A Vulnerability Detection Method for Internet Cross-site Scripting Based on Relationship Diagram Convolutional Networks

Zhida Guo[1,*], Xiaoli Li[1], Ran Hu[1], Dapeng Wang[2] and Weijie Song[2]

[1]*Huizhou Power Supply Bureau of Guangdong Power Grid Co., Ltd, Huizhou 516000, China*
[2]*Zhuhai Power Supply Bureau of Guangdong Power Grid Co., Ltd, Zhuhai 519000, China*
E-mail: guozhida773@163.com
[*]*Corresponding Author*

Received 03 September 2024; Accepted 03 March 2025

Abstract

The aim of this research is to quickly detect cross-site scripting (XSS) attacks on the internet based on relationship diagram convolutional networks. Based on the principle and attack process of cross-site scripting attacks, domain knowledge is used to build an XSS ontology to conduct high-level modeling of cross-site scripting attacks, obtain data that can reflect XSS attacks, normalize these attack data, extract attack data word vectors, use them as the input of the relationship diagram convolution networks added to the attention mechanism, and learn attack feature word vectors. After further extracting node characteristics through convolution and pooling, all node characteristics are aggregated and fed into the fully connected neural network. XSS vulnerability detection results are obtained through classification

Journal of Web Engineering, Vol. 24_2, 243–266.
doi: 10.13052/jwe1540-9589.2424

of the activation function, and malicious domain name and malicious IP information are combined as supplementary rules to improve the effectiveness of the vulnerability detection in internet cross-site scripting based on the relationship graph convolution network. Experiments show that this method can accurately detect XSS vulnerabilities, provide comprehensive and accurate attack details, and its performance is better than that of the literature method, which is reflected in the higher accuracy, recall, accuracy and F1 value, and the leading area of the ROC curve. Its detection speed is extremely fast, only 0.03 s, and by combining malicious domain name and IP information, the detection efficiency is further improved, realizing rapid response and effectively maintaining Internet security.

Keywords: Relationship diagram, convolutional network, Internet, cross-site scripting, vulnerability detection, word vector.

1 Introduction

The internet has become increasingly complex and its scale is also constantly increasing. At the same time, the number of security vulnerabilities has been increasing year by year. Vulnerability refers to specific defects or omissions in the internet. Attackers can exploit these vulnerabilities for malicious operations, expose or modify sensitive information, disrupt or control computer systems, and pose a great threat to information security. Therefore, cross-site scripting vulnerability detection on the Internet is essential.

Nancy et al. proposed a vulnerability detection method for cross-site scripting in a wireless sensor network based on dynamic feature selection and fuzzy time decision tree classification [1]. This method used the dynamic recursive feature selection algorithm to select the optimal number of features from the dataset, and combined the decision tree algorithm and convolutional neural network to build an intelligent fuzzy time decision tree algorithm to achieve vulnerability detection of cross-site scripting in the network. But the detection results take a long time; Anjinappa et al. studied an unsupervised learning vulnerability detection method [2] using state-of-the-art manifold learning techniques. They identified network vulnerabilities in an unsupervised manner through unified manifold approximation and projection. The key idea was to preserve the inherent local connectivity structure in the unlabeled channel samples collected so that vulnerabilities from various service regions of the network could be detected. Although the detection time for vulnerabilities is shortened, the detection effect is not ideal. Modi

et al. proposed a vulnerability detection model based on the restful api [3], which implemented feature based Webshell script vulnerability detection for HTTP traffic. The accuracy of vulnerability detection through this method is high, but the maintenance amount of this method is huge; Qasem et al. studied the automatic vulnerability detection method in embedded devices and firmware based on survey and hierarchical classification [4], crawled the web page links of the specified depth of the whole site through web crawler technology and analyzed them, then constructed attack fuzzy test cases and selected more potential attack payloads from them, and finally analyzed the website response to complete vulnerability detection. This method greatly reduces the vulnerability detection time, but at the same time the false alarm rate of detection results is also high. Krishnaveni et al. proposed an intrusion detection method based on set classification and feature selection for a cloud computing network [5]. This method combines feature selection and classification with integrated technology to achieve efficient and accurate intrusion detection and has a high detection rate for cross-site scripting vulnerabilities, but also has a high false alarm rate.

The relationship diagram convolutional neural network model [6] can predict much exact information through the structured relationship information of local domains in the knowledge graph. Therefore, this paper proposes a vulnerability detection method for internet cross-site scripting based on relationship diagram convolutional networks. This method not only improves the performance of cross-site scripting vulnerability detection, but also shortens the detection time required.

2 Principles and Process of a Cross-site Scripting Attack

In order to avoid confusion with CSS in HTML language, XSS (cross-site scripting) is referred to for short. It is a security vulnerability [7] of web applications that attackers exploit (usually caused by the Internet's failure to properly filter the data entered by users) to steal user information by inserting malicious code into the page link visited by users. The attack process involves three aspects: the attacker, the victim and the web application with vulnerabilities. Only the victim will actually run the attacker's code. The Internet is just a carrier to launch the attack, which will not be affected in general. In addition, XSS attacks are often used to attack web forms whose input is not properly verified or cleaned up. If the web application does not strictly verify and clean up the data entered by the user, the attacker can take advantage of these vulnerabilities by submitting malicious code.

The XSS attack steps are as follows:

(1) The attacker selects a web site of interest, and normal users need to use their ID card to access the site. The site tracks authenticated users based on the user's cookie and session ID.
(2) The attacker searches for a page with an XSS vulnerability on the site, such as: http//trusted.org/account.jsp.
(3) The attacker carefully crafts a special link to this page and then posts the link to the potential victim's email or pop-up advertisement on the page.
(4) The attacker inserts malicious code into the link [8] and copies the user's cookie. For example: <img src=" http://trusted.org/account.jsp?ak=<script>document.location.replace ('http://evil.org/steal.jsp?'+document.cookie); </script>">.
(5) When the victim clicks on this link, the attacker obtains the user's cookie, allowing them to impersonate the user to access the site.

The existing XSS attacks are divided into three basic types based on their utilization, namely reflective/non-persistent XSS, stored/persistent XSS, and DOM-based XSS. Reflective XSS refers to the implementation of XSS attacks by attackers by enticing users to click on a URL link that carries malicious script parameters and points to a page that does not exist in the normal internet. Storage-based XSS refers to attackers transmitting malicious code to normal internet servers (such as messages, blogs, etc.) through some means, thereby subjecting users accessing the internet to XSS attacks. DOM-based XSS, also known as local XSS attacks, is usually included in reflective XSS attacks, which have similar principles to reflective XSS attacks and a relatively low probability of occurrence.

3 Vulnerability Detection in Internet Cross-site Scripting Based on Relationship Diagram Convolutional Networks

Figure 1 illustrates the process of XSS detection using the proposed method.

This method first constructs an ontology for high-level modeling of XSS attacks, obtains data that can reflect XSS attacks, normalizes these attack data, and extracts attack data word vectors. Then, these attack data word vectors are used as inputs to the relationship diagram convolutional network [9], and the attack feature word vectors are learned through the network. After further extraction of node features through convolution and pooling, all node features are aggregated and fed into the full connection neural network, and the final XSS vulnerability detection tag is obtained through classification

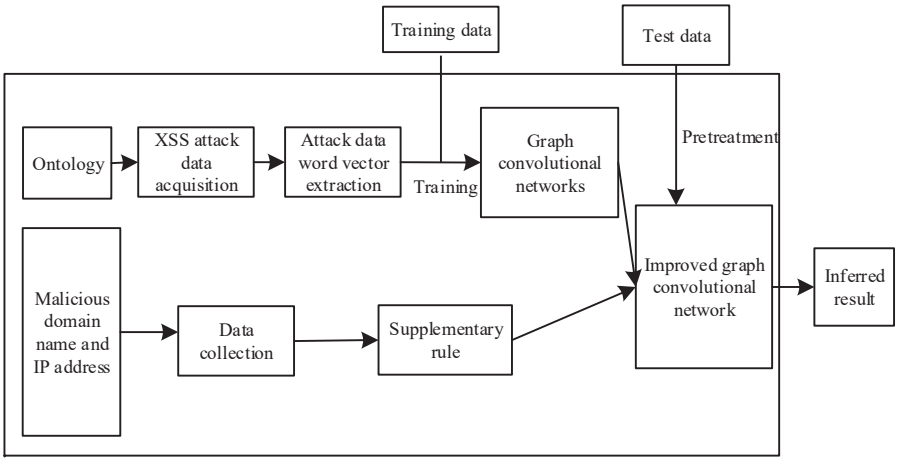


Figure 1 Framework of the model application process.

of the activation function. After the basic model is built, malicious domain names and malicious IP information are collected and crawled [10], and this open-source threat intelligence serves as a supplementary rule to improve the detection effect of the model. In practical applications, when the data to be judged is input into the model, the data will be judged by both the basic model and supplementary rules. The characteristic of the white box model for the relationship diagram convolutional network means that its results are well interpretable to users, and supplementary rules derived from malicious domain names and malicious IPs can timely discover hidden attack types. The combination of the model and rules significantly improves the detection ability of attacks, which can effectively help security personnel handle events.

In practical applications, relying solely on machine learning models for detection while ignoring threat intelligence can hide dangers for attackers' attacks. Cyber threat intelligence is an important supplement to traditional cybersecurity. The National Protective Security Authority of the UK categorizes threat intelligence into four categories: tactical, technical, operational, and strategic. This classification can effectively assist researchers in macro grasping threat intelligence, but in reality, this threat intelligence is mostly controlled by for-profit organizations and belongs to the internal assets of the group, which cannot be obtained by non-corresponding group members. For researchers, the available threat intelligence is mainly obtained from open-source websites. In the detection process of XSS attacks, this paper utilizes a malicious IP address library and a malicious domain name library.

A malicious IP address library is one of the most easily obtained threat intelligence for researchers, which is used in the browser's access blacklist [11]. When a client sends a request containing these addresses, the browser will block them. The malicious domain name library is mainly used to prevent botnet, phishing pages and malware. When a user visits a page address that is included in the malicious domain name list, it will alert the user for security reasons.

Open source threat intelligence can serve as supplementary rules to improve the model's effectiveness, with $C = \{C_1, \ldots, C_m\}$ being the set of supplementary rules, $X = \{X_1, \ldots, X_n\}$ being the set of features extracted based on ontology, $R = \{R_1, \ldots, R_N\}$ being the set of data records, $r = \{r_1, \ldots, r_N\}$ being the value extracted based on specific rules, and $T = \{T_1, \ldots, T_N\}$ being the predicted result of the corresponding records. The calculation equation for T_i is:

$$T_i = \begin{cases} 1, & if \max\{I_C(r_i), P(T_i = 1|X_1, \ldots, X_n)\} \geq \mu \\ 0, & if \max\{I_C(r_i), P(T_i = 1|X_1, \ldots, X_n)\} < \mu \end{cases} \tag{1}$$

In the equation, 0 represents that the current record is normal, 1 represents that the current record is an XSS attack, and the value of μ is 0.5; $I_C(r_i)$ is an indicator function. When the value extracted according to specific rules is detected by supplementary rules, the value of this indicator function is 1, otherwise it is 0; $P(T_i = 1|X_1, \ldots, X_n)$ represents the posterior probability calculated by the connection tree method after the corresponding evidence is given.

3.1 Construction of XSS Ontology Based on Domain Knowledge

Ontology is an important representation of domain knowledge, and the classic concept of ontology in the computer field is given by Gruber. Ontology is a formal description of concepts and relationships between entities, providing a generalization of specific domain knowledge. In this paper, ontology provides a knowledge base for subsequent modeling, which stores facts in the form of triples (subject, predicate, object). This knowledge base helps to grasp the core concepts and characteristics of XSS attacks in subsequent modeling. Using ontology modeling language directly to construct ontology is a huge work, so graphical tools are usually used to assist in the actual construction process. Graphical tools can enable ontology builders to focus on streamlining concepts and relationships between entities without the need to master underlying programming languages. This paper uses Protégé as an ontology

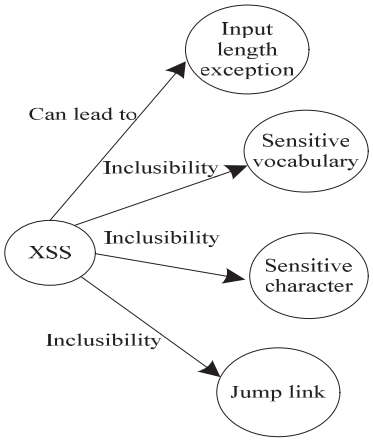


Figure 2 XSS high-level ontology.

modeling tool [12], which is a free open source ontology building framework. Using the web ontology language (OWL) can help researchers build ontology efficiently. The high-level ontology of the XSS attack constructed in this paper is shown in Figure 2.

For this study, we extracted XSS attack data [13]. Each word vector appears as a node in the relationship diagram convolutional network.

3.2 XSS Attack Data Normalization and Word Vector Extraction

3.2.1 XSS attack data normalization

Based on the XSS attack data obtained above, this paper establishes a preprocessing process for XSS attack feature data, as shown in Figure 3.

The XSS attack dataset obtained in this paper is not plaintext readable, but it is URL encoded data. To analyze the specific representation of the data, the unquote function is used to encode the dataset and obtain a clear and readable attack script. Considering the complex and diverse format of cross-site scripting statements, which is not conducive to word segmentation processing, without affecting the expression of script statements, the URL links and numbers within the cross-site scripting are standardized to the same symbol (links are unified as http://u; numbers are unified as 0). After standardization, word segmentation operations are performed. The word segmentation operation includes a simple word segmentation principle that can partition malicious script expressions [14], label representations, URL link representations, etc. in the dataset. Specifically: (1) The content contained in

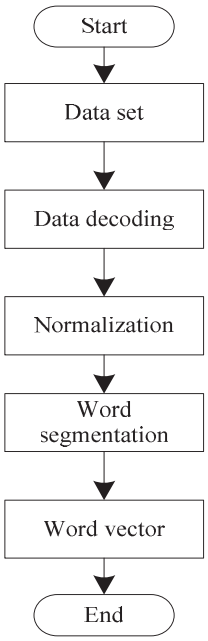


Figure 3 XSS attack feature data preprocessing process.

single quotation marks and double quotation marks; (2) URL link; (3) HTML tags; (4) parameter name; (5) function body.

3.2.2 Overview of word vector models

Word2vec is an open-source tool developed by Google for word vector computation. This tool can be efficiently trained on a large order of magnitude dictionary or dataset. After training, the word vector results of the corresponding dataset can be obtained, which can serve as a high-level ontology for XSS and can effectively describe the similarity between adjacent single words or multiple words. The traditional One-hot vector represents every word in the dataset discretely, and there is no interpretability between words. Moreover, when the capacity of the dataset reaches tens of thousands of levels, the representation of each word is only represented by taking one dimension from the vector of tens of thousands of dimensions. This method leads to abnormally high matrix sparsity [15], and there is a possibility of dimension explosion in the subsequent deep learning training process. Therefore, this paper uses Word2vec to express word vectors, enhancing the interpretability of scripts while reducing computational dimensions.

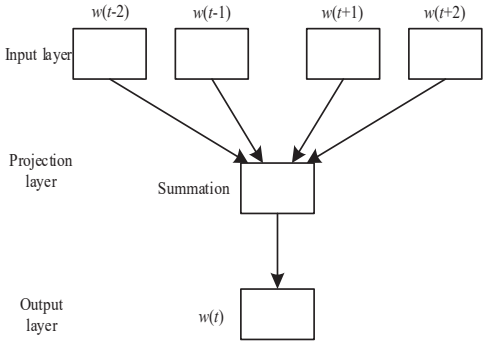


Figure 4 CBOW model.

The Word2vec algorithm consists of two models, namely the CBOW model and the Skip-gram model. CBOW model multiplies the vector of a bag-of-words model and a word embedding matrix to obtain a continuous word embedding vector [16]. This model infers the target word from several words in the context, as shown in Figure 4.

The CBOW model is divided into three layers: input layer, projection layer, and output layer. This model constructs a word vector using four words: $w(t-2)$, $w(t-1)$, $w(t+1)$), and $w(t+2)$ as input conditions, and accumulates the four vectors of the input layer to obtain the vector of the mapping layer. Finally, the output layer adopts a negative sampling method to distinguish between positive and negative samples and obtain the output result. The calculation process of the CBOW three-layer model is shown in Equations (2) to (4).

$$w' = T_i v(Context(w)) \tag{2}$$

$$T_w = \sum_{i=1}^{2n} (Context(w')_i) \tag{3}$$

$$G(w) = \prod_{u \in \{w\} \cup NEG(w)} p(u|T_w) \tag{4}$$

where Context is the set of input words w, and u is the word of the input positive sample. Equation (2) represents the input word vector, Equation (3) accumulates $2n$ vectors in the input layer, and Equation (4) maximizes whether the output result is a positive or negative sample to obtain an appropriate output result. Similarly, the Skip-gram model shown in Figure 5 is the

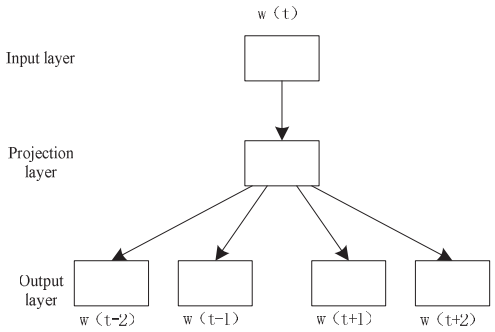


Figure 5 Skip-gram model.

output result obtained from the CBOW model, as shown in Equation (5),

$$G'(w) = \prod_{u \in \{w\} \cup NEG(w)} p(G(w)|u) \tag{5}$$

This paper uses the Skip-gram model to process the collected malicious script data. This method can reduce the input dimension of sample data and accelerate the training speed of the model. For any word that produces a word vector, this paper calculates the eight words closest to the target word in the word vector and displays the output. This method can provide a more detailed understanding of the possible forms of malicious cross-site scripting statements to a certain extent, providing support for subsequent analysis.

3.2.3 Entity description representation and feature fusion

This paper represents the fact triplets in the knowledge base as directed labeled multiple graphs $G = (V, E, R)$, where node $v_i \in V$ and V is the set of nodes (entities); the marked edge $(v_i, r, v_j) \in E$, and E is the set of relationships; the directed edge relationship type represents $r \in R$, and R is the set of relationship types. This paper continues to use the Word2vec algorithm for entity text processing and obtains x_i. Using the distributed vector of entity description paragraphs obtained from the trained model to represent entity word vectors in the knowledge graph can provide richer information supplementation for entities, better preserve the value information contained in the semantic text of entity description in the knowledge graph, and better mine out the potential feature information contained in the entities.

This paper considers fusing the features of node information and entity description information. The entire knowledge graph includes relationships in R and N entity nodes. Each entity node *i* is represented by a feature vector

h_i. In addition, the features of all entity nodes form a feature matrix X,

$$X = G'(w)[h_1, h_{2,}, \ldots, h_n], X \in R^{N \times F} \tag{6}$$

The paragraph vector x_i of entity i description information obtained using the Word2vec algorithm replaces the word vector corresponding to entity i as the feature vector volume of entity i, i.e. $h_i = x_i$, where X is:

$$X = [x_1, x_{2,}, \ldots, x_n], X \in R^{N \times F} \tag{7}$$

3.2.4 Intermediate representation

Different relationships convey different information. Under relationship r, each node obtains a unique intermediate representations feature vector $g_i^{(r)} \in R^{N \times F}$ through the weight matrix $W^{(r)}$:

$$G^{(r)} = XW^{(r)} = [g_1^{(r)}, g_2^{(r)}, \ldots, g_N^{(r)}] \tag{8}$$

where $G^{(r)}$ is the intermediate representation feature matrix under the relationship r, and $W^{(r)} \in R^{F \times F}$ is the learnable shared linear transformation matrix.

3.3 Vulnerability Detection Model of Internet Cross-site Scripting Based on Relationship Diagram Convolutional Networks

The R-GCN model proposed in this paper represents fact triplets in the knowledge base as directed labeled multiple graphs, and completes the knowledge graph by convolutional learning of their local domain structured information. Figure 6 shows the structure of the R-GCN model.

According to Figure 6, a special case of a simple differentiable message propagation framework is obtained:

$$h_i^{l+1} = \sigma\left(G^{(r)} \sum_{m=M_i} g_m(h_i^{(l)} + h_j^{(l)})\right) \tag{9}$$

where $h_i^{(l)}$ represents the potential implicit state of node v_i in the lth layer of the neural network; $g_m(\cdot\cdot)$ represents a class of neural network functions; $\sigma(\cdot)$ represents the element activation function; M_i represents the input information set of node v_i of this layer. This equation can be understood as the input information of the lth layer neural network processed by the function

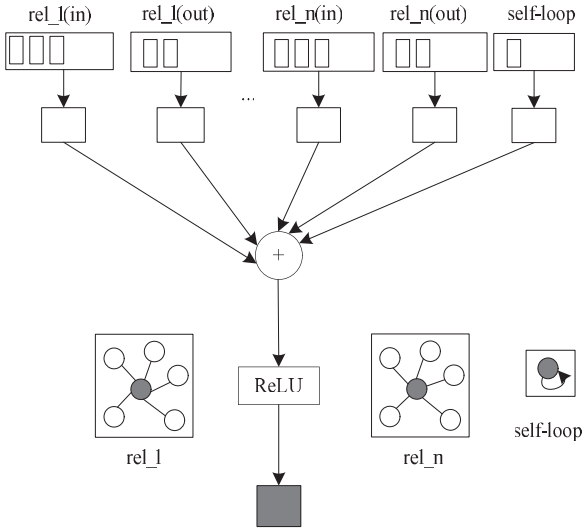


Figure 6 R-GCN model.

$g_m(\cdot\cdot)$, then accumulated and activated, and finally the potential implicit state representation $h_i^{(l+1)}$ of nodes in the $(l+1)$th layer neural network is obtained. Generally, the activation function $\sigma(\cdot)$ can be $\mathrm{ReLU}(\cdot) = \max(0, \cdot)$, and the function $g_m(\cdot\cdot)$ can be a linear transformation $g_m(h_i, h_j) = Wh_j$ with a weight matrix W.

It has been proven that this type of transformation is excellent in handling the accumulation and encoding of features in structured information representation learning of local domain in the knowledge base. Inspired by this architecture, Schlichtkrull et al. proposed an R-GCN model based on the star structure of local domains in the knowledge graph to handle large-scale high-dimensional and multi relational data in the knowledge graph:

$$H^{(l)} = \sigma\left(\sum_{r\in R}\sum_{j\in N_i^r}\frac{W^{(r)}h_j^{(l)}}{C_{i,r}} + W^{(0)}h_i^{l+1}\right) \tag{10}$$

where N_i^r represents the neighbor index set of node i under relationship $r \in R$; $c_{i,r}$ is a standardized constant specific to the problem, which can be prelearned or selected, and can generally be selected as $c_{i,r} = |N_i^{(r)}|$. It can be simply expressed as:

$$H^{(l+1)} = \sigma(H^{(l)}, W^{(l)}) \tag{11}$$

3.3.1 Attention mechanism

Based on the different characteristics of each node, the impact of different neighboring nodes on the central node varies. The independent $E_{i,j}^{(r)}$ of each relationship r is:

$$E_{i,j}^{(r)} = aH^{(l+1)}(g_i^{(r)}, g_j^{(r)}) \tag{12}$$

$E_{i,j}^{(r)}$ is composed of queries and key, and the potential feature vector representation $h_i^{(l+1)}$ of the update node is composed of the intermediate feature vector representation $g_i^{(r)}$. The intermediate feature vector representation $g_i^{(r)}$ is projected onto the D-dimensional space using the query kernel $Q^{(r)} \in R^{F\times D}$ and the key kernel $K^{(r)} \in R^{F\times D}$;

$$q_i^{(r)} = E_{i,j}^{(r)} g_i^{(r)} Q^{(r)} \in R^D, k_j^{(r)} = g_j^{(r)} K^{(r)} \in R^D \tag{13}$$

By paying attention to logic through multiplication, $\lg itE_{i,j}^{(r)}$ is obtained:

$$\lg itE_{i,j}^{(r)} = q_i^{(r)} k_j^{(r)} \tag{14}$$

The softmax mechanism is independently used to standardize $\lg itE_{i,j}^{(r)}$ for each relationship r:

$$\mathrm{softmax}(\lg itE_{i,j}^{(r)}) = a_{i,j}^{(r)} = \frac{\sum_{k\in N_i^r} \exp(E_{i,j}^{(r)})}{\exp(E_{i,j}^{(r)})}, \quad \forall i, \sum_{j\in N_i^r} a_{i,j}^{(r)} : j = 1 \tag{15}$$

3.3.2 Attention layer of the relationship diagram

The input of this layer is a graph of relationship type R and N nodes. The ith node is represented by the feature vector h_i, and the features of all nodes are summarized into a feature matrix H. The output of this layer is the implicit feature matrix $H_i^{(l+1)}$ obtained after transformation, and $h_i^{(l+1)}$ is the transformation feature vector of the ith node. When $l = 0$, $H = X$, that is, the first layer input of the neural network is the fusion feature of each initial node, it can be obtained that:

$$H^{(l+1)} = \mathrm{softmax}(\lg itE_{i,j}^{(r)})(W^{(l)}, X^{(l)}) \tag{16}$$

It is represented as the form of the intermediate feature matrix $G^{(l)}$:

$$H^{(l+1)} = \sigma(G^{(l)}) \tag{17}$$

The attention mechanism is added to it [17] and combined with R-GCN model neighborhood update rules to obtain the output feature vector representation of node i:

$$\sigma\left(\sum_{j\in N_i^r}\sum_{r\in R} g_j^{(r)} a_{i,j}^{(r)} \in R^{F'}\right) = h_i^{(l+1)} \tag{18}$$

This paper refers to the relationship diagram convolutional neural network model with the attention mechanism as the DR-GAT model.

3.3.3 Implementation of XSS vulnerability detection

As the current mainstream framework of graph neural networks [18, 19], the DR-GAT model can provide multiple graph neural network models and sampling methods. The propagation equation of the DR-GAT model is as follows:

$$h_i^{(l+1)} = \sigma\left(\sum_{j\in N_i^r}\sum_{r\in R}\frac{1}{c_{i,r}} W_r^{(l)} h_j^{(l)}\right) + W_0^{(l)} h_i^{(l)} \tag{19}$$

where $c_{i,r}$ is the normalization factor, which can be learned or specified in advance; $W_r^{(l)}$ is the linear transformation function under the relationship $r \in R$; $h_j^{(l)}$ is the hidden state of the neighboring node i in relation $r \in R$ in the previous layer; $W_0^{(l)} h_i^{(l)}$ represents learning the hidden state of the layer above node i.

Firstly, the initialization feature vector of the node is used as the initialization state vector of the node. The graph is divided into different subgraphs based on type and direction, and information is propagated independently on each subgraph. Then, the neighboring node features under different subgraphs are aggregated to update the next hidden layer state $h_i^{(l+1)}$ of node i, and the updates of each node can be calculated in parallel. After iterating T steps, the state vector calculated in the last step is used as the final representation of the node. The difference between the DR-GAT model and conventional GCN lies in the introduction of relationship specific transformations, which consider the direction and type of edges.

Different from the node level classification task, the vulnerability detection goal of this paper is to perform the neighborhood aggregation process under different relationships, take the XSS attack data word vector extracted from the XSS domain knowledge model as the input of each node in the relationship graph convolution network, learn a new set of embeddings for

Table 1 Statistics of the UNSW-NB15 dataset

Label	Quantity/Piece	Percentage
Normal record	3328496	95.45%
Forgery	4059	0.22%
Fuzzy attack	2580	0.16%
Back door	254870	7.10%
Flow analysis	841	0.13%
Denial of service	37894	2.09%
Tamper with message	2049	1.04%
Worm	382	0.11%
Buffer overflow	389	0.11%
Reconnaissance attack	2589	0.16%
Eavesdropping	241	0.11%

each node, and then further extract node features through convolution and pooling, and aggregate all node features and feed them into the fully connected neural network. The final vulnerability detection results in Internet XSS are classified and output through the activation function [20]. And combined with malicious domain names and malicious IP information as supplementary rules, it can improve the detection results of the network and improve the effectiveness of vulnerability detection of Internet XSS.

4 Experimental Analysis

Taking a bicycle production and service company network as the experimental object, the network consists of 20 nodes irregularly distributed within a 60×60 m area, the base station is located 40 m to the right of the center point, with a network transmission rate of 150 mbps and a protocol standard of 792.06. XSS attacks are simulated using attack data from the UNSW-NB15 dataset to test the effectiveness of the XSS vulnerability detection method proposed in this paper.

The UNSW-NB15 dataset contains a total of 3554129 pieces of data, each with 100 dimensional features. The statistical information of the UNSW-NB15 dataset is described in Table 1.

To verify the effectiveness of the method proposed in this paper, the results of detecting XSS vulnerabilities using the proposed method are presented in Table 2 by analyzing the above test dataset.

According to Table 2, this method can accurately detect many types of XSS attack, including persistent XSS, reflective XSS, storage XSS, DOM

Table 2 Results of XSS vulnerability detection

IP Address	Attack Time	Attack Intensity	XSS Attack Variant	Whether it is Consistent With the Reality
192.168.1.10	22.02.25	1500	Persistent XSS	Yes
10.0.0.5	22.03.12	1800	Reflected XSS	Yes
172.16.0.2	22.04.05	2100	Stored XSS	Yes
192.168.2.15	22.02.28	1200	DOM-based XSS	Yes
10.0.0.100	22.03.15	2500	Blind XSS	Yes

Table 3 Number of XSS and number of missing and misreported messages at various sites

Various Web Sites	Number of XSS Vulnerability Detections	Number of Missed Reports	Number of False Positives
Government website	8	1	0
Hospital website	9	0	0
News entertainment	3	0	0
Online shopping	25	1	1
Book website	10	0	0
Other	3	0	0
Total	58	2	1

based XSS and blind XSS. These attack types come from different IP addresses, and their attack time and attack intensity are recorded. By comparing with the actual attack types, it is found that the detection results of this method are completely consistent with the actual situation, which shows that this method has high accuracy and reliability in XSS vulnerability detection. Therefore, the method in this paper can provide strong technical support for the network security team to help them discover and respond to XSS attacks in time, so as to protect the security and stability of the network system.

To verify the efficiency of vulnerability detection in internet XSS under the method proposed in this paper, the experiment shows the number of XSS vulnerabilities detected by the bicycle production and service company's network accessing various websites within a month, as well as the results of missed and false alarms detected, as shown in Table 3.

According to Table 3, this method successfully detected 58 XSS vulnerabilities, covering websites in government, hospitals, news and entertainment, online shopping, books and other fields, proving its wide application ability and efficiency in different scenarios. In a total of 58 XSS vulnerabilities, there are only 2 false negatives, and the false negatives rate is 3.4%, which shows that this method has high accuracy in detecting XSS vulnerabilities. The number of false positives is only 1, and the false positive rate is 1.7%,

which further proves the reliability of this method in detecting XSS vulnerabilities. To sum up, this method performs well in detecting Internet cross site script vulnerabilities, and has the characteristics of high efficiency, low false negative rate and low false positive rate. These results not only verify the effectiveness of this method, but also provide a strong guarantee for the security of network environment.

Experimental comparison and analysis of the method proposed in this paper, the vulnerability detection method for cross-site scripting based on support vector machine in reference [1], the detection method based on single classification support vector machine classifier combined with feature vectorization in reference [2], the malicious code extraction vulnerability method based on the Web shell in reference [3], the reflective vulnerability detection method for cross-site scripting based on fuzzy theory testing in reference [4], and the association rule mining model based on the FGrowth algorithm for XSS vulnerability detection in reference [5] are carried out. Table 4 and Figure 7 respectively represent the performance comparison of different methods and the ROC curve. The ROC is a curve drawn under different classification thresholds, with false positivity as the horizontal axis and true positivity as the vertical axis. The area enclosed by the ROC curve and the coordinate axis is the area under the curve, and the larger the area is, the better the detection effect is. In order to comprehensively evaluate the detection effectiveness of vulnerabilities, this paper selects accuracy (Acc), recall (Rec), accuracy ($\Pr e$), and F_1 values as evaluation indicators. The relevant calculation equations are as follows:

$$Acc = \frac{TP + TN}{TP + FP + FN + TN} \tag{20}$$

$$Rec = \frac{TP}{TP + FN} \tag{21}$$

$$\Pr e = \frac{TP}{TP + FP} \tag{22}$$

$$F1 = \frac{Rec \times \Pr e \times 2}{Rec + \Pr e} \tag{23}$$

where samples containing vulnerabilities are represented as positive classes, while samples without vulnerabilities are represented as negative classes; TP represents the number of positive samples predicted by the model as positive; FP represents the number of positive samples predicted by the model as negative classes; TN represents the number of negative samples predicted by

Table 4 Performance comparison of different methods

Method	Acc	Re c	Pr e	F_1
Textual method	68.99	62.91	90.27	73.78
Method of reference [1]	66.40	58.71	73.41	65.09
Method of reference [2]	58.44	54.66	62.33	57.88
Method of reference [3]	60.22	59.86	15.89	32.68
Method of reference [4]	61.44	57.64	80.34	46.32
Method of reference [5]	65.44	60.41	85.21	56.37

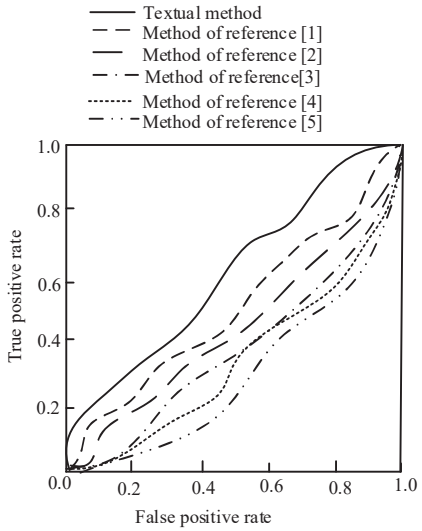


Figure 7 ROC curve.

the model as negative classes; FN represents the number of negative samples predicted by the model as positive.

It is not difficult to see from the analysis of Table 4 and Figure 7 that the detection accuracy (Acc), recall rate (Rec), accuracy (Pr e), and F_1-value of the proposed method for XSS vulnerabilities are higher than those of the methods in references, especially the accuracy which is as high as 90.27%, indicating that this method has high accuracy in identifying samples with XSS vulnerabilities. The area under the curve formed by the ROC curve and the coordinate axis of the XSS vulnerability detection method in this paper is the largest, which further proves the superior performance of this method in XSS vulnerability detection. These results not only verify the effectiveness of this method, but also provide a strong guarantee for the security of the network environment.

Table 5 Vulnerability detection time of different methods

Method	Consume Time/s
Method of this paper	0.03
Method of reference [1]	0.12
Method of reference [2]	0.11
Method of reference [3]	0.23
Method of reference [4]	0.21
Method of reference [5]	0.13

The experiment provides the time required for detecting specific XSS vulnerabilities using the method proposed in this paper and the comparison methods of references, as shown in Table 5.

According to Table 5, it can be seen that under specific XSS vulnerability attacks, the method proposed in this paper takes less time to detect vulnerabilities than the literature methods, indicating that the method proposed in this paper has a high recognition effect on XSS vulnerability attacks.

5 Conclusion

This paper studies a vulnerability detection method of Internet XSS based on relationship diagram convolutional networks. This method can not only achieve vulnerability detection of Internet XSS, but also reduce the leakage rate of network privacy information and the loss rate of important assets, ensuring the security of the internet environment. In addition, the method proposed in this paper has outstanding performance and high efficiency in detecting vulnerabilities on the internet XSS. It can complete vulnerability detection in a very short time, and the false positive and false positive rates of vulnerabilities are low. It can be seen that the method proposed in this paper has a crucial impact on the security of the cyberspace environment.

Declarations

Ethics Approval and Consent to Participate

Not Applicable.

Consent for Publication

Not Applicable.

Availability of Data and Materials

The data are within the manuscript.

Competing Interests

The authors declare that no conflict interests.

Funding

Not Applicable.

Authors' Contributions

Zhida Guo, conceptualization, methodology, data curation, writing-original draft preparation; Xiaoli Li, writing-review and editing, validation, resources, formal analysis; Ran Hu, investigation, resources, data curation, data curation, writing-review and editing; Dapeng Wang, resources, visualization, supervision, investigation; Weijie Song, investigation, formal analysis, resources, data curation.

Acknowledgements

Not Applicable.

References

[1] Nancy, P., Muthurajkumar, S., Ganapathy, S., Kumar, S., Selvi, M., and Arputharaj, K. (2020). Intrusion detection using dynamic feature selection and fuzzy temporal decision tree classification for wireless sensor networks. IET Communications, 14(5), 888–895.

[2] Anjinappa, C. K., and Guvenc, I. (2021). Coverage hole detection for mmwave networks: an unsupervised learning approach. IEEE Communications Letters, 25(11):3580–3584.

[3] Modi, B., Chourasia, U., and Pandey, R. (2022). Design and implementation of restful api based model for vulnerability detection and mitigation. IOP Conference Series: Materials Science and Engineering, 1228(1), 012010.

[4] Qasem, A., Shirani, P., Debbabi, M., Wang, L., Lebel, B., and Agba, B. L. (2022). Automatic vulnerability detection in embedded devices and

firmware: survey and layered taxonomies. ACM Computing Surveys, 54(2):25.1–25.42.

[5] Krishnaveni, S., Sivamohan, S., Sridhar, S., and Prabhakaran, S. (2022). Network intrusion detection based on ensemble classification and feature selectionmethod for cloud computing. Concurrency and Computation: Practice and Experience, 34(11):1–29.

[6] Liu, Z., Fang, Y., Huang, C., and Han, J. (2022). Graphxss: an efficient xss payload detection approach based on graph convolutional network. Computers & Security, 114.

[7] Liu, S., Lin, G., Han, Q., Wen, S., Zhang, J., and Xiang, Y. (2020). Deepbalance: deep-learning and fuzzy oversampling for vulnerability detection. IEEE Transactions on Fuzzy Systems, 28(7), 1329–1343.

[8] Mao, Y., and Cheng, X. (2020). Trace data monitoring and simulation of local area network malicious code intrusion process. Computer Simulation, 37(01): 263–266+271.

[9] Hosseiny, B., and Shah-Hosseini, R. (2020). A hyperspectral anomaly detection framework based on segmentation and convolutional neural network algorithms. International Journal of Remote Sensing, 41(18), 6946–6975.

[10] Alshdadi, A., Alghamdi, A., Daud, A., and Hussain, S. (2021). Blog backlinks malicious domain name detection via supervised learning. International Journal on Semantic Web and Information Systems, 17(3), 1–17.

[11] Zhao, F., and Ni, Z. (2021). Research on lightweight web intrusion active defense key technology and visual measurement model based on dynamic ip black list. Journal of Physics Conference Series, 1802(4), 042072.

[12] Qiu, H., Zhang, F., Li, G., Lin, Z., Zhou, X., and Li, J., et al. (2023). First principles of in-situ generated interfaces-cohesive force modeling. Weapon Materials Science and Engineering, 46 (01): 94–100.

[13] Fu, S., Liu, W., Li, S., and Zhou, Y. (2020). Two-order relationship diagram convolutional networks for semi-supervised classification. IET Image Processing, 13(14), 2763–2771.

[14] Song, X., Chen, C., Cui, B., and Fu, J. (2020). Malicious javascript detection based on bidirectional lstm model. Applied Sciences, 10(10), 3440.

[15] Dvali, G. (2021). S-matrix and anomaly of de sitter. Symmetry, 13(1), 3.

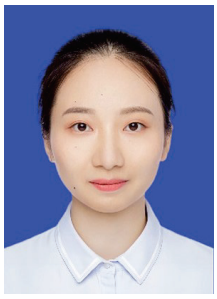
[16] Tripathi, S., and Kansal, V. (2020). Machine translation evaluation: unveiling the role of dense sentence vector embedding for

morphologically rich language. International Journal of Pattern Recognition and Artificial Intelligence, 34(1), 2059001.1–2059001.18.
[17] Alis, D., Alis, C., Yergin, M., Topel, C., Asmakutlu, O., Bagcilar, O., Oksuz, I., Kizilkilic, O., Karaarslan, E. (2022). A joint convolutional-recurrent neural network with an attention mechanism for detecting intracranial hemorrhage on noncontrast head ct. Scientific Reports, 12(1), 2084.
[18] Chen, H., Qi, B., and Zhao, H. (2022). Relationship diagram convolutional neural network gesture recognition based on pooling algorithm. Journal of Circuits, Systems and Computers, 31(15).
[19] Yhc, A., Cl, B., Sang, M., and Envelope, P. (2022). Graph neural network based multiple accident diagnosis in nuclear power plants: data optimization to represent the system configuration-sciencedirect. Nuclear Engineering and Technology, 54(8), 2859–2870.
[20] An, F. P., Liu, J. E., and Bai, L. (2022). Object recognition algorithm based on optimized nonlinear activation function-global convolutional neural network. The Visual Computer, 38(2), 541–553.

Biographies

Zhida Guo gained a master's degree from Sun Yat-sen University of Computer Science in 2013. His research interests include network security. From 2013 to present, he has worked at Huizhou Power Supply Corporation of Guangdong Power Grid Co. Ltd., Huizhou, China. He has published 6 academic papers and 7 patents.

Xiaoli Li gained a Bachelor's degree in Electrical Engineering and Automation from South China University of Technology in 2003. Her research interests include network security and digitalization. From 2003 to present she has worked at Huizhou Power Supply Corporation of Guangdong Power Grid Co. Ltd., Huizhou, China. She has published 7 academic papers.



Ran Hu gained a Bachelor's degree in Network Engineering from Guangdong University of Technology in 2015. Her research interests include network security. From 2015 to present she has worked at Huizhou Power Supply Corporation of Guangdong Power Grid Co. Ltd., Huizhou, China. She has published 4 academic papers and 5 patents.

Dapeng Wang gained a Bachelor's degree in Business Management and Business Information Systems from Northeast Electric Power University in 2005. His research interests include network security. From 2006 to present he has worked at Zhuhai Power Supply Corporation of Guangdong Power Grid Co. Ltd., Zhuhai, China. He has published 3 academic papers.

Weijie Song gained a Bachelor's degree in Network Security from Guangdong University of Technology in 2010. His research interests include network security. From 2010 to present he has worked at Zhuhai Power Supply Corporation of Guangdong Power Grid Co. Ltd., Zhuhai, China. He has published 3 academic papers.