonality between them which is that they belong to the same domain name “appspot.com”. The fact that so many sub domains belonging to bad domains share the same domain name means that the domain itself has been heavily compromised and its events need to be carefully scrutinized.



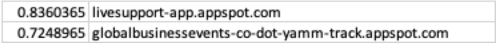
dvzmzgyahrxwzjjvqzkdjbhtrl-dot-project2-297402.rj.r.appspot.com	0.385507
wldcbzfjdwrsfdkrvdjfwqirjh-dot-project2-297402.rj.r.appspot.com	0.385507
ltkxxlawncslgrjjjiggmccpds-dot-project2-297402.rj.r.appspot.com	0.3843825
irdqfyyyvmynngqnydcghwwirh-dot-project2-297402.rj.r.appspot.com	0.3843825
btumsztykfobmhnpzhflrbdjtm-dot-project2-297402.rj.r.appspot.com	0.3843825
hgwcpqlrqixzpfdrmbobtmhjmq-dot-project2-297402.rj.r.appspot.com	0.3843825
plzjrhxmnqayzfdfhhnntuzzch-dot-project2-297402.rj.r.appspot.com	0.3843825
wrpklwytckpfzrrknaggdahgyz-dot-project2-297402.rj.r.appspot.com	0.3843825
budhctjxthcxfdavljtveveewk.project2-297402.rj.r.appspot.com	0.38367
zlcvbyakhrnffpfraseitjyufs.project2-297402.rj.r.appspot.com	0.38367
fcecdfvrsmbhtqwcahcwrajfbx-dot-project2-297402.rj.r.appspot.com	0.382034
brpebzwvjrdbxopcppsdwtohgt-dot-project2-297402.rj.r.appspot.com	0.382034
fcoqhlusplspllfyzlmlpopjrc-dot-project2-297402.rj.r.appspot.com	0.382034
khdkacplwgrzpxandbffkdmqux-dot-project2-297402.rj.r.appspot.com	0.382034
gxzdlwtztilpqcysmvtnngapuq-dot-project2-297402.rj.r.appspot.com	0.382034
jxvcmfhkyvsofpfmjtldxzicpo-dot-project2-297402.rj.r.appspot.com	0.382034
zzffixtbcxfdxdusawrwpxkvbl-dot-project2-297402.rj.r.appspot.com	0.382034
rxgfjarhwhpnmdfhjmxgmdiaxj-dot-project2-297402.rj.r.appspot.com	0.382034
m.facebook.com------account--sec--center----repxk112412.schmittfamilyfarm.com	0.3814635
ldfhszudetznzrghfcomnsjcot-dot-project2-297402.rj.r.appspot.com	0.3810265
tuvmdpzvjmwnoythptliopxqnf-dot-project2-297402.rj.r.appspot.com	0.3810265
epfflpawpdxyrgezuffkvnsdkt-dot-project2-297402.rj.r.appspot.com	0.3810265
xskokzstgofhqgwtjomweqlfxk-dot-project2-297402.rj.r.appspot.com	0.3810265
xlfxlbedzfqszhabliyfzaksbc-dot-project2-297402.rj.r.appspot.com	0.3810265
cmlftxknuxbnghzlxkdkmiusif-dot-project2-297402.rj.r.appspot.com	0.3810265
leolqgrfrzudcwnbuchrtgsfrw-dot-project2-297402.rj.r.appspot.com	0.3810265
mvkwxzjsydotyeaeyyqlcfddwg-dot-project2-297402.rj.r.appspot.com	0.3810265
dhbjlcaappsfnzkmeklnvcipkh-dot-project2-297402.rj.r.appspot.com	0.3810265
pfmfagcnzarefjumwspzqzwcth-dot-project2-297402.rj.r.appspot.com	0.3810265
vsulfzodzctcgslgdvydnfaedy-dot-project2-297402.rj.r.appspot.com	0.3810265
jnupzrizpcnsmzbkhipofzjwbr-dot-project2-297402.rj.r.appspot.com	0.3810265

We tracked similar domain names in Alexa. These are just a few



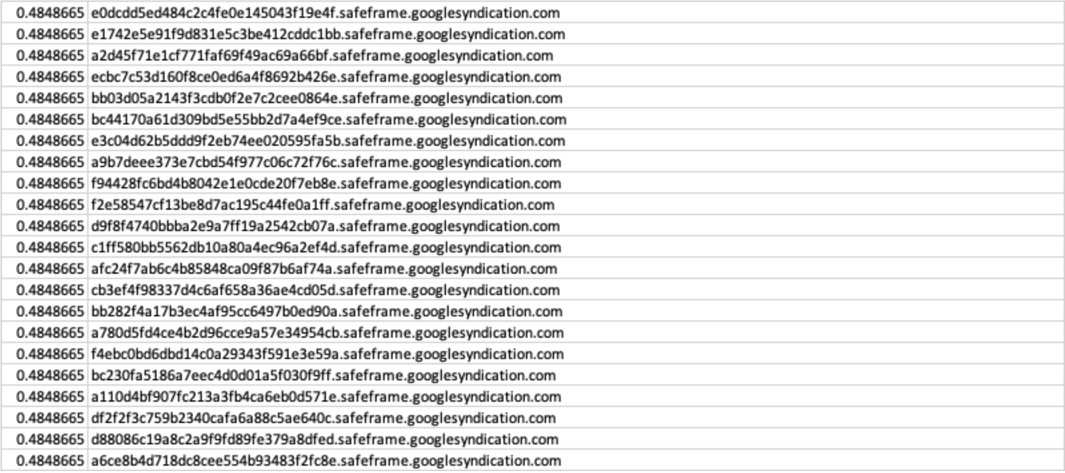
mailfoogae.appspot.com
rec-dot-ddd-model-gap.appspot.com
iron-dot-cobalt-antenna-219709.appspot.com
bdg-analytics.appspot.com
nyt-games-prd.appspot.com
assets-dot-dapio-prod.appspot.com
counter-dot-spine-insights.uc.r.appspot.com

And OMF data set



0.8360365	livesupport-app.appspot.com
0.7248965	globalbusinessevents-co-dot-yamm-track.appspot.com

Lastly the model can be harsh to some of the domains based on how they look but a look at the domain name indicates that the domain itself could be harmless. A good example is the following events from OMF dataset



0.4848665	e0dcdd5ed484c2c4fe0e145043f19e4f.safeframe.googlesyndication.com
0.4848665	e1742e5e91f9d831e5c3be412cddc1bb.safeframe.googlesyndication.com
0.4848665	a2d45f71e1cf771faf69f49ac69a66bf.safeframe.googlesyndication.com
0.4848665	ecbc7c53d160f8ce0ed6a4f8692b426e.safeframe.googlesyndication.com
0.4848665	bb03d05a2143f3cdb0f2e7c2cee0864e.safeframe.googlesyndication.com
0.4848665	bc44170a61d309bd5e55bb2d7a4ef9ce.safeframe.googlesyndication.com
0.4848665	e3c04d62b5ddd9f2eb74ee020595fa5b.safeframe.googlesyndication.com
0.4848665	a9b7deee373e7cbd54f977c06c72f76c.safeframe.googlesyndication.com
0.4848665	f94428fc6bd4b8042e1e0cde20f7eb8e.safeframe.googlesyndication.com
0.4848665	f2e58547cf13be8d7ac195c44fe0a1ff.safeframe.googlesyndication.com
0.4848665	d9f8f4740bbba2e9a7ff19a2542cb07a.safeframe.googlesyndication.com
0.4848665	c1ff580bb5562db10a80a4ec96a2ef4d.safeframe.googlesyndication.com
0.4848665	afc24f7ab6c4b85848ca09f87b6af74a.safeframe.googlesyndication.com
0.4848665	cb3ef4f98337d4c6af658a36ae4cd05d.safeframe.googlesyndication.com
0.4848665	bb282f4a17b3ec4af95cc6497b0ed90a.safeframe.googlesyndication.com
0.4848665	a780d5fd4ce4b2d96cce9a57e34954cb.safeframe.googlesyndication.com
0.4848665	f4ebc0bd6dbd14c0a29343f591e3e59a.safeframe.googlesyndication.com
0.4848665	bc230fa5186a7eec4d0d01a5f030f9ff.safeframe.googlesyndication.com
0.4848665	a110d4bf907fc213a3fb4ca6eb0d571e.safeframe.googlesyndication.com
0.4848665	df2f2f3c759b2340cafa6a88c5ae640c.safeframe.googlesyndication.com
0.4848665	d88086c19a8c2a9f9fd89fe379a8dfed.safeframe.googlesyndication.com
0.4848665	a6ce8b4d718dc8cee554b93483f2fc8e.safeframe.googlesyndication.com

These events are marked down because of their unappealing structure but a look at the domain "googlesyndication.com" indicates it is one of the top domains in Alexa 1 Million.

Deductions

Based on the results obtained from the current model we have pointed some successes of the model in identifying very different domain names. We have also identified areas where it fails because lexical characteristics do not give the whole picture. To improve the model, we need to incorporate other non-lexical features or build another complimentary model using DNS and historical / usage features of domains. To minimize false alarms, we can use the Alexa 1 million as an "allow list" to clear some domains which have been falsely captured by the model. Querying each and every domain event using Alexa 1 million might be computationally challenging but querying just a few domain names with low prediction is practically easier to implement.