


# An Abnormal External Link Detection Algorithm Based on Multi-Modal Fusion

Zhiqiang Wu, Henan Police College, China\*

 <https://orcid.org/0000-0002-5373-3983>

## ABSTRACT

Website link detection is an important means to ensure the security of the external chain. In the past, it was mainly realized through blacklisting and feature engineering-based machine learning, which has the problems of slow detection speed and weak model generalization ability. The development of neural networks has brought a new solution to the security detection of the external chain of the website. To address the performance bottleneck caused by the variable content length of web pages, this article introduces an innovative approach: a website external link security detection algorithm based on multi-modal fusion. It extracts text, dynamic script, and image features separately, and constructs a deep fusion model that combines these multi-modal features. Compared with the previous research results, the proposed method is superior to the traditional single-mode method, and can quickly and accurately identify malicious web pages. The accuracy and F1 value are improved by 2.7% and 0.026.

## KEYWORDS

feature extraction, malicious content, multimodal fusion, security detection, website external links

## INTRODUCTION

With the rapid development of information technology and the popularization of the Internet, the number of websites on the Internet has increased exponentially. In order to provide users with richer information resources and promote cooperation and interaction with other websites or institutions, a lot of external links are generally introduced into the website. Due to information updates, domain name changes, hacker attacks, and other reasons, if you link to an insecure external website, it will pose a security risk to users. Such risks can include malicious links, erotic gambling sites, or web pages containing malicious code that may lead to the disclosure of the user's personal information, computer infection, economic losses, and other problems (Tenis & Santhosh, 2021). In addition, if you link to external websites containing harmful information, it will seriously damage the reputation of the organization, and users may doubt the professionalism, trust, and network security capabilities of the organization, which will affect user's access to and use of the organization's website. Therefore, ensuring the security of the external link of the website is crucial for the website.

DOI: 10.4018/IJISP.337894

\*Corresponding Author

This article published as an Open Access article distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/4.0/>) which permits unrestricted use, distribution, and production in any medium, provided the author of the original work and original publication source are properly credited.

It is an important means to carry out regular inspections of the external chain of the website to ensure the security of the external chain. However, due to the large number of websites and pages, it is undoubtedly unrealistic for website security managers to use manual inspection. With the development of computer technology, the research on the security detection of external links of websites by computer programs has been widely concerned, and many detection schemes have been proposed by scholars at home and abroad. The earliest detection method used the blacklist technique, which preconstructed a blacklist listing all known harmful domain names. When a user visits a website, they check whether its domain address is in the blacklist to detect harmful external links. This method has the advantage of high detection accuracy, but it needs to ensure the timely maintenance of the black and white list, which has certain limitations and lag and cannot effectively judge the security of unknown web pages (Darwish et al., 2023). To solve this problem, some scholars have proposed a method based on dynamic behavior analysis, which analyzes the behavior of the website host, such as access records, execution processes, etc., to analyze whether the website host has abnormal behavior and find out the abnormal external chain. This method has the ability to detect unknown viruses and malicious codes, but the detection speed is slow because it needs to simulate the running state of malicious web pages and analyze them.

With the development of data mining and machine learning technology, a website off-link security detection method based on machine learning has been proposed (Jerjes et al., 2023; Venugopal et al., 2021). This method has a certain generalization ability, but due to the great impact of the selection of webpage features on the model recognition effect, the workload in the feature engineering stage is relatively large. At the same time, the traditional machine learning technology cannot learn the contextual semantic features of web text, resulting in a certain bottleneck in the recognition effect.

In the past few years, the field of external chain detection has witnessed a shift toward deep learning-based approaches driven by the rapid advancements in machine learning and artificial intelligence technology. According to the existing literature, text features are mostly used, and due to the variable length of Chinese text on web pages (Naim et al., 2023), in order to achieve the feasibility of model training, in addition to short text features such as Uniform Resource Locator(URL) and tags, part of text content from web pages is generally extracted for model training, resulting in poor practicability of the trained model. In addition, with the development of communication technology, a large number of web pages contain not only text information but also a lot of multimedia information, such as pictures, videos, and sounds. It is not good to judge whether a web page has malicious information only through text information. In view of these problems, in this research, the website link security detection is regarded as a binary classification problem. By integrating the features of webpage text, dynamic script, and image, an innovative intelligent detection algorithm for website link security based on multimodal fusion is proposed. The main work of this paper includes:

1. The FastText model is used to extract the text features of web pages, aiming at the problem that the content length of web pages is not fixed.
2. Aiming at the performance bottleneck of web page detection using text alone, this paper introduces an innovative approach: a website external link security detection algorithm based on multimodal fusion. It extracts text, dynamic script, and image features separately and constructs a deep fusion model that combines these multimodal features.
3. Through comparative experiments, the effectiveness of the web security intelligent detection algorithm proposed in this paper based on multimodal feature fusion is verified.

The following organizational structure of this paper is as follows: The Related Work section briefly introduces the related research work; the Methodology section introduces the feature extraction method and the multimodal feature fusion model. The Experiment section discusses the effectiveness of the proposed algorithm verified by comparison experiments. The Discussion section summarizes and looks forward to the work of this paper.

## RELATED WORK

The purpose of website external link detection is to deal with the webpage containing malicious content in time to ensure the security of website content. In the past, this was mainly achieved through blacklisting and feature engineering-based machine learning. The nature of web threats is constantly evolving. The development of neural networks has brought many new solutions to the security detection of external links of websites (Zhu et al., 2018).

Zhu et al. (2018) introduced a novel model called OFSNN for detecting malicious websites. This model utilized an optimal feature selection method and neural network to achieve effective detection. Notably, they introduced a new index known as the “feature effective value” to assess the influence of sensitive features on identifying malicious websites. This innovative approach greatly mitigated the overfitting issue commonly associated with neural networks.

Chen et al. (2021) enhanced a multilayer recursive convolutional neural network model utilizing the YOLO algorithm to detect malicious URLs. They incorporated word embedding to map individual characters to dense vectors and incorporated these dense vectors into the training process of the entire model based on the structural characteristics of URLs. Additionally, they proposed a novel CSPDarknet neural network model using the improved YOLO algorithm to extract features to identify malicious URLs via the bidirectional long short-term memory (BiLSTM) network.

Yuan et al. (2021) used the visualization algorithm to realize the visual mapping of a URL to a grayscale image and extracted the lexical and character features of a URL. They adopted a parallel joint neural network to simultaneously capture multimodal vectors of visual and semantic information and combined this with the attention mechanism to further filter the deep features, effectively improving the classification accuracy.

Raja et al. (2023) used natural language processing techniques, such as term frequency and inverse document frequency, to vectorize URLs and used weighted classifiers to improve the robustness of malicious link detection. However, the feature selection also needs some manual intervention and has some limitations.

In 2018, the large-scale text-based Transformer pretraining model BERT came out, which has refreshed a number of records of natural language processing tasks. Zhang and Zhang (2022) applied BERT to the malicious domain name detection task, which strengthens the character’s decision-making ability for the model and improves the model’s detection performance. However, the BERT model is not used to identify malicious web pages based on the text content of web pages. Wang et al. (2020) extracted binary file similarity, used word vector tool word2vec (Word to Vector) to train URL word vector features, used a convolutional neural network (CNN) to extract deep local features, and proposed a parallel joint algorithm model of a convolutional neural network and bidirectional independent recurrent neural network combined with an attention mechanism. With the increasing concealment and complexity of malicious websites, the identification of malicious web pages based solely on a URL and its related features is far from meeting the current demand for the identification of malicious web pages.

Feng et al. (2021) proposed a machine learning method to identify malicious web pages based on the features of traditional web pages, such as a URL and hypertext markup language (HTML), and the text features of web pages. However, the traditional machine learning classification method does not consider the contextual information of text in the vector construction of text features and cannot reflect the deep-level features of polysemy in text. There is some information missing.

Lian et al. (2021) proposed research on malicious website identification technology based on network behavior analysis. By studying the multistep and sequential network data flow of users, this method extracts the dominant traffic characteristics and recessive network behavior characteristics, establishes a malicious website recognition model based on a backpropagation (BP) neural network, and adopts a genetic algorithm to optimize the model, so as to realize the dynamic recognition of malicious websites. However, this method takes a long time for identification and has low efficiency.

According to the existing literature, there is much research on malicious web page detection based on deep learning. These methods generally regard web page detection as a binary classification problem and carry out research on malicious web page detection based on deep learning based on web page text information. With the improvement of computer computing power, the selection of text information has gradually developed from a single URL feature to multifeature fusion, such as page title, label, and page text content. Meanwhile, with the development of deep learning applications in image, audio, and video, some studies have begun to focus on the impact of multimodal information on malicious web page detection. Based on relevant research, this paper proposes a deep learning model based on multimodal feature fusion to detect and identify malicious web pages and realize intelligent detection of web security.

## METHODOLOGY

### Multimodal Feature Fusion Model

The features used to detect malicious web pages mainly include text, dynamic scripts, and images. How to comprehensively apply these features to improve the effect of the detection model is the focus of research. To solve this problem, this paper proposes a malicious web page detection model based on multimodal feature fusion, as shown in Figure 1. The innovation of this paper lies in selecting appropriate models to extract text, dynamic script, and web image features, respectively, constructing a feature fusion model, and utilizing multiple web page features to improve the performance of external link detection. Firstly, FastText, BiLSTM, and ResNet (Residual Neural Network) models are used to obtain the text, dynamic script, and image features of web pages, and then feature level fusion is carried out. Finally, the final decision result is output through the model. There are many methods to achieve feature fusion. One simple and effective method is to splice the feature vectors to get the fused feature vectors. Subsequent model networks can learn useful feature content for classification tasks from this fused feature vector. Although this method is simple, it can achieve good results. In addition, other more complex methods can be employed, such as the introduction of attention mechanisms to further optimize the process of feature fusion. No matter what method is used, the goal of feature fusion is to extract effective feature information for classification tasks. In this paper, Concatenate is used for feature connection, and the connected features are processed in Dense layers. At the same time, Dropout layers are added to reduce the possibility of overfitting.

The process of fusing Concatenate features can be represented as  $z = [a, b, c]$ , where  $[ ]$  represents the concatenation operation. In this way, a new feature vector  $z$  can be obtained, which contains feature information from text, dynamic scripts, and images. Assuming that  $n$  represents sample size,  $z_i$  represents the feature vector of the  $i$ -th sample,  $y_i$  represents the corresponding label, and  $\theta$  represents the model parameters, the objective of the model is to minimize the loss function  $L(\theta)$  as Equation (1)

$$L(\theta) = -1/n * \sum (y_i * \log(p_i) + (1 - y_i) * \log(1 - p_i)). \quad (1)$$

Here,  $p_i$  represents the predicted probability of the model for the  $i$ -th sample, which can be calculated using the Softmax function. The calculation formula is shown in Equation (2)

$$p_i = \text{softmax}(z_i * \theta). \quad (2)$$

When training machine learning models, iterative optimization algorithms, such as stochastic gradient descent (SGD), are frequently employed to update the model's parameters. During each iteration, a small batch (minibatch) of samples is randomly selected from the training set for training. The loss function's gradient is then computed, and the model parameters are updated in the direction of the gradient. Specifically, for the  $t$ -th iteration, the  $k$ -th minibatch of samples is  $\{z_{k_1}, z_{k_2}, \dots, z_{k_m}\}$  with corresponding labels  $\{y_{k_1}, y_{k_2}, \dots, y_{k_m}\}$ .  $m$  represents the size of training samples per batch. The gradient of the loss function can be expressed as shown in Equation (3)

$$\nabla L(\theta) = -1 / m * \sum (y_{k_i} - p_{k_i}) * z_{k_i}. \quad (3)$$

According to the update rule of gradient descent, the model parameters  $\theta_{t+1}$  for the next iteration can be updated as in Equation (4)

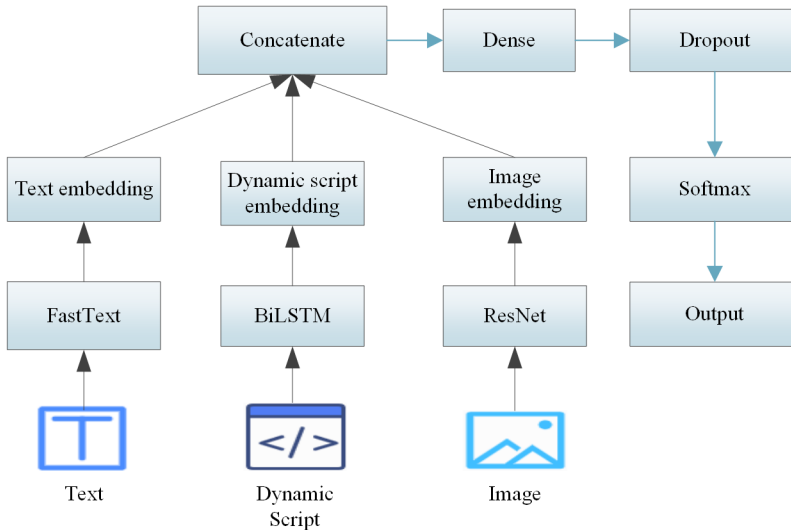
$$\theta_{t+1} = \theta_t - \alpha * \nabla L(\theta). \quad (4)$$

Here,  $\alpha$  is the learning rate that controls the step size of each update. During the training process, it is important to monitor the performance metrics of the model on both the training set and the validation set. This allows us to promptly identify issues, such as overfitting or underfitting. When the model's performance on the validation set no longer improves, we stop the training process and apply the final model to the test set for evaluation.

## Web Text Feature Extraction

Web page text mainly refers to the title of the web page and the text content in the web page. With the popularity of responsive programming technology, the content of web pages is mainly controlled by dynamic scripts, and a web page may introduce multiple web pages through iframe tags, resulting in the actual display content of web pages not being obtained from the HTML source code of a single web page. In order to obtain the actual display content of the web page, Selenium WebDriver is used

Figure 1. Malicious website detection model based on multimodal feature fusion



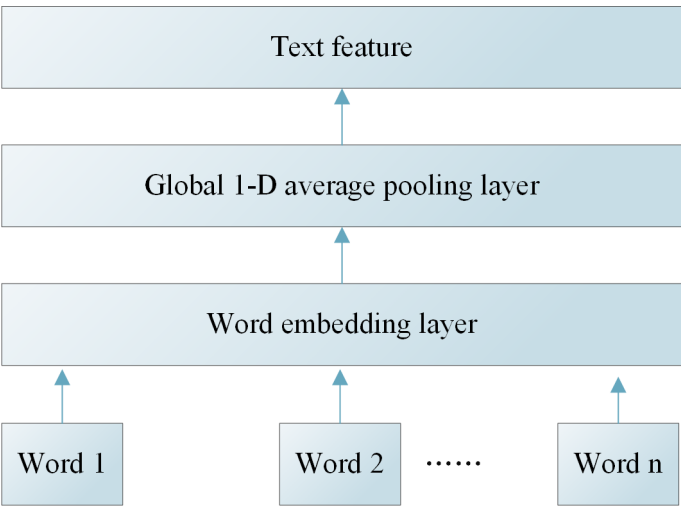
to simulate the browser to access the web page, trigger the execution of the dynamic content loading script by controlling the scroll of the web page, and climb the source code of the web page after the loading of the web page is completed and analyze it to obtain the text content contained in the title of the web page and the content labels in the web page. The text data of the web page are obtained after concatenation.

The text data length of the web page is not fixed, generally very long, and belongs to long text. Currently, commonly used text feature extraction models, such as Bert and Ernie, are not good at processing long text (Artene et al., 2021; Li et al., 2023). First, common models need to limit the length of input text data because the number of nodes in each layer of the model is fixed at design time and can only handle fixed-length input data. To make the best use of the information in long text, you must design to accept the longest input data length possible, but this results in inefficient operation of the entire model. At the same time, even if the accepted input length is maximized, it can only be a fixed value and cannot deal with the situation of uncertain length. Web malicious information detection generally does not need to dig deep semantic information of text to obtain a good detection effect. The FastText model is a model suitable for processing text data of any length, which consists of three parts: input layer, hidden layer, and output layer (Khasanah, 2021). The input layer is the word embedding layer, which receives the preprocessed text corpus and maps the words or word vectors into a low-dimensional vector. The hidden layer is a one-dimensional global average pooling layer, which is responsible for superimposing words or word vectors in the text corpus and averaging them to obtain the overall feature vector representation of the text corpus (Zhou et al., 2020). This model is simple and efficient, so the FastText model is chosen to extract text features, as shown in Figure 2. “1-D” represents one-dimensional, and “word  $n$ ” indicates the  $n$ -th word.

### Web Dynamic Script Feature Extraction

While injecting dynamic effects into web pages, JavaScript itself also has a lot of vulnerabilities, which provide an opportunity for some attackers. Attackers can implant malicious code in web scripts, spread viruses through URL redirection, JavaScript hijacking, cross-site scripting attacks, etc., steal personal privacy, and lure users to phishing websites to commit fraud. The dynamic script security detection of web pages has become an important part of malicious web page detection. JavaScript has the feature of long-distance dependence, and one-way LSTM can only transfer dependency from front to back but cannot solve the dependency of the following on the above (Sherstinsky, 2020). In

Figure 2. Text feature extraction based on the FastText model



the process of learning features, the dependency of context in JavaScript code is ignored. Feature learning is not sufficient. Therefore, bidirectional LSTM is used to extract dynamic script features, and the process is shown in Figure 3.

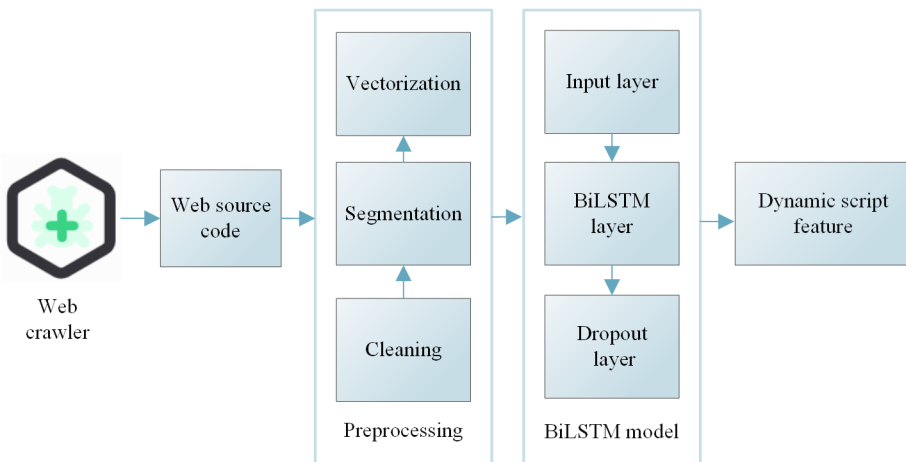
Firstly, the data of JavaScript code are cleaned to delete the data with repeated errors, and then the code is de-obfuscated to improve the readability of the code. Then the term frequency-inverse document frequency (TF-IDF) algorithm is used to build a vocabulary database, and an id is set for each word. The code is divided into words according to the vocabulary database, and the data of word segmentation are converted into numerical values. The vectorization of the data is processed by a Word2Vec tool. Finally, input the BiLSTM model to extract the abstract features of the code. The input is  $X_i = \{x_1, x_2, \dots, x_n\}$ ,  $n$  represents the dimension of the input vector, and the output vector is obtained through the input layer  $I_i = \{i_1, i_2, \dots, i_n\}$ , take  $I_i$  as the input vector of the BiLSTM layer. Through the forward LSTM of this layer, a forward implicit vector is obtained  $L_i = \{l_1, l_2, \dots, l_{64}\}$ . Passing through the backward LSTM of this layer yields a backward implicit vector  $R_i = \{r_1, r_2, \dots, r_{64}\}$ . Concatenate these two vectors into one vector  $h_i = \{[l_1, r_1], [l_2, r_2], \dots, [l_{64}, r_{64}]\}$ . In order to prevent overfitting problems during later model training, the input is put into the Dropout layer and several neurons are randomly deactivated and get the output vector  $d = \{d_1, d_2, \dots, d_{64}\}$ . The calculation method is shown as Equation (5).  $g^l$  represents the state of all neurons, with values consisting of 0 and 1

$$d = g^l * h^l. \quad (5)$$

## Web Image Feature Extraction

With the development of the Internet infrastructure, web pages not only contain a lot of text information but also increasingly use pictures as a way of information expression. Web page images have a lot of semantic information, which can be used as an important factor in the judgment of malicious web pages. Through the analysis, it is found that the ratio of the number of pictures and documents in the web page is about 9:1, and how to select the pictures becomes the key to the feature extraction of the web page image. In this paper, the images in the web page are first extracted by crawler, then

Figure 3. Web dynamic script feature extraction process



the image is normalized, and the size is uniformly scaled to  $224 \times 224$ . Then the data are enhanced by random rotation, color jitter, and image clipping to reduce the influence of position, color, and light factors on feature extraction. Finally, in different epochs of model training, images from web pages are randomly selected and input into a ResNet model pretrained by an ImageNet data set to obtain image feature vectors.

The ResNet model was proposed by the Microsoft Research Asia team (Alsabhan & Alotaiby, 2022) and won first place in the ImageNet Large Scale Visual Recognition Challenge in 2016. ResNet50 belongs to one of the ResNet family models and has 49 convolution layers and 1 maximum pooling layer. Different from the traditional neural network structure, ResNet50 adds a residual learning module to solve the problems of gradient disappearance and model degradation encountered in training deep neural networks. These residual learning modules can make the network deeper and reduce the number of parameters and the amount of computation while maintaining the same accuracy. According to the application requirements and experimental computing power conditions, ResNet50 is selected as the image feature extraction model.

## EXPERIMENT

### Experimental Data Collecting

We downloaded positive sample data from the Alexa data set and malicious website data from the Phish Tank and OpenPhish websites as negative sample data. In order to maintain the balance of positive and negative samples, the ratio of the number of positive and negative samples downloaded was about 1:1. Next, to solve the problem of false positives, the downloaded sample data were cleaned to remove redundant data, and the accessibility of URLs was tested to remove invalid URLs. In the end, we got a list of URLs containing both normal and malicious web pages. In order to get the text and image data of the web page, we used the Beautiful Soup toolkit in Python to crawl. After crawling, the text and image data were preprocessed, such as denoise and unified format, and then stored in folders, respectively. In the end, 24,436 folders were retained, each containing text, JavaScript scripts, and image information extracted from a URL.

After randomly shuffling the data, 80% of the data were used as the training set, and 20% were used as the test set. The final training set contained 19,548 pieces of data, and the test set contained 4,888 pieces of data. In the training set, the number of normal web pages was 9,792, and the number of malicious web pages was 9,756. In the test set, the number of normal web pages was 2,510, and the number of malicious web pages was 2,378. The specific data set is shown in Table 1.

### Experimental Setup

All the experiments in this paper were conducted on an HP graphics workstation. The network model was built using the open-source deep learning tool Caffe. The experimental steps are as follows:

- a) Combined with the size of the experimental data set, the dimension of the FastText model and BiLSTM hyperparameter word vector used in the experiment was set to 300, the size of the lexis was set to 10,000, and the activation function is ReLu. In order to explore the effect of the FastText model and BiLSTM model on the detection of malicious web pages, the two models

Table 1. Data set details

Classification	Code	Training set	Test set	Total
Normal	1	9,792	2,510	12,302
Malicious	0	9,756	2,378	12,134

were used separately for web page detection experiments. In addition, considering that text and dynamic scripts in web pages belong to text, the FastText model and BiLSTM model were integrated, and the detection effect of fusion model was verified by experiments.

- b) Once the training set was established and the input-output layers were defined, a crucial and challenging task is optimizing the number of hidden layers. Selecting an appropriate number of hidden nodes is vital for ensuring the network possesses the necessary learning and information processing capabilities. Insufficient hidden nodes may limit the network's capacity, while an excess can lead to increased complexity, making it more susceptible to getting trapped in local minima during the learning process and slowing down the network's learning speed. Striking the right balance is essential. To this end, the web page detection effect data, when the node was 40–320, were obtained through the selection experiment, from which the optimal number of hidden nodes was selected.
- c) In order to explore the influence of different models on web page detection performance, comparison and ablation experiments were designed. For the text model, the support vector machine (SVM) and Bert model were used to compare with the FastText model and BiLSTM model. For the image model, VGG16 and CNN were used to compare with the ResNet50 model in this paper.
- d) The algorithm proposed in this paper was used to carry out web page detection experiments, and the experimental results were compared with those of the previous experiments to verify the effectiveness of the multimodal feature fusion algorithm proposed in this paper.

## Metric Standard

The accuracy rate, recall rate, and *F1* value were selected to evaluate the performance of the model. The metric consists of four values: *TP*, *TN*, *FP*, and *FN*, where *TP* is the number of positive samples predicted; *TN* is the number of negative class samples predicted to be negative; *FP* is the number that is predicted to be positive in a negative sample; *FN* is the number of positive samples that are predicted to be negative. The calculation formula of accuracy *A* is shown as Equation (6), which represents the percentage of all samples that can be correctly classified. The formula for calculating the recall rate *R* is shown as Equation (7), which represents the percentage of positive samples predicted to be positive. The formula for calculating the *F1* value is shown as Equation (8). When *F1* is larger, the classification effect of the proposed algorithm is better. In addition, the convergence rate of the model is selected as another index of the evaluation performance of the model. The faster the convergence rate of the model, the higher the stability of the model and the better the evaluation effect

$$A = \frac{TP + TN}{TP + TN + FP + FN} \times 100\% \quad (6)$$

$$R = \frac{TP}{TP + FN} \times 100\% \quad (7)$$

$$F1 = \frac{TP}{TP + \frac{1}{2}(FP + FN)} \quad (8)$$

## DISCUSSION

### Training Process Analysis

As shown in Figure 4, when the number of nodes in the hidden layer was 200, each index was optimal. This was because the number of nodes in the hidden layer was too small, and the learning and information processing ability of the proposed algorithm was poor. However, if the number of nodes was too large, the structure of the network would become more complex, and it was prone to

limitations, and the learning rate would decrease. Therefore, when the hidden layer node was set to 200, the proposed algorithm had the best identification effect and could accurately find out the bad information of the website.

Figure 5 is the model training diagram. It can be seen that the convergence speed of the model designed in this paper reaches a stable state when the number of training steps is about 400, and the convergence speed is fast, which ensures the evaluation effect of the model.

Single-Mode Comparative Analysis

To evaluate the effectiveness of various text models for web page detection, we conducted model training and detection experiments following the procedure outlined in the Experimental Setup section.

Figure 4. The experimental results of different hidden layer nodes are used

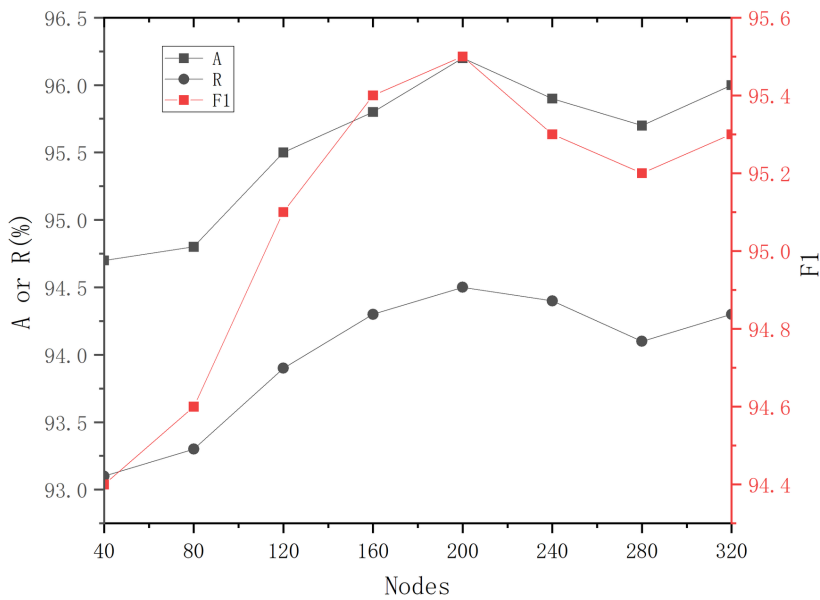
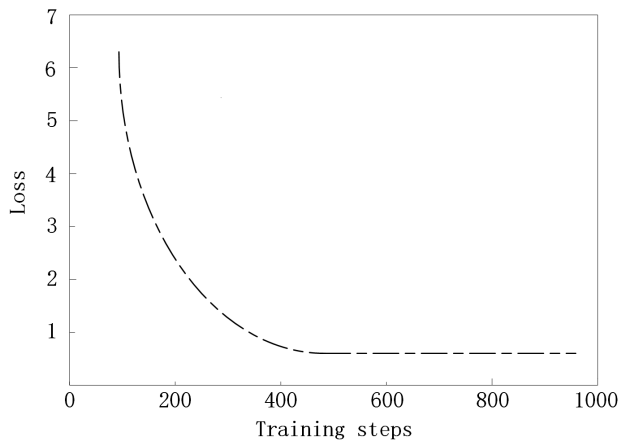


Figure 5. Multimodal model training diagram



The results of these experiments are presented in Table 2. The experimental findings indicate that both the Bert and BiLSTM models outperform the traditional SVM model in terms of performance. Compared with SVM and Bert models, FastText and BiLSTM models have better detection results, which may be due to the fact that FastText uses all the text features in the web page, while BiLSTM uses the dynamic script features of the web page. When the Bert model is used, it takes into account the character length that the model can receive. The data used for training and testing are shown in Figure 6, which are the last 100 characters of the web page. The extraction of text features of the web page is insufficient, and there is a certain deviation in feature extraction. Compared with other models, the FastText-BiLSTM model proposed in this paper has higher accuracy and F1 value. Compared with the traditional SVM, the model has improved by 11 percentage points, which is mainly due to the comprehensive use of text features and dynamic script features.

The experimental results of web page detection based on image features are shown in Table 3, and the comparison of single-mode experimental results is shown in Figure 7. According to the results, compared with other models, the pretrained ResNet50 model has a better detection effect, which is mainly due to the deeper network of ResNet50. Using the idea of residual learning, the training time of the model can be reduced, and the feature data of the image can be obtained better. Compared with the detection results based on text features, the detection accuracy rate, recall rate, and F1 value of malicious web pages based on image features are all decreased mainly because the semantic information contained in the pictures of malicious web pages is less than that in the text, and the pictures of some malicious web pages may not necessarily contain malicious information.

Multimodal Comparative Analysis

Approach B represents the BiLSTM model, F represents the FastText model, C represents the CNN model, V represents the VGG16 model, and R represents the ResNet50 model, as shown in

Table 2. Detection results based on text models

Approaches	A (%)	R (%)	F1
SVM	81.9	81.2	0.821
FastText	88.6	87.8	0.872
Bert	85.2	86.5	0.813
BiLSTM	91.1	94.3	0.927
FastText-BiLSTM	93.6	94.8	0.928

Figure 6. Examples of data used by Bert models

门太阳城娱乐城 香港六合彩网站老虎机 官网捕鱼达人技巧 电子游戏澳门现金网 注册澳门赌球美高梅澳门巴黎人澳门总统澳门新葡京娱乐城澳门银河娱乐城澳门金沙娱乐城澳门威尼斯人娱乐城版权所有 普腾商贸有限公司 1  
官网赌博网站线上赌博网站博彩游戏真人博彩网上百家乐百家乐代理金冠娱乐城娱乐城注册送 钱能娱乐城皇家娱乐城处女皇号娱乐城女福国际娱乐城香港六合彩开奖六合彩开奖澳门百家乐博彩网站澳门威尼斯人官网澳门百家乐 1  
业科学教育 文化卫生其他行业农 林 渔 牧业事业单位单位名称 所在部门 单位地址 证 码 南力码请照图片输入 如果看不清 请点击图片刷新 我已看过并同意 中国通信股份有限公司河南分公司用户注册协议 0  
告招商 排名联系我们网站地图全球机构官网微博微信微博 网盛科技 联盟网站 本网热线电话 版权所有 华夏收藏网 本站常年法律顾问 杨天城律师事务所 陈先生 发密不上网 上网不涉密分享到 分享到百度一下 0  
频类网站收藏 以方便网友观看 版权归原创作者或原公司所有本站不承担任何法律责任和连带责任 如果已经涉及到您的版权 请速与本站管理员联系 我们将第一时间为您删除 粤 备 号 综艺巴士 联系我们 站长统计 0

Table 3. Detection results based on image features

Approaches	A (%)	R (%)	F1
CNN	80.1	80.5	0.809
VGG16	81.6	81.5	0.811
ResNet50	82.9	80.7	0.818

Figure 7. Comparison of single-mode experimental results: (a) Accuracy comparison, (b) F1 comparison

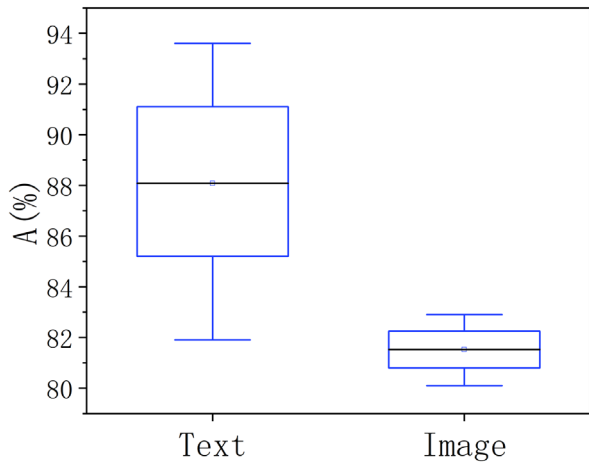


Table 4 and Figure 8. From the experimental results, it can be seen that the FBR model proposed in this paper has higher accuracy and F1 value than other models. Compared to other single-modal feature models, the accuracy has improved by 2.7%, and the F1 value has increased by 0.026. It is mainly because the fusion model can better extract and learn the malicious information features of web pages. Although the experimental results using image information alone are poor, the fusion of image features can effectively improve the effect of model detection and help to accurately identify malicious web pages.

### Application

In order to facilitate application, the SpringMVC framework encapsulates the algorithm implemented in this paper into a Restful interface. After receiving the URL address, the interface will detect the webpage corresponding to the URL address. Without considering the crawler's time to fetch the webpage, the average detection time of a webpage is about 2 seconds. Compared with other detection algorithms (Yan & Kun, 2019), the proposed algorithm can not only quickly detect malicious information on web pages but also ensure the accuracy of detection.

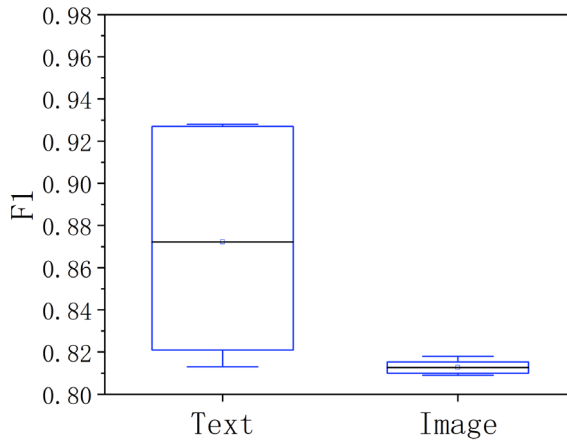
### CONCLUSION

In this paper, an abnormal external link detection algorithm based on multimodal fusion was proposed. By preprocessing, invalid data were removed from web pages, the FBR model was used to

Table 4. Multimodal detection results

Approaches	A (%)	R (%)	F1
FR	92.2	93.4	0.929
BR	93.9	91.6	0.927
FBC	95.4	94.0	0.942
FBV	96.2	94.3	0.953
FBR	96.6	94.5	0.955

Figure 8. Comparison of multimodal experimental results



extract malicious information from web pages, and a deep fusion model of multimodal features was established to update the learning parameters. From the experimental results, the algorithm designed in this paper could quickly and effectively detect malicious information on web pages. However, the proposed algorithm still has some shortcomings. The current experimental data sets may not be comprehensive and representative enough, and larger and more diverse data sets are needed in future studies to improve the robustness and accuracy of the algorithm. Future work could optimize and accelerate the algorithm to improve real-time and scalability and continuously improve the practical application of the algorithm.

## CONFLICTS OF INTERESTS

All authors of this article declare there are no competing interest.

## FUNDING STATEMENT

This paper is supported by the Academic Research Project of Henan Police College (Grant: HNJY-2023-29) and the Key Technology Research and Development Program of Henan Province (Grant: 222102210041).

## REFERENCES

- Alsabhan, W., & Alotaiby, T. (2022). Automatic Building Extraction on Satellite Images Using Unet and ResNet50. *Computational Intelligence and Neuroscience*, 5008854, 1–12. Advance online publication. doi:10.1155/2022/5008854 PMID:35222630
- Artene, C. G., Tibeică, M. N., & Leon, F. (2021, 28-30 Oct. 2021). Using BERT for Multi-Label Multi-Language Web Page Classification. *2021 IEEE 17th International Conference on Intelligent Computer Communication and Processing (ICCP)*. IEEE. doi:10.1155/2021/9994127
- Darwish, S. M., Farhan, D. A., & Elzoghbi, A. A. (2023). Building an Effective Classifier for Phishing Web Pages Detection: A Quantum-Inspired Biomimetic Paradigm Suitable for Big Data Analytics of Cyber Attacks. *Biomimetics*, 8(2), 197. doi:10.3390/biomimetics8020197 PMID:37218783
- Feng, K., Luo, Q., Zheng, M., & Li, C. (2021). Research on detection method of malicious web pages based on multi-feature fusion. [Natural Science Edition]. *Journal of Hubei Minzu University*, 39(1), 80–85. <https://link.cnki.net/urlid/11.4762.TP.20240119.1730.023>
- Jerjes, A., Dawod, A. Y., & Abdulqader, M. F. (2023). Detect Malicious Web Pages Using Naive Bayesian Algorithm to Detect Cyber Threats. *Wireless Personal Communications*. doi:10.1007/s11277-023-10713-9
- Khasanah, I. N. (2021). Sentiment Classification Using fastText Embedding and Deep Learning Model. *Procedia Computer Science*, 189, 343–350. doi:10.1016/j.procs.2021.05.103
- Li, Y., Xiong, H., Wang, Q., Kong, L., Liu, H., Li, H., Bian, J., Wang, S., Chen, G., Dou, D., & Yin, D. (2023). COLTR: Semi-supervised Learning to Rank with Co-training and Over-parameterization for Web Search. *IEEE Transactions on Knowledge and Data Engineering*, 35(12), 1–14. doi:10.1109/TKDE.2023.3270750
- Lian, Y., Sheng, M., Yuan, Y., & Zhou, S. (2021). Research on Fraud Website Identification Technology Based on network Behavior Analysis. *Modern Computer*, 27(28), 33–38.
- Naim, O., Cohen, D., & Ben-Gal, I. (2023). Malicious website identification using design attribute learning. *International Journal of Information Security*, 22(5), 1207–1217. Advance online publication. doi:10.1007/s10207-023-00686-y
- Raja, A. S., Balasubaramanian, S., Ganesan, P., Rajasekaran, J., & Karthikeyan, R. (2023). Weighted ensemble classifier for malicious link detection using natural language processing. *International Journal of Pervasive Computing and Communications*. doi:10.1108/IJPC-09-2022-0312
- Sherstinsky, A. (2020). Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) network. *Physica D. Nonlinear Phenomena*, 404, 132306. doi:10.1016/j.physd.2019.132306
- Tenis, A. A., & Santhosh, R. (2021). Challenges and Security Issues of Online Social Networks (OSN). *Lecture Notes on Data Engineering and Communications Technologies* [Mobile computing and sustainable informatics]. International Conference on Mobile Computing and Sustainable Informatics (ICMCSI), Tribhuvan Univ, Kirtipur, NEPAL.
- Venugopal, S., Panale, S. Y., Agarwal, M., Kashyap, R., & Ananthanagu, U. (2021). Detection of Malicious URLs through an Ensemble of Machine Learning Techniques. [2021 IEEE Asia-Pacific conference on computer science and data engineering (csde)]. 8th IEEE Asia-Pacific Conference on Computer Science and Data Engineering (IEEE CSDE), Electr Network.
- Yan, L., & Kun, C. (2019). Analysis of Bad Website Filtering System on Intelligent Terminal under Cloud Computing. *Proceedings of the 2019 3rd International Conference on Economic Development and Education Management (ICEDEM 2019)*. IEEE. doi:10.1109/ACCESS.2021.3049625
- Zhang, F., & Zhang, W. (2022). Malicious domain names detection based on BERT and hierarchical attention. *Journal of CAEIT*, 17(3), 290–296.
- Zhou, T., Wang, Y., & Zheng, X. (2020). Chinese text classification method using FastText and term frequency-inverse document frequency optimization. *Journal of Physics: Conference Series*, 1693(1), 012121. doi:10.1088/1742-6596/1693/1/012121

Zhu, E., Ye, C., Liu, D., Liu, F., Wang, F., & Li, X. (2018, 11-13 Dec. 2018). An Effective Neural Network Phishing Detection Model Based on Optimal Feature Selection. *2018 IEEE Intl Conf on Parallel & Distributed Processing with Applications, Ubiquitous Computing & Communications, Big Data & Cloud Computing, Social Computing & Networking, Sustainable Computing & Communications (ISPA/IUCC/BDCloud/SocialCom/SustainCom)*. IEEE.

Zhiqiang Wu received his Ph.D. from Northwest Minzu University. He is currently working in the Department of Network Security, Henan Police College. His main research interests: data analysis and visualization.