Is This URL Safe: Detection of Malicious URLs Using Global Vector for Word Representation

Rohit Bharadwaj[1] Ashutosh Bhatia[1] Laxmi Chhibbar[1]
Kamlesh Tiwari[1] Ankit Agrawal[1]

[1]Department of Computer Science and Information System,
Birla Institute of Technology and Science Pilani, Pilani, Jhunjhunu, Rajasthan, 333031 India

International Conference on Information Networking (ICOIN 2022)

ICOIN 2022
The 36th International Conference on Information Networking (ICOIN 2022)
January 12-15, 2022, Jeju Island, Korea & Virtual Conference

Background

Users are frequently exposed to many unknown links through advertisements and emails.

[1] *Help Net Security*. https://www.helpnetsecurity.com/2019/10/09/phishing-increase-2019/. Accessed: 2021-09-28.

[2] *Kaspersky Security Bulletin 2020-2021*. https://securelist.com/category/kaspersky-security-bulletin/. Accessed: 2021-09-28.

Background

Users are frequently exposed to many unknown links through advertisements and emails.

- Malicious URLs pose a prime threat to the information security of organizations.

[1] *Help Net Security*. https://www.helpnetsecurity.com/2019/10/09/phishing-increase-2019/. Accessed: 2021-09-28.

[2] *Kaspersky Security Bulletin 2020-2021*. https://securelist.com/category/kaspersky-security-bulletin/. Accessed: 2021-09-28.

Background

Users are frequently exposed to many unknown links through advertisements and emails.

- Malicious URLs pose a prime threat to the information security of organizations.
- According to a report, nearly 1 in 50 URLs are infected with malicious code.[1]

[1] *Help Net Security*. https://www.helpnetsecurity.com/2019/10/09/phishing-increase-2019/. Accessed: 2021-09-28.

[2] *Kaspersky Security Bulletin 2020-2021*. https://securelist.com/category/kaspersky-security-bulletin/. Accessed: 2021-09-28.

Background

Users are frequently exposed to many unknown links through advertisements and emails.

- Malicious URLs pose a prime threat to the information security of organizations.
- According to a report, nearly 1 in 50 URLs are infected with malicious code.[1]
- Trusted domain URLs are being used to host malicious code.

[1] *Help Net Security*. https://www.helpnetsecurity.com/2019/10/09/phishing-increase-2019/. Accessed: 2021-09-28.

[2] *Kaspersky Security Bulletin 2020-2021*. https://securelist.com/category/kaspersky-security-bulletin/. Accessed: 2021-09-28.

Background

Users are frequently exposed to many unknown links through advertisements and emails.

- Malicious URLs pose a prime threat to the information security of organizations.
- According to a report, nearly 1 in 50 URLs are infected with malicious code.[1]
- Trusted domain URLs are being used to host malicious code.
- In 2019, 85% of detected web threats were malicious URLs.[2]

[1] *Help Net Security*. https://www.helpnetsecurity.com/2019/10/09/phishing-increase-2019/. Accessed: 2021-09-28.

[2] *Kaspersky Security Bulletin 2020-2021*. https://securelist.com/category/kaspersky-security-bulletin/. Accessed: 2021-09-28.

Current Approach

- IDS (Intrusion Detection Systems):

Current Approach

- IDS (Intrusion Detection Systems): uses URL blacklisting or signature blacklisting.

Current Approach

- IDS (Intrusion Detection Systems): uses URL blacklisting or signature blacklisting.
- These blacklist URLs maybe secured behind a paywall.

Current Approach

- IDS (Intrusion Detection Systems): uses URL blacklisting or signature blacklisting.
- These blacklist URLs maybe secured behind a paywall.
- Keeping updated list of blacklist URLs is challenging.

Current Approach

- IDS (Intrusion Detection Systems): uses URL blacklisting or signature blacklisting.
- These blacklist URLs maybe secured behind a paywall.
- Keeping updated list of blacklist URLs is challenging.
- Attackers have started using DGAs which renders blacklist method ineffective.

Machine Learning Based Approaches

- Train a learning model based on URLs' static and dynamic features.

Machine Learning Based Approaches

- Train a learning model based on URLs' static and dynamic features.

Dynamic Features

- Activities performed in the target system due to the URL.
- Features associated with the content of the webpage referred to by the URL.

Machine Learning Based Approaches

- Train a learning model based on URLs' static and dynamic features.

Dynamic Features

- Activities performed in the target system due to the URL.
- Features associated with the content of the webpage referred to by the URL.

Static Features

- Directly derived from the URL.
- Ex: URL length, presence of unsafe extension.

Related Work

- Static classifiers have been modeled using lexical features of URL. (i.e. host properties, network traffic, etc.)
- The bag-of-words model is often used with these approaches, but not helpful in our case due to significant latency.
- The bag-of-words model leads to the loss of contextual information from the data.

Related Work

- (James et al., 2013)[3] detected phishing websites using lexical features. However, their approach depends on specially crafted features that are not suitable for large datasets.
- (Verma & Dyer, 2015)[4] built machine learning classifiers using URL's lexical features. However, it is not robust to spelling errors in the malicious URLs.
- Our work builds upon the work by (Hai & Hwang, 2018)[5], but we use different feature extraction methods, different dataset sizes and introduce GloVe based embedding learning instead of Word2Vec embedding used by previous authors.

---

[3] Joby James, L Sandhya, and Ciza Thomas. "Detection of phishing URLs using machine learning techniques". In: *2013 international conference on control communication and computing (ICCC)*. IEEE. 2013, pp. 304–309.

[4] Rakesh Verma and Keith 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*. 2015, pp. 111–122.

[5] Quan Tran Hai and Seong Oun Hwang. "Detection of malicious URLs based on word vector representation and ngram". In: *Journal of Intelligent & Fuzzy Systems* 35.6 (2018), pp. 5889–5900.

Word2Vec

Word2Vec focuses on learning word embedding. For this, a two-layer neural network is trained to remodel the word's semantic contexts.

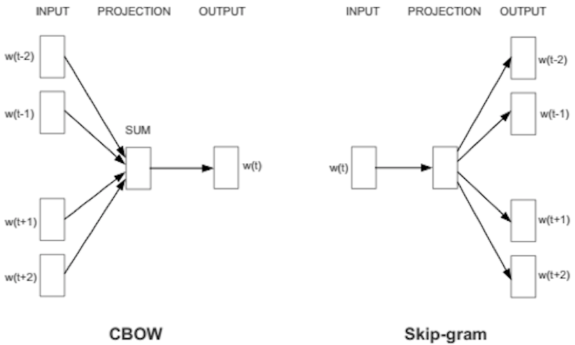


Figure: The two variants of Word2Vec model. CBOW predicts center word given the context words, while skip-gram predicts the context words given the center word.

Word2Vec Analogy Tasks

These word embeddings are capable of identifying similarities between words. They use vector arithmetic to reconstruct the syntactic and linguistic patterns.

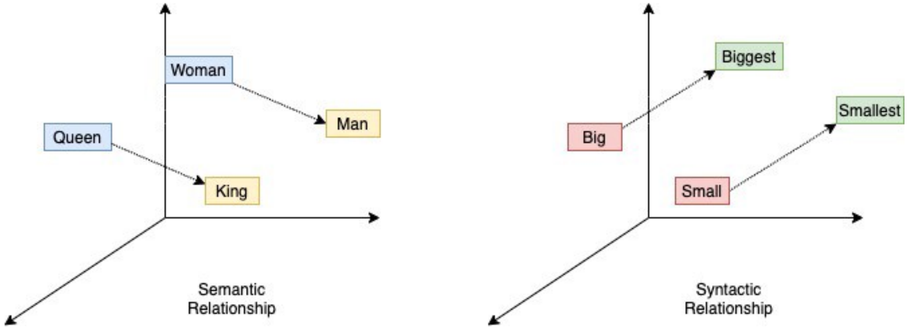


Figure: The figure shows the different relationship captured by Word2Vec models. Embedding("King") - Embedding("Man") + Embedding("Woman") yields a vector which is closest in similarity to the vector Embedding("Queen").

GloVe

Word2Vec model considers only local perspectives to learn the embedding. Whereas, GloVe model builds upon the work of Word2Vec by also considering the global word-word co-occurrence statistics.

GloVe based models have also been shown to give better results in many natural language processing tasks compared to Word2Vec.[6,7]

With this work we show that the ability to incorporate global word co-occurrence statistics through GloVe model helps to better discriminate between malicious and benign URLs.

[6] Jeffrey Pennington, Richard Socher, and Christopher D Manning. "Glove: Global vectors for word representation". In: *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*. 2014, pp. 1532–1543.

[7] Christopher Ifeanyi Eke et al. "The significance of global vectors representation in sarcasm analysis". In: *2020 International Conference in Mathematics, Computer Engineering and Computer Science (ICMCECS)*. IEEE. 2020, pp. 1–7.

Proposed Approach

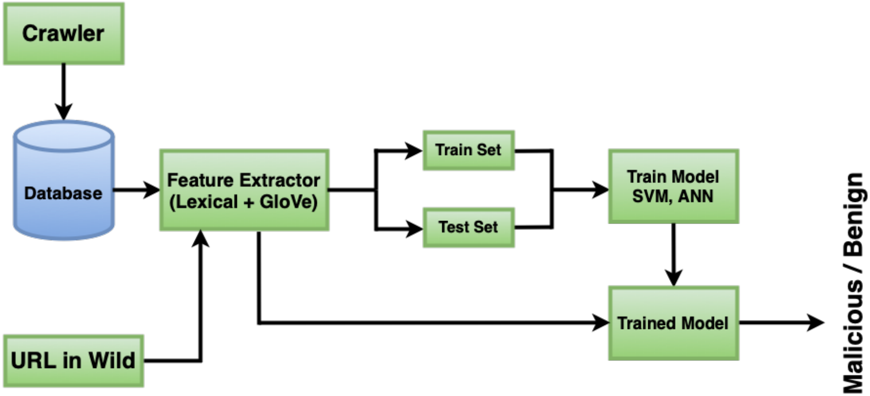


Figure: Proposed Malicious URL Detector.

Getting the Data

The data for benign URLs were collected from DMOZ. The website contains several categories and subcategories where there are many links.

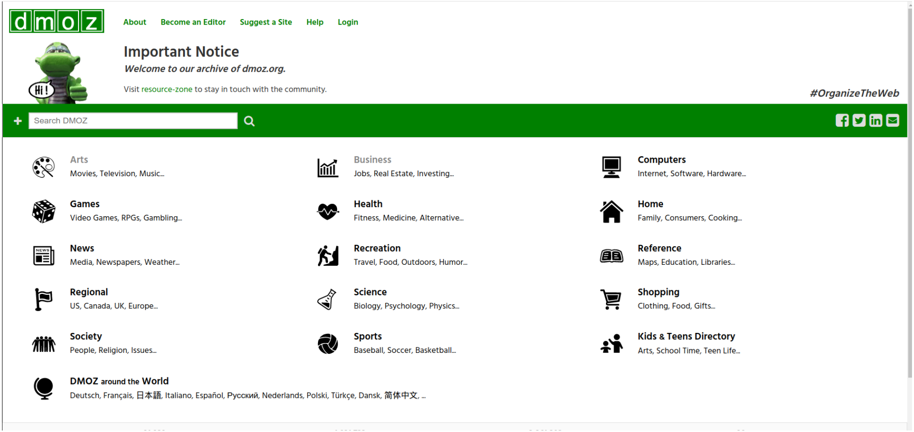


Figure: DMOZ website with several top-level categories.

Getting the Data

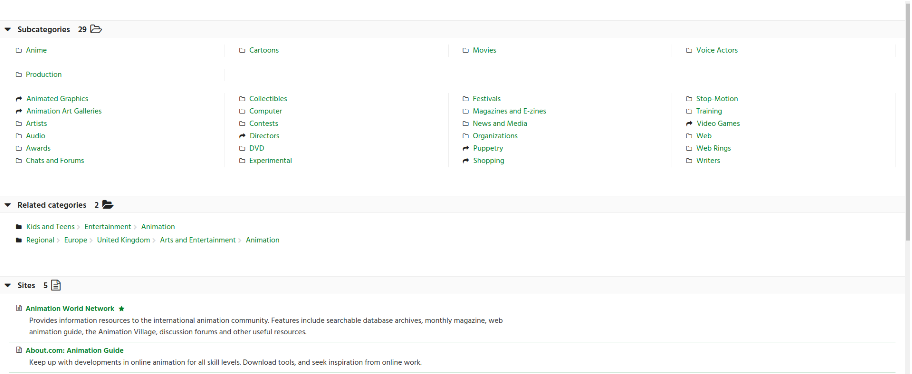


Figure: Subcategories and website links found in DMOZ.

Getting the Data

```python
try:
        url = "https://dmoz-odp.org"
        myreq=req.Request(url,headers=headers)
        resp=req.urlopen(myreq)
        respData=str(resp.read())
        patt=r'<h2 class="top-cat"><a href="[^"]+'              ###Regular Expression for match
        links=re.findall(patt,respData) #Parsing Data Using Regex
        links = links[1:-1]
        print('The total of %i links are as follows : '%len(links))



        for i in links:
                i = i.replace(r'<h2 class="top-cat"><a href="','')
                i = url + i
                getSubCat(i,True)
        print('\n')
```

Figure: Python code to extract categories from DMOZ in a DFS based manner.

In this work, **80,128** benign URLs were collected using the above-stated algorithm. Further, **147,781** malicious URLs were obtained and used in this work.[8,9]

[8] AK Singh. "Malicious and Benign Webpages Dataset". In: *Data in brief* 32 (2020), p. 106304.

[9] URL: https://www.kaggle.com/antonyj453/urldataset.

Getting the Data

```
headers={}
headers['user-agent']="Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/50.0.2661.102 Safari/537.36"

visited = {}
visited["a"] = -1

def getSubCat(link, isNotFirstWorld):
        #saveURL(link)

        myreq=req.Request(link,headers=headers)
        resp=req.urlopen(myreq)
        respData=str(resp.read())
        #  <div class="cat-item">
        #<a href=
        #print(respData[0:10])
        patt=r'<div class="cat-item">[\\r\\n]+([^>]+)'          ###Regular Expression for matching and searching Relevant Link on the Page
        links=re.findall(patt,respData) #Parsing Data Using Regex
        for i in range(len(links)):
                links[i] = str(links[i]).strip()[:-1]

        #Need to remove /World entries
        links2 = []
        for e in links:
                if ( ( (not e.startswith('<a href="/World') ) and (isNotFirstWorld) ) or (not isNotFirstWorld) ):
                        links2.append(e)


        #links2 = links

        #links = links2
        link = "https://dmoz-odp.org"
        # print(links2)
        for i in links2:
                i = i.replace(r'<a href="','')
                i = link + i
                print("Saving All Links in: ", i)
                try:
                        if visited.get(i) == None:
                                saveURL(i)
                                visited[i] = 1
                                getSubCat(i, True)
```

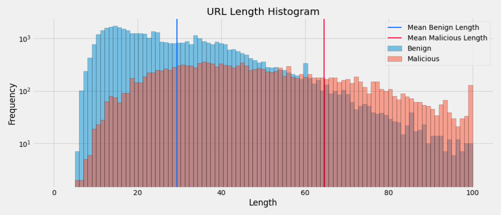
Figure: Code to extract URL from categories.

Extracting Features

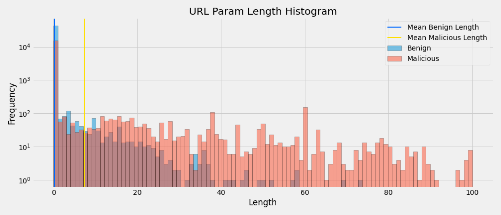
Feature Set	
Statistical Features	**Type**
URL Length	Continuous
Path Length	Continuous
Parameter Length	Continuous
is_exe	Categorical
is_dll	Categorical
letter-count	Continuous
digit-count	Continuous
count@	Continuous
count#	Continuous
count%	Continuous
use_of_ip	Categorical
vowByCons	Continuous
DigByLetter	Continuous
blackNGScore	Continuous
New Feature	**Type**
GloVe	Continuous

Table: URL Lexical Features considered for training the ML Model.

Histogram



(a) Histogram of URL length.



(b) Histogram of URL parameter length.

Figure: Histogram of URL length and URL parameter length.

Proposed GloVe Feature

Following from (Hai & Hwang, 2018)[10], we train the GloVe model on only benign URL tokens. By doing so, the malicious URL tokens would be far from benign tokens in the vector space.

The GloVe score is calculated using Algorithm 1.

Algorithm 1: Computing GloVe scores.

Input: tokenList // Obtained from URL
Output: s // GloVe score for the URL

foreach *token* $w \in$ *tokenList* **do**
    if $w \in$ *embeddingsIndex* **then**
        $myVec \leftarrow myVec + embeddingsIndex[w]$
    end
    else
        $myVec \leftarrow myVec + \mathbf{0}$
    end
end
$myVec \leftarrow \frac{myVec}{len(tokenList)}$
return $s \leftarrow \|myVec\|$

[10] Quan Tran Hai and Seong Oun Hwang. "Detection of malicious URLs based on word vector representation and ngram". In: *Journal of Intelligent & Fuzzy Systems* 35.6 (2018), pp. 5889–5900.

Classifiers

SVM with linear kernel and a simple ANN with architecture shown in the below figure were used as classifiers.

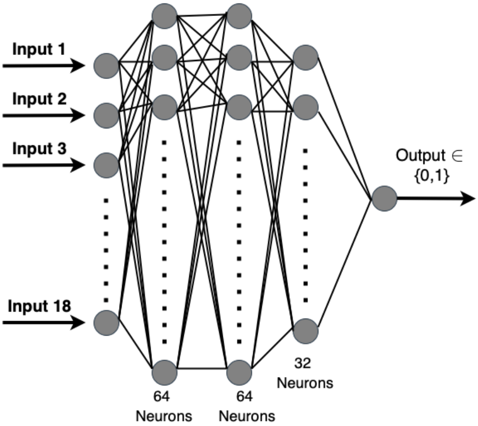


Figure: Neural Network Architecture

Results

Classifier	Feature Type	Accuracy (%)	Precision (%)	F1
Support Vector Machine (SVM)	Only Statistical Features	69.72	0.69	0.80
Artificial Neural Network (ANN)	Only Statistical Features	86.00	0.86	0.90
Support Vector Machine (SVM)	GloVe with Statistical Feature	**77.26**	**0.80**	**0.83**
Artificial Neural Network (ANN)	GloVe with Statistical Feature	**89.00**	**0.88**	**0.92**

Table: Performance of the proposed system as compared with other setting.

ROC Curve

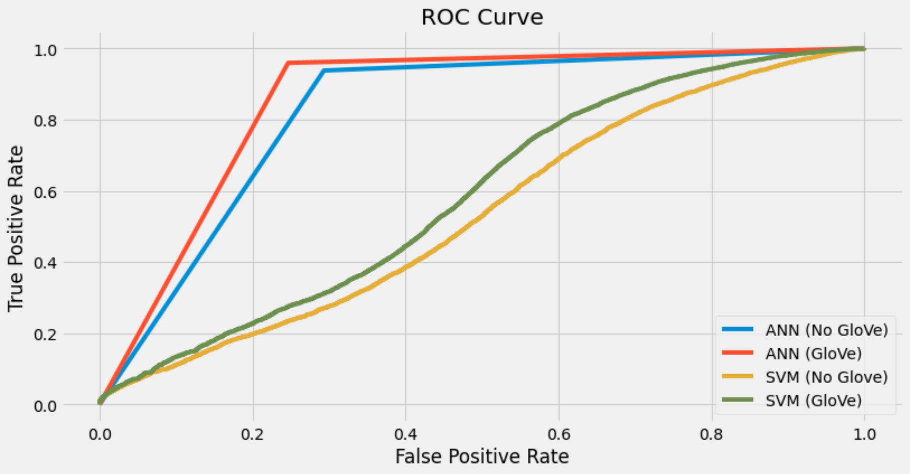


Figure: Receiver Operating Curve (ROC) while using the proposed feature on SVM and ANN classifier. Higher is the area under the curve, the better is the model.

Conclusion

We proposed using GloVe as one of the features to improve the accuracy of machine learning models for detecting malicious URLs. We observe that the proposed system has improved the overall performance of malicious URL detection, as it has reduced the error by **63.33%** when compared to the traditional approach.

The performance gain is observed in both ANN and SVM, by including the GloVe based feature. This validates our hypothesis and demonstrates the effectiveness of using GloVe based features to classify malicious URLs.