Chrome Extension For Malicious URLs detection in Social Media Applications Using Artificial Neural Networks And Long Short Term Memory Networks

Shivangi S [1]
Computer Science and Engineering
PES Institute of Technology
Bangalore, India
shivangi2197@gmail.com

Pratyush Debnath [2]
Computer Science and Engineering
PES Institute of Technology
Bangalore, India
pratyush.debnath93@gmail.com

Sajeevan K [3]
Computer Science and Engineering
PES Institute of Technology
Bangalore, India
sajeevan@pes.edu

D. Annapurna [4]
Information Science and Engineering
PES Institute of Technology
Bangalore, India
annapurnad@pes.edu

***Abstract*—Social media applications have become an integral part of our life, business and society today. Due to their increasing number of users and growing popularity, many organisations use them as a medium to generate income. The businesses use social media analytics and advertisements to increase their revenue. Although social media applications were initially built to connect people across the globe, they have now turned into one of the most favoured ways of propagation of cyber crimes. Most users lack cyber awareness and fall prey to the malicious activities distributed via Uniform Resource Locators (URLs) and advertisements. When a user visits the malicious URL, it makes the hackers privy to a lot of personal and sensitive information of the user. To overcome the problem of malicious URLs victimising users we propose a tool deployed as a chrome extension. This tool, analyses URLs and classifies them using two different neural networks, Artificial Neural Networks (ANN) and Long Short-Term Memory (LSTM) networks which is a specific type of Recurrent Neural Network (RNN). The major objective of the proposed model is to aid the users to avoid becoming a victim of malicious and fraudulent activities like malicious URLs, phishing and social engineering that favour social media as their target medium by detecting them accurately. The model proposed is scalable, easy to train and compatible with devices of varied hardware specifications. The proposed model gives excellent accuracy and overcomes several issues faced by the existing systems.**

I. Introduction

The advent of the internet has transformed the world into a global village, thus revolutionising the communication technologies that have impacted the lives of people as well as the growth of businesses. The exponential rise in the number of Internet users is attributed to the availability of various user-friendly web and mobile applications. Online presence plays a vital role in the success of any business venture today. Organisations invest a significant measure of time and assets in the advancement of applications like banking, social media, e-commerce and others. Social media applications are defined as web-based and mobile-based applications that permit the creation, access, and trade of user-generated content (UGC) that is universally available [1], [2]. Data generated by social media is by far the most extensive, dynamic and valuable information base of human behaviour presenting new opportunities to comprehend the individuals, groups, and society.

Researchers are designing innovative methods for consequently collecting, combining and analysing the data generated by social media applications. Some of the most popular social media applications like Facebook, Twitter, and Instagram are extensively used and constitute the largest source of user-generated content [1]. Social media applications have changed the face of social interaction and has become an integral part of the society. People use these applications to share information and multimedia [2]. In order, to ease the process of decision making and facilitate an increase in the revenue, companies analyse the content generated by these applications.

Social media is used for various purposes like to sell and promote products, events, people or organisations. People spend a substantial amount of time on different social media platforms resulting in the production of a large amount of user-generated content. Strict and detailed monitoring of such massive amounts of data for anomalies would require a lot of computation power and may violate the privacy of the users. Unfortunately, due to an enormous user base and privacy laws, most hackers use social media as the preferred medium of delivering malicious and fraudulent content without the fear of being traced or blacklisted easily [3]. The malicious content could be a macro-enabled document, executable, virus, javascript code, malware, adware, shell scripts, bat files or set of commands [4]. Fraudulent content could be phishing sites, financial scams that trick you into parting with your money by

978-1-5386-5314-2/18/$31.00 ©2018 IEEE

convincing you to buy a product or by making you take part in lucky draws, events, games or polls. Many people become the target of such malicious and fraudulent activities as there is a lack of awareness [5].

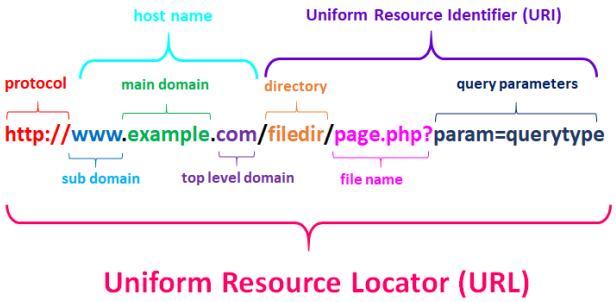

Fig. 1. Uniform Resource Locator Structure

The content on the World Wide Web is accessed, in general, via Uniform Resource Locators or URLs. URLs are unique strings used to access resources on the internet. Fig. 1, clearly indicates the general structure of the URL and its major components. Hackers compromise URLs to utilise it for malicious activities.

The existing approaches to combat these issues are blacklists, heuristic and machine learning techniques [6]. Blacklisting and heuristic approaches are not suitable for this highly dynamic problem [7], [8]. The most popular approaches, create a Bag of Words from the URL string by extracting the lexical features and apply machine learning algorithms like Support Vector Machines, Logistic regression, Naive Bayes and Decision trees [6], [9]. The existing machine learning approaches require a lot of feature engineering and it is easy to bypass the security setup if these features are known. URLNet was implemented using a deep learning technique called Convolutional Neural Network to overcome the same problem recently [9]. They had a high accuracy rate, but their model is computationally very expensive, requires a lot of time for feature extraction and training. To provide a scalable, efficient, portable and easy to use solution for our problem we propose a tool that will overcome the shortcomings of the existing approaches. The proposed model is capable of learning features via two different deep learning algorithms, Artificial Neural Network (ANN) and Long Short Term Memory (LSTM) network.

II. Case Study

A. Shortened URLs

It has been observed that shortened URLs are often shared on social media applications. For example, let us consider the Twitter application. In order, to fit into twitter's 140 character limit and to bypass the security mechanisms installed by the application hackers use shortened URLs [5]. These URLs are capable of hiding the actual URL and convince users to visit them.

B. Misleading Advertisements

Advertisements (ads) are showed and shared on social media applications with the intent of tempting the user to click on it. These advertisements may contain content that will capture the user's attention and convince them to visit the URL. The nature of the content may range from attractive or free deals, lucky draws, games to money earning schemes and so on. Generally, all applications, use an ad personalisation service which helps to ensure the ads presented to the user are relevant. These advertisements are also sources of malicious content as some applications use new or small advertisement servers in order to get more revenue. These ad servers are easily compromised by hackers. Hence, hackers get the user to visit the malicious URLs by targeting the user's interests.

C. Usage Of Accented Characters and Similar URLs

Attackers use creative techniques to evade blacklists and fool users by modifying the URL to make them appear authentic via obfuscation [6]. Garera et. al. identified four types of obfuscation [7].The identified types of obfuscation were obfuscating the host with an IP, obfuscating the host with another domain, obfuscating the host with large hostnames, and wrong spelling [7]. All of these try to hide the malicious intentions of the website, by masking the malicious URL [8]. The increasing popularity of URL shortening services, it has become a new and most common form of obfuscation used to hide the malicious URL behind a short URL [5]. Another technique for spreading malicious content is social engineering. Social engineering is used to manipulate users into divulging confidential or personal information that may be used for fraudulent purposes. In this technique, the hackers use a domain name that is very similar to the original domain by changing a few letters to the accented characters. For instance, instead of a, e, i, o and u they may use á, ẽ, ì, ò and ú or use hyphens and other symbols.

D. Frequent Redirection and Obnoxious Pop-ups

When a user clicks on a malicious URL generally, the user may experience many redirections. They may also have to deal with obnoxious and stubborn pop-ups. If the user clicks on these pop-ups either the user is again redirected to some other page or some malicious content gets downloaded on to the user's device. These pop-ups could be in the form of advertisements or banners on the web pages. Also, attackers make use of conditional redirections [5]. In this case, the attacker creates a long URL redirect chain by using public URL shortening services, such as bit.ly and t.co, and their own private redirection servers to redirect visitors [5]. The attacker then uploads a tweet including the initial URL of the redirect chain to Twitter. The redirection servers check whether the current visitor is a normal browser or a crawler. When a user or a crawler visits the initial URL, he or she will be redirected to an entry point of intermediate URLs that are associated with private redirection servers. If the link is visited by a crawler, it is redirected to a benign page. But if the link is visited by a normal browser,it is redirected to a malicious

page. Hence, the attacker can selectively attack normal users while deceiving investigators [5].

III. PROPOSED MODEL SPECIFICATIONS

A. PROBLEM DEFINITION

We define the malicious URL detection as a binary classification problem having two classes malicious and non-malicious. Given a data set with N URLs, where a training example d is represented as $\{u_i , y_i\}$, where u_i represents the i^{th} URL of training data, and y_i denotes the class of the URL where 0 is for malicious URLs and 1 is for non-malicious [6], [9].

B. PROCESSING FRAMEWORK

The major processes involved in the development of the proposed model are data gathering, feature engineering and model selection, training, testing and validation, deploying, taking feedback and updating of the models as shown in Fig. 2 [6].

1) Data Gathering: We have collected several URLs from various sources by web scraping. The sources for the URLs are search engines, existing data repositories like PhishTank [10] and commoncrawl [11] as well as twitter stream API for the model.

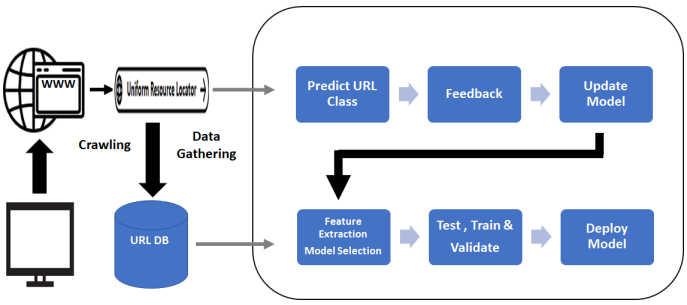

Fig. 2. High-Level Architecture Diagram

2) Feature Engineering and Model Selection: We clean the data set and process the URLs to obtain the features. We chose the models we wanted to implement. The models that were realised are Dense Artificial Neural Network and Long Short-Term Memory (LSTM) Network, which is a special kind of Recurrent Neural Network (RNN).

3) Training, Testing and Validation: We train the models over a random sample of varying sizes of the data set of over 2 million URLs. Deep neural nets with a large number of parameters are very powerful machine learning systems. Small neural networks are easier to train, whereas larger neural networks are more computationally expensive and difficult to train. Overfitting is a major issue in these models. Overfitting can be addressed by a technique called dropout [12]. The performance of the proposed models and the results are validated.

4) Deployment, Feedback and Updates: The model is deployed as a chrome extension to provide the user with a safer browsing experience by analyzing the URL browsed by the user. Then, predicting if the URL is malicious or not and displaying the confidence score. After, this it is left to the user's discretion whether they want to visit the URL or not. If they visit the site and find that it belonged to the other class they can provide us with the feedback which will be used to update the model.

C. WORKING OF THE TOOL

The working of the tool can be broadly classified into two major phases as shown in Fig. 3. The first phase deals with the development of the model and the second deals with the deployment and use of the tool.

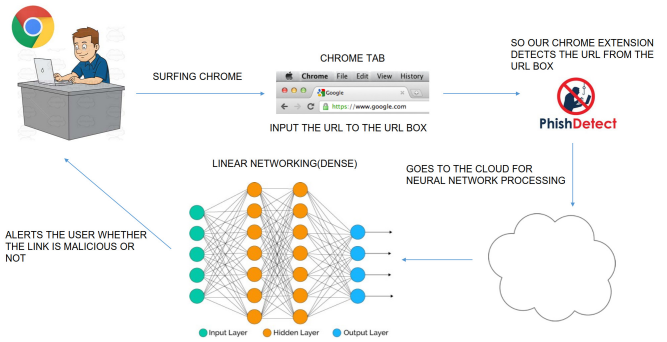

Fig. 3. Overview Of the Working Of the Tool

Phase 1: Tool Development

In this phase, we converted the data set into a form that could be fed into the model. We extracted the lexical features from the URLs [6], [9] and created a dictionary to account for the unique words used in URLs along with their count. We also created a categorical variable for two of the classes for training. We implemented the two models ANN and LSTM.

Dense Artificial Neural Network: The model implemented is a sequential multi-layer densely connected Artificial neural network. The model has 6 dense layers. The input to this model is a matrix of features. The first four dense layers are hidden layers that process the input using RELU as the activation function and produces the output. The output has reduced dimensionality and is fed to the next layer. The fifth dense layer of the model is the hidden layer that applies the activation function as sigmoid to process the input and produces the output within the range of 0 and 1. Then the output of the sigmoid layer is fed to the last dense layer which uses softmax as the activation function and produces the final output of by assigning a class to the train example. To overcome the problem of overfitting we make use of dropout [12].

Dense Bidirectional Long Short Term Memory Networks: The model implemented is a multi-layer densely connected Bi-Directional LSTM(Bi-LSTM) [13]. The input fed to the model, is the variable length URL, it can be denoted as U = $\{c_1, c_2, ..., c_n\}$.The words are represented as a vector that is extracted from a word embedding matrix [14].The obtained output sequence is $\{e(c_1), e(c_2), ..., e(c_n)\}$ which is the input to the first Bi-LSTM [15] layer and the output is $o^1 = \{o_1^1, o_2^1, ..., o_n^1\}$. The output $o_t = [\overrightarrow{o_t} : o_t]$, where $\overrightarrow{o_t}$ = lstm$(\overrightarrow{o_t}, e(c_t))$ and o_t = lstm$(o_t, e(c_t))$ represents the forward and backward hidden sates. The second layer Bi-LSTM layer, the input is the concatenation of previous layers outputs as $\{[e(c_1) : o_1^1], [e(c_2) : o_2^1], ..., [e(c_n) : o_n^1]]\}$ and the output is $o^2 = \{o_1^2, o_2^2, ..., o_n^2\}$. Similarly, the other third and fourth layer process the input and output. We compute the average pooling of the of the 4 layer Bi-LSTM as $o*$ = average$(o_1^4, o_2^4, ..., o_n^4)$. After pooling, we used a softmax classifier, which predicts the class using $o*$ as input.

The optimal weights computed by the models after several epochs are saved in a file. This file was used to the build of the chrome extension.

Phase 2: Tool Deployment

In this phase, after the construction of the chrome extension. We upload it to chrome app store. The user can download and install this extension. When the user enables this extension and visits a URL it will try to predict if the URL is malicious or not with a confidence score. Then, it is left to the user's discretion whether they want to visit that URL or not. Also, if the URL visited by the URL is not as predicted by the model then the user can provide feedback. The feedback obtained will be used to retrain and update the model.

IV. Experimental Results And Analysis

The proposed models differ from existing models in several ways. First, although we focus here on social media applications the model is generic and can be extended to various domains. Also, we want this model to be highly scalable and easy to train so that it could be trained easily on devices with varied hardware specifications and yet provide a good accuracy to aid the users in avoiding malicious content. The tableI compares the performance of both models on different data set sizes.

TABLE I
Comparison Of Models Over the Subset Of Data set

Data Set Size		512	2343	11234	45630	456300
Accuracy	**ANN**	82.13	89.5	91.34	92.66	**95.57**
	LSTM	80.23	85.17	90.23	93.50	**96.89**

From table I and Fig 4, we can observe that ANN is showing a higher accuracy over LSTM, this clearly indicates the occurrence of over-fitting for smaller data set even though LSTM is better at generalising [16].

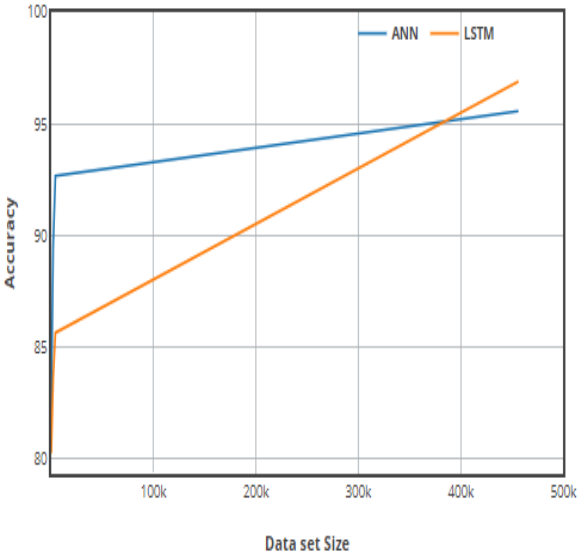

Fig. 4. Performance Comparison Of the Proposed models

To overcome this, we need to train the ANN over a larger data set which suitable for systems equipped with high computation power. LSTM is suitable for smaller data sets as well as larger data sets, although LSTM RNNs with a single layer does not perform very well for large-scale modelling tasks [16].

Analysis Of Non-Functional Requirements: The proposed models are highly scalable, adaptable, easy to train and deploy and requires lesser computation power in comparison to models with more layers or Convolutional Neural Networks [9].The training time is considerably less compared to the existing systems. It can easily be deployed to devices with varied hardware specifications.

V. CONCLUSIONS

Our main objective is to allow users access social media applications safely that will be achieved with our tool. From the results, it is clear that the proposed models give very good accuracy when trained over large datasets. The existing techniques required manual feature engineering to detect sequences in the URLs is a very difficult, computationally expensive and time-consuming task. The other objectives were to build a scalable, efficient, portable model that is easy to use all of which are clearly satisfied by our proposed tool.

References

[1] Andreas M Kaplan and Michael Haenlein. Users of the world, unite! the challenges and opportunities of social media. *Business horizons*, 53(1):59–68, 2010.

[2] Bogdan Batrinca and Philip C. Treleaven. Social media analytics: a survey of techniques, tools and platforms. *AI & SOCIETY*, 30(1):89–116, Feb 2015.

[3] Hongyu Gao, Yan Chen, Kathy Lee, Diana Palsetia, and Alok N Choudhary. Towards online spam filtering in social networks. In *NDSS*, volume 12, pages 1–16, 2012.

[4] Dharmaraj Rajaram Patil and JB Patil. Survey on malicious web pages detection techniques. *International Journal of U-and E-service, Science and Technology*, 8(5):195–206, 2015.

[5] Sangho Lee and Jong Kim. Warningbird: Detecting suspicious urls in twitter stream.

[6] Doyen Sahoo, Chenghao Liu, and Steven CH Hoi. Malicious url detection using machine learning: A survey. *arXiv preprint arXiv:1701.07179*, 2017.

[7] Sujata Garera, Niels Provos, Monica Chew, and Aviel D Rubin. A framework for detection and measurement of phishing attacks. In *Proceedings of the 2007 ACM workshop on Recurring malcode*, pages 1–8. ACM, 2007.

[8] Jonghyuk Song, Sangho Lee, and Jong Kim. Spam filtering in twitter using sender-receiver relationship. In *International workshop on recent advances in intrusion detection*, pages 301–317. Springer, 2011.

[9] Hung Le, Quang Pham, Doyen Sahoo, and Steven CH Hoi. Urlnet: Learning a url representation with deep learning for malicious url detection. *arXiv preprint arXiv:1802.03162*, 2018.

[10] LLC OpenDNS. Phishtank: An anti-phishing site. *Online: https://www. phishtank. com*, 2016.

[11] L Green. common crawl enters a new phase. *Common Crawl blog http://www. commoncrawl. org/common-crawl-enters-a-new-phase*, 2011.

[12] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

[13] Zixiang Ding, Rui Xia, Jianfei Yu, Xiang Li, and Jian Yang. Densely connected bidirectional lstm with applications to sentence classification. *arXiv preprint arXiv:1802.00889*, 2018.

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

[15] Alex Graves, Abdel-rahman Mohamed, and Geoffrey Hinton. Speech recognition with deep recurrent neural networks. In *Acoustics, speech and signal processing (icassp), 2013 ieee international conference on*, pages 6645–6649. IEEE, 2013.

[16] Haşim Sak, Andrew Senior, and Françoise Beaufays. Long short-term memory recurrent neural network architectures for large scale acoustic modeling. In *Fifteenth annual conference of the international speech communication association*, 2014.