XSS Detection Technology Based on LSTM-Attention

Li Lei
Hubei Key Laboratory of Intelligent Robot, School of Computer Science and Engineering
Wuhan Institute of Technology
Wuhan 430070, China
e-mail: leidashuai520@163.com

Chengwan He
School of Computer Science and Engineering
Wuhan Institute of Technology
Wuhan 430205, China
e-mail: hechengwan@hotmail.com

Ming Chen
School of Computer Science and Engineering
Wuhan Institute of Technology
Wuhan 430205, China
e-mail: 1039443823@qq.com

Duojiao Li
School of Computer Science and Engineering
Wuhan Institute of Technology
Wuhan 430205, China
e-mail: 1349018183@qq.com

Abstract—Cross-site scripting (XSS) is one of the main threats of Web applications, which has great harm. How to effectively detect and defend against XSS attacks has become more and more important. Due to the malicious obfuscation of attack codes and the gradual increase in number, the traditional XSS detection methods have some defects such as poor recognition of malicious attack codes, inadequate feature extraction and low efficiency. Therefore, we present a novel approach to detect XSS attacks based on the attention mechanism of Long Short-Term Memory (LSTM) recurrent neural network. First of all, the data need to be preprocessed, we used decoding technology to restore the XSS codes to the unencoded state for improving the readability of the code, then we used word2vec to extract XSS payload features and map them to feature vectors. And then, we improved the LSTM model by adding attention mechanism, the LSTM-Attention detection model was designed to train and test the data. We used the ability of LSTM model to extract context-related features for deep learning, the added attention mechanism made the model extract more effective features. Finally, we used the classifier to classify the abstract features. Experimental results show that the proposed XSS detection model based on LSTM-Attention achieves a precision rate of 99.3% and a recall rate of 98.2% in the actually collected dataset. Compared with traditional machine learning methods and other deep learning methods, this method can more effectively identify XSS attacks.

Keywords-cross-site scripting (XSS); LSTM; attention mechanism; word2vec

I. INTRODUCTION

With the rapid development of Internet technology, the Internet has become an important tool for people to exchange and share of information. However, while the network brings convenience to people's lives, it also brings risks to information security [1]. The issue of network security has attracted increasing attention. Cross-site scripting (Cross-Site Scripting, XSS) is a serious web vulnerability that ranks among the top three in the Top10 vulnerability of the Open Web Application Security Project Group. The hazards of XSS attacks mainly include stealing user cookies, session hijacking, phishing, and denial of service attacks, resulting in serious consequences such as website inaccessibility, user's privacy information leakage, and property loss.

The XSS attack is due to the attacker injecting malicious HTML code or malicious JavaScript code in the client. Once the user visits the tampered page, the browser does not detect and process the maliciously injected code, the malicious script will be executed to complete the attack. The types of XSS attacks can be divided into three categories: reflective XSS, stored-XSS, and DOM-Based XSS [2]. Reflective XSS is the most common. Attackers construct a URL containing malicious code and then trick users into clicking to access, then a one-time attack is implemented, so it is also called non-persistent XSS; Stored-XSS generally appears in interactions such as website messages, comments, blog, and malicious scripts are stored in the client or server database, so it is also called persistent XSS; DOM-Based XSS is an XSS based on the document object model. Attackers use front-end scripts to modify pages and inject malicious code into DOM objects or attributes to complete XSS attacks in the client browser.

In order to deal with the above three different types of XSS attacks, traditional rule-based filtering detection methods are widely used, but due to the complex and variable forms of XSS attacks, the traditional detection methods for XSS attacks are inefficient, especially for the malicious XSS code after the obfuscation process is more difficult to detect. XSS detection problems have also tried to use machine learning methods to solve. How to improve the accuracy of detection has become a huge challenge. Therefore, in this paper, we propose a method for detecting XSS attacks based on the attention mechanism of Long Short-Term Memory (LSTM) recurrent neural network. This method applies the attention mechanism to the LSTM network to form a new LSTM-Attention network detection model, and the experiment proves that the addition of

978-1-7281-8624-5/20/$31.00 ©2020 IEEE

Attention plays a role in the classification of the detection model. The experimental results show that the method in this paper has higher precision rate and recall rate in XSS detection.

II. Related Work

The traditional XSS detection technology is mainly divided into static analysis method and dynamic analysis method [3]. The static detection method refers to the analysis of the source code of the web application, and the possible security vulnerabilities in the web application can be detected by viewing the code. Jovanovic N et al. [4] used static source code analysis to determine whether XSS attacks exist by detecting whether stain data is processed before output and designed a XSS vulnerability detection tool called Pixy. The dynamic detection method refers to analyzing the behavior of the Web application to determine whether there is an XSS vulnerability. Gupta S et al. [5] used dynamic analysis method to design a server-side framework called XSS-SAFE, which injects attack vector purification into JavaScript source code to prevent attack vectors from being sent to the server.

Choi J et al. [6] proposed a method that uses Support Vector Machine (SVM) as a machine learning classifier to effectively extract n-gram features from malicious code to classify executable files as malicious or normal. Wang R et al. [7] constructed an alternative decision tree (ADTree) classifier with better detection accuracy than other decision trees and an adaptive boosting (Adaboost) classifier combining multiple weak classifiers for XSS detection, classification effect is directly affected by the quality of extracted features. Mokbal F et al. [8] proposed a multi-layer perceptron (MLP) scheme based on robust artificial neural network integrated with dynamic feature extractor for XSS attack detection.

Li L et al. [9] used the DQN algorithm to transform the original malicious XSS parameters to enhance learning and created a more efficient automatic XSS detection tool. Fang Y et al. [10] used the LSTM network to construct the XSS detection model. The experimental results show that the LSTM-based detection method can be effectively applied to XSS detection, but it is not accurate enough for malicious confusion XSS detection. Pan J et al. [11] proposed a detecting framework employing hybrid analysis to detect DOM-based cross-site scripting. But it is limited to identifying other types of XSS attacks.

III. LSTM and Attention Mechanism

A. Long Short-Term Memory (LSTM)

LSTM (Long Short-Term Memory) is a variant of RNN (Recurrent Neural Network) [12]. RNN can keep memory based on historical information and deal with certain short-term dependence problems, so that it can use the distance feature to predict the current output. In CNN, the signal of each layer of neurons can only be propagated to the upper layer, and the processing of the samples is independent at each moment, so it is called feed-forward neural networks. But in RNN, the output of the neuron can directly affect itself in the next time period and can better deal with problems that are highly related to time series. However, due to the defect of the gradient explosion and gradient disappearance problems, RNN can't solve the long-term dependence problem. Therefore, we used LSTM for experiments in this paper. LSTM is controlled by introducing a cell state and setting three gates: the forget gate (f_t), the input gate (i_t), and the output gate (o_t). The cell status of LSTM will help decide to retain relevant important information and forget the unimportant information. The structure of LSTM model is shown in Figure 1.

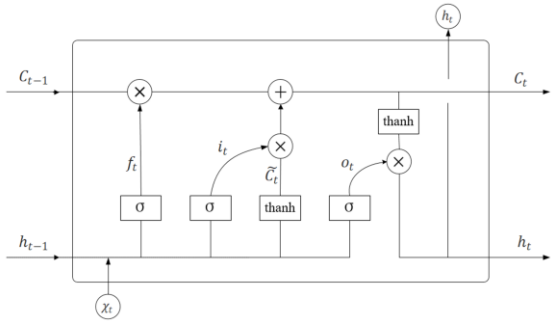

Figure 1. Structure of LSTM model.

The first step in LSTM is to decide what information needs to be discarded in the cell state. The forget gate (f_t) is calculated by the hidden state (h_{t-1}) at the last moment and the current input (x_t). The formula is as follows:

$$f_t = \sigma\left(W_f \cdot \left[h_{t-1}, x_t\right] + b_f\right) \tag{1}$$

The next step is to decide what new information to add to the cell state. First, we need to decide which information to update. The input gate (i_t) is calculated by h_{t-1} and x_t . The formula is as follows:

$$i_t = \sigma\left(W_i \cdot \left[h_{t-1}, \chi_t\right] + b_i\right) \tag{2}$$

Then the temporary cell state ($\tilde{C}_t$) that may be updated to the cell information is calculated by h_{t-1} and x_t . The formula is as follows:

$$\tilde{C}_t = \tanh\left(W_c \cdot \left[h_{t-1}, x_t\right] + b_C\right) \tag{3}$$

The current cell state (C_t) is calculated by f_t , i_t and the cell state (C_{t-1}) at the last moment. The formula is as follows:

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \tag{4}$$

After updating the cell state, the output gate (o_t) is calculated by h_{t-1} and x_t .the formula is as follows:

$$o_t = \sigma\left(W_o \cdot \left[h_{t-1}, x_t\right] + b_o\right) \tag{5}$$

Finally, the hidden state (h_t) at the current moment is calculated by o_t and C_t . The formula is as follows:

$$h_t = o_t * \tanh(C_t) \tag{6}$$

In the formula, W_f , W_i , W_c , W_o are weights matrix; b_f , b_i , b_C , b_o are the corresponding bias; σ: sigmoid activation function; tanh: activation function.

B. Attention Mechanism

The attention mechanism theory [13] was first proposed in the field of images, and the motivation for research was inspired by the human attention mechanism. In the NLP field, this mechanism was first applied to tasks related to machine translation. In this paper, we add the attention mechanism to the LSTM model and uses the LSTM-Attention model for the detection of XSS attacks.

Attention is added in the hidden layer of the LSTM output, and the hidden state (h_t) is converted to the target attention weight (u_t) through the fully connected layer. The formula is as follows:

$$u_t = \tanh(h_t) \tag{7}$$

Probability of attention weights. The attention probability distribution values a_1 , a_2 , a_3 , …, a_t are generated by the softmax function. The formula is as follows:

$$a_t = \frac{\exp(u_t)}{\sum_{t=1}^{m} \exp(u_t)} \tag{8}$$

Distribution of attention weights. The context vector v is calculated based on a_t and h_t . The formula is as follows:

$$v = \sum_{t=1}^{m} a_t \cdot h_t \tag{9}$$

IV. XSS Detection Method Based on LSTM-Attention

A. Overview

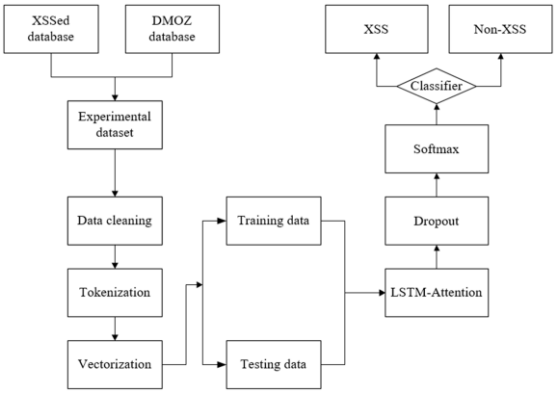

Figure 2. Structure of the detection system.

The XSS detection method proceeds as follows. Firstly, we preprocess the experimental data, which includes data cleaning and tokenization. Then we use word2vec to extract XSS payload features and map them to feature vectors for vectorization. Finally, we use LSTM-Attention detection model to train and test the data for distinguishing XSS samples from normal samples. The detection steps are shown in Figure 2.

B. Data Preprocessing

1) Data cleaning

Attackers usually use encoding techniques to confuse XSS codes to evade traditional filters or validation mechanisms, mainly including HTML encoding and URL encoding. Therefore, we clean the original data and try to convert data into the original form by a decoder. For example, a confused XSS payload by URL encoding "%3Cscript%3Ealert(%2F123%2F)%3C%2fscript%3E" is restored to "<script>alert(/123/)</script>".

2) Tokenization

In order to reduce the interference of useless information and reduce the vectorized data dimension. Firstly, we replace the number in the code with "0". Then, we replace the various uniform resource locators (URL) in the input data with "http: // u". In order to fully retain the XSS text information, we design a series of custom regular expressions to segment the normalized results based on the features of the XSS code in accordance with the scripting language. The category of tokenization is shown in Table I.

TABLE I. Tokenization

Category	example
Start Label	<script>, <body>, <title>, <h1>, <a>, etc
End Label	</script>, </body>, </title>, </h1>, , etc
Function Name	alert(, prompt(, document.write(, etc
Script URL	javascript:, vbscript:, etc
Windows Event	onload=, onerror=, onblur=, etc
Others	>,), &, #, etc

3) Vectorization

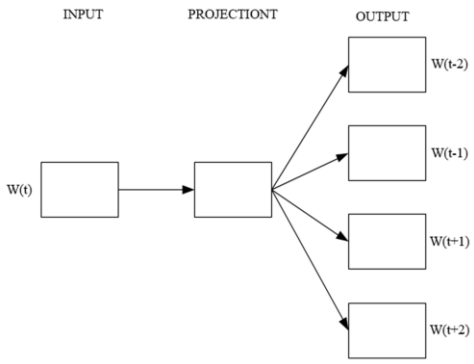

Figure 3. Structure of Skip-gram model.

Word2vec [14] is a deep learning tool released by Google in 2013, including CBOW (Continuous Bag-of-Words) model and Skip-gram model. The CBOW model predicts the current word based on the context. Contrary to CBOW, the Skip-gram model predicts the context based on the current word [15]. The model includes three layers: input layer, projection layer and output layer (softmax layer). The input $W_{(t)}$ is the central word of the corpus which as word target, and the output layer is normalized by softmax to achieve the probability of the context distribution of the $W_{(t)}$. The typical Skip-gram model architecture is shown in Figure 3.

C. Detection Model Based on LSTM-Attention

In order to improve the precision rate of XSS attacks detection, we build a network detection model based on LSTM-Attention by studying the existing neural network model. The structure is shown in Figure 4.

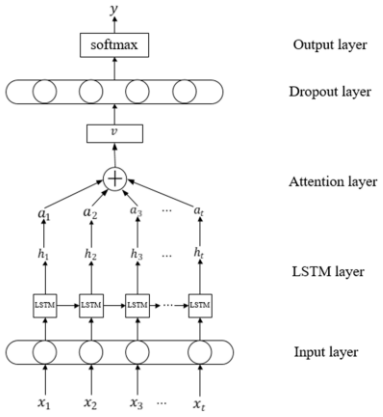

Figure 4. Detection model based on LSTM-Attention.

1) Input layer

For example, there is an experimental data sample. The content of data is first represented by vectors x_1, x_2, x_3, ..., x_t after tokenization. Because the data vector matrix embedding training has been completed in advance in the data preprocessing process, the embedding layer is no longer used in this part.

2) LSTM layer

This layer can pass each input to the LSTM unit to obtain the output of the corresponding hidden layer h_1, h_2, h_3, ..., h_t. It uses LSTM to solve the long-term dependency problem and automatically learns the abstract features of XSS attacks.

3) Attention layer

In this layer, we apply attention mechanism for the hidden layer to obtain the attention probability distribution values a_1, a_2, a_3, …, a_t.

4) Dropout layer

In this layer, we use dropout technology to prevent overfitting.

5) Output layer

The classification result y is output through the softmax classifier.

V. Experiment and Discussion

A. Experimental Dataset

In the experiment, a total of 32,168 standard data are obtained, and normal samples from the DMOZ database are used as negative samples, a total of 78,652 normal samples are obtained. Finally, all the data is randomly divided into 70% training data and 30% testing data, which are used for neural network training and testing. The experimental dataset is shown in Table II.

TABLE II. Experimental Dataset Settings

Category	Training data	Testing data	Total
XSS	22518	9650	32168
Normal	55056	23596	78652

B. Experimental Environment

This paper used the TensorFlow framework, the experiment was conducted on an Intel(R) Core(TM) i7-8550U CPU @ 1.80GHz, 16G RAM, and NVIDIA GeForce GTX 1050 Max-Q.

C. Results and Discussion

In order to evaluate the performance of the detection method in this paper, the proposed method is evaluated from three indicators: Precision, Recall, and F1 score. Their calculations are as follows:

$$\text{Precision} = \frac{TP}{TP + FP} \tag{10}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{11}$$

$$\text{Fl} = \frac{2 * \text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} \tag{12}$$

In the formula, *TP* represents the number of positive samples predicted and actually positive samples, *FP* represents the number of predicted positive samples but actually negative samples, and *FN* represents the number of predicted negative samples but actually positive samples.

In order to objectively evaluate and analyze the validity and advantages of the detection model, we designed two comparative experiments.

1) Comparison of machine learning methods

We compared our XSS attacks approach with the traditional machine learning methods. Traditional machine learning methods (ADTree and AdaBoost) are used to detect XSS by Wang R [7], and their experimental dataset and the dataset in this paper come from the same database. At the same time, in order to increase the contrast, another set of SVM algorithm used in XSS detection was added for comparative experiments. Results are shown in Table III.

TABLE III. EXPERIMENTAL COMPARISON RESULTS OF THE PROPOSED METHOD AND THE MACHINE LEARNING METHODS

Classifier	Precision / %	Recall / %	F1 / %
ADTree	93.8	93.6	93.6
AdaBoost	94.1	93.9	93.9
SVM	96.5	95.3	95.9
LSTM-Attention	99.3	98.2	98.5

The precision rates of ADTree and AdaBoost are 93.8% and 94.1%, and the precision rate of SVM is 96.5%. The precision rate of the detection method proposed in this paper is as high as 99.3%, the recall rate is 98.2%, and the F1 score is 98.5%. Our proposed method is obviously better than traditional machine learning methods.

2) Comparison of deep learning methods

We compared our XSS attacks approach with the other deep learning methods. Results are shown in Table IV.

TABLE IV. EXPERIMENTAL COMPARISON RESULTS OF THE PROPOSED METHOD AND OTHER DEEP LEARNING METHODS

Classifier	Precision / %	Recall / %	F1 / %
RNN	97.2	90.5	93.7
GRU	95.6	95.8	95.7
LSTM	96.4	96.3	96.4
LSTM-Attention	99.3	98.2	98.5

The results show that RNN, GRU and LSTM have relatively good performance on XSS detection, but compared with the LSTM-Attention method proposed in this paper, both in precision rate, recall rate and F1 score are lacking. After adding the attention mechanism to the LSTM network, the precision rate increased by about 2.9%, the recall rate increased by about 1.9%, and the F1 score increased by about 2.1%. This is because the Attention model calculates the attention probability of each word in the input text and assigns different attention weights to distinguish the importance of the text information. The results show that Attention plays a certain role in improving the classification performance.

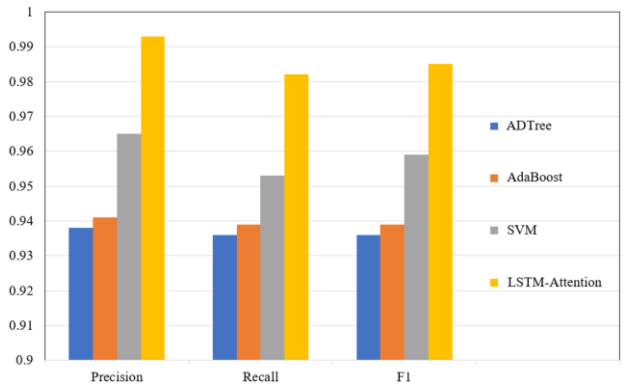

Figure 5. Comparison of the proposed method and the traditional machine learning methods.

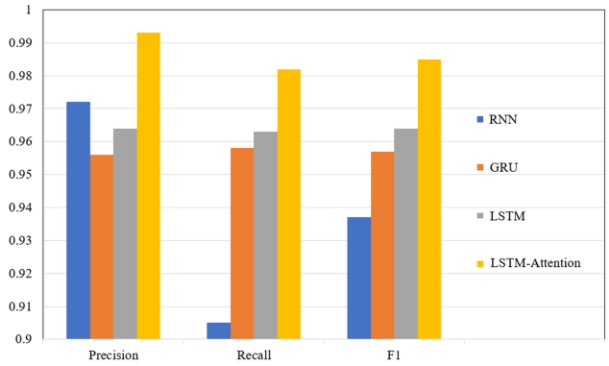

Figure 6. Comparison of the proposed method and other deep learning methods.

In order to more intuitively compare the performance of our proposed method with traditional machine learning methods and other similar deep learning methods in terms of Precision, Recall, and F1 score, according to the above experimental results, Figure 5 and Figure 6 are drawn.

VI. CONCLUSION AND FUTURE WORK

Cross-site scripting detection is an important part of Web security. In this paper, we proposed a method for detecting XSS attacks based on LSTM-Attention. First, obfuscated data was restored through decoding technology, and we used word2vec to extract the semantic information of XSS payloads and mapped them to the feature vector as the input of the neural network. Secondly, the abstract features of XSS attacks were extracted using LSTM recurrent neural network. Then, we added attention mechanism to improve the classification effect. Finally, we used the classifier to output the classification results to complete the XSS detection. The experimental results prove the effectiveness of the detection method based on the LSTM-Attention model proposed in this paper. Compared with traditional machine learning methods and other deep learning methods, our method has higher precision rate and recall rate. However, due to the complex and variable form of XSS, we will collect more other types of XSS data to improve the generalization ability of the detection model, we will continue to study how to optimize the network model to improve the accuracy and performance of classification.

ACKNOWLEDGMENT

This work was supported by Hubei key Laboratory of Intelligent Robot Hubei Technology Innovation Project (2019AAA045).

REFERENCES

[1] Li X , Xue Y . A survey on server-side approaches to securing web applications[J]. Acm Computing Surveys, 2014, 46(4):1-29.

[2] Rocha T S, Souto E. ETSSDetector: a tool to automatically detect Cross-Site Scripting vulnerabilities[C]//2014 IEEE 13th International Symposium on Network Computing and Applications. IEEE, 2014: 306-309.

[3] Hydara I, Sultan A B , Zulzalil H , et al. Current state of research on cross-site scripting (XSS) - A systematic literature review[J]. Information and software technology, 2015, 58(feb.):170-186.

[4] Jovanovic N . Pixy : A static analysis tool for detecting web application vulnerabilities[C]// Proc. of 2006 IEEE Symposium on Security and Privacy. IEEE, 2006.

[5] Gupta S, Gupta B B. XSS-SAFE: a server-side approach to detect and mitigate cross-site scripting (XSS) attacks in JavaScript code[J]. Arabian Journal for Science and Engineering, 2016, 41(3): 897-920.

[6] Choi J , Kim H , Choi C , et al. Efficient Malicious Code Detection Using N-Gram Analysis and SVM[C]// The 14th International Conference on Network-Based Information Systems, NBiS 2011, Tirana, Albania, September 7-9, 2011. IEEE, 2011.

[7] R. Wang, X. Jia, Q. Li and S. Zhang, "Machine Learning Based Cross-Site Scripting Detection in Online Social Network," 2014 IEEE Intl Conf on High Performance Computing and Communications, 2014 IEEE 6th Intl Symp on Cyberspace Safety and Security, 2014 IEEE 11th Intl Conf on Embedded Software and Syst (HPCC,CSS,ICESS), Paris, 2014, pp. 823-826.

[8] Mokbal F M M, Dan W, Imran A, et al. MLPXSS: An Integrated XSS-Based Attack Detection Scheme in Web Applications Using Multilayer Perceptron Technique[J]. IEEE Access, 2019, 7: 100567-100580.

[9] Li L, Wei L. Automatic XSS Detection and Automatic Anti-Anti-Virus Payload Generation[C]//2019 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC). IEEE, 2019: 71-76.

[10] Fang Y, Li Y, Liu L, et al. DeepXSS: Cross Site Scripting Detection Based on Deep Learning[C]//Proceedings of the 2018 International Conference on Computing and Artificial Intelligence, Chengdu, Mar 12-14, 2018. New York : ACM, 2018: 47-51.

[11] Pan J , Mao X . Detecting DOM-Sourced Cross-Site Scripting in Browser Extensions[C]// IEEE International Conference on Software Maintenance & Evolution. IEEE, 2017.

[12] Yu Y, Si X, Hu C, et al. A review of recurrent neural networks: LSTM cells and network architectures[J]. Neural computation, 2019, 31(7): 1235-1270.

[13] Mnih V, Heess N, Graves A. Recurrent models of visual attention[C]//Advances in neural information processing systems. 2014: 2204-2212.

[14] Distributed Representations of Words and Phrases and their, Compositionality[J]. Advances in Neural Information Processing Systems, 2013, 26:3111-3119.

[15] Mikolov T, Chen K, Corrado G, et al. Efficient estimation of word representations in vector space[J]. arXiv preprint arXiv:1301.3781, 2013.