DeepAnti-PhishNet: Applying Deep Neural Networks for Phishing Email Detection

CEN-AISecurity@IWSPA-2018

Vinayakumar R[a], Barathi Ganesh HB[a,b], Anand Kumar M[a], Soman KP[a]
[a]Center for Computational Engineering and Networking(CEN),
Amrita School of Engineering, Coimbatore
Amrita Vishwa Vidyapeetham, India
vinayakumarr77@gmail.com

[b]Arnekt Solutions Pvt. Ltd., Pentagon P-3, Magarpatta City
Pune, Maharashtra, India
barathiganesh.hb@arnekt.com

Prabaharan Poornachandran
Center for Cyber Security Systems and Networks,
Amrita School of Engineering, Kollam
Amrita Vishwa Vidyapeetham, India
prabasuja@gmail.com

Abstract

Phishing represents a genuine risk to the Internet economy. Email has turned out to be a necessary verbal exchange tool in contemporary lifestyles. In recent days, email remains as the foremost generally utilized medium to dispatch phishing attacks. As a result, detection of phishing emails has been considered as an important task in the field of Cybersecurity. In this working note, we use word embedding and Neural Bag-of-ngrams with deep learning methods such as convolutional neural network (CNN), recurrent neural network (RNN), long short-term memory (LSTM) and multi-layer perceptron (MLP) to detect phishing email. Both word embedding and Neural Bag-of-ngrams facilitates to extract syntactic and semantic similarity of emails. Deep learning algorithms facilitate to extract the abstract and optimal feature representation and fully connected layer with non-linear activation function for classifcation. All the experiments are done on anti-phishing shared task corpus at IWSPA-AP 2018[1]. All the models performed well during training phase. Moreover, word embedding with LSTM obtains 10-fold cross validation accuracy of 0.991 on sub task 1[2] and 0.971 on sub task 2[3]. Based on the experimental results, we claim that word embedding with deep learning, specifcally LSTM is appropriate for the anti-phishing task.

In: R. Verma, A. Das (eds.): Proceedings of the 1st AntiPhishing Shared Pilot at 4th ACM International Workshop on Security and Privacy Analytics (IWSPA 2018), Tempe, Arizona, USA, 21-03-2018, published at http://ceur-ws.org

1 Introduction

Email or electronic mail is an effective type of communication through electronic devices over the Internet. It is one of the fastest and reasonable method of information exchange varying from interpersonal to intercorporational levels, ranging across continents as well

[1]dasavisha.github.io/IWSPA-sharedtask/
[2]Email without header
[3]Email with header

as to even outer space missions. The first message was sent through ARPANET (Advanced Research Projects Agency Network) from computer to computer on October 29th 1969 by the US Department of Defense. In 1971 the currently known electronic mail was developed by Ray Tomlinson while creating ARPANET's networked email system. In the modern era, the usage of email has been increasing rapidly because of its fast, low cost, effective and very convenient to use properties. Nowadays, as the use of smart phones and the availability of networks is in abundance in most places, usage of emails has also increased accordingly. As email became a primary communication tool for people, it finds application in almost all fields. Hence the marketers also have found its potential as a primary marketing tool. In 2018, the number of email users worldwide increased to 3.8 billion users as per the Radicati research group Inc. and is expected to rise to 4.1 billion users by 2021 as per the trend followed[4]. The number of business and consumer emails produced and delivered per day in 2018 reached 279 billion and is set to grow at a rate of 4.4% annually resulting to 319.6 billion emails by the end of 2021. So, almost half of the population uses email as a mode of communication these days. With its increasing popularity and ease of use, many people use it for inappropriate activities by sending illegitimate or spam emails[CL98].

Through spam emails people deliver all kinds of malicious attacks. The frequently used type of malware attack through spam emails are blended attacks. It uses more than one method to deliver malware to an internal network. Blended attacks often starts from illegitimate emails, which may not contain malware but provide links to compromised websites. Usually attackers send emails in such a way that it looks legitimate to a normal user by mixing authentic links and false links that will contain URLs to some fake website. As per the survey produced by IBM's X-Force research team, more than half of the emails produced worldwide are scam. The percentage of spam email amounted to 55.9% in the first quarter of 2017 and shows gradual increase in the coming years. Spam mails may also consist of phishing mails hence resulting in leakage of sensitive information at times. As reported by APWG, the number of phishing email has increased from 68270 in 2014 to 106421 in 2015[5]. According to Gartner report[6], 109 million users received phishing email. It can be delivered using several ways, by attaching files with malicious content or by sending a link to a compromised website. There are various types of phishing attacks that exist and these are discussed in detail by[7].

In most cases, the internet users fails to check the authenticity of the emails and land in compromised websites due to lack of education regarding security. There is no method found till now, which provide 100% accuracy in checking if someone has fell in the trap of phishing emails. But analyzing the header and checking the content of the email body for spelling or grammar mistakes or identifying emails seeking for personal information will help in most of the cases. Many works have been proposed to handle such scenarios[CNU06], [FST07], [ZDL07], [ANNWN07], [S$^+$08], [TC10], [HAK13]. Recently, [CYT18] has conducted a comprehensive literature survey on phishing attacks and its approaches. Additionally, [GAP18] discussed the current issues, future directions and taxonomy of methods for defending against phishing attacks. The methods which are based on the blacklisting and heuristic approaches completely fails to detect the new or the variants of existing phishing email[AGA$^+$13].

With all these traditional methods that we have been following since ages, artificial intelligence (AI) is another technique which became popular in last few decades. AI uses supervised learning classification algorithms to do binary classification of phishing emails. Machine learning methods rely on feature engineering to extract body, header based features and a hybrid of both to detect phishing email. This can perform well in comparision to any of the previous methods used for phishing email detection. It is because training and building a classifier based on a given data is much easier than to build a set of filtering rules. Additionally, they have the capability to detect variants of existing phishing email or the new email itself. However, the recent development of machine learning models, typically called as 'deep learning' models have performed well in various long-standing artificial intelligence (AI) tasks that exist in the fields of natural language processing, image processing and speech recognition. The application of deep learning techniques has transformed to various Cybersecurity tasks [VSP18b], [VSP18a], [VSPSK18a], [VSPSK18b], [VSP17d], [VSP17i], [VSP17f], [VSVG17], [VSP17b], [VSP17h], [VSP17c], [VSP17a], [VSP17e], [VSP17g]. The application of word embedding is largely used in text classification related to various domains [VKS17], [BVP]. Recently, the application of deep learning with word embedding is used for email spam detection[RK], [EC]. Following, in this paper, we use word embedding and Neural Bag-of-ngrams with deep learning models and MLP respectively to distinguish an email as phishing or legitimate.

The sections in this paper are arranged as follows:

[4]https://www.radicati.com/wp/wp-content/uploads/2017/01/Email-Statistics-Report-2017-2021-Executive-Summary.pdf

[5]APWG, Apwg attack trends report, 2014

[6]Gartner Survey Shows Phishing Attacks Escalated in 2007

[7]PCworld, 2016 mobile world congress, 2016

Section 2 discusses the mathematical details of algorithms. Section 3 includes task description, email representation and proposed architecture. Section 4 provides results. At last, the conclusion is placed in Section 5.

2 Background

The purpose of this section is to discuss the various deep learning approaches namely multi-layer perceptron (MLP), convolutional neural network (CNN), recurrent neural network (RNN) and long short-term memory (LSTM) architectures concisely.

2.1 Artificial neural networks (ANNs)

An artificial neural network (ANN) is a computational model influenced by the characteristics of biological neural networks. Feed forward neural network (FFN), convolutional neural network, recurrent neural network (RNN) and long short-term memory (LSTM) are various types of ANN.

2.1.1 Feed forward neural network (FFN)

Feed forward neural network (FFN) forms a directed graph which is composed of nodes and edges. FFN passes information along edges from one node to another without forming a cycle. Multi-layer perceptron (MLP) is one type of feed forward neural network that contains 3 or more layers, specifically input layer, one or more hidden layer and an output layer in which each layer has many neurons, called as units in mathematical notation. The number of hidden layers is selected through following fine-tuning mechanism. The information is transformed from one layer to another layer in forward direction without considering the past values. Moreover, neurons in each layer are fully connected. MLP is defined mathematically as $O : R^m X R^n$ where m is the size of the input vector $x = x_1, x_2, x_3....x_m$ and n is the size of the output vector $O(x)$ respectively. The computation in each hidden layer hi_i is mathematically defined as

$$hi_i = f(w^T x + b_i) \tag{1}$$

$hi_i : R^{d_i - 1} \rightarrow R^{d_i}$, $f : R \rightarrow R$, where $w \in R^{d \times d_{i-1}}$, $b \in R^d$ d_i denotes the size of the input, f is non-linear activation function, either *sigmoid* (values in the range [0, 1]) or *tangent* function (values in the range [1, -1]).

$$\sigma(x) = \frac{1}{1 + exp(-z)} \tag{2}$$

2.1.2 Convolutional neural network (CNN)

Convolution neural network (CNN) is most commonly used in the field of computer vision[LBH15]. This has been used for text classification tasks[Kim14], [KGB14]. This has been transformed for email spam detection[LNRW]. CNN is fairly effective and considerable faster for training and predictive evaluation in sequential data modeling problems. CNN network contains convolution1d layer, pooling1d layer (maxpooling1d or minpooling1d) and fully connected network with non-linear activation function. Generally, convolution1d layer extracts the optimal features, maxpooling1d reduces the dimension of the CNN layer features and fully connected layer is used for classification. A CNN may involve varying numbers of such convolutional layers and may finally be terminated with linear fully connected or partially connected layers. The number of layers and the number of filters decides the performance of theses networks. More and more abstract features are extracted in the higher layers of such networks and thus the number of such layers required heavily depends on the complexity and non linearity of the data under analysis. Further, the number of filters in each stage decides the number of features extracted from each stage. Proper choice of these numbers and tuning is a difficult task and several literature discusses this[Che90]. More the number of layers and filters, more is the computation required; hence it is important that concise designs are to be selected. Also, there is a high chance for the selection of an over fitted model which results in poor prediction accuracy. Techniques like 'dropout' are implemented during training to avoid this[SHK+14].

2.1.3 Recurrent neural network (RNN)

Recurrent neural network is a variant of traditional FFNs introduced in the 1980's for time-series data modeling[Elm90]. As RNN has a cyclic connection in its units it facilitates to carryout previous time step information in computing the current states. This has obtained good performance in long standing artificial intelligence tasks related to the field of computer vision, natural language processing, speech processing and others[BSF94]. The values hidden layer units are estimated recurrently by a transition function tf according to the current input vector x_t and the previous hidden state hi_{t-1}.

$$hi_t = \begin{cases} 0 & t = 0 \\ tf(hi_{t-1}, x_t) & otherwise \end{cases} \tag{3}$$

where tf is a mix of affine transformation of x_t and hi_{t-1} with the element wise non-linearity. This type of transition function tf is trained using the backpropagation through time (BPTT). While in the process of backpropagating error across many time-steps, the weight matrix has to be multiplied with the gradient signal. This causes the vanishing issue when

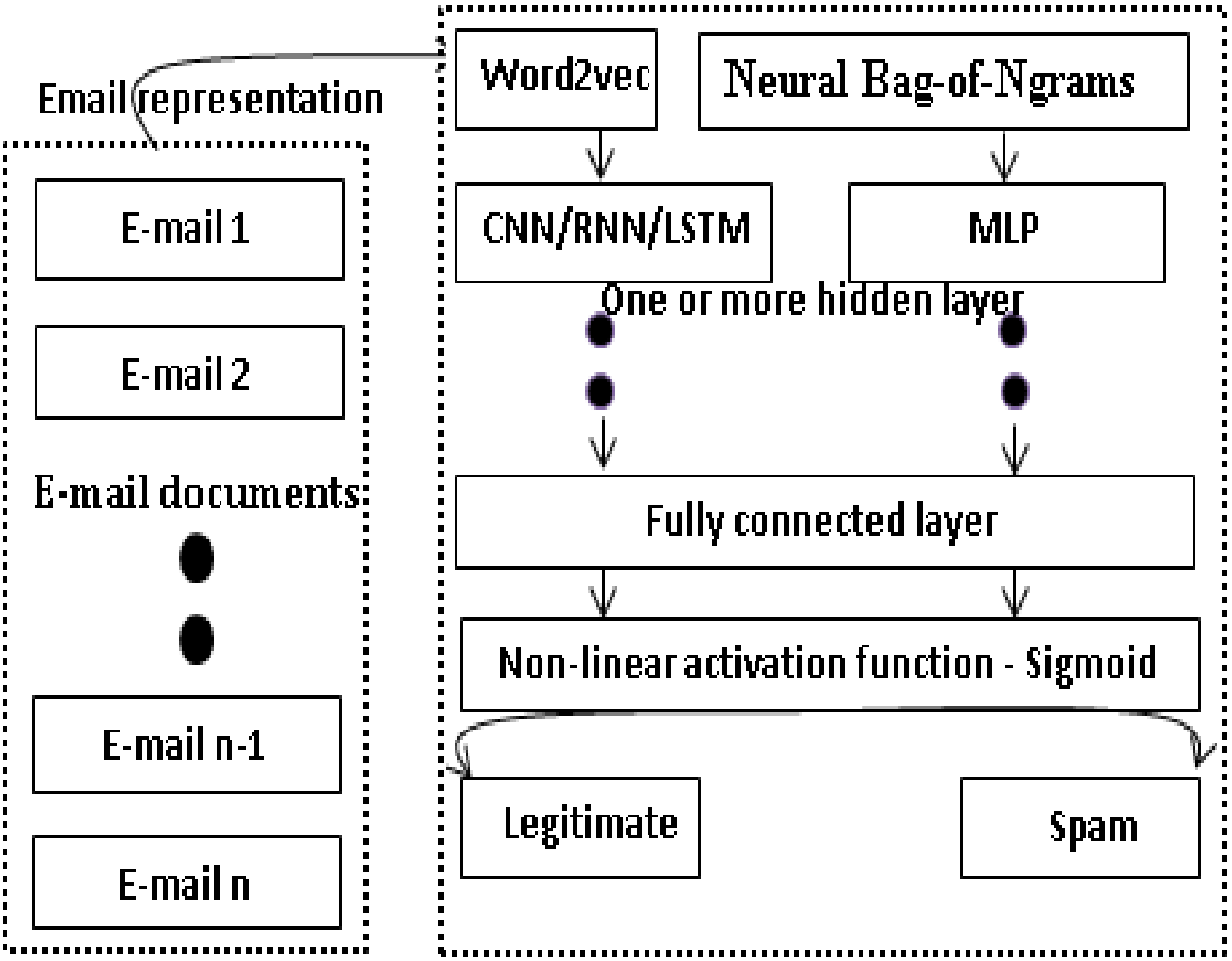

Figure 1: Proposed architecture for phishing email detection (All connections and inner elements of deep learning architecture are not shown)

a gradient becomes too small and exploding gradient issue when a gradient becomes too large[HS97]. Long short-term memory (LSTM) is an extension of RNN[GSC99], [GSS02], [MCCD13], memory block in LSTM facilitates to handle the vanishing and exploding gradient problem by forcing the constant error flow.

Generally, a memory block in LSTM is composed of input gate (ig), forget gate (fg), output gate (og), memory cell (m) and hidden state vector (hi) at each time step t. The values of input gate (ig), forget gate (fg), output gate (og) are in the range [0, 1]. The transition function (tf) for each LSTM units is written as follows

$$ig_t = \sigma(w_{ig}x_t + P_{ig}hi_{t-1} + Q_{ig}m_{t-1} + b_{ig}) \quad (4)$$

$$fg_t = \sigma(w_{fg}x_t + P_{fg}hi_{t-1} + Q_{fg}m_{t-1} + b_{fg}) \quad (5)$$

$$og_t = \sigma(w_{og}x_t + P_{og}hi_{t-1} + Q_{og}m_{t-1} + b_{og}) \quad (6)$$

$$m_t = \tanh(w_m x_t + P_m hi_{t-1} + b_m) \quad (7)$$

$$hi_t = og_t \odot \tanh(m_t) \quad (8)$$

where x_t is the input at time step t, σ is sigmoid non-linear activation function, $\odot$ denotes element-wise multiplication.

Table 1: Confguration details of word2vec model

parameter	value	description
batch_size	128	The number of training samples required in the operation
embedding_size	200	Word vector dimension
skip_window	5	Context window, five words before and after each word
num_skips	10	How many prediction pairs are selected from the window
num_samples	64	Number of negative samples
learning_rate	0.01	learning rate
n_epoch	100	Repeat training for all samples

Table 2: Summary of test results in terms of accuracy, precision, recall and F1-score

Method	Task	Accuracy	Precision	Recall	f1-score
Word embedding + RNN	sub task 1	0.86	0.962	0.876	0.917
Word embedding + LSTM	sub task 1	0.876	0.957	0.90	0.928
Neural Bag-of-ngrams + MLP	sub task 2	0.864	0.936	0.91	0.922

3 Experiments

All experiments are run on GPU enabled TensorFlow[ABC+16] in conjunction with Keras[C+15]. All deep learning architectures are trained using the back propagation through time (BPTT)[Wer90] technique. TensorFlow is an open source library for numeric computation using data flow graphs. TensorFlow is the second generation of machine learning platforms developed by the Google Brain team after DistBelief. As the name suggests, TensorFlow represents a problem with a data flow model acting on N dimensional arrays (tensors). The key advantage of the framework is its flexibility; the model can be mapped onto a range of hardware platforms ranging from a mobile device to massive GPU clusters.

3.1 Task description

The phishing email detection is a task on anti-phishing shared task at first security and privacy analytics anti-phishing shared task (IWSPA-AP 2018) as co-located with 8th ACM Conference on Data and Application Security and Privacy[8]. Anti-phishing shared task is an exercise in the field of applied machine learning and text analysis in the domain of Cybersecurity. The email corpus was provided by the organizers of the (IWSPA-AP 2018)[EDMB+18]. The aim of the anti-phishing shared task is to build a classifier to detect phishing email from spam and legitimate ones. The given email corpus is highly unbalanced, this is primarily to make the task relatable to a real world situation. Both of the sub tasks belongs to unconstrained category which means participants can use any of the other external corpus during training. The Anti-phishing shared task contains two sub tasks, The sub task 1 contains email samples without header and sub task 2 contains email samples with header. The detailed statistics of training and testing email corpus of each task is summarized in Table 5 and Table 6. The detailed description of both sub task 1 and sub task 2 corpus and the baseline detection methodologies is summarized by the shared task organizers[EDB+18].

[8]http://www.ycheng.org/codaspy/2018/index.html

3.2 Problem definition

Let $E = \{e_1, e_2, \cdots e_n\}$ be a set of email samples and $C = \{c_1, c_2, \cdots c_n\}$ be a set of email types such as legitimate or phishing, where n denotes the number of email samples. The task is to classify each given email sample into either legitimate or phishing.

3.3 Text Representation

The first step is to map each email sample into their corresponding numeric vector representation. Two types of distributed text representation are mapped for email representation, they are (1) Word2vec (2) Neural Bag-of-ngrams

1. **Word2vec** : Representing a word in dense vector form is called as word embedding, which is a projection of words (tokens) into vector space. The vector we got from projection will capture semantic properties which will be very helpful in improving the performance of the natural language processing (NLP) system to get a better result than traditional bag-of-words representations. For this work, we implement skip-gram

Table 3: Summary of test results

Method	Task	TP	TN	FP	FN
Word embedding + RNN	sub task 1	3239	369	127	460
Word embedding + LSTM	sub task 1	3329	346	150	370
Neural Bag-of-ngrams + MLP	sub task 2	3479	237	238	346

Table 4: Summary of 10-fold cross validation accuracy

Method	Task	Accuracy
Word embedding + CNN	sub task 1	0.974
Word embedding + RNN	sub task 1	0.975
Word embedding + LSTM	sub task 1	0.991
Neural Bag-of-ngrams + MLP	sub task 1	0.962
Word embedding + CNN	sub task 2	0.965
Word embedding + RNN	sub task 2	0.962
Word embedding + LSTM	sub task 2	0.971
Neural Bag-of-ngrams + MLP	sub task 2	0.979

module[MCCD13], [MSC$^+$13], [Ron14] so as to predict the context based on the given current word.

2. **Neural Bag-of-ngrams**: Traditional bag-of-words is a representation of multiset belonging words where the word order and grammar are ignored, on the other hand in Bag-of-ngram a token is represented with one-hot representation with the sum of n-gram vectors. Bag-of-ngram is a sparse vector representation where the semantics of the text is disregarded. To overcome this, Neural Bag-of-ngrams is introduced [LLZ$^+$17], [JGBM16]. Neural Bag-of-ngrams vectors is a dense, real-valued vector representation and also captures the semantics of the context. It is the combination of Bag-of-ngram and neural word embedding. It is robust, simple and flexible.

3.4 Proposed Architecture

The proposed architecture to identify phishing emails is shown in Fig. 1. This same architecture is used for both the sub tasks. The proposed tool is DeepAntiPhishNet[9] made publically available. This can be adopted to any of the security text classification. This can work on any other language as well as the code mixed language. The architecture contains the following modules.

- **Representation of emails**: Two types of email representation are used. They are (1) word2vec (2) Neural Bag-of-ngrams. Table 1 incudes the detailed configuration details of word2vec. In word embedding, we append every word's embedding so the text's representation is a variable length embeddings. Neural Bag-of-ngrams sum every word's embedding up to a fixed one.

- **Deep learning**: The dense vector that is obtained from the word2vec and Neural Bag-of-ngrams are passed as input to the CNN/RNN/LSTM and MLP network. All these various algorithms capture the appropriate feature representation and pass into the fully connected layer for classification. The detailed configuration details of MLP, CNN, RNN and LSTM architecture is reported in Table 7, 8, 9 and 10 respectively.

- **Classification**: The units in this layer have connection to every other unit in the succeeding layer. That's why this layer is called as fully-connected layer. It contains *sigmoid* non-linear activation function, which gives values 0 for legitimate and 1 for phishing. The prediction loss for both the sub tasks is estimated using binary cross entropy, as given below

$$loss(p, e) = -\frac{1}{N}\sum_{i=1}^{N}[e_i \cdot log(p_i) + (1-e_i) \cdot log(1-p_i)] \quad (9)$$

[9] github.com/vinayakumarr/IWSPA-AP-2018

Table 5: Training email corpus details

Task	Type	Legitimate	Phish	Total
sub task 1	No header	5088	612	5700
sub task 2	With header	4082	501	4583

Table 6: Testing email corpus details

Task	Type	Total
sub task 1	No header	4300
sub task 2	With header	4195

Table 7: Confguration details of MLP model

parameter	value
n_epoch	500
batch_size	128
n_units	200
cost_function	binary cross entropy
optimizer	Adam
Dropout	0.01

where p is a vector of predicted probability for all samples in testing corpus, e is a vector of expected class label, values are either 0 or 1.

4 Results

Though the sub tasks are unconstrained, only the private training corpus is used with word2vec and Neural Bag-of-ngrams. Initially, preprocessing is done on the corpus. Preprocessing includes conversion of all characters to lower case, ignoring punctuation marks and special characters and assigning a unique number for the unknown word. This is primarily due to the fact that distinguishing between the lowercase and higher case letters might ends up in regularization issue. Word vector of dimension 200 is estimated on the preprocessd data by using the word2vec and Neural Bag-of-ngrams. These methods captures the syntactic and semantic similarity of words exist in phishing and legitimate emails. These word vectors are passed to the deep learning algorithms. These algorithms learns the abstract and high level feature representation and in turn passes into the fully connected layer for classification.

To find the hyper parameters existing in word2vec and deep learning model, the given training data has been randomly shuffled and split into 73% training and 27% testing. The best hyperparameters of word2vec, MLP, CNN, RNN and LSTM is displayed in Table 1, 7, 8, 9 and 10 respectively. The 10-fold cross validation accuracy of both the email representation with deep learning methods and MLP model is displayed in Table 4.

Table 8: Confguration details of CNN model

parameter	value
learning_rate	0.1
cost_function	binary cross entropy
optimizer	Adam
n_epoch	100
batch_size	128
activation function in CNN	ReLU
flter_size	3
strides	3
Dropout	0.1
number_of_flters	128

Table 9: Confguration details of RNN model

parameter	value
Dropout	0.01
batch_size	128
n_epoch	1000
cost_function	binary cross entropy
optimizer	Adam
units in RNN	128

We have submitted two runs for sub task 1, where the first run is based on RNN with word2vec and second run is based on LSTM with word2vec. This is due to the fact that the 10-fold cross validation of RNN and LSTM model are closer. We have submitted one run for sub task 2. It is based on the MLP with Neural Bag-of-ngrams. The model performance on the submitted runs has been evaluated by the shared task organizers. Shared task organizers evaluated submitted runs based on the true positive (TP), true negative (TN), false positive (FP) and false negative (FN) and are reported in Table 3. From the Table 3, we estimated the accuracy, precision, recall and f1-score and are reported in Table 2. Accuracy, precision, recall and f1-score are estimated using the following equations

$$accuracy = \frac{(tp + tn)}{(tp + fp + tn + fn)} \tag{10}$$

Table 10: Confguration details of LSTM model

parameter	value
Dropout	0.1
batch_size	256
n_epoch	1000
cost_function	binary cross entropy
optimizer	Adam
units in RNN	128

$$precision = \frac{tp}{(tp + fp)} \tag{11}$$

$$recall = \frac{tp}{(tp + fn)} \tag{12}$$

$$f1 - score = \frac{(2 * tp)}{(2 * tp + fp + fn)} \tag{13}$$

From the Table 2, 3 and 4, we can observe that the model has shown less performance during testing when compared to training. This is due to that the email corpus used during training process is very less. Moreover, the unbalanced email corpus has made the model to be biased. This can be alleviated by training a word2vec model on the large number of email corpus and training by using the highly complex deep learning model.

5 Conclusion

Detecting phishing email from spam and legitimate ones is considered as an important aspect in the field of Cybersecurity. This is primarily due to the fact that most of the internet's traffic was caused by phishing emails in the previous years. In this work, we use word embedding and Neural Bag-of-ngrams with deep learning algorithms such as CNN/RNN/LSTM and traditional neural network, MLP. The performance of deep learning methods with word embedding and Neural Bag-of-ngrams with MLP is closer. Moreover, the LSTM network has performed well in both the sub tasks. Due to computational cost and other constraints, we were not able to train complex deep learning architecture. The performance of the system can be enhanced with more complex deep learning architecture. These architectures can be trained by using advanced hardware and following distributed approach in training that we are incompetent to try.

Both the sub tasks belong to the unconstrained category which means any other corpus can be used during training. The given corpus for both the sub tasks are highly imbalanced. Even though the tasks are unconstrained, we haven't used any other external data sources. With highly imbalanced corpus, the proposed methodology achieve considerable phishing email detection rate in both the sub tasks. The phishing email detection rate of the proposed methodology can be easily enhanced by adding additional extra publically available or private data sources. This will be considered as one of the significant direction towards the future work.

Acknowledgements

This research was supported in part by Paramount Computer Systems. We are grateful to NVIDIA India, for the GPU hardware support to the research grant. We are grateful to Computational Engineering and Networking (CEN) department for encouraging the research.

References

[ABC+16] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. Tensorflow: A system for large-scale machine learning. In *OSDI*, volume 16, pages 265–283, 2016.

[AGA+13] Ammar Almomani, BB Gupta, Samer Atawneh, A Meulenberg, and Eman Almomani. A survey of phishing email filtering techniques. *IEEE communications surveys & tutorials*, 15(4):2070–2090, 2013.

[ANNWN07] Saeed Abu-Nimeh, Dario Nappa, Xinlei Wang, and Suku Nair. A comparison of machine learning techniques for phishing detection. In *Proceedings of the anti-phishing working groups 2nd annual eCrime researchers summit*, pages 60–69. ACM, 2007.

[BSF94] Yoshua Bengio, Patrice Simard, and Paolo Frasconi. Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks*, 5(2):157–166, 1994.

[BVP] Barathi Ganesh Hullathy Balakrishnan, Anand Kumar Madasamy Vinayakumar, and Soman Kotti Padannayil. Nlp cen amrita@ smm4h: Health care text classification through class embeddings.

[C+15] François Chollet et al. Keras, 2015.

[Che90] F-C Chen. Back-propagation neural networks for nonlinear self-tuning adaptive control. *IEEE Control systems magazine*, 10(3):44–48, 1990.

[CL98] Lorrie Faith Cranor and Brian A LaMacchia. Spam! *Communications of the ACM*, 41(8):74–83, 1998.

[CNU06] Madhusudhanan Chandrasekaran, Krishnan Narayanan, and Shambhu Upadhyaya. Phishing email detection based on structural properties. In *NYS Cyber Security Conference*, volume 3, 2006.

[CYT18] Kang Leng Chiew, Kelvin Sheng Chek Yong, and Choon Lin Tan. A survey of phishing attacks: Their types, vectors and technical approaches. *Expert Systems with Applications*, 2018.

[EC] Louis Eugene and Isaac Caswell. Making a manageable email experience with deep learning.

[EDB+18] Ayman Elaassal, Avisha Das, Shahryar Baki, Luis De Moraes, and Rakesh Verma. Iwspa-ap: Anti-phising shared task at acm international workshop on security and privacy analytics. In *Proceedings of the 1st IWSPA Anti-Phishing Shared Task*. CEUR, 2018.

[EDMB+18] Ayman Elaassal, Luis De Moraes, Shahryar Baki, Rakesh Verma, and Avisha Das. Iwspa-ap shared task email dataset, 2018.

[Elm90] Jeffrey L Elman. Finding structure in time. *Cognitive science*, 14(2):179–211, 1990.

[FST07] Ian Fette, Norman Sadeh, and Anthony Tomasic. Learning to detect phishing emails. In *Proceedings of the 16th international conference on World Wide Web*, pages 649–656. ACM, 2007.

[GAP18] BB Gupta, Nalin AG Arachchilage, and Kostas E Psannis. Defending against phishing attacks: taxonomy of methods, current issues and future directions. *Telecommunication Systems*, 67(2):247–267, 2018.

[GSC99] Felix A Gers, Jürgen Schmidhuber, and Fred Cummins. Learning to forget: Continual prediction with lstm. 1999.

[GSS02] Felix A Gers, Nicol N Schraudolph, and Jürgen Schmidhuber. Learning precise timing with lstm recurrent networks. *Journal of machine learning research*, 3(Aug):115–143, 2002.

[HAK13] Isredza Rahmi A Hamid, Jemal Abawajy, and Tai-hoon Kim. Using feature selection and classification scheme for automating phishing email detection. *Studies in Informatics and Control*, 22(1):61–70, 2013.

[HS97] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[JGBM16] Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. Bag of tricks for efficient text classification. *arXiv preprint arXiv:1607.01759*, 2016.

[KGB14] Nal Kalchbrenner, Edward Grefenstette, and Phil Blunsom. A convolutional neural network for modelling sentences. *arXiv preprint arXiv:1404.2188*, 2014.

[Kim14] Yoon Kim. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*, 2014.

[LBH15] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436, 2015.

[LLZ+17] Bofang Li, Tao Liu, Zhe Zhao, Puwei Wang, and Xiaoyong Du. Neural bag-of-ngrams. In *AAAI*, pages 3067–3074, 2017.

[LNRW] Christopher Lennan, Bastian Naber, Jan Reher, and Leon Weber. End-to-end spam classification with neural networks.

[MCCD13] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[MSC+13] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.

[RK] Tim Repke and Ralf Krestel. Bringing back structure to free text email conversations with recurrent neural networks, ecir 2018.

[Ron14] Xin Rong. word2vec parameter learning explained. *arXiv preprint arXiv:1411.2738*, 2014.

[S+08] A Syed et al. Feature selection for spam and phishing detection. In *NIPS 2007 Workshop on Machine Learning in Adversarial Environments for Computer Security*, 2008.

[SHK+14] 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.

[TC10] F Toolan and J Carthy. A survey of phishing email filtering techniques. *eCrime Researchers Summit*, pages 1–12, 2010.

[VKS17] PV Veena, M Anand Kumar, and KP Soman. An effective way of word-level language identification for code-mixed facebook comments using word-embedding via character-embedding. In *Advances in Computing, Communications and Informatics (ICACCI), 2017 International Conference on*, pages 1552–1556. IEEE, 2017.

[VSP17a] R Vinayakumar, KP Soman, and Prabaharan Poornachandran. Applying convolutional neural network for network intrusion detection. In *Advances in Computing, Communications and Informatics (ICACCI), 2017 International Conference on*, pages 1222–1228. IEEE, 2017.

[VSP17b] R Vinayakumar, KP Soman, and Prabaharan Poornachandran. Applying deep learning approaches for network traffic prediction. In *Advances in Computing, Communications and Informatics (ICACCI), 2017 International Conference on*, pages 2353–2358. IEEE, 2017.

[VSP17c] R Vinayakumar, KP Soman, and Prabaharan Poornachandran. Deep android malware detection and classification. In *Advances in Computing, Communications and Informatics (ICACCI), 2017 International Conference on*, pages 1677–1683. IEEE, 2017.

[VSP17d] R Vinayakumar, KP Soman, and Prabaharan Poornachandran. Deep encrypted text categorization. In *Advances in Computing, Communications and Informatics (ICACCI), 2017 International Conference on*, pages 364–370. IEEE, 2017.

[VSP17e] R Vinayakumar, KP Soman, and Prabaharan Poornachandran. Evaluating effectiveness of shallow and deep networks to intrusion detection system. In *Advances in Computing, Communications and Informatics (ICACCI), 2017 International Conference on*, pages 1282–1289. IEEE, 2017.

[VSP17f] R Vinayakumar, KP Soman, and Prabaharan Poornachandran. Evaluating shallow and deep networks for secure shell (ssh) traffic analysis. In *Advances in Computing, Communications and Informatics (ICACCI), 2017 International Conference on*, pages 266–274. IEEE, 2017.

[VSP17g] R Vinayakumar, KP Soman, and Prabaharan Poornachandran. Evaluation of recurrent neural network and its variants for intrusion detection system (ids). *International Journal of Information System Modeling and Design (IJISMD)*, 8(3):43–63, 2017.

[VSP17h] R Vinayakumar, KP Soman, and Prabaharan Poornachandran. Long short-term memory based operation log anomaly detection. In *Advances in Computing, Communications and Informatics (ICACCI), 2017 International Conference on*, pages 236–242. IEEE, 2017.

[VSP17i] R Vinayakumar, KP Soman, and Prabaharan Poornachandran. Secure shell (ssh) traffic analysis with flow based features using shallow and deep networks. In *Advances in Computing, Communications and Informatics (ICACCI), 2017 International Conference on*, pages 2026–2032. IEEE, 2017.

[VSP18a] R Vinayakumar, KP Soman, and Prabaharan Poornachandran. Detecting malicious domain names using deep learning approaches at scale. *Journal of Intelligent & Fuzzy Systems*, 34(3):1355–1367, 2018.

[VSP18b] R Vinayakumar, KP Soman, and Prabaharan Poornachandran. Evaluating deep learning approaches to characterize and classify malicious urls. *Journal of Intelligent & Fuzzy Systems*, 34(3):1333–1343, 2018.

[VSPSK18a] R Vinayakumar, KP Soman, Prabaharan Poornachandran, and S Sachin Kumar. Detecting android malware using long short-term memory (lstm). *Journal of Intelligent & Fuzzy Systems*, 34(3):1277–1288, 2018.

[VSPSK18b] R Vinayakumar, KP Soman, Prabaharan Poornachandran, and S Sachin Kumar. Evaluating deep learning approaches to characterize and classify the dgas at scale. *Journal of Intelligent & Fuzzy Systems*, 34(3):1265–1276, 2018.

[VSVG17] R Vinayakumar, KP Soman, KK Senthil Velan, and Shaunak Ganorkar. Evaluating shallow and deep networks for ransomware detection and classification. In *Advances in Computing, Communications and Informatics (ICACCI), 2017 International Conference on*, pages 259–265. IEEE, 2017.

[Wer90] Paul J Werbos. Backpropagation through time: what it does and how to do it. *Proceedings of the IEEE*, 78(10):1550–1560, 1990.

[ZDL07] Jian Zhang, Zhen-Hua Du, and Wei Liu. A behavior-based detection approach to mass-mailing host. In *Machine Learning and Cybernetics, 2007 International Conference on*, volume 4, pages 2140–2144. IEEE, 2007.