Multilabel URL Classification using Feature Learning

Enzo Casamassima

CMPE-789

Spring 2021

Motivation

Traditional detection of malicious URLs is done by using blacklists [11-13], which by definition are finite and cannot possibly contain newly generated URLs and some of them are usually only up for a few hours and the only data left might be the URL [8].

Feature learning, also known as representation learning, is defined by Bengio et al. [9] as “learning representations of the data that make it easier to extract useful information when building classifiers or other predictors".

By training a machine learning algorithm, ideally without manually engineered features, it has the potential to generalize better to unseen and evolving data.

Problem statement and hypotheses

With new malicious URLs being created every second, it is important that we devise a way to detect and block them as fast as possible to protect users.

Hypotheses:

1. Multilabel URL classification is possible by only using the URL itself, without any additional information about the website itself.
2. The use of feature learning can have comparable performance to using manually engineered features on a specific dataset,
3. The use of feature learning could possibly find a better representation to classify unseen/evolving data and might generalize better to very distinct datasets.

Breakdown of hypotheses

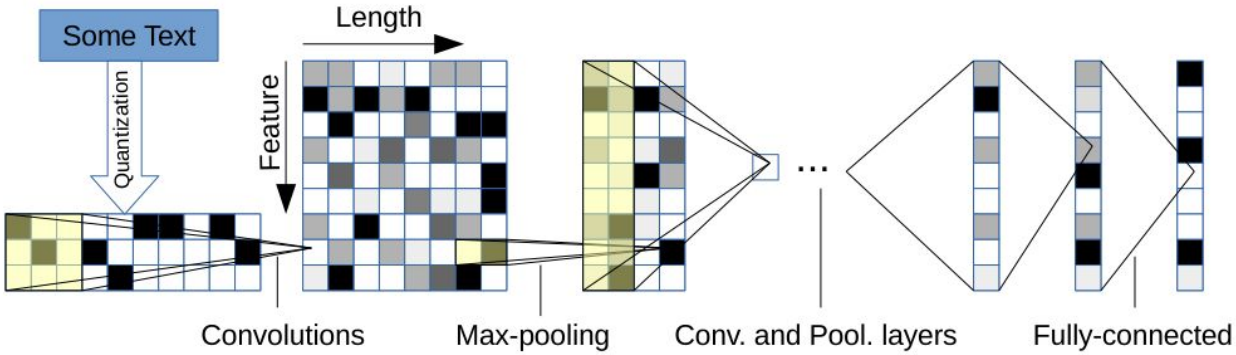
1. Multilabel URL classification is possible by only using the URL itself, without any additional information about the website itself.
    - By the time we get the URL the site might be down already
2. The use of feature learning can have comparable performance to using manually engineered features on a specific dataset,
    - If we can achieve similar results with both approaches, it might not be worth investing the time and effort to manually engineer these features.
3. The use of feature learning could possibly find a better representation to classify unseen/evolving data and might generalize better to very distinct datasets.
    - Since there is no universally accepted features for URL classification, a general big set of features could be learned stochastically by a deep neural network or other ML algorithm.

Related works

- Mamum et al. [4] create the ISCX2016 dataset and propose lexical analysis of the different parts of the URL to classify it. Used KNN and Random Forest.
- Le et al. [12] improve over existing methods (Bag-of-words features) by using a combination of a char level and word level CNN - treats the URL as a sequence. They achieve slightly better AUC than various SVM models
- Many other methods also include looking at P Address properties, WHOIS information, Location, DNS [10]

Methodology

- TF-IDF: Sparse matrix features
  - Random Forest
  - Multinomial Bayes
- Character Level CNN: Embedding layer -> 1D convolutions
  - Padded input sequence 400 chars (max URL length)
  - "alphabet": "abcdefghijklmnopqrstuvwxyz0123456789-,;.!?:'\"/\\|_@#$%^&*˜`+-=<>()[]{}"
  - Code: https://github.com/rashimo/ChCNN



Methodology: TF-IDF

- **TF*IDF** (term frequency–inverse document frequency) is a numerical statistic intended to reflect how important a word (feature) is to a document (sample) in a collection (dataset).
- Intuition [2]:
  - If a word appears frequently in a document, then it should be important and should have a high score.
  - But if the same word appears in too many other documents, it's probably not a unique identifier, therefore we should assign a lower score.
- This gives us a way to represent our URLs as a vector and feed it to a classifier!

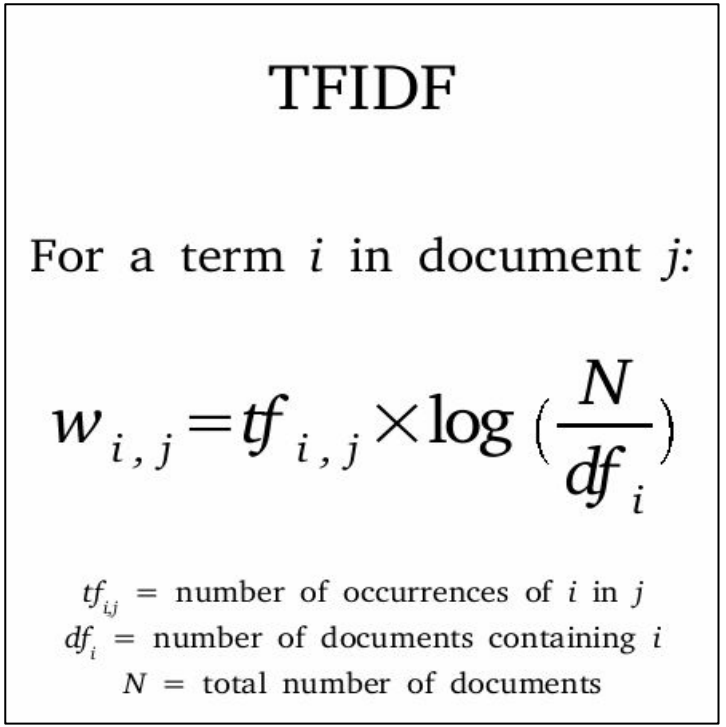


Image credit to [1]

Example (credit to [3])

Sentence 1 : The car is driven on the road. ||| **Sentence 2:** The truck is driven on the highway.

Word	TF		IDF	TF*IDF	
	A	B		A	B
The	1/7	1/7	log(2/2) = 0	0	0
Car	1/7	0	log(2/1) = 0.3	0.043	0
Truck	0	1/7	log(2/1) = 0.3	0	0.043
Is	1/7	1/7	log(2/2) = 0	0	0
Driven	1/7	1/7	log(2/2) = 0	0	0
On	1/7	1/7	log(2/2) = 0	0	0
The	1/7	1/7	log(2/2) = 0	0	0
Road	1/7	0	log(2/1) = 0.3	0.043	0
Highway	0	1/7	log(2/1) = 0.3	0	0.043

Dataset description

ISCX-URL2016:

```
5332  http://secure.runescape.com.d.weblagon.loginforn.com/mod/?/&amp;
5333  http://secure.runescape.com.nnweb.asia/m=weblogin/loginform.html
5334  http://soloclick.com.ve/49/config/login_verify2&amp;.src=/ym.htm
5335  http://spectrumofabomination.co.uk/wp-content/assistance/freebo/
5336  http://tant.co.kr/letsmall/admin/Template_c/properties/index.htm
5337  http://tibia.com.subtopic.community.guilds.w.interia.pl/acc.html
5338  http://us.battle.net.login.en.rrweb.asia/login/en/login.html?amp
5339  http://www.3kz.us/wp-content/themes/folioway/css/remax/index.htm
5340  http://www.520168.cn/style/?ref=http://us.battle.net/d3/en/index
```

- ISCX-URL2016 Dataset [4]:
  - 35,378 benign URLs
  - 11,566 malware URLs
  - 9,965 phishing URLs
  - 12,000 spam URLs
  - Engineered features: Domain_token_count; executable; NumberofDotsinURL; Arguments_LongestWordLength; NumberRate_Domain; NumberRate_FileName; NumberRate_AfterPath; Entropy_Domain
- Phishing Dataset [5]:
  - 38,758 phishing URLs
- FWAF Dataset [6]:
  - Queries, not URLs
  - Benign = 1,294,531
  - Malicious = 48,126

FWAF:

```
1   /top.php?stuff='uname >q36497765 #
2   /h21y8w52.nsf?<script>cross_site_scripting.nasl</script>
3   /ca000001.pl?action=showcart&hop=\"><script>alert('vulnerable')</script>&path=acatalog/
4   /scripts/edit_image.php?dn=1&userfile=/etc/passwd&userfile_name= ;id;
5   /javascript/mta.exe
6   /examples/jsp/colors/kernel/loadkernel.php?installpath=/etc/passwd\x00
7   /examples/jsp/cal/feedsplitter.php?format=../../../../../../../../../../etc/passwd\x00&debug=1
8   /phpwebfilemgr/index.php?f=../../../../../../../../../etc/passwd
9   /cgi-bin/script/cat_for_gen.php?ad=1&ad_direct=../&m_for_racine=</option></select><?phpinfo();?>
10  /examples/jsp/cal/search.php?allwords=<br><script>foo</script>&cid=0&title=1&desc=1
11  /moodle/filter/tex/texed.php?formdata=foo&pathname=foo"+||+echo+db+4d+5a+50+00+02+00+00+00+04+00+0f+00
```

Data preprocessing and Metrics

- TF-IDF
  - Replace number sequences with “NUMBER_TOKEN”
  - Split URLs at the word (or char) level (tokenization)
- Char CNN
  - Calculate maximum length of any URL and set that as input size
  - Define an alphabet
  - Mapping every character in URL to integer value to feed into the Embedding input layer

- A model that produces no false positives has a precision of 1.0.
- A model that produces no false negatives has a recall of 1.0.

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

$$F1 = 2 \cdot \frac{precision \cdot recall}{precision + recall}$$

Results - ISCX2016 Dataset

Their Approach [4] (Engineered Features):

Random Forest

	precision	recall	f1-score
benign	0.94	0.96	0.95
malware	0.93	0.94	0.94
phishing	0.92	0.89	0.90
spam	0.98	0.97	0.97
weighted avg	0.94	0.94	0.94

My Approach (Learned Features):

Multinomial Bayes Classifier

	precision	recall	f1-score
benign	0.96	0.99	0.98
malware	0.95	0.92	0.94
phishing	0.93	0.89	0.91
spam	0.99	0.98	0.99
weighted avg	0.96	0.96	0.96

Char CNN

	precision	recall	f1-score
benign	1.00	1.00	1.00
malware	0.99	0.99	0.99
phishing	0.98	0.98	0.98
spam	1.00	1.00	1.00
weighted avg	0.99	0.99	0.99

Results - FWAF Dataset

Multinomial Bayes

	precision	recall	f1-score
benign	0.98	0.92	0.95
malicious	0.18	0.5	0.26
weighted avg	0.95	0.9	0.92

Char CNN

	precision	recall	f1-score
benign	1.00	0.00	0.01
malicious	0.04	1.00	0.07
weighted avg	0.97	0.04	0.01

Results - Phishing Dataset (Predictions and F-1 Score)

Multinomial Bayes:

Correct = 61.41%

Incorrect = 38.59%

	precision	recall	f1-score
phishing	1.00	0.48	0.65

Char CNN:

Correct = 78.16%

Incorrect = 21.84%

	precision	recall	f1-score
phishing	1.00	0.78	0.88

Conclusion

- Lexical analysis (TF-IDF) and CNNs are a solid option to classify URLs not found in the blacklist.
- In the long term, TF-IDF or other related methods that keep a list of terms will increase the dimensionality of the features too much when new samples are added; Char CNN is a more flexible and more accurate approach since it doesn't require retraining for new unseen words.
- There is a need for better datasets that contain URLs and not just specific features.
    - The ones I have found so far are already encoded
- If perfect classification accuracy is desired, using the URL alone is not enough. However, in a real world scenario, since these are usually up for a very brief moment, the URL remains the only accessible part and can be processed very fast [4,7].

References

[1]I. Mamun, “Creating a TF-IDF in Python,” Medium, Jan. 14, 2020. https://medium.com/@imamun/creating-a-tf-idf-in-python-e43f05e4d424 (accessed May 04, 2021).

[2]E. Liu, “TF-IDF, Term Frequency-Inverse Document Frequency,” 2015. https://ethen8181.github.io/machine-learning/clustering_old/tf_idf/tf_idf.html (accessed May 04, 2021).

[3]M. Tripathi, “How to process textual data using TF-IDF in Python,” freeCodeCamp.org, Jun. 06, 2018. https://www.freecodecamp.org/news/how-to-process-textual-data-using-tf-idf-in-python-cd2bbc0a94a3/ (accessed May 04, 2021).

[4]M. S. I. Mamun, M. A. Rathore, A. H. Lashkari, N. Stakhanova, and A. A. Ghorbani, “Detecting Malicious URLs Using Lexical Analysis,” in Network and System Security, Cham, 2016, pp. 467–482, doi: 10.1007/978-3-319-46298-1_30.

[5]M. Krog, mitchellkrogza/Phishing.Database. 2021.

[6]F. Ahmad, faizann24/Fwaf-Machine-Learning-driven-Web-Application-Firewall. 2021.

[7] R. Verma and K. Dyer, “On the Character of Phishing URLs: Accurate and Robust Statistical Learning Classifiers,” in Proceedings of the 5th ACM Conference on Data and Application Security and Privacy, New York, NY, USA, Mar. 2015, pp. 111–122, doi: 10.1145/2699026.2699115.

[8]R. Verma and A. Das, “What’s in a URL: Fast Feature Extraction and Malicious URL Detection,” in Proceedings of the 3rd ACM on International Workshop on Security And Privacy Analytics, New York, NY, USA, Mar. 2017, pp. 55–63, doi: 10.1145/3041008.3041016.

References

[9]Y. Bengio, A. Courville, and P. Vincent, “Representation Learning: A Review and New Perspectives,” arXiv:1206.5538 [cs], Apr. 2014, Accessed: Mar. 01, 2021. [Online]. Available: http://arxiv.org/abs/1206.5538.

[9]“Phishing Statistics (Updated 2021) | 50+ Important Phishing Stats,” Tessian, Feb. 10, 2021. https://www.tessian.com/blog/phishing-statistics-2020/ (accessed Mar. 01, 2021).

[10]D. Sahoo, C. Liu, and S. C. H. Hoi, “Malicious URL Detection using Machine Learning: A Survey,” arXiv:1701.07179 [cs], Aug. 2019, Accessed: Mar. 01, 2021. [Online]. Available: http://arxiv.org/abs/1701.07179.

[11]J. Zhang and Y. Wang, “A real-time automatic detection of phishing URLs,” in Proceedings of 2012 2nd International Conference on Computer Science and Network Technology, Dec. 2012, pp. 1212–1216, doi: 10.1109/ICCSNT.2012.6526142.

[12]H. Le, Q. Pham, D. Sahoo, and S. C. H. Hoi, “URLNet: Learning a URL Representation with Deep Learning for Malicious URL Detection,” arXiv:1802.03162 [cs], Mar. 2018, Accessed: Mar. 01, 2021. [Online]. Available: http://arxiv.org/abs/1802.03162.

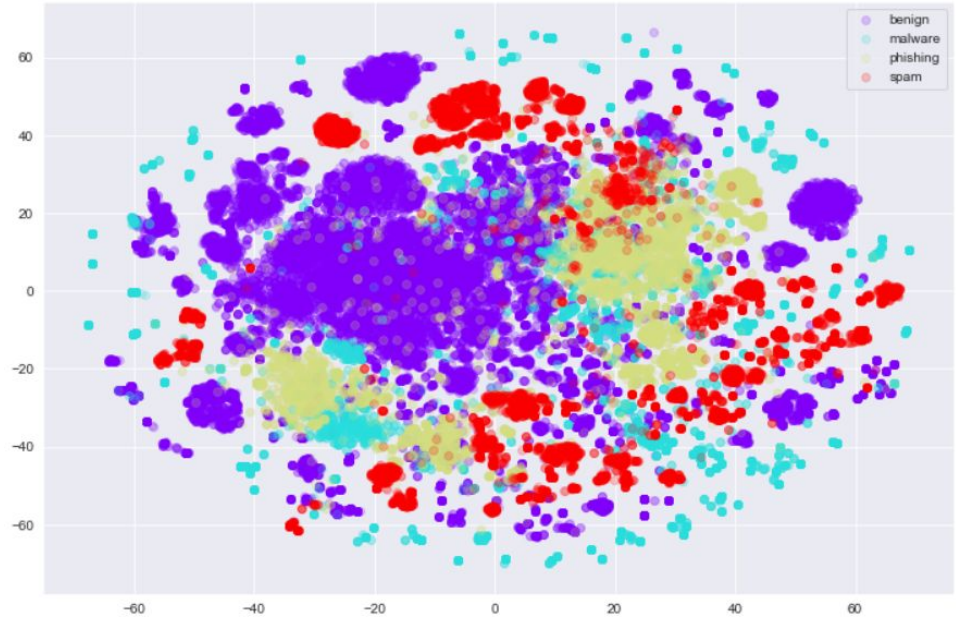
Backup Slides

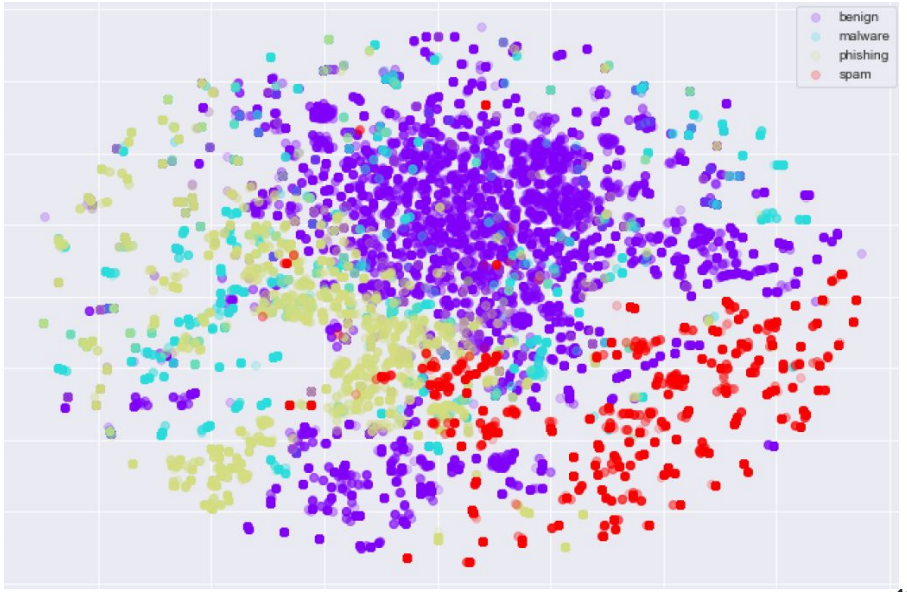
Precision vs Recall (further insight of metrics on results)

- It depends on what you think it's more important to your classifier
  - Precision: Cost of false positives is high
  - Recall: Cost of false negatives is high
- In this work we look at F1 score instead of accuracy because it is a better metric when the classes are imbalanced
- FWAF Results:
  - The data input structure (sequence) is important to the CNN, so it performs poorly in terms of metrics and predicts everything as malicious since this dataset is not URLs but queries (i.e. it is too different).
  - Although for this type of task, predicting everything as malicious is preferable since it means bad URLs don't get shown to the user.
  - TF-IDF is only looking at words out of context.

t-SNE

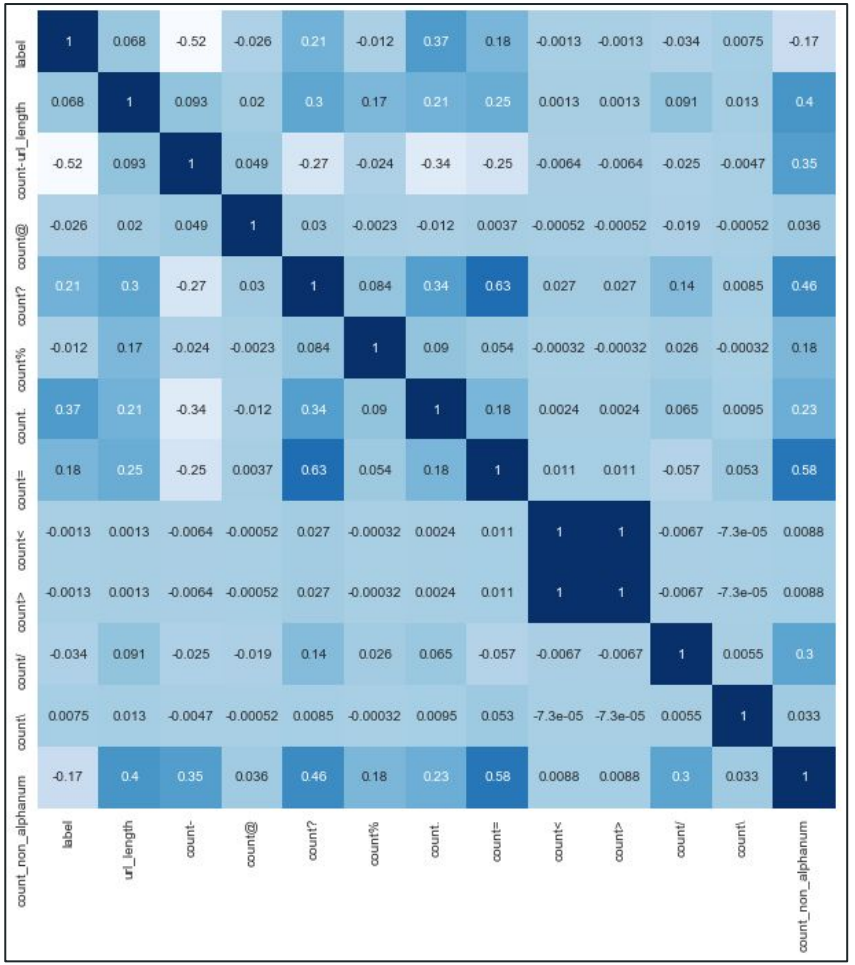
- Compared spread of data using char level (left) vs word level (right) TF-IDF tokenization
- Word level splitting of words (features) shows more defined, separate clusters.





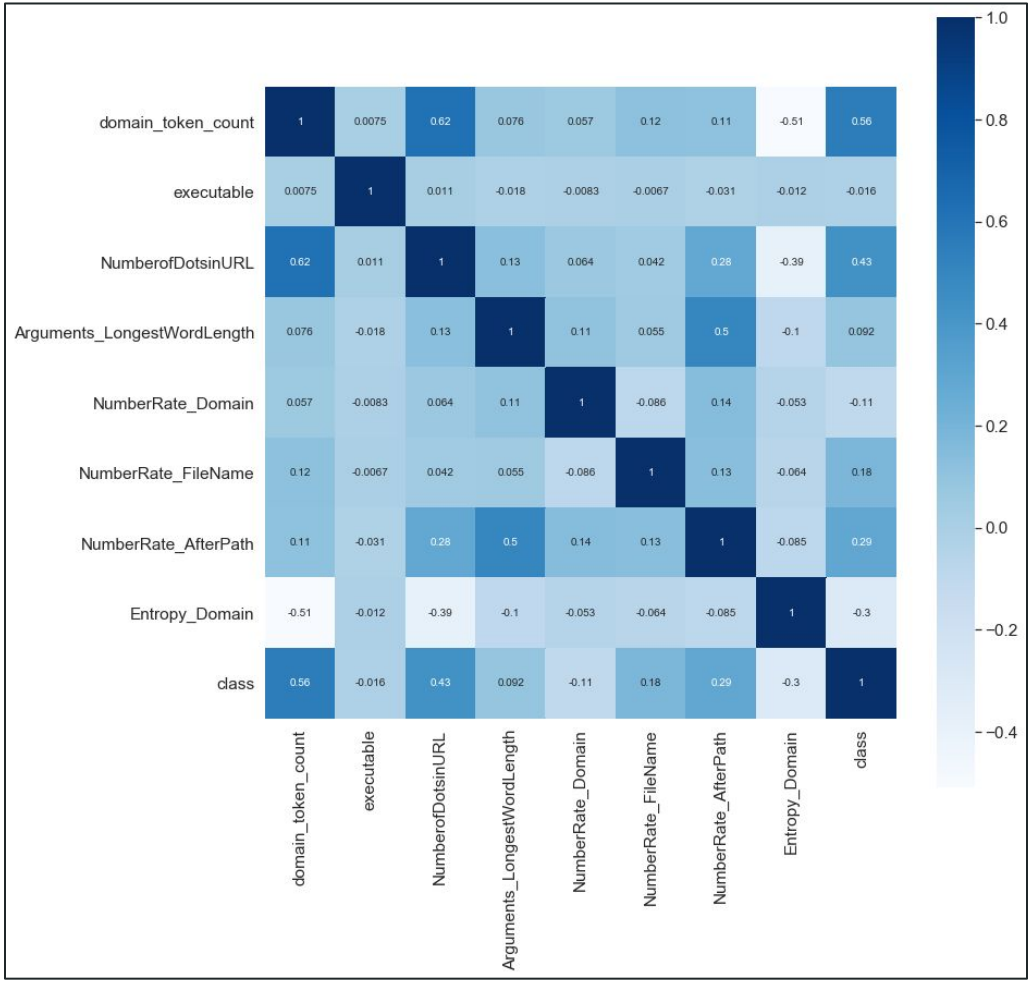
Exploration of ISCX2016 Dataset

- Image shows correlation matrix for the following features:
  - Calculated URL length
  - Counted number of special characters (-,@,?,%,.,=,<,>,/,\)
  - Count total number of non alphanumeric characters
- Analysis showed no strong correlation between label (category of URL) and the rest of the features.



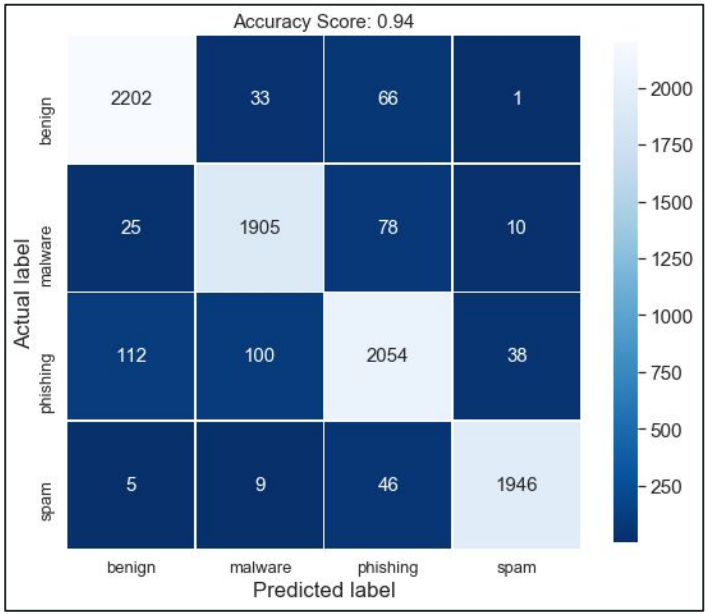
ISCX2016 Dataset

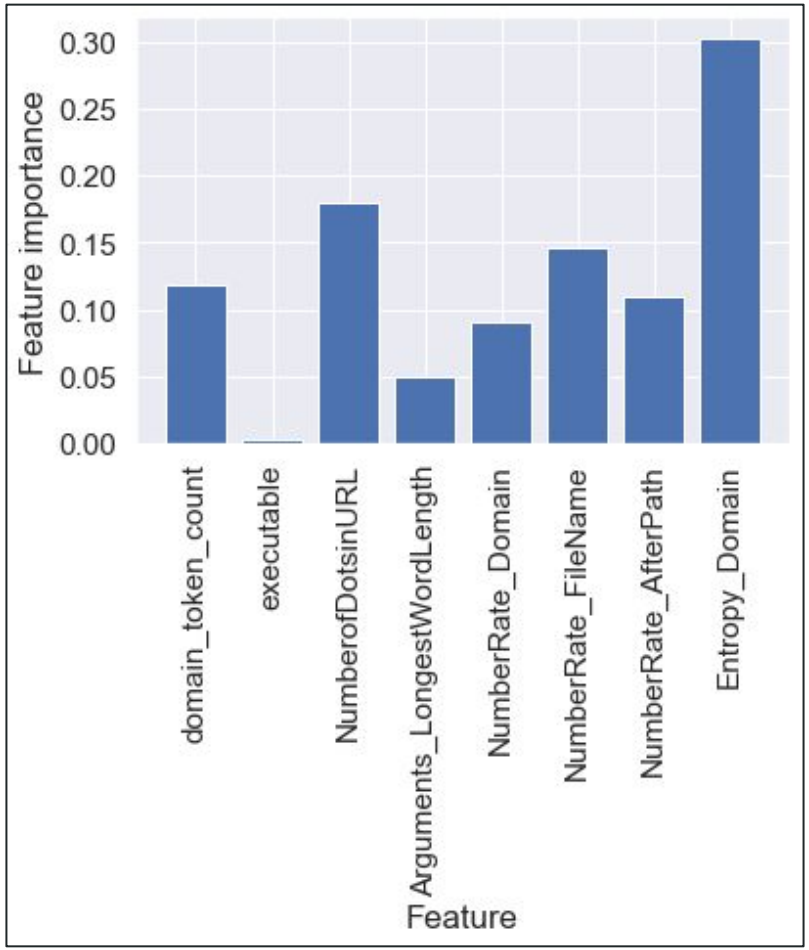
- The dataset came with manually engineered features, but not the corresponding raw URL they were calculated from, preventing reproduction of the original results using these features.
- Image shows correlation matrix



ISCX2016 Dataset - RF Engineered Features:

- The image shows feature importance after being fed into the Random Forest classifier, achieving 94% accuracy. More metrics can be seen in slide 11.

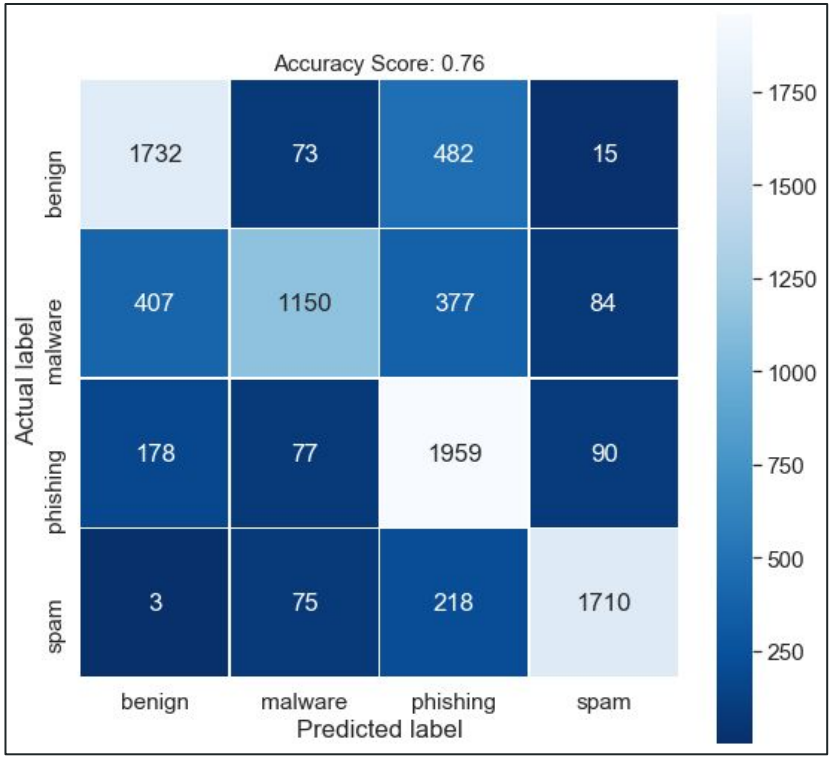




ISCX2016 Dataset - SVM - Engineered Features:

- Results for using an SVM classifier on the dataset, with the manually engineered features. Results show worse metrics compared to using a Random Forest classifier.

	precision	recall	f1-score
0	0.75	0.75	0.75
1	0.84	0.57	0.68
2	0.65	0.85	0.73
3	0.90	0.85	0.88
accuracy			0.76
macro avg	0.78	0.76	0.76
weighted avg	0.78	0.76	0.76



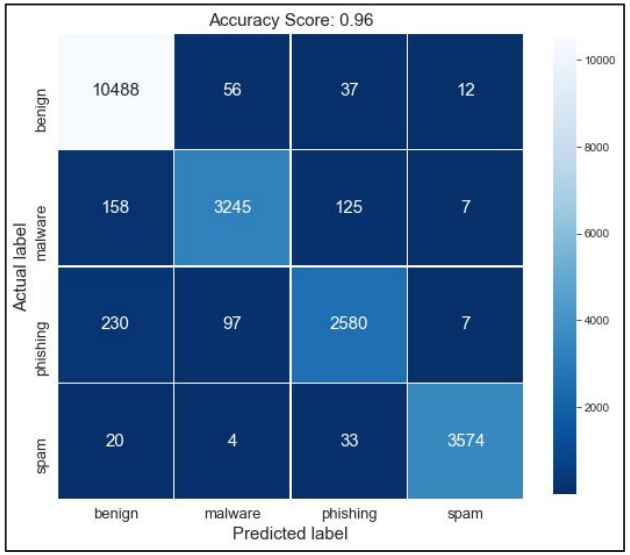
ISCX2016 Dataset - IDF

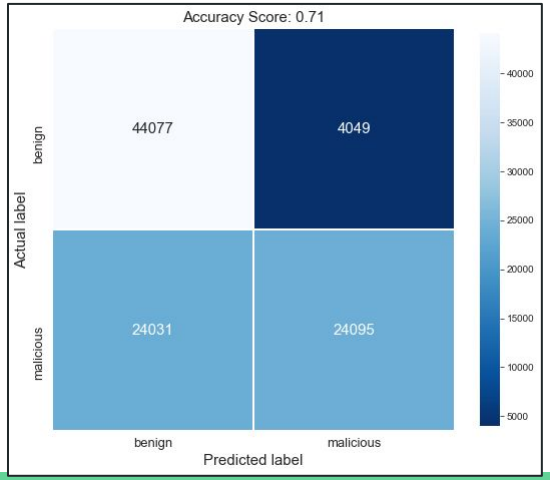
- Image shows inverse document frequency calculation for the top 10 and bottom 10 words (features) extracted.
- The significance here is that the bottom ranking ones align with what we would expect, since they are so common they don't provide much information and are therefore have a lower score. Unique words are ranked near the top.

	Term	IDF
0	E119	7.871859
1	Homewares	7.871859
2	OSI	7.871859
3	ShowItem	7.871859
4	banner	7.871859
5	beautiful	7.871859
6	bigtruckstopseek	7.871859
7	box	7.871859
8	case	7.871859
9	college	7.871859
1367	to	3.680085
1368	html	3.394749
1369	www	3.169743
1370	php	3.136187
1371	net	3.092399
1372	co	2.846663
1373	uk	2.623888
1374	com	1.746388
1375	NUMBERSPECIALTOKEN	1.695867
1376	http	1.029069

MNB - TFIDF results

- Top image: confusion matrix for Multinomial Bayes Classifier (see slide 11) using TF-IDF learned features on the ISCX2016 Dataset.
- Bottom image: confusion matrix on FWAF Dataset using the same model (Multinomial Bayes trained on ISCX Dataset) but as a binary classifier.
    - Results show low recall for malicious URLS (bottom table)





	precision	recall	f1-score
benign	0.65	0.92	0.76
malicious	0.86	0.50	0.63
accuracy			0.71
macro avg	0.75	0.71	0.70
weighted avg	0.75	0.71	0.70

MNB - TFIDF results - Phishing Dataset

- Results of Multinomial Bayes Classifier trained using TF-IDF learned features on the ISCX2016 Dataset.
- Prediction on the Phishing Dataset (see slide 13)
- We create the matrix with the predicted labels being benign, phishing (only type contained in this dataset) and malicious (predicted label by the model was malware or spam)
- Results suggest that this classifier and TF-IDF features are not generalizable to other different datasets and it predicts ‘benign’ too often.