Neural Network for Phishing Email Detection

Amir Hoshen, Bar Goldshtein, Peleg Zborovsky

Abstract

One of the internet-based identity thefts is called phishing. Phishing is a method of trying to gather personal information using deceptive e-mails and websites, while phishing can also be used to gain access to a system and even infect the system with a virus. The information the attackers seek for might be personal information including password, point of interest and even credentials. Due to this kind of attacks, people and companies may lose their money, self-reputation and humiliated in public. Despite all state-of-the-art solution to detect phishing attacks, there is still a lack of accuracy for the detection systems, in an ever growing fields of cyber-attacks the tools we have are very often out dated , approximately at the same time we all still uploading our information every day in every given minutes of each day. The goal is to develop a neural network in-order to detect a possibility of a potential phishing email attack on users.

Keywords: Convolutional neural network, Phishing, LSTM, Bi-LSTM, RCNN

Introduction

As shown in Anti-Phishing Working Group (APWG) Apwg (2021) report, in January 2021 we can see that the number of unique phishing web sites detected reached to a new record highs of 245,771 before declining later in the quarter. **Unique phishing sites,**- the victim is redirected to those sites by clicking a unique URL from a phishing Email. as we can seee in figure 1

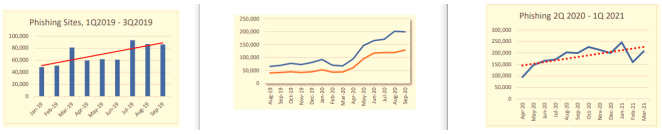

Figure 1 number of phishing sites

the number of phishing sites keep rising up

with the increasing number of attacks we can only assume that the financial damage is going in the same direction where the average amount requested in wire transfer Business Email Compromise (BEC) attacks increased from 48,000$ in Q3 2020 to 75,000$ in Q4 2020 and up to 85,000 $ in Q1 2021. according to PURPLESEC statistics pur (2021) 98% of cyber attacks rely on social engineering, phishing is a part of this kind of attacks. social engineering is when criminals deceive or manipulate victims into taking actions, anywhere human interaction is involved social engineering can take place. When it comes to Cyber security the human awareness factor and security education plays the most important role in identifying and addressing attacks related to social engineering.

Phishing - phishing can be done via email and text message that aimed to create sense of urgency, curiosity or raise fear in the victims and push the victims into revealing sensitive information and even open attachment that contain malware or click a link to malicious site.

in the past ,various of methods have been propose for detecting phishing emails, this methods can be divided into different class types like blacklist mechanism that rely on people to identify the phishing link and then report them, but this method requires a large amount of manpower and time. Another method is classification algorithms that based on machine learning which may not require that much manpower although it do require an extraction of features and mainly based on natural language processing (NLP) in order to translate the text into vectors that the machine can be trained on

in this paper our goal is to create various neural network models based machine learning and deep learning in order to detect if an email is spam , fraud , phishing or normal message. Deep learning is used in order to allow the model to find the best features in the textual content analyzing process in supervised learning, i.e. FRAUD email message ,this email representation in the data is structured as two columns, the first one is the email message content and the second one is the label corresponding('Normal','Fraud','Spam','Phish'). Few different models have been built and tested in order to increase accuracy in an attempt to reduce the vulnerabilities of email users in the near future. Every one of this attacks: fraud, spam or phishing is part of the social engineering attacks, and they are all based on textual content, in order to deal with textual content in computational models, word embedding is used. word embedding will be further discussed in the Proposed Approach section.

we'd love to thank Prof Rakesh Verma for the contribution to this project and sharing the data.Verma *et al.* (2012). the Dataset hold a MIME format with headers and without headers. we have our MIME Messages in two different folders, one for benign messages and the other for phishing messages.

Related Work

We are witnessing the ever-growing use of emails in our personal and business daily routine. this massive use of the email platform revealed vulnerability, mainly the innocence

and lack of awareness in aspects of cyber security on the part of users,that led to this types of phishing attacks through emails. in the past few years ,the field of phishing detection methods has been propose though articles like Fang *et al.* (2019) who's goal was to classify an email into two categories phishing or legitimate the article authors split the email into two parts, one is the header and the other is the body of the email message they also note that all textual content has two basic unit levels.

char-level-Characters forming all kind of word and then couple of word become a sentence and therefore Characters are the basic units of sentence.

Word-level-the must common use of word embedding is when words become sentences, using word embedding to transform the words into sentence vector of words. the authors mentioned that the body is where most of the important information resides, because the body is controlled only by the people who write the message. they also noted that when it comes to phishing attack in order to achieve the purpose of the attack there is often some **warning information in the body of phishing emails** then they combined the word embedding and an RCNN machine into forming the THEMIS model which have 99.848% accuracy for a phishing email detection model. This article deal with word embedding and the email structure

we would like to note Alotaibi *et al.* (2020) whose goal is to investigate the effectiveness of Convolutions Neural Networks for the classification of phishing emails. the article's data-set wasn't well balanced and they had an imbalance ratio of 54.91% for the sake of using that data, the author used Borderline-SMOTE oversampling which the article Dattagupta (2018) has proven to be the most effective over several over-sampling methods. Then they build a CNN machine to classify if a given email is legit one or phishing. Note that in our article we didn't have any over-sampling therefore we used this article to try and get more solid understanding of CNN machines.

Also should be noted, Salloum *et al.* (2021) which study phishing detection using NLP techniques. this article points out that the features for detecting phishing emails and split this email into five categories
Email body-based characteristics, Subject-based features, URL-based characteristics, Script-based features, Sender-based characteristics.
The paper present a critical analysis of research about phishing email detection techniques. they also mention Fang *et al.* (2019) and his THEMIS model who got an accuracy of 99.848% according to the outcomes of Salloum *et al.* (2021) study.

another paper worth to mention is Kulikova *et al.* (2021) in which we got insight on phishing in 2021 and how the corona virus changes the content of the phishing email additionally, we also use the Statistics that this paper provides. this article also provides a little insight into how scammers get corporate usernames and passwords. the attackers will write their message in a way that the email will look respectable and like a business tool or service, so the email will blend into the workflow of the corporate and in that email, they will try to persuade the worker into following a link to a fake page and enter the desire data on that page

the next paper is Rosander and Ahlstrand (2018) this paper explain the LSTM machines and methods of Word Embeddings. a point worth to note is that Rosander and Ahlstrand (2018) found that any NLP model that they used did not significantly affect the classification performance because the LSTM seem to compensate for the difference between them. the article also found out that LSTM outperform the other non-sequential models that they check in the article

in this paper We would like to harness the power of LSTM with different activation functions while we use the Data-set from Verma *et al.* (2012) in order to Comparing the results and determine the correctness of each of the models.

Proposed Approach

this article will use Long short term memory on sanitize text that been extract from the MIME format. and the first layer of the machine will be word embedding and we use word embedding because people read a sentence a 'bit' different from our Personal computer(PC), our computers world consists only numbers and every sentence we have in the MIME format can be 'embed' into vectors.

Word embedding

Word embedding is the representation of text where each word or Character(as we can see in Fang *et al.* (2019)). where each word with the same meaning will get a similar representation. Each word is represented by a real-valued vector, and often the dimensions of this vector can be even the size of tens and hundreds . which is contrasted to the fact that we need these vectors to be in the size of hundreds and millions because of the sparse required for the word representations like when using one-hot encoding. the answer for this contrast is in the way we represent the word is also based on the usage of the words and by this way we allow the word that is used in a similar way to have similar representation in other words the vector also get the meaning of this word.

extract data from MIME format

because the model need the body of the email for our machine the first step in our work is the extract the text from the MIME format our model check if our data contain HTML tags if it does we remove them using html2text module if the data contains an attachment for now we ignore it

pre-processing the text

this stage contains few steps
remove any punctuation symbol from the text and then remove any stop words (a, an, the, etc) to stay with clean data as much as possible

Long short term memory

LSTM develop to address the gradient problems in RNNs machines. **gradient problems** - in RNN when learning long data sequences, the gradient carry information to update the RNN parameters and in each step, the gradient becomes more

smaller and at some point, the gradient becomes so small that the parameter update become almost zero and that means that no real learning is done to address the gradient problems LSTM introduces gates that act as filters

Our LSTM machine

first, we sensitize the text we of the massage and delete special letter that could damage our data(like #)

Our LSTM machine looks like the following: the first layer is the input layer which input vectors of tokens whose been padded so every token will be at the same size as the other in order the get a union size for the input layer and these tokens also have been sanitized in the pre-processing stage the second layer of the model is the embedding layer
this layer transpose the sensitize data using embedding technique to get the text into vectors the next layer is doing dropout to reduce overfitting the next layer LSTM layer which will output results into the hidden layer that will perform a series of **convolution** and **pooling** operations and sent the result into the output layer that will classification the text into the four categories

Experimental And Evaluation

All of our test done on the same computer with GPU:Rtx 3080 64GB of RAM
and CPU: i7 10870h

LSTM with tanh function Loss: 0.109 Accuracy: 95.511% F1-Score:96.009% Precision:96.363% Recall:95.673%

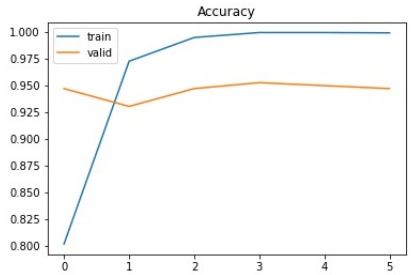

Figure 2 Accuracy using tanh function

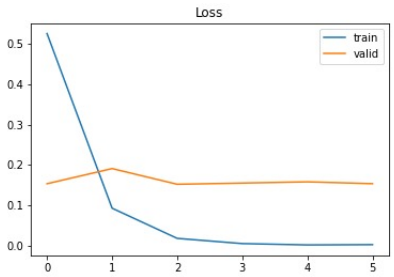

Figure 3 Loss using tanh function

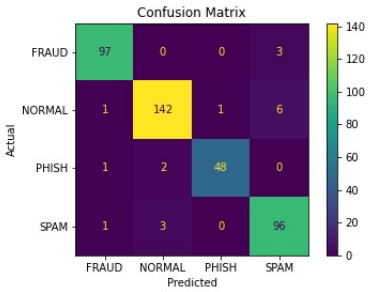

Figure 4 Confusion matrix using tanh funcion

LSTM with RELU function Loss: 0.119 Accuracy: 96.259% F1-Score:96.182% Precision:96.182% Recall:96.182%

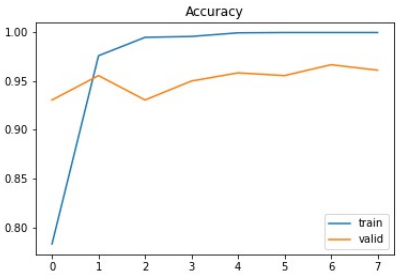

Figure 5 Accuracy using RELU function

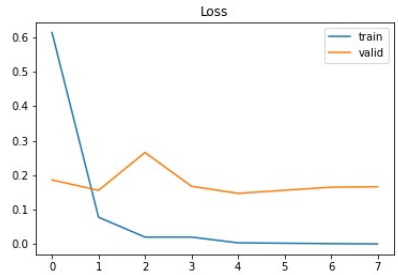

Figure 6 Loss using RELU function

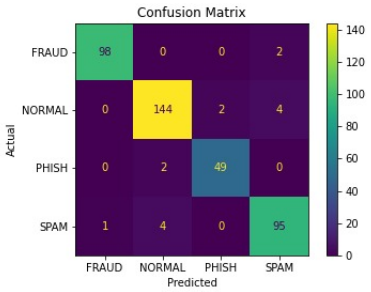

Figure 7 Confusion matrix using RELU funcion

BI-LSTM instad of using normal LSTM we try to use BI-LSTM wich is LSTM that can go also backward insted of only forward Loss: 0.098 Accuracy: 97.257% F1-Score:97.581% Precision:97.813% Recall:97.356%

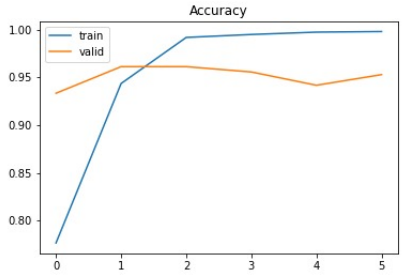

Figure 8 Accuracy using BI-LSTM

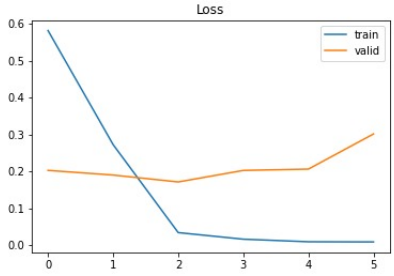

Figure 9 Loss using BI-LSTM

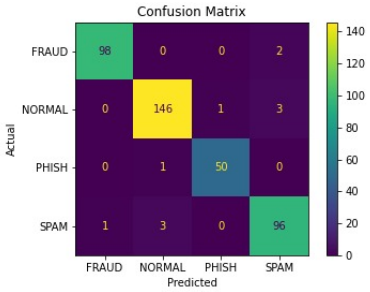

Figure 10 Confusion matrix using BI-LSTM

Conclusion

from the result, it is clear that the accuracy of the BI-LSTM machine in our model is higher than LSTM with different activation functions we test in this article. the main difference is the structure of the BI-LSTM and that is what made the BI-LSTM result higher than any tested LSTM

future work

Real world data - Balance the e-mails data quantity for each label according to real-world statistics

Literature cited

2021. 2021 cyber security statistics trends & data.

Alotaibi R, Al-Turaiki I, Alakeel F. 2020. Mitigating email phishing attacks using convolutional neural networks. In: . pp. 1–6. IEEE.

Apwg. 2021. Apwg q1 2021 report: Detected phishing websites maintain historic high in q1 2021, after doubling in 2020.

Dattagupta SJ. 2018. *A performance comparison of oversampling methods for data generation in imbalanced learning tasks*. Ph.D. thesis.

Fang Y, Zhang C, Huang C, Liu L, Yang Y. 2019. Phishing email detection using improved rcnn model with multilevel vectors and attention mechanism. IEEE Access. 7:56329–56340.

Kulikova T, Shcherbakova T, Sidorina T. 2021. Spam and phishing in q1 2021.

Rosander O, Ahlstrand J. 2018. Email classification with machine learning and word embeddings for improved customer support.

Salloum S, Gaber T, Vadera S, Shaalan K. 2021. Phishing email detection using natural language processing techniques: A literature survey. Procedia Computer Science. 189:19–28. AI in Computational Linguistics.

Verma R, Shashidhar N, Hossain N. 2012. Detecting phishing emails the natural language way. In: . pp. 824–841. Springer.