Vulnerability Severity Prediction With Deep Neural Network

1st Kai Liu
Science and Technology on Information Systems Engineering Laboratory
National University of Defense Technology
Changsha, China
Liukai18@nudt.edu.cn

2nd Yun Zhou
Science and Technology on Information Systems Engineering Laboratory
National University of Defense Technology
Changsha, China
zhouyun@nudt.edu.cn

3rd Qingyong Wang
Science and Technology on Information Systems Engineering Laboratory
National University of Defense Technology
Changsha, China
wang@cnu.ac.cn

4th Xianqiang Zhu
Science and Technology on Information Systems Engineering Laboratory
National University of Defense Technology
Changsha, China
zhuxianqiang@nudt.edu.cn

***Abstract*—High frequency of network security incidents has also brought a lot of negative effects and even huge economic losses to countries, enterprises and individuals in recent years. Therefore, more and more attention has been paid to the problem of network security. In order to** evaluate the newly included vulnerability text **information accurately, and to reduce the workload of experts and the false negative rate of the traditional method. Multiple deep learning methods for vulnerability text classification evaluation are proposed in this paper. The standard Cross Site Scripting (XSS) vulnerability text data is processed first, and then classified using three kinds of deep neural networks (CNN, LSTM, TextRCNN) and one kind of traditional machine learning method (XGBoost). The dropout ratio of the optimal CNN network, the epoch of all deep neural networks and training set data were tuned via experiments to improve the fit on our target task. The results show that the deep learning methods evaluate vulnerability risk levels better, compared with traditional machine learning methods, but cost more time. We train our models in various training sets and test with the same testing set. The performance and utility of recurrent convolutional neural networks (TextRCNN) is highest in comparison to all other methods, which classification accuracy rate is 93.95%.**

***Index Terms*—Network security, XSS vulnerability, Text classification, Deep neural networks**

I. INTRODUCTION

With the development of internet, it has penetrated into all aspects of human production and life. However, the problem of network security affects economic losses to countries, enterprises and individuals. Therefore, the network security is paid attention rapidly, mainly reasons are security vulnerabilities in hardware, software, protocols and so on in computer systems. These vulnerabilities are attacked so that caused loss to the target system [1], [2]. The total number of vulnerabilities counted about 112946 by National Vulnerability Database (NVD) in the United States in the past two decades, and the growth trend is obvious, with the emergence of a large number of vulnerabilities [3], [4]. In order to deal with the increasingly serious network security situation, China also set up the China National Vulnerability Database of Information Security (CNNVD) to analyze and assess the vulnerabilities. The release of these vulnerabilities information plays a very positive role in improving the security protection of information systems. However, due to the large new increment of vulnerabilities, it has become an important challenge in the field of network security analysis to complete the accurate and efficient assessment of the threat degree of security vulnerabilities in the network. NVD, CNNVD and other open source leak libraries for network security personnel timely summary of network threat intelligence, timely update of the leak library can enable technicians to find new vulnerabilities in a timely manner [5]. The traditional security vulnerability analysis method needs a lot of manual participation, which is not only time-consuming

and laborious, but also omits a lot of vulnerabilities, and there is a high underreporting rate [6]. In general, the analysis and repair of vulnerabilities should be determined on the basis of the degree of threat and the resources available to security personnel, and the vulnerabilities with higher levels of threat should be dealt with as a matter of priority. In order to realize the rapid analysis of vulnerabilities, it should be based on the classification of vulnerability information. However, the latest vulnerability information usually does not have a corresponding threat assessment, which affects the efficiency of security personnel. Therefore, prediction vulnerabilities is an effective way to provide priority for technicians. At present, many kinds of vulnerabilities are found, and many methods from different angles are derived to solve these problems. NVD uses the Common Vulnerability Scoring System (CVSS) to assess the threat degree of vulnerability, and gives a qualitative judgment on the severity of the vulnerability according to the level of the score [7].

In recent years, machine learning has been widely used in the field of vulnerability evaluation, and has become a research focus. The implicit latent dirichlet allocation (LDA) and support vector machine (SVM) are used to classify the network vulnerabilities [8]. Yamamoto et al. combine machine learning with text mining technology [9]. On the crawling NVD data, LDA, SLI and SLDA models are used to extract the topic of NVD text, and the topic is used to evaluate the characteristics of vulnerabilities. At the same time, linear function and sigmoid function are introduced for weight distribution, which improves the accuracy of classification and prediction. Spanos and Ghaffarian et al. compared the three methods of decision tree, support vector machine and neural network to analyze the vulnerability description text, and classified the corresponding evaluation value. The prediction accuracy is about 80% [10]. Toloudis et al. studied two vulnerability threat degree scoring methods (CVSS, WIVSS), which is composed of sub-scores, by text mining, and constructed a word vector that can represent a vulnerable text description text [11]. The Spelman correlation coefficient was used to depict the relationship between the word vector and the sub-scores of the vulnerability. The fuzzy entropy theory and SVM multi-classification algorithm are used to classify vulnerabilities on the basis of vulnerability text feature extraction [12]. Meanwhile, text mining technology extracted the characteristics of different vulnerability texts, and the self-proposed SVM binary tree algorithm to automatically divide the categories of vulnerabilities [13]. Wang et al. compared multiple machine learning methods, such as SVM, Logistic Regression (LR), Random Forest (RF) and eXtreme Gradient Boosting (XGboost), to intelligently predict the vulnerability security level. Among these methods we observe that the XGBoost prediction accuracy reaches 80.48%, which has advantages than other algorithms. After gathering the sparse features with principle component analysis (PCA), the mentioned accuracy improve to 92.928% [14]. As the increasing of vulnerability data, deep neural networks show more advantages than traditional machine learning. At present, deep learning technology has been widely used in image processing, speech recognition and natural language processing. For example, Convolutional neural network (CNN) to text classification [15], vulnerability detection system based on deep learning [6]. The vulnerability description text is one kind of unstructured time series data. Zhou et al. utilized long short-term memory networks (LSTM) in text classification, which achieve excellent performance [16]. S. Lai et al. introduce a recurrent convolutional neural network (RCNN) for text classification without human-designed features, and the proposed method outperforms the state-of-the-art methods in several datasets [17]. In addition, the combination of traditional machine learning technology and text mining to classify vulnerability description text often requires a large number of complex feature engineering, and different feature selection also has a great impact on the final classification prediction accuracy [18]. Therefore, deep learning has become an important development direction to deal with the problem of feature engineering [19]. In order to solve these problems, this paper uses several deep learning methods to classify the vulnerability description text, and realizes the evaluation of the threat degree of the vulnerability. The main work is organized as follows:

1) the main source and composition of the text data of the vulnerability are described;

2) The experiment used XGBoost, CNN, LSTM and TextRCNN methods to classify the vulnerability description texts for assessing the vulnerability severity level and compare their predictions;

3) The parameter setting of the models is discussed for problem of vulnerable detection classification, and the advantages of the TextRCNN model are evaluated and analyzed on the experimental part.

II. METHODS

The summary of a large number of vulnerability descriptions, XGBoost, CNN, LSTM and TextRCNN

methods are used in this paper to solve classification of XSS [20], [21]. Fig. 1 is flow chart. We firstly crawl the XSS vulnerability data from NVD form a date set followed by data preprocessing. Then we use text mining to implement feature extraction. The above mentioned machine learning methods are applied to five training sets with different volumes. Their classification results are evaluated with the same testing set. XGBoost algorithm is one classic machine learning classification methods. The idea of the algorithm is to constantly add trees, constantly transforming features to grow a tree, and adding a tree each time is actually learning a new function to fit the residual of the last prediction. When we get the k tree in training, we need to predict the score of a sample. In fact, according to the characteristics of this sample, we will fall into the corresponding leaf node in each tree. Each leaf node corresponds to a score, and finally only the score corresponding to each tree needs to be added to the predicted value of the sample. The CNN model contains four layers. The first layer is the input layer, which aims to embed the word vector into a low-dimensional vector. The second layer network is the convolution and pooling layer, which uses multiple convolution kernels to carry out convolution operations on the previous layer network. Three kinds of convolution kernels are selected for convolution in this paper, respectively the convolution kernels covering 3, 4 or 5 words each time by sliding. The convolution operation function used in this paper is *conv2()*, the specific operational formula is shown in (1).

$$\begin{cases} V = conv2(W, X) + b \\ Y = \varphi(V) \end{cases} \quad (1)$$

where V is the convolution result, W is the convolution kernel matrix, X is input matrix, and b is the bias, φ(V) is the activation function, where the activation function uses the most commonly used linear rectification ReLU funtion, i.e. Y=max(0,V). Then the Max-pool layer is used to reduce the spatial operation of its long and high direction, so as to get a long vector. The dropout layer is used to normalize the convolution neural network. Dropout [22] method is the most popular regular convolution neural network method at present. The output layer uses the softmax function, as in (2), to satisfy the probability that the output text belongs to different categories after normalizing the input value. Assuming that there are n neurons in the output layer, the output of the k neuron is calculated, and the molecule is the exponential function of the input signal a_k , and the denominator is the sum of the exponential functions of all the input signals, in order to avoid excessive values in the operation of the exponential function. Therefore, the constant C, which does not change the result of the operation, is added to prevent the overflow of the operation.

$$\mathrm{y_k} = \frac{\exp(a_k + C)}{\sum_{i=1}^{n} \exp(a_i + C)} \quad (2)$$

The loss function of classification problem usually uses cross entropy loss function as in (3), to calculate the cross entropy loss value of each class based on the built-in cross entropy loss function module of TensorFlow. The cross entropy loss function of m groups of samples trained by CNN network model, and the actual output value of softmax function is represented byy_k. $x^{(k)}$and $y^{(k)}$ are the data and its corresponding label of the k group.

$$\mathrm{Loss} = -\frac{1}{m}\sum_{k=1}^{m} y^{(k)} \log(y_k) + (1 - y^{(k)}) \log(1 - y_k) \quad (3)$$

LSTM model is a special recurrent neural network (RNN), which is better than the traditional RNNs, because of it solves the problem of gradient disappearance and gradient explosion during long sequence training, as the XSS description text. The former is well-suited to learn from experience to classify, process and predict time series when there are very long time lags of unknown size between important events. Because its good performance at text classification, we apply the LSTM algorithm to the XSS description text classification. In the follow experiment we use the BasicLSTMCell of TensorFlow to achieve the target task. As we know, CNNs tends to use simple convolutional kernels, such as a fixed window, which is difficult to determine the window size small window sizes may result in the loss of some critical information, whereas large windows result in an enormous parameter space, which could be difficult to train. Therefore, TextRCNN was proposed to learn more contextual information than conventional window-based neural networks and represent the semantic of texts more precisely for text classification. In the TextRCNN model, convolution layer is replaced by a bidirectional RNN, so formed a model combining RNN with pooled layer. Considering the above various situations, we conducted a comparative test using the above proposed models on the same XSS description data. To the best of our knowledge, it is the first time to use deep learning models to solve the XSS description text classification problem.

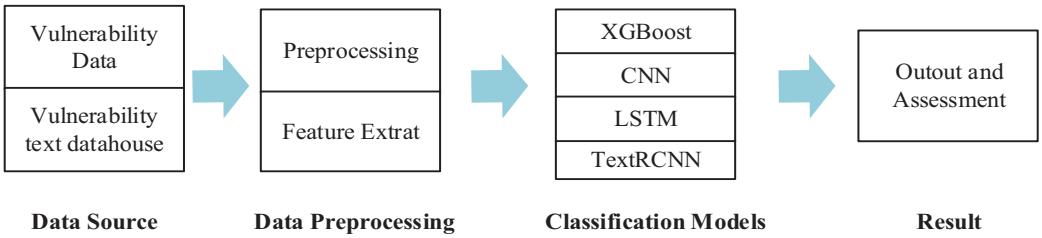

Fig. 1. Flow chart of vulnerability classification based on multiple classification algorithms.

III. EXPERIMENT RESULTS

A. Data description

The XSS vulnerability accounts for a large proportion in CNNVD vulnerability database. As a common web vulnerability, it can be exposed in large quantities every year and has been listed as one of the top ten most threatening vulnerabilities by the Open Web Application Security Project (OWASP). Therefore, this paper mainly studies the evaluation of XSS vulnerability data. The NVD vulnerability library contains all the exposed CVE vulnerabilities, which contain a large number of XSS vulnerability information. This paper studies this part of XSS vulnerability data. This article focuses on vulnerability information, which consists of three parts: vulnerability number, vulnerability description summary, and threat rating, as shown in Table I. Based on threat rating score, the **vulnerability risk levels** can be divided into 3 levels : low, medium, high. In this paper, we mainly use vulnerability to describe the text information and threat degree of the summary part of the data, and use this to train the models, and finally obtain the optimal parameters suitable for XSS vulnerability text classification.

TABLE I EXAMPLE OF XSS VULNERABILITY IN NVD

NO.	Briefly description of vulnerability	Threat rating score
CVE-2019-5727	Splunk Web in Splunk Enterprise 6.5.x before 6.5.5, 6.4.x before 6.4.9, 6.3.x before 6.3.12, 6.2.x before 6.2.14, 6.1.x before 6.1.14, and 6.0.x before 6.0.15 and Splunk Light before 6.6.0 has Persistent XSS, aka SPL-138827.	V3:5.4 MEDIUM V2:3.5 LOW

B. Experiments design

In the process of using various models to train vulnerability text data, the bias and weight of the model are updated and optimized according to the training data [23]. In addition, there are many hyper-parameter that need to be set, such as the ratio of dropout, the number of neurons in each layer, the size of batch, the learning rate or weight attenuation when the parameters are updated [24]. If these hyper-parameter do not select the right values, the performance of the model will be very poor. In this paper, we select four kinds of dropout ratios for comparison in the CNN model, and use the setting of the value of dropout keep probable to determine the dropout ratio, which is 0.4, 0.6, 0.8 and 1, respectively. For other models, the hyper-parameters are tuned to get the best accuracy. After randomly scrambling the data, 10% of the data is used as test data set. The remaining 90% of the data is used as training data set. In order to study the influence of amount of data on the training effect, 20%, 40%, 60%, 80%, 100% of training data set are used to train the models. Training time is recorded to evaluate the time cost of the models. For the all the deep learning algorithms，they are trained with dual-core CPU on Windows platform based on TensorFlow.

IV. RESULTS AND DISCUSSES

Table II calculates the prediction accuracy and Loss value of the corresponding model under different dropout keep probable ratio values. It can be seen that when the model is less than 0.8, the accuracy of the model increases with the increase of the value. When its value is greater than 0.8, the prediction accuracy of CNN model begins to decrease until no dropout. That is to say, the dropout keep probable value is 1. It can be seen that when the dropout keep probable ratio is 0.8 in a certain range, the classification accuracy of XSS vulnerability text data set by CNN model is higher. This also means that when using the model for classification and prediction, a certain dropout ratio is needed to avoid the occurrence of overfitting phenomenon. From the above analysis, it can be seen that when the other parameters remain unchanged, 0.8 is the most suitable dropout keep probable ratio. No matter what the value of dropout selection parameters, the accuracy of the model for the classification

of vulnerable text is always between 90.89% and 92.04%, showing a good classification accuracy.

TABLE II
ACCURACY AND LOSS VALUES CHANGE WITH DROPOUT KEEP PROBABLE RATIO

Dropout Keep Ratio		0.4	0.6	0.8	1.0
Parameters	Accuracy	91.0125	90.8987	92.0364	91.4676
	Loss	0.44694	0.35276	0.27512	0.23663

For the XGBoost algorithm, feature extraction is the first step to achieve data dimensionality reduction and keep more information of original data. The word frequency of each word in the text is counted, and the high frequency vocabulary is selected as the feature to classify the vulnerable text data and their respective classification results are obtained. The deep learning models choose word2vec as the way to complete word embedding. Table III shows the accuracy of using multiple classification models to classify vulnerable text information. As can be seen, the mean performance of TextRCNN is the highest in comparison to all other methods, but it does not show a big advantage compared to LSTM methods. In terms of loss value, LSTM is better than the other methods. However, the TextRCNN method and LSTM method are cost more training time than traditional CNN method and XGBoost method obviously. The former two ways take 30min01s and 22min59s respectively, compared to the latter two methods, which only takes 4m19s and 1min59s.

TABLE III
COMPARISON OF PREDICTION ACCURACY BETWEEN DIFFERENT CLASSIFICATION METHODS

Methods	XGBoost	CNN	LSTM	TextRCNN
Accuracy (%)	87.30	92.04	93.73	**93.95**
Loss	0.391	0.275	**0.056**	0.170
Time	1m59s	4m19s	22m59s	30m01s

In Table IV we compared across all 4 models all models are trained with 5 kinds of training data size, in order to find the influence of training data size on the performance of methods. The performance gain show that the accuracy of models increase with using more training data. However, when we used the XGBoost and LSTM models, there is a phenomenon that did not meet this trend, which use 80% of original training data size. That is maybe because of the training data is not ideal. From the experimental results, the classification of vulnerability text based on deep learning model has a good accuracy, which can effectively support the evaluation of vulnerability threat degree in XSS. Traditional machine learning method is not suitable to the target task.

TABLE IV
COMPARISON OF PREDICTION ACCURACY OF METHODS WITH VARIOUS TRAINING DATA SIZE

Methods	Training Data Size				
	20%	*40%*	*60%*	*80%*	*100%*
XGBoost	80.84	83.23	84.82	84.65	87.30
CNN	88.79	86.46	89.82	90.08	92.04
LSTM	90.7	91.0	92.3	91.3	93.73
TextRCNN	91.60	91.86	92.06	93.10	93.95

V. CONCLUSION

Text detection vulnerability threat based on XSS vulnerability description is a typical text categorization problem. The accuracy of traditional categorization methods is unsatisfactory, resulting in a large number of manual participation. In this paper, 4 kinds of vulnerability text categorization and evaluation methods are used to solve this this problems. The methods are trained by XSS vulnerability text description data provided by NVD, and uses CNN, LSTM and TextRCNN models to categorize it. Compared with the traditional text mining and machine learning methods (XGBoost), the deep learning methods not only avoid the heavy feature engineering, but also improve the classification accuracy and achieves good classification results. TextRCNN model performs best among three deep learning models, the test accuracy of which reach 93.95%. What's more, the LSTM also have good performance with smaller loss value and shorter time. All of these can play an important role in vulnerability threat assessment.

VI. ACKNOWLEDGE

This work was partially supported by National Natural Science Foundation of China (Grant No. 61703416), the Natural Science Foundation of Hunan Province, China (Grant No.2018JJ3614), the Research and Innovation Project for Postgraduate of Hunan Province (Grant No.CX2018B023).

REFERENCES

[1] Y. Xiang, L. Wang, and N. Liu, “Coordinated attacks on electric power systems in a cyber-physical environment,” Electric Power Systems Research,

vol. 149, pp. 156–168, 2017.

[2] L. Wang, M. Zhang, S. Jajodia, A. Singhal, and M. Albanese, “Modeling network diversity for evaluating the robustness of networks against zeroday attacks,” in European Symposium on Research in Computer Security. Springer, 2014, pp. 494–511.

[3] M. Abomhara et al., “Cyber security and the internet of things: vulnerabilities, threats, intruders and attacks,” Journal of Cyber Security and Mobility, vol. 4, no. 1, pp. 65–88, 2015.

[4] B. Kordy, L. Pietre-Cambac ` ed ´ es, and P. Schweitzer, “Dag-based attack and defense modeling: Dont miss the forest for the attack trees,” Computer science review, vol. 13, pp. 1–38, 2014.

[5] Q.-q. WU, L.-h. WEI, Z.-q. LIANG, Z.-w. YU, C. Min, Z.-h. CHEN, and J.-j. TAN, “Patching power system software vulnerability using cnnvd,” DEStech Transactions on Computer Science and Engineering, no. ccme, 2018.

[6] Z. Li, D. Zou, S. Xu, X. Ou, H. Jin, S. Wang, Z. Deng, and Y. Zhong, “Vuldeepecker: A deep learning-based system for vulnerability detection,” arXiv preprint arXiv:1801.01681, 2018.

[7] P. Mell, K. Scarfone, and S. Romanosky, “Common vulnerability scoring system,” IEEE Security & Privacy, vol. 4, no. 6, pp. 85–89, 2006.

[8] B. Shuai, H. Li, M. Li, Q. Zhang, and C. Tang, “Automatic classification for vulnerability based on machine learning,” in 2013 IEEE International Conference on Information and Automation (ICIA). IEEE, 2013, pp. 312–318.

[9] Y. Yamamoto, D. Miyamoto, and M. Nakayama, “Text-mining approach for estimating vulnerability score,” in 2015 4th International Workshop on Building Analysis Datasets and Gathering Experience Returns for Security (BADGERS). IEEE, 2015, pp. 67–73.

[10] S. M. Ghaffarian and H. R. Shahriari, “Software vulnerability analysis and discovery using machine-learning and data-mining techniques: A survey,” ACM Computing Surveys (CSUR), vol. 50, no. 4, p. 56, 2017.

[11] D. Toloudis, G. Spanos, and L. Angelis, “Associating the severity of vulnerabilities with their description,” in International Conference on Advanced Information Systems Engineering. Springer, 2016, pp. 231– 242.

[12] X. Zhan, T. Xiang, and H. Chen, “The application of weighted entropy theory in vulnerability assessment and on-line reconfiguration implementation of microgrids,” Entropy, vol. 16, no. 2, pp. 1070–1088, 2014.

[13] J. A. Wang and M. Guo, “Vulnerability categorization using Bayesian networks,” p. 29, 2010.

[14] P. Wang, Y. Zhou, B. Sun, W. Zhang, “Intelligent prediction of vulnerability severity level based on text mining and XGBoost” The Eleventh International Conference on Advanced Computational Intelligence, June 7-9,2019,Guilin, China.

[15] Y. Kim, “Convolutional neural networks for sentence classification,” empirical methods in natural language processing, pp. 1746–1751, 2014.

[16] Zhou P , Qi Z , Zheng S , et al. Text Classification Improved by Integrating Bidirectional LSTM with Two-dimensional Max Pooling[J]. 2016.

[17] Lai, S., Xu, L., Liu, K., & Zhao, J. Recurrent Convolutional Neural Networks for Text Classification. In AAAI (Vol. 333, pp. 2267-2273). January, 2015.

[18] L.-Y. Xia, Q.-Y. Wang, Z. Cao, and Y. Liang, “Descriptor selection improvements for quantitative structure-activity relationships,” International Journal of Neural Systems, 2019.

[19] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Dropout: a simple way to prevent neural networks from overfitting,” Journal of Machine Learning Research, vol. 15, no. 1, pp. 1929–1958, 2014.

[20] T. Hsien-De Huang and H.-Y. Kao, “R2-d2: Color-inspired convolutional neural network (cnn)-based android malware detections,” in 2018 IEEE International Conference on Big Data (Big Data). IEEE, 2018, pp. 2633–2642.

[21] X. Zhang and D. Wu, “On the vulnerability of cnn classifiers in eegbased bcis,” IEEE Transactions on Neural Systems and Rehabilitation Engineering, 2019.

[22] Y. Shuhan, X. Yang, and E. Shijia, “Text big data content understanding and development trend based on feature learning,” Big Data Research, vol. 1, no. 3, p. 2015030, 2015.

[23] H. Shin, H. R. Roth, M. Gao, L. Lu, Z. Xu, I. Nogues, J. Yao, D. J. Mollura, and R. M. Summers, “Deep convolutional neural networks for computer-aided detection: Cnn architectures, dataset characteristics and transfer learning,” IEEE Transactions on Medical Imaging, vol. 35, no. 5, pp. 1285–1298, 2016.

[24] H. Xie, D. Yang, N. Sun, Z. Chen, and Y. Zhang, “Automated pulmonary nodule detection in ct images using deep convolutional neural networks,” Pattern Recognition, vol. 85, pp. 109–119, 2019.