Malicious URL detection using Ensemble Methods

Chouliaras Andreas and Pappas Apostolos
Department of Electrical and Computer Engineering, University of Thessaly, Greece

ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ
1988 – 2018 χρόνια δημιουργίας

Τμήμα Ηλεκτρολόγων Μηχανικών & Μηχανικών Υπολογιστών
Πανεπιστήμιο Θεσσαλίας

Abstract

With the continuous expansion of the Internet and its users, more and more security issues come to light. As much as security measures increase to protect systems that make up the Internet, the user remains the most vulnerable part of it. Malicious URLs leading to malicious websites, are a common and serious threat to cybersecurity. Malicious URLs host unsolicited content (spam, phishing, drive-by exploits, etc.) and lure unsuspecting users to become victims of scams (monetary loss, theft of private information, and malware installation), causing losses of billions of dollars every year.

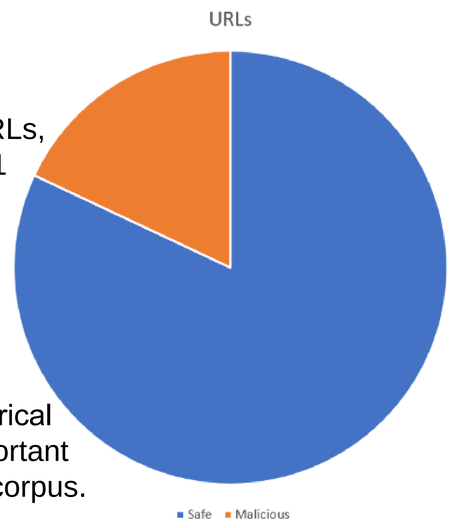
The most classical way of dealing with this threat is the usage of blacklists. Blacklists tend to lack the scalability and the overall ability to to detect newly generated malicious URLs. In this report, we examine the effectiveness of several Machine Learning algorithms on detecting such Malicious URLs taking training time into account as well. We also examine ensemble methods, in order to further improve our model's accuracy.

Introduction

- **Dataset and Preprocessing**

  Our dataset contains a total of 420,464 URLs, 75,643 of which are malicious and 344,821 that are safe. Each URL contains only the host name and the path, excluding the HTTP Protocol at the beginning.

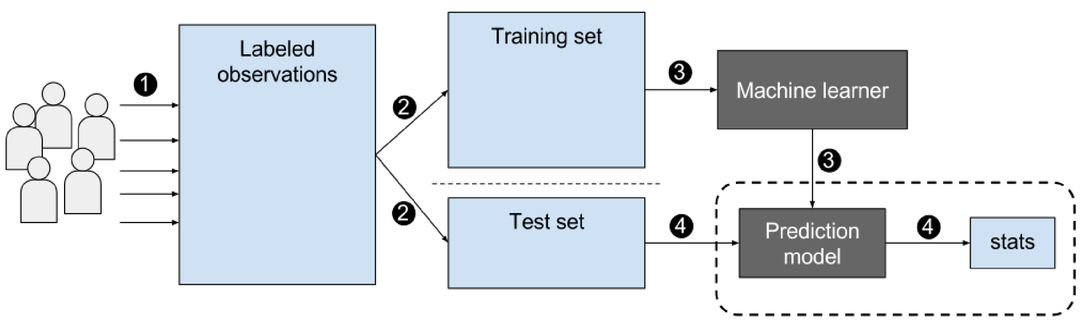
  The algorithm we used for preprocessing the data is tf-idf, short for "term frequency-Inverse document frequency". It is a numerical statistic that is intended to reflect how important a word is to a document in a collection or corpus.



- **Models Evaluated**

  We applied several ensemble and non-ensemble algorithms for our experiments. These were:
  - Logistic Regression
  - Naïve Bayes (Bernoulli)
  - Naïve Bayes (Multinomial)
  - K-Nearest-Neighbors
  - Multilayer Perceptron Classifier (MLP)
  - Random Forest

  For the models that had a very slow training procedure, we applied feature reduction using the TruncatedSVD algorirthm. We provide the results of both versions



Evaluation Methods

In this report, we evaluated our models using the following metrics:
- Accuracy
- Precision
- Recall
- F1 score and
- Area Under Curve (known as AUC)

In the Report the Receiver Operating Characteristic curve is provided for each model.

Experimental Results

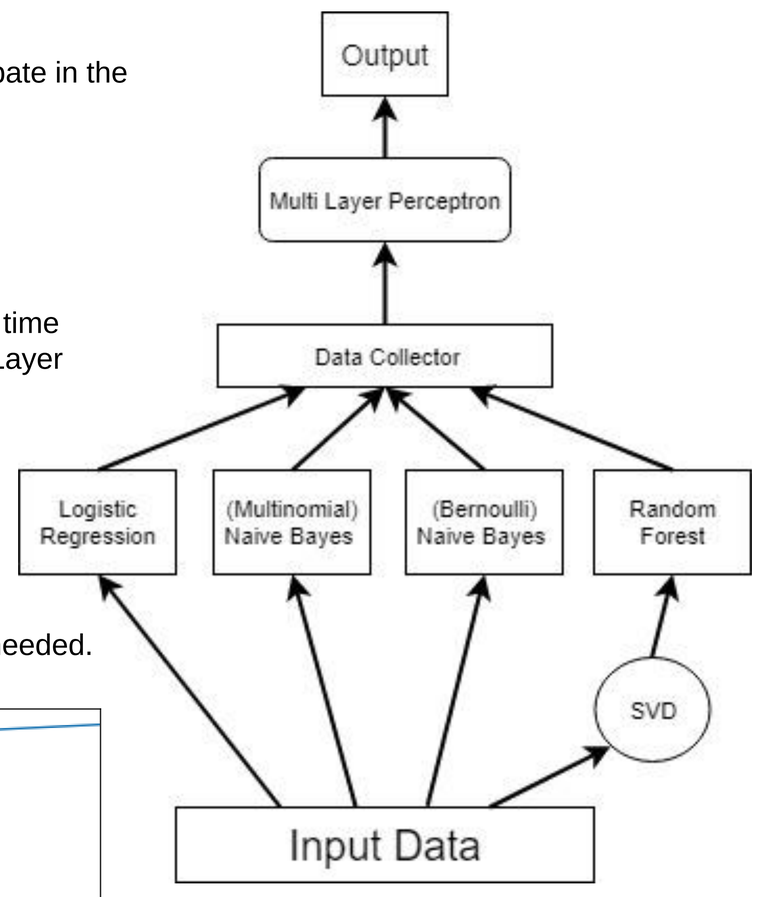
Algorithm	Precision	Recall	F1 score	AUC	Accuracy(%)
MLP (SVD)	0.895	0.674	0.769	0.828	92.7
NB Bernoulli	0.979	0.745	0.846	0.871	95.12
NB Multinomial	0.993	0.801	0.887	0.899	96.32
KNN (SVD)	0.892	0.825	0.857	0.901	95.58
Logistic Regression	0.973	0.816	0.888	0.905	96.12
KNN	0.993	0.801	0.886	0.909	96.12
MLP	0.970	0.826	0.892	0.910	98.0
Random Forest	0.979	0.856	0.914	0.926	97.1
Random Forest (SVD)	0.944	0.874	0.908	0.931	96.8
Custom Ensemble Model	0.943	0.922	0.933	0.955	97.61

Custom Ensemble Model

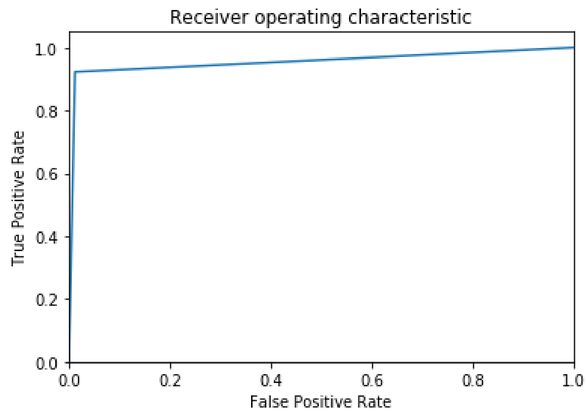
The idea here is to exploit the best of our models and use their predictions to further improve the results.

The four models we chose to participate in the ensemble model were:
- Logistic Regression
- Naïve Bayes (Multinomial)
- Naïve Bayes (Bernoulli)
- Random Forest after SVD

We based our choice on the training time performance. The input of the Multi Layer perceptron consists of the prediction vectors of each algorithm stacked together forming a matrix.



Even though Naïve Bayes models weren't very efficient to reduce FN, the model was effected into making opposite predictions to theirs when needed.



The results were better than initially expected. We achieved the best accuracy among the constituent models at 97.61% falling short behind the sklearn MLP model. It has lower precision than the constituent models at 94.3% but has by far the best recall at 92.2% and the best F1 score at 93.3&. Finally it has the best AUC at 0.955.

Conclusions

Malicious URLs are a serious threat to cybersecurity. The detection of such URLs plays a critical role for many cybersecurity applications, and clearly machine learning approaches are a promising direction. We tested a lot of machine learning algorithms, more than the included, like SVM which we excluded because it was so slow that we never actually managed to finish the training. We saw how interesting and effective ensemble methods can be and even a custom-made ensemble model that consists of fast and diverse models can produce impressive results.

Refer to the written report for more information.

References:
1. A. Defazio, F. Bach, and S. Lacoste-Julien. Saga: A fast incremental gradient method with support for non-strongly convex composite objectives. Advances in Neural Information Processing Systems, 2, 07 2014.
2. J. Hong. The state of phishing attacks. Commun. ACM, 55: 74–81, 01 2012. doi: 10.1145/2063176.2063197
3. M. E. Maron. Automatic indexing: an experimental inquiry. Journal of the ACM (JACM), 8(3):404–417, 1961.
4. D. Sahoo, C. Liu, and S. C. Hoi. Malicious url detection using machine learning: a survey. arXiv preprint arXiv:1701.07179, 2017.
5. S. Sinha, M. Bailey, and F. Jahanian. Shades of grey: On the effectiveness of reputation-based blacklists. In 2008 3rd International Conference on Malicious and Unwanted Software (MALWARE), pages 57–64, Oct 2008. doi: 10.1109/MALWARE.2008.4690858.
6. S. Yang and A. Browne. Neural network ensembles: Combining multiple models for enhanced performance using a multistage approach. Expert Systems, 21:279 – 288, 11 2004. doi: 10.1111/j.1468-0394.2004.00285.x.