Computers, Materials & Continua

Tech Science Press

DOI: 10.32604/cmc.2024.048637

ARTICLE

FusionNN: A Semantic Feature Fusion Model Based on Multimodal for Web Anomaly Detection

Li Wang[1,2,3,*], Mingshan Xia[1,2,*], Hao Hu[1], Jianfang Li[1,2], Fengyao Hou[1,2] and Gang Chen[1,2,3]

[1]Institute of High Energy Physics, Chinese Academy of Sciences, Beijing, 100049, China

[2]Spallation Neutron Source Science Center (SNSSC), Dongguan, 523803, China

[3]School of Nuclear Technology, University of Chinese Academy of Sciences, Beijing, 100049, China

*Corresponding Authors: Li Wang. Email: wangli320@ihep.ac.cn; Mingshan Xia. Email: xiams@ihep.ac.cn

Received: 13 December 2023 Accepted: 10 April 2024 Published: 15 May 2024

ABSTRACT

With the rapid development of the mobile communication and the Internet, the previous web anomaly detection and identification models were built relying on security experts' empirical knowledge and attack features. Although this approach can achieve higher detection performance, it requires huge human labor and resources to maintain the feature library. In contrast, semantic feature engineering can dynamically discover new semantic features and optimize feature selection by automatically analyzing the semantic information contained in the data itself, thus reducing dependence on prior knowledge. However, current semantic features still have the problem of semantic expression singularity, as they are extracted from a single semantic mode such as word segmentation, character segmentation, or arbitrary semantic feature extraction. This paper extracts features of web requests from dual semantic granularity, and proposes a semantic feature fusion method to solve the above problems. The method first preprocesses web requests, and extracts word-level and character-level semantic features of URLs via convolutional neural network (CNN), respectively. By constructing three loss functions to reduce losses between features, labels and categories. Experiments on the HTTP CSIC 2010, Malicious URLs and HttpParams datasets verify the proposed method. Results show that compared with machine learning, deep learning methods and BERT model, the proposed method has better detection performance. And it achieved the best detection rate of 99.16% in the dataset HttpParams.

KEYWORDS

Feature fusion; web anomaly detection; multimodal; convolutional neural network (CNN); semantic feature extraction

1 Introduction

With the rapid development of the mobile communication and the Internet, web applications have penetrated into every corner of people's work and lives, becoming the main way for ordinary users to access information, conduct online transactions, and enjoy entertainment services and other daily activities. The openness and convenience of web applications have greatly enriched users' online lives. However, this has also frequently made web applications the target of hacker attacks and network intrusions. In recent years, attack means have become increasingly complex and diverse, and variant attacks have emerged endlessly, posing great challenges to the security protection of web applications [1,2].

As an important means to ensure web system security and normal operation, intrusion detection is crucial for the timely discovery of web attacks. Traditional intrusion detection relies on detection rules and feature models designed by security experts. These rules and models contain signatures of known attack patterns. When network traffic or event logs contain parts that match the defined signatures, corresponding attacks can be effectively identified [3–5]. Although such methods can efficiently detect recorded attack samples, their capability in detecting unknown and variant attacks is very limited. To improve web attack detection, researchers have proposed many anomaly detection methods relying on web traffic analysis. For example, Kruegel et al. [6–8] studied the statistical patterns of parameter value lengths, feature integrity, access order, etc., and built models that can detect anomalous web requests based on these statistical characteristics. However, these methods only utilize traffic-level statistical features and can hardly fully represent the semantic information behind the traffic. As a result, detection models face difficulties in generalizing to attack variants, often causing false alarms in detection results.

In recent years, the development of deep learning and natural language processing technologies has brought new opportunities for web anomaly detection. By deeply analyzing the parameters of network requests, the content of accessed pages, and other natural language information, the semantics of network access behaviors can be inferred to establish an in-depth understanding of network activities. For example, references [9–11] successfully utilized various semantic analysis techniques to process web log texts and implemented automatic identification of network intrusions by combining anomaly detection models. In general, deep learning technologies fully tap the semantic information contained behind the data to achieve automated feature learning and model optimization. This not only greatly improves the detection efficiency, but also endows detection systems with certain adaptability and scalability, providing new ideas for building intelligent and interpretable network security detection systems.

However, existing detection systems are still limited to single semantic representations in semantic feature extraction, such as simple word segmentation or character-level representations, which cannot fully express semantic information. This easily leads to the problems of single and missing semantic features. To obtain richer and more comprehensive semantic features, one feasible approach is to represent information from multiple semantic perspectives simultaneously and integrate the information from these "perspectives" to fully explore the complementarity between different semantic representations.

Based on the above issues, this paper proposes a web anomaly detection model based on semantic feature fusion. The proposed fusion method enhances the complementary advantages between different features. Specifically, our contributions can be summarized as follows:

- We utilize the advantages of word semantics and character semantics to achieve multi-granularity semantic feature extraction and fusion, which enhances the constraint and discrimination of semantic representations.
- From a multi-modal perspective, we carry out semantic feature engineering under multiple views, achieve unified semantic feature representation, and give full play to the complementary effect of multi-source semantics.
- We constructed an end-to-end semantic feature learning, mapping and detection model. By designing three loss functions to reduce semantic feature distance loss, semantic space loss and

classification loss, the goal of fusing semantic features is achieved, which ultimately improves the final detection effect.

The rest of this paper is structured as follows: Section 2 summarizes related work on anomaly detection of web applications. Section 3 introduces the data processing methodology. Section 4 shows the architecture of the proposed model. Section 5 summarizes the datasets, evaluation methods and comparative experiments used, and discusses the experimental results in detail. Finally, we conclude this paper in Section 6.

2 Related Works

Web anomaly detection has been a key issue that has received great attention and research in the network security field. With the development of machine learning methods and deep learning, a large amount of research work on web anomaly detection based on deep learning has been done. In order to achieve better detection performance, researchers focus on different feature representation methods and detection model designs, with the main work summarized in the following two aspects.

2.1 Embedding Techniques

Embedding techniques map discrete symbols to continuous vector spaces to achieve vectorized semantic representation. In recent years, the application of various embedding techniques in web anomaly detection has made great progress.

References [12–14] use word embedding-based deep learning methods to learn features from the data, and take them as model inputs for web anomaly detection. Reference [12] proposes an improved word embedding-based deep learning (WEDL-NIDS) method that learns features from structurally complex data. Reference [13] first decomposes web requests into individual words and does not directly generate word vectors using tools like Word2Vec. Instead, an embedding layer is added in the CNN model to automatically learn word features, and it is verified that automated feature extraction is more effective. Reference [14] adopts one-hot word embedding and Text-CNN to extract significant features from the payload.

References [15–18] study character embedding methods to detect web attacks like malicious URLs, malicious domain names, and suspicious web requests. References [15,16,18] all propose character embedding models that divide web requests into individual characters and automatically learn features through model embedding layers. Differently from the above methods, reference [17] first extracts character vectors through the Skip-gram model, then inputs them into a deep learning model for automated feature learning.

References [19,20] adopt splitting URLs into characters and embedding each character ASCII code for detecting suspicious URLs.

References [21–23] analyze from the n-gram semantic perspective and demonstrate the effectiveness of n-grams in network anomaly detection. Among them, reference [21] uses n-grams to decompose and represent domain names, while the latter two are based on HTTP traffic, one focusing on payload and the other on packets.

In summary, semantic feature extraction enhances the semantic recognition capability of texts and is widely applied to web security detection tasks. However, current research mainly focuses on single-context semantic feature engineering and lacks multi-granularity semantic understanding, which can easily lead to word segmentation errors that have some impact on semantic analysis.

2.2 Semantic Detection Models

Reference [24] adopts the Transformer-based BERT model to learn word features for URL semantic analysis, and uses a CNN model to classify suspicious URLs. Bokolo et al. [25] build a web intrusion detection system using DistilBERT, RNN and LSTM models to identify body attacks, URL attacks and user-data attacks, respectively. Experimental results show that it can recognize various mixed attacks. Halbouni et al. [26] study a CNN and LSTM hybrid intrusion detection system that fuses CNN's advanced semantic feature extraction with LSTM's temporal dependency modeling. Reference [16] studies character-level convolutional neural networks (CLCNN) to detect web application attacks, where each character is represented by an 8-bit number and each HTTP request is represented as a 128-dimensional vector based on the characters it contains. Finally, the CLCNN model is used for classification.

In general, semantic detection models have obvious advantages in detecting web attacks. However, detection models still rely on single semantic modes (word segmentation or character segmentation) for detection, leading to problems of semantic expression singularity. Although reference [27] simultaneously represents URLs at both character and word levels and then concatenates the two representations before anomaly detection using CNN, achieving multi-granularity semantic fusion, such a simple concatenation fusion method can still easily lead to semantic missing.

3 Data Processing

As a text segment, web request contains rich semantic information that fully reflects user intentions and purposes. For example, it contains: Uppercase letters [A–Z], lowercase letters [a–z], numbers [0–9], and special symbols (such as @, $*$, =, $-$). The character composition of the request packet also implies the semantics of user behavior. For example, a large number of numbers and special characters may indicate requests from automated programs or crawlers; while consecutive combinations of lowercase letters are more like manually written query requests. By analyzing character semantic information, the motivations and behavioral attributes of the requesting party can be directly understood. Compared with word semantics, character semantics provides a finer-grained understanding of user behaviors. Therefore, the two-sided processing of the data mainly includes the following processes.

3.1 Character Segmentation

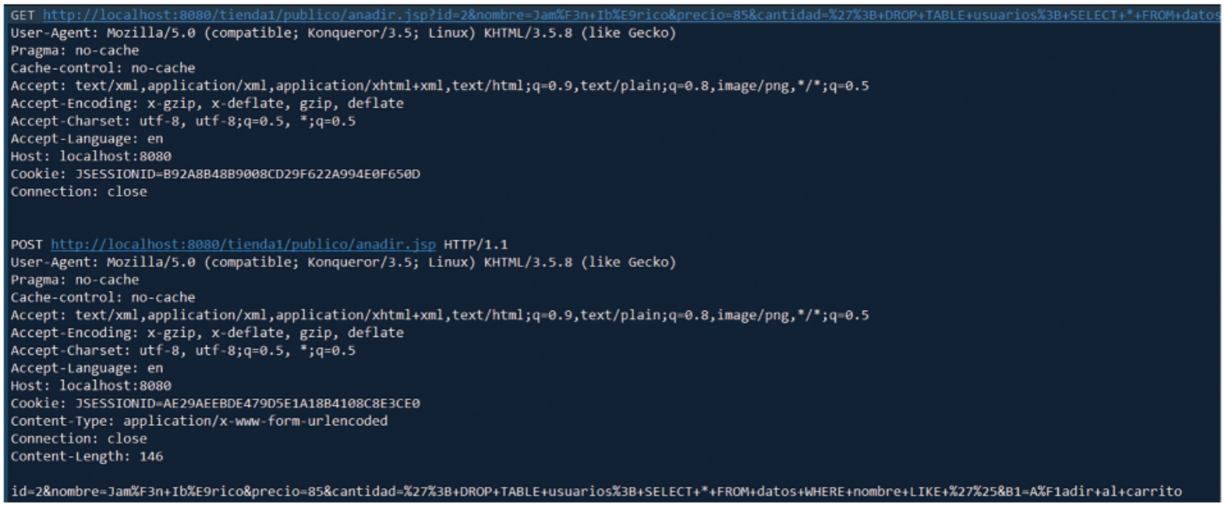
To obtain the character vector representation of a web request, the web request needs to be segmented into characters first. This processing is relatively simple, only requiring splitting the request query string into a sequence of individual characters. For example, after character segmentation, the query string "getpage.php?home=../../../etc/passwd" becomes the sequence ("g", "e", "t", "p", ..., "a", "s", "s", "w", "d"). After performing character segmentation on all web requests in the dataset, the character corpus of this dataset can be obtained, where m represents the number of characters.

3.2 Word Segmentation

To obtain the word vector representation of a web request, the web request needs to be segmented into words first. Query strings generally consist of several key-value pairs separated by the symbol "&", while the key and value in each pair are connected by the symbol "=". Therefore, this paper uses "&", "=", and other special symbols as delimiters to divide the request query string into sequences of several words and symbols, in order to maintain the integrity of the web request semantics as much as possible. For example, after word segmentation, the query string "getpage.php?home=../../etc/passwd" becomes the sequence (“getpage”, “php”, “home”, “..”, “..”, ..., “etc”, “passwd”). After word segmentation of all web requests in the dataset, the word corpus of this dataset can be obtained, where n represents the number of words.

3.3 *Processing of CSIC 2010*

Different from Malicious URLs [28] and HttpParams [29], the HTTP CSIC 2010 [30] dataset is relatively complex as shown in Fig. 1. The requests contain four basic HTTP methods: GET, POST, PUT and DELETE. This paper first needs to extract key contents from the logs, including the URL and Body Parts after the GET, POST, and PUT request methods. Then perform word segmentation and character segmentation on the obtained logs.



```
GET http://localhost:8080/tienda1/publico/anadir.jsp?id=2&nombre=Jam%F3n+Ib%E9rico&precio=85&cantidad=%27%3B+DROP+TABLE+usuarios%3B+SELECT+*+FROM+datos
User-Agent: Mozilla/5.0 (compatible; Konqueror/3.5; Linux) KHTML/3.5.8 (like Gecko)
Pragma: no-cache
Cache-control: no-cache
Accept: text/xml,application/xml,application/xhtml+xml,text/html;q=0.9,text/plain;q=0.8,image/png,*/*;q=0.5
Accept-Encoding: x-gzip, x-deflate, gzip, deflate
Accept-Charset: utf-8, utf-8;q=0.5, *;q=0.5
Accept-Language: en
Host: localhost:8080
Cookie: JSESSIONID=B92A8B48B9008CD29F622A994E0F650D
Connection: close


POST http://localhost:8080/tienda1/publico/anadir.jsp HTTP/1.1
User-Agent: Mozilla/5.0 (compatible; Konqueror/3.5; Linux) KHTML/3.5.8 (like Gecko)
Pragma: no-cache
Cache-control: no-cache
Accept: text/xml,application/xml,application/xhtml+xml,text/html;q=0.9,text/plain;q=0.8,image/png,*/*;q=0.5
Accept-Encoding: x-gzip, x-deflate, gzip, deflate
Accept-Charset: utf-8, utf-8;q=0.5, *;q=0.5
Accept-Language: en
Host: localhost:8080
Cookie: JSESSIONID=AE29AEEBDE479D5E1A18B4108C8E3CE0
Content-Type: application/x-www-form-urlencoded
Connection: close
Content-Length: 146

id=2&nombre=Jam%F3n+Ib%E9rico&precio=85&cantidad=%27%3B+DROP+TABLE+usuarios%3B+SELECT+*+FROM+datos+WHERE+nombre+LIKE+%27%25&B1=A%F1adir+al+carrito
```

Figure 1: Log of HTTP CSIC 2010

4 The Proposed Model

Since word semantics represents the associations between words, and character semantics represents fine-grained semantic information within words, we propose a semantic feature fusion model for web anomaly detection. The overall architecture of the proposed model is shown in Fig. 2. The model mainly contains three components named CharCNN for extracting character semantic features, WordCNN for extracting word semantic features, and the fusion neural network FusionNN for feature fusion. We obtain word and character semantic features from the first two parts, then fuse them in the third part to obtain the final fused features.

4.1 *FusionNN*

The proposed FusioNN is a feedforward neural network, containing two CNNs, namely CharCNN for extracting character semantic features and WordCNN for extracting word semantic features, and is trained in an end-to-end manner. The two CNNs have the same internal structure, defined to contain 2 hidden layers and one fully connected layer. Finally, features are output through two fully connected layers. The model is validated using mean Average Precision (mAP). The Sigmoid activation function is used during the final classification.

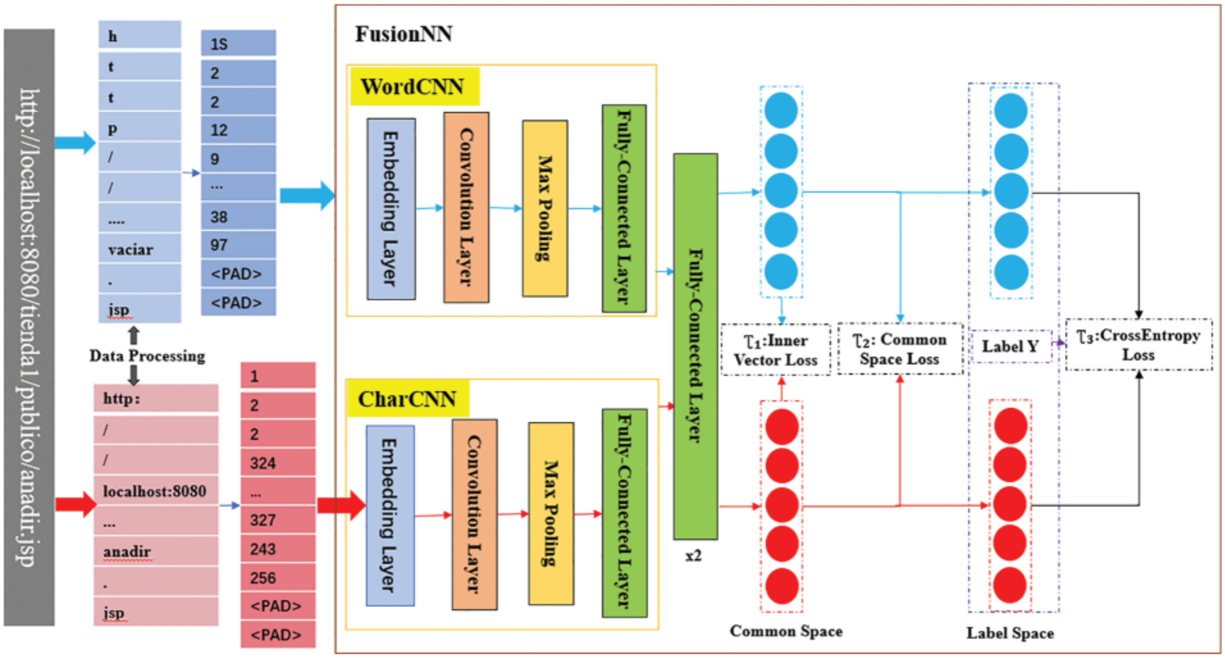


Figure 2: The proposed model

1) Embedding Layer

Embedding layers can convert text into mathematical vectors, serving as a bridge for language analysis between humans and computers, and playing an important role in text classification. Similarly, in web anomaly detection, embedding models play a key role in representing deep models for web anomaly detection. This paper uses the CNN model to learn the vector representation of web requests. We map each word or each character to 256 dimensions.

2) Convolution Layer

In order to improve the expressive ability of CNN and fully learn the diversity of features during classification, the input data is extracted through three convolutional layers, then the outputs of the three layers are concatenated to convert the input data into feature representations of higher levels. Each Convolution Layer is followed by a ReLU and a Max Pooling Layer. The kernel size is set to {3, 4, 5} and stride = 1.

3) Full-Connected Layer

Full-connected layers exist in WordCNN, CharCNN and FusionNN. Among them, in the two CNNs, the fully connected layers are equivalent to the last layer of CNN neural networks, outputting the character-level and word-level features of each web request, as inputs to the fully connected layers inside FusionNN. The roles of the two fully connected layers in FusionNN are: The first layer extracts features for mean Average Precision (mAP) calculation after feature fusion, and the last layer obtains classification features to calculate the final classification result.

The fully connected layer is the critical part of the model. Through the design of three losses in this layer, the word-based and character-based semantic features are fused into a joint representation.

4.2 Feature Fusion

To formalize the problem studied in this work, the following notation is used: $U = \{u_1, u_2 \ldots u_n\}$, $V = \{v_1, v_2 \ldots v_n\}$, $Y = \{y_1, y_2 \ldots y_n\}, y_i = \{0,1\}$. u_i and v_i represent the tokenized words of a web request and embedded character vectors, respectively. y_i is the label matrix. Therefore, when the number of input web requests is n, all web requests can be represented as pairs of words and character vectors and denoted as: $\Psi = \{(u_i, v_i)\}_{i=1}^{n}$

Common feature fusion methods like concatenation or direct addition of multiple feature vectors, though simple and efficient, can cause loss of information. To avoid the information loss caused by simple concatenation or the addition of features, we designed a semantic feature fusion method to combine dual-granularity semantics and achieve the goal of enhancing semantic features. Inspired by multi-modal technologies [31], we perform semantic feature fusion. Therefore, to ensure decreasing distance between the same labels and increasing distance between different labels, three loss functions are established during the mapping process.

1) Semantic Feature Loss

U and V represent the word-based and character-based semantic features. In order to eliminate the differences between the two semantics, we propose to minimize the distance between the representations of all Word-Char pairs. Therefore, the loss function is obtained by calculating the distance between the two features. Technically, we formulate the semantic invariance loss as follows:

$$\mathcal{T}_1 = \frac{1}{n}||U - V||_{F.} \tag{1}$$

2) Discriminative Loss in the Common Space [31]

We hypothesize that the two semantic features are mapped to the same public space. The designed loss function can make the vector representations of Web requests from the same category closer, while increasing the vector distance between Web requests from different categories and making the differences more distinct.

$$\begin{aligned}\mathcal{T}_2 = & \frac{1}{n^2}\sum\nolimits_{i,j=1}^{n}\left(\log\left(1 + e^{\Gamma_{ij}}\right) - S_{ij}^{uv}\Gamma_{ij}\right) + \frac{1}{n^2}\sum\nolimits_{i,j=1}^{n}\left(\log\left(1 + e^{\Phi_{ij}}\right) - S_{ij}^{uu}\Phi_{ij}\right) \\ & + \frac{1}{n^2}\sum\nolimits_{i,j=1}^{n}\left(\log\left(1 + e^{\Theta_{ij}}\right) - S_{ij}^{vv}\Theta_{ij}\right)\end{aligned} \tag{2}$$

In the formula $\Gamma_{ij} = \frac{1}{2}\cos(u_i, v_j)$, $\varnothing_{ij} = \frac{1}{2}\cos(u_i, u_j)$, $\theta_{ij} = \frac{1}{2}\cos(v_i, v_j)$ and $\cos(\cdot)$ is used to calculate the cosine similarity between two vectors. $S_{ij}^{uv} = 1\{u_i, v_j\}$, $S_{ij}^{uu} = 1\{u_i, u_j\}$, $S_{ij}^{vv} = 1\{v_i, v_j\}$. $1\{.\}$ is an indicator function, which takes the value 1 when the representations of the two elements belong to the same category, and 0 otherwise. The formula contains three parts: Word and character modality calculations, loss between words and between characters. The goal of this design is to maximize the distance between word and character representations from different categories, and minimize the distance between word and character representations from the same category.

3) Classification Loss

In order to minimize the loss between features and labels, a cross entropy loss function is adopted. This loss function calculates the sum of character classification loss and word classification loss, which is formulated as follows:

$$\mathcal{T}_3 = \mathcal{T}_w + \mathcal{T}_c = \frac{1}{n}\sum_i -[y_i \cdot \log(p_i) + (1 - y_i) \cdot \log(1 - p_i)] + \frac{1}{n}\sum_j -[y_j \cdot \log(p_j) + (1 - y_j) \cdot \log(1 - p_j)] \quad (3)$$

$\mathcal{T}_w$ and $\mathcal{T}_c$ represent the classification loss for characters and words, respectively.

4) Final Loss

Based on formulas (1) to (3), the final loss function is obtained:

$$\mathcal{T} = \lambda\mathcal{T}_1 + \eta\mathcal{T}_2 + \mathcal{T}_3 \quad (4)$$

The parameters λ and η are the loss weight coefficients optimized by the Adma algorithm. Through the final loss function, better representations can be generated in the common representation space, so that words and characters with the same semantics are closer in the common space, and the similarity between different types of data can be measured.

5 Experiments and Evaluation

We conducted different experiments to evaluate the performance of the proposed method on three publicly available standard datasets and our experimental environment is Win10 operating system with i7 CPU processing and 128 GB RAM.

5.1 Datasets

The proposed model is validated on the public dataset HTTP CSIC 2010, Malicious-URLs, and HttpParams Dataset, which have been used for web attack detection in multiple papers [13,15–17,23,24].

HTTP CSIC 2010 [30]: The dataset contains 36,000 legitimate web requests and over 25,000 malicious requests automatically generated by CSIC (Spanish National Research Council). HTTP requests are labeled as normal or anomalous. The dataset contains various attacks such as SQL injection, buffer overflow, information gathering, file leakage, CRLF injection, cross-site scripting, parameter tampering, etc. For the original HTTP request data, this paper mainly extracts the GET, POST and PUT request data for intrusion detection. 80% of the dataset is used for training and 20% is used for testing.

Malicious-URLs [28]: As a labeled dataset, the Malicious-URLs dataset is an open source project on Github for malicious URL detection. The data contains two files ending with csv. We use the data.csv file as the dataset, containing over 340,000 normal data and over 70,000 anomalous data. The specific distribution of test set and training set is shown in Table 1.

Table 1: Malicious-URLs experiment data distribution

Dataset	Good data		Bad data	
Malicious-URLs dataset	Train	Test	Train	Test
	50400	10000	20000	7600

HttpParams Dataset [29]: This dataset is the anomaly detection dataset in Morzeux's graduation thesis, recording parameter values in HTTP requests, with over 30,000 normal and anomalous data. The anomalous data contains four types of attacks. payload_train.txt is used for training and payload_text.txt is used for testing.

5.2 Evaluation Methods and Model Parameters

The validity of the model is compared by Accuracy, Precision, Recall and F1-Score to compare the performance of each method. Accuracy and F1 are used to evaluate the effectiveness of the model. Accuracy refers to the proportion of correctly predicted positive and negative examples out of the total examples. Precision is the proportion of correctly predicted positive examples out of examples predicted positive based on prediction results. Recall refers to the proportion of correctly predicted positive examples out of actually positive examples based on actual samples. F1-Score considers both Precision and Recall, achieving harmony between them. The Accuracy, Precision, Recall and F1 of our proposed method are the mean of word classification evaluation values and character classification evaluation values, respectively.

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{5}$$

$$Precision = \frac{TP}{TP + FP} \tag{6}$$

$$Recall = \frac{TP}{TP + FN} \tag{7}$$

$$F1 = \frac{2 * (Precision * Recall)}{(Precision + Recall)} \tag{8}$$

Since user requests may contain multiple words or characters, we set sentence length to 200 for word requests and 1024 for character requests. 0 padding is used if the sentence length is less than the set length. The experimental parameters are as shown in Table 2 and the settings of λ and η are inspired by [26].

Table 2: Experiment parameters

Learning rate	η	λ	Batch size	EPOCH
0.0001	0.1	0.001	512	50

5.3 Experiments and Result

Our goal is to improve web anomaly detection performance through the semantic feature fusion model. Therefore, we conduct comparisons from three aspects. First, we compare against manually extracted features and machine learning classifiers. Second, we compare against automated semantic feature extraction methods and anomaly detection models [24]. Third, we compare against deep learning models. According to current research, CNN models have general applicability for web attack detection [13,14,16,19,25]. Hence, experiments also compared the CNN model combined with semantic features. The results are listed below.

5.4 Comparisons Study

We clearly compare the proposed method to machine learning baselines with manually engineered features on a relevant dataset (CSIC 2010). The features in Table 3 cover important aspects like URL length and request types. Table 4 shows our experimental results with an accuracy of 96.68%, precision of 96.64%, recall of 97.84%, and F1-Score of 97.23%, demonstrating great advantages over traditional machine learning methods like K-nearest neighbor (KNN), Logistic regression (LR), Support vector machine (SVM) and Decision tree (DT) using manually extracted features. Quantitative results in Table 4 demonstrate superior performance of the proposed method over several ML methods across accuracy, precision, recall and F1.

Table 3: Ten features by handcrafted

Parameters	Description
Url_length	The length of the URL
Request_type	Request type (Post, Get . . .)
Parameter_length	The length of request parameters
Parameter_number	The number of request parameters
Digits_number	The number of digits in the parameter value
Digits_proportion	The proportion of digits in the parameter value
Special_char_number	The number of special characters in the parameter value
Special_char_ proportion	The proportion of special characters in the parameter value
Special_char_url_number	The number of special characters in the URL
Special_char_url_ proportion	The proportion of special characters in URL

Table 4: COMPARISON of traditional machine learning methods on CSIC 2010

Methods	Dataset	Accuracy	Precision	Recall	F1-Score
KNN	CSIC 2010	86.36%	87.85%	89.23%	88.53%
LR	CSIC 2010	69.31%	78.04%	66.75%	71.95%
SVM	CSIC 2010	71.11%	76.94%	72.89%	74.86%
DT	CSIC 2010	80.42%	78.23%	92.59%	84.81%
Our method	CSIC 2010	**96.68%**	**96.64%**	**97.84%**	**97.23%**

Moreover, since the FusionNN model designed in this paper internally includes two CNN models, and adopts a fusion mechanism of word-based and character-based semantic features. In order to verify the advantages of the proposed fusion method, this paper selected character-embedded CNN, word-embedded CNN, and CNN-GRU models based on character embedding, such as Word-CNN-GRU [32], Word-CNN [20], Char-CNN [17]. In order to more fully contrast the advantages of the proposed fused semantic features, we added a comparison of the n-gram [21] semantic feature method. The comparison results are given in Table 5. However, we achieve higher accuracy, precision and F1-Score, except for recall compared to these deep learning studies. The low recall rate may be caused by two reasons: Insufficient positive sample learning, and representing unknown words or characters as zero vectors in Word2Vec, which leads to confusion in data learning. To address this, we have conducted a study on dynamic unknown word embedding in another work, and achieved significantly improved recall results. This work has been accepted by a journal.

Table 5: COMPARISON of deep learning methods on CSIC 2010

Methods	Dataset	Accuracy	Precision	Recall	F1-Score
CHAR-CNN	CSIC 2010	92.99%	89.80%	99.43%	94.36%
WORD-CNN	CSIC 2010	93.86%	95.82%	93.68%	94.73%
CHAR-CNN-GRU	CSIC 2010	94.22%	95.55%	94.61%	96.86%
WORD-CNN-GRU	CSIC 2010	93.05%	89.69%	99.69%	94.43%
N-gram	CSIC 2010	91.57%	90.43%	95.86%	93.07%
Our method	CSIC 2010	**96.68%**	**96.64%**	**97.84%**	**97.23%**

We also used the Malicious-URLs and HttpParams datasets to evaluate the robustness of the model by increasing the number of positive samples. The results in Tables 6 and 7 show that our proposed method still has better results on different datasets and the proposed semantic fusion method can obtain richer semantic information compared to character-level or word-level semantics alone, improving the detection performance of the model.

Table 6: COMPARISON of deep learning methods on malicious-URLs dataset

Methods	Dataset	Accuracy	Precision	Recall	F1-Score
CHAR-CNN	Malicious-URLs	94.94%%	96.14%	94.92%	95.53%
WORD-CNN	Malicious-URLs	95.40%	96.40%	95.48%	95.93%
Our method	Malicious-URLs	**97.93%**	**96.64%**	**99.84%**	**98.21%**

Table 7: COMPARISON of deep learning methods on HttpParams dataset

Methods	Dataset	Accuracy	Precision	Recall	F1-Score
CHAR-CNN	HttpParams	97.79%	98.14%	96.28%	97.20%
WORD-CNN	HttpParams	98.50%	98.83%	97.18%	97.99%
Our method	HttpParams	**99.16%**	**99.95%**	**97.83%**	**98.88%**

To fully demonstrate the effectiveness of the model, we compared with semantic models. Reference [24] proposed using BERT model for web-based attack detection, verified on both HTTP CISC2010 and HttpParams datasets, with the highest accuracy of only 96%, similar to our experimental result on HTTP CISC2010. But for HttpParams, their result is lower than ours by over 3%, fully proving that our proposed feature fusion method enables better representation of semantic features. Since the HttpParams dataset is for payload anomaly detection, the semantic feature fusion effect is significantly better than HTTP CICS2010. Figs. 3 to 5 show the accuracy and loss during model training and validation. The comparative experimental results on the three datasets are shown in Fig. 6 that model achieves the best validation performance on the HttpParams dataset. Through in-depth study, we found this dataset does not contain domain name parts in the request and has larger differences between positive and negative samples, which may explain why detection on this dataset has better results over others.

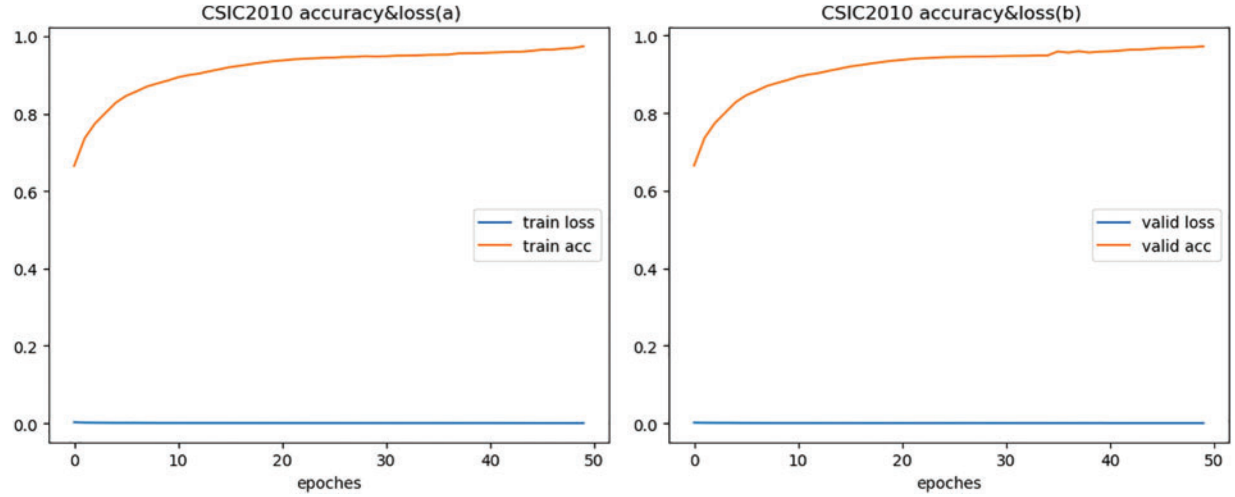


Figure 3: Accuracy & loss of HTTP CSIC 2010

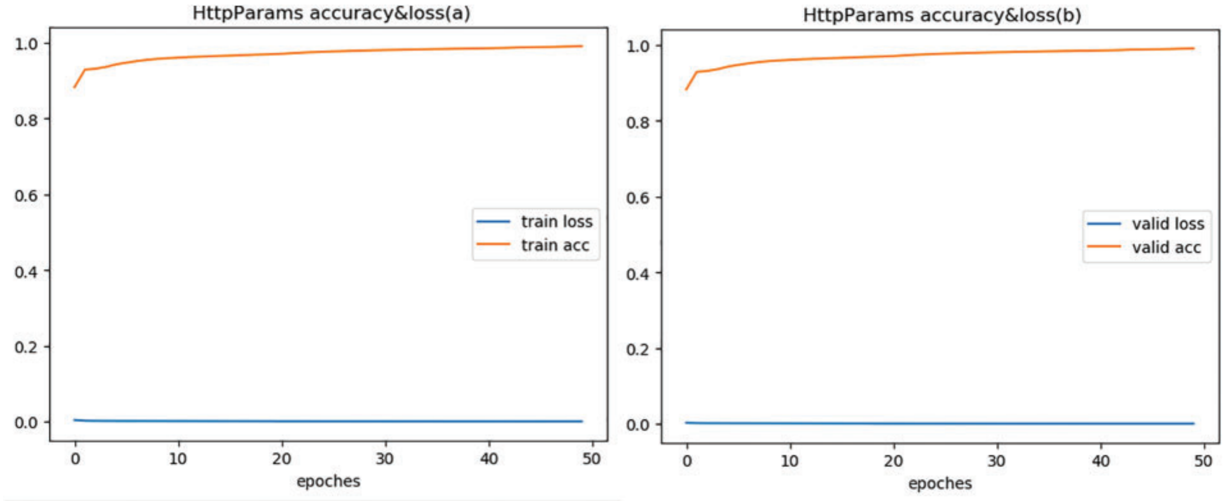


Figure 4: Accuracy & loss of HttpParams

5.5 Ablation Experiments

In the training process, three loss functions were used for learning optimization. In order to further analyze the contributions and influences of the three loss functions in the model on the final results, we conducted ablation experiments. The experimental results are shown in Table 8.

In the aforementioned Table 8, L1 represents not using the inter-semantic feature loss, L2 represents not including the spatial discrimination loss, focusing only on the fusion of semantic features without explicitly constraining the distribution of semantic features; L3 represents the inclusion of all three loss functions, integrating mono-modal semantic feature learning, multi-modal semantic feature difference elimination, and constraints on the distribution of semantic features.

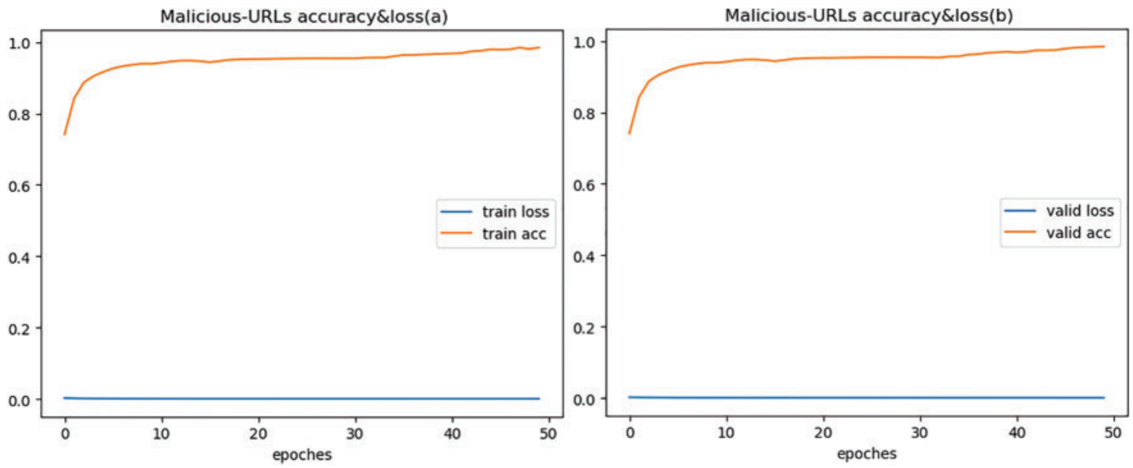


Figure 5: Accuracy & loss of Malicious-URLs

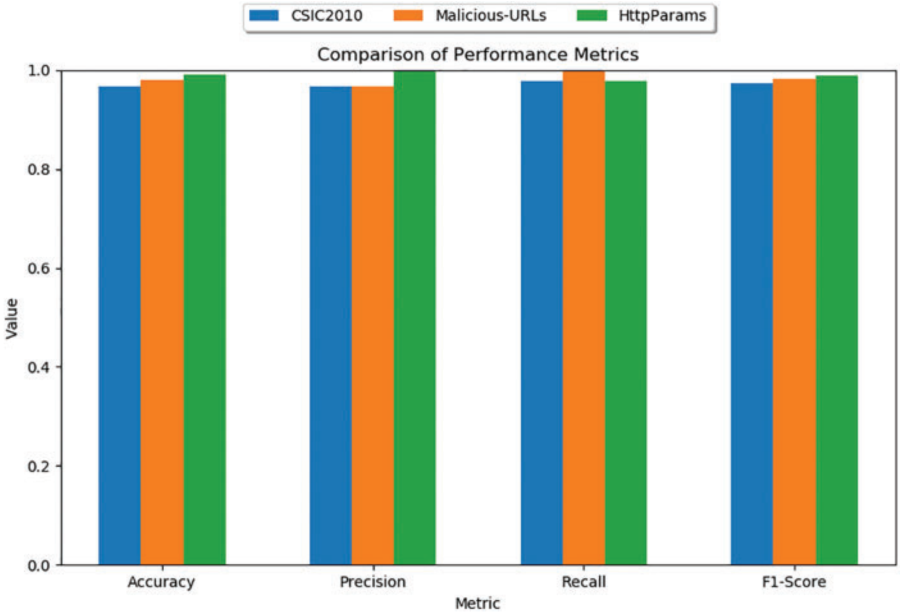


Figure 6: Results of three datasets

Table 8: Performance of three loss on HttpParams dataset

Methods	Dataset	Accuracy	Precision	Recall	F1-Score
L1	HttpParams	98.94%	99.93%	97.27%	98.57%
L2	HttpParams	99.04%	99.91%	97.55%	98.71%
L3	HttpParams	**99.16%**	**99.95%**	**97.83%**	**98.88%**

The experimental results on the HttpParams dataset indicate:

1) L3 achieves the best detection performance metrics, demonstrating that the rational design and integration of the three loss functions are beneficial to the final detection results. They can complement each other and play to their respective strengths;
2) L3 shows a significant improvement over L1, which validates the importance of the inter-semantic feature loss L1. It effectively reduces the differences between semantic features from different modalities, leading to better feature fusion;
3) L3 also outperforms L2, indicating that explicitly constraining the distribution of semantic features and introducing discriminative losses have a positive effect on guiding the learning of the fused semantic features, contributing to the extraction of more discriminative feature representations.

In summary, by rationally designing and integrating three complementary loss functions, the constructed multi-modal anomaly detection model can fully utilize multi-modal input data to learn more discriminative semantic feature representations, thereby achieving superior detection performance.

6 Conclusion

This paper proposes a web anomaly detection model with semantic feature fusion. The model treats request URLs with certain semantics as strings, and extracts text models of the URL requests from word and character aspects, respectively. It utilizes CNN to extract semantic features, then studies common vector representations through multi-granularity semantic feature fusion technology, and finally uses them for classification. Multiple comparative experiments demonstrate that the method has good performance on the CSIC 2010, Malicious-URLs, and HttpParams datasets, and can learn discriminative and expressive vector representations of URL requests. Among them, the results on the HttpParams dataset are the best. Next, we will continue to explore more effective classification algorithms with better performance.

Acknowledgement: The author would like to extend sincere gratitude to the methodologies that have significantly contributed to and facilitated this research.

Funding Statement: This work was supported by a grant from the National Natural Science Foundation of China (Nos. 11905239, 12005248 and 12105303).

Author Contributions: The authors declare their individual contributions to this paper as follows: Li Wang was responsible for conceiving and designing the study as well as collecting the data. Mingshan Xia and Hao Hu performed an analysis and interpretation of the results. Li Wang drafted the initial manuscript. Jianfang Li and Gang Chen modified the final manuscript. All authors subsequently reviewed the findings, results and approved the final manuscript prior to submission.

Availability of Data and Materials: Not applicable.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] S. Applebaum, T. Gaber, and A. Ahmed, "Signature-based and machine-learning-based web application firewalls: A short survey," in *Proc. 2021 5th Int. Conf. AI Comput. Linguist.*, vol. 189, no. 10, pp. 359–367, 2021. doi: 10.1016/j.procs.2021.05.105.

[2] S. Kim, J. S. Kim, and H. P. In, "Multitier web system reliability: Identifying causative metrics and analyzing performance anomaly using a regression model," *Sensors*, vol. 23, no. 4, pp. 1919, 2023. doi: 10.3390/s23041919.

[3] M. Roesch, "Snort-Lightweight intrusion detection for networks," in *Proc. USENIX LISA '99 Conf.*, Seattle, WA, USA, Nov. 1999.

[4] G. Vigna, W. Robertson, V. Kher, and R. A. Kemmerer, "A stateful intrusion detection system for world-wide web servers," in *Proc. Annual Comput. Secur. Appl. Conf. (ACSAC 2003)*, Las Vegas, NV, USA, Dec. 2003, pp. 34–43.

[5] V. Paxson, "Bro: A system for detecting network intruders in real-time," in *Proc. 7th USENIX Secur. Symp.*, San Antonio, TX, USA, Jan. 1998.

[6] C. Kruegel and G. Vigna, "Anomaly detection of web-based attacks," in *Proc. 10th ACM Conf. Comput. Commun. Secur.*, 2003, pp. 251–261.

[7] C. Kruegel, G. Vigna, and W. Robertson, "A multi-model approach to the detection of web-based attacks," *Comput. Netw.*, vol. 48, no. 5, pp. 717–738, 2005. doi: 10.1016/j.comnet.2005.01.009.

[8] W. Robertson, G. Vigna, C. Kruegel, and R. A. Kemmerer, "Using generalization and characterization techniques in the anomaly-based detection of web attacks," in *Proc. Netw. Distrib. Syst. Secur. Symp., NDSS*, San Diego, California, USA, 2006.

[9] M. Siwach and M. Suman, "Anomaly detection for web log data analysis using weighted PCA technique," *J. Inform. Optim. Sci.*, vol. 43, no. 1, pp. 131–141, 2022.

[10] J. Yu, D. Tao, and Z. Lin, "A hybrid web log based intrusion detection model," in *Proc. CCIS*, Beijing, China, 2016, pp. 356–360.

[11] M. Siwach and S. Mann, "Anomaly detection for web log based data: A survey," in *Proc. DELCON*, New Delhi, India, 2022, pp. 1–5.

[12] J. Cui, J. Long, E. Min, and Y. Mao, "WEDL-NIDS: Improving network intrusion detection using word embedding-based deep learning method," in *Proc. 15th Int. Conf. Model. Deci. Artif. Intell.*, Mallorca, Spain, 2018, vol. 11144, pp. 283–295.

[13] M. Zhang *et al.*, "A deep learning method to detect web attacks using a specially designed CNN," in *Proc. Int. Conf. Neural Inform. Process. ICONIP 2017*, Guangzhou, China, 2017, vol. 10638, pp. 828–836.

[14] E. Min, J. Long, Q. Liu, J. Cui, and W. Chen, "TR-IDS: Anomaly-based intrusion detection through text-convolutional neural network and random forest," *Secur. Commun. Netw.*, vol. 2018, no. 1, pp. 9–19, 2018. doi: 10.1155/2018/4943509.

[15] J. Saxe and K. Berlin, "eXpose: A character-level convolutional neural network with embeddings for detecting malicious urls, file paths and registry keys," arXiv:1702.08568, 2017.

[16] J. Wu, Z. Yang, L. Guo, Y. Li, and W. Liu, "Convolutional neural network with character embeddings for malicious web request detection," in *Proc. 2019 IEEE Int. Conf. Parallel & Distrib. Process. Appl., Big Data & Cloud Comput., Sustain. Comput. & Commun., Social Comput. & Netw. (ISPA/BDCloud/SocialCom/SustainCom)*, Xiamen, China, 2019, pp. 622–627.

[17] M. Ito and H. Iyatomi, "Web application firewall using character-level convolutional neural network," in *Proc. 2018 IEEE 14th Int. Colloquium Signal Process. Appl. (CSPA)*, Penang, Malaysia, 2018, pp. 103–106.

[18] J. Wang, Z. Zhou, and J. Chen, "Evaluating CNN and LSTM for web attack detection," in *Proc. 2018 10th Int. Conf. Mach. Learn. Comput. (ICMLC '18)*, New York, NY, USA, 2018, pp. 283–287.

[19] I. Jemal, M. A. Haddar, O. Cheikhrouhou, and A. Mahfoudhi, "M-CNN: A new hybrid deep learning model for web security," in *Proc. 2020 IEEE/ACS 17th Int. Conf. Comput. Syst. Appl. (AICCSA)*, Antalya, Turkey, 2020, pp. 1–7.

[20] I. Jemal *et al.*, "ASCII embedding: An efficient deep learning method for web attacks detection," in *Proc. Mediterranean Conf. Pattern Recognit. Artif. Intell.*, Instanbul, Turkey, 2020, vol. 1322, pp. 286–297.

[21] R. Pal and N. Chowdary, "Statistical profiling of n-grams for payload based anomaly detection for HTTP web traffic," in *Proc. the 2018 IEEE Int. Conf. Adv. Netw. Telecommun. Syst. (ANTS)*, Indore, India, 2018, pp. 1–6.

[22] W. Khreich, B. Khosravifar, A. Hamou-Lhadj, and C. Talhi, "An anomaly detection system based on variable N-gram features and one-class SVM," *Inform. Software. Tech.*, vol. 91, no. 5, pp. 186–197, 2017. doi: 10.1016/j.infsof.2017.07.009.

[23] Z. Zhang, R. George, and K. Shujaee, "Efficient detection of anomolous HTTP payloads in networks," in *Proc. SoutheastCon 2016*, Norfolk, VA, USA, 2016, pp. 1–3.

[24] Y. E. Seyyar, A. G. Yavuz, and H. M. Ünver, "Detection of web attacks using the BERT model," in *Proc. 2022 30th Signal Process. Commun. Appl. Conf. (SIU)*, Safranbolu, Turkey, 2022, pp. 1–4.

[25] B. G. Bokolo, L. Chen, and Q. Liu, "Detection of web-attack using DistilBERT, RNN, and LSTM," in *Proc. 2023 11th Int. Symp. Digital Foren. Secur. (ISDFS)*, Chattanooga, TN, USA, 2023, pp. 1–6.

[26] A. Halbouni, T. S. Gunawan, M. H. Habaebi, M. Halbouni, M. Kartiwi and R. Ahmad, "CNN-LSTM: Hybrid deep neural network for network intrusion detection system," *IEEE Access*, vol. 10, pp. 99837–99849, 2022. doi: 10.1109/ACCESS.2022.3206425.

[27] H. Le, Q. Pham, D. Sahoo, and S. C. Hoi, "URLNet: Learning a URL representation with deep learning for malicious URL detection," arXiv:1802.03162, 2018.

[28] F. Ahmad, "Malicious-URLs Dataset," Feb. 18, 2017. [Online]. Available: https://github.com/faizann24/Using-machine-learning-to-detect-malicious-URLs

[29] Morzeux, "HttpParams Dataset," May 17, 2016. [Online]. Available: https://github.com/Morzeux/HttpParamsDataset

[30] march0, "HTTP CSIC 2010," Jun. 11, 2019. [Online]. Available: https://github.com/march0/CSIC-2010-HTTP-dataset-classificatin-with-TFidf/tree/master/data

[31] L. Zhen, P. Hu, X. Wang, and D. Peng, "Deep supervised cross-modal retrieval," in *Proc. 2019 IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Long Beach, CA, USA, 2019, pp. 10386–10395.

[32] Q. Niu and X. Li, "A high-performance web attack detection method based on CNN-GRU model," in *Proc. 2020 IEEE 4th Inform. Technol. Netw. Elect. Autom. Control Conf. (ITNEC)*, Chongqing, China, 2020, pp. 804–808, doi: 10.1109/ITNEC48623.2020.9085028.