Journal of Network Intelligence ©2024 ISSN 2414-8105 (Online)
Taiwan Ubiquitous Information Volume **9**, Number **3**, August **2024**

Web Threat Recognition Model Based on Anomaly Detection and Attention Mechanism Seq2Seq Network

Guang-Hua Zhang, Kai-Di Zhang
School of Information Science and Engineering
Hebei University of Science Technology, Shijiazhuang 050018, China
xian_software@163.com, kaidi_zhang2017@126.com

Jia Yan*
Finance Department
Hebei Youth Management Cadre College, Shijiazhuang 050031, China
yanjia0327@163.com

*Corresponding author: Jia Yan

Received September 12, 2023, revised December 22, 2023, accepted March 20, 2024.

Abstract. *With the rapid development of the Internet, Web threat identification is crucial. However, the existing Web threat recognition models have the following problems: inability to identify untrained attacks, poor overall performance, and limited generalization ability. To address these problems, this paper proposes a web threat detection model based on anomaly detection and the attention mechanism of Seq2Seq networks. Firstly, a vocabulary based on ASCII codes is established for data processing, and the Seq2Seq network is utilized for encoding and decoding operations to enhance the model's generalization ability. Secondly, an anomaly detection approach is employed to train the model only on normal requests, thereby detecting attacks that have not been encountered during training. Finally, the attention mechanism is introduced to allocate different weights and scores to each time step, enhancing the model's performance.Experimental results demonstrate that the proposed model performs better than existing models.*

Keywords: anomaly detection; attention mechanism; Seq2Seq network; LSTM; Web threats

1. **Introduction.** The Internet has become an integral component of contemporary lifestyles, and numerous socially significant activities rely on Web applications. Therefore, protecting Web applications from intrusion is essential. When developing Web-based applications, however, security is not sufficiently considered. The majority of all types of vulnerabilities are caused by the transmission of malevolent HTTP requests that have been manipulated. According to the most recent OWASP report [1], up to 94%of Web applications have been attacked in some way, such as Cross-Site Scripting (XSS) [2], SQL injection [3], file upload vulnerabilities [4], and authentication vulnerabilities [5].

For Web attack defense, various threat identification models have been proposed. The vast majority of conventional models are rule-based, relying on expert knowledge, building domain-specific rules based on known threats, and identifying malicious threats by comparing extracted signatures with predefined rules using an efficient engine. On the one hand, rule-based models have unsatisfactory accuracy; on the other, they are excessively dependent on existing knowledge; and the burden of manually identifying threats is immense. Its applicability in information security is progressively expanding as artificial intelligence advances [6-8]. In the field of threat recognition, AI is primarily of two types: the first is based on supervised learning and searches for specific attack patterns in requests. Li et al. [9] employed TF-IDF weighted mapping of HTTP traffic and supervised algorithms LightGBM and CatBoost to build an HTTP traffic detection model. Li et al. [10] extracted the features of web images and introduced TSVM to detect phishing pages. Rout et al. [11] used the PU learning method to improve the classification F-score metric to detect comment spam. The second strategy was based on anomaly detection, establishing normal request description methods to differentiate abnormal requests from regular ones. Maseer et al. [12] used the unsupervised SOM algorithm to test it by launching a web attack on the CICIDS2017 data set. Yang et al. [13] showed that unsupervised learning and semi-supervised learning were the development directions of network anomaly detection through statistics. Vigna et al. [14] improved the detection rate of the system by using a serial combination of a web anomaly detector and a SQL query anomaly detector. The false alarm rate and accuracy of supervision-based methods are lower and higher, respectively, but they cannot identify attacks other than those for which the model is well-trained. The approach based on anomalies can identify attacks that are not trained, but its detection is inadequate.

This paper aims to overcome the limitations of the above methods and improve the performance of Web threat recognition by using a network based on anomaly detection with attention mechanism LSTM structure as a threat recognition model. The advantages of the LSTM network structure based on anomaly detection with attention mechanism are as follows: First, on the basis of anomaly detection, an unsupervised anomaly detector trained solely on normal request data is applicable to a variety of scenarios and capable of detecting various malicious activities. Once the model is trained, only deviations from normal data must be searched for in new datasets without accessing the maliciously labeled dataset, thereby overcoming the problem of limited availability of anomalously requested data. Second, the model is not limited to trained attacks alone. It is impossible to model all possible attack types in advance, so anomaly detection models play a more significant role in practice than supervised models, which can only identify attack types that have been trained. Thirdly, the proposed LSTM deep structure introduces an attention mechanism that automatically extracts more pertinent and representative features from the request data, thereby eliminating the need to manually extract features. Fourthly, the proposed attention mechanism LSTM structure incorporates sequence-to-sequence (Seq2Seq) dynamic structure for improved time series data modeling and understanding.

1.1. **Related work.** Web threat recognition usually includes two approaches: supervision-based [15-17] and anomaly detection-based [18-20]. Supervised detection methods typically involve training on both normal and malicious data, classifying based on well-trained malicious attacks. However, these methods may not perform optimally in identifying untrained attacks. Supervision-based Web threat recognition includes shallow machine-learning supervision and deep machine-learning supervision. The majority of shallow supervision models are less complex machine learning models, including KNN [21], SVM [22], and logistic regression [23]. Calzavara et al. [24] proposed a new supervised learning-based detection method that trains a set of binary classifiers, providing new insights into how web authentication is implemented in practice. Lucas et al. [25] compared Minimum Redun-dancy Maximum Relevance (mRMR) and Permutation Feature Importance (PFI) for Web attack feature selection. Using logistic regression and Bayesian point machines, PFI test results are used to select the most suitable features for machine learning methods with a 97% accuracy rate based on their suitability. Shaheed and Kurdy [26] evaluated the model by ana-lyzing and parsing HTTP requests that came into a web server. This was done to get four features from the HTTP request part: the length of the request, the percentage of allowed characters, the percentage of unique characters, and the attack weights, which addressed the issue of limited features and improved detection performance. The limitation of shallowly supervised models is that they require manual feature extraction and selection, and the accuracy of the models could be enhanced.

Deep supervised models are mainly deep neural network models with more complex models, such as CNN [27] and RNN [28,29]. Deep-supervised models reduce the complexity of manually extracting features and have an overall increase in model accuracy compared to shallow-supervised models. Liu et al. [30] employed vocabulary indexing to convert URIs derived from HTTP requests into code sequences that are supplied to GRU in order to monitor harmful Web traffic. Jemal et al. [31] utilized ASCII embedding to preprocess the inputs to the neural network CNN to detect web server attacks with a model accuracy of 98.2434%. Junior et al. [32] employed the BERT model to discover HTTP request patterns and added a bidirectional BiLSTM layer to represent information for detecting HTTP request attacks. However, the limitation of either shallow or deep supervision is that they are trained on normal and abnormal requests; the model relies on the type of attack that it has been trained for, and when the attack is not in the trained range, the supervised model is not the best choice; it is only for the trained attack, and the model cannot recognize the unknown attack.

Models based on anomaly detection are trained using only normal requests to discover malicious requests other than normal and are no longer limited to trained attack types. Mac et al. [33] adopted an unsupervised learning autoencoder algorithm, which used Regularized Deep autoencoder (RDA) to detect malicious HTTP requests, and achieved better performance compared with one-class SVM and stacked autoencoder. Moradi et al. [34] learned features using deep belief networks and stacked autoencoders. Normal data is utilized for training, and a one-class SVM is employed to detect anomalies. Liang et al. [35] utilized LSTM or GRU units to train two recurrent neural networks, unsupervised, using only normal requests for model training for Web attack detection. Gniewkowski et al. [36] proposed a method to detect and analyze anomalous HTTP traffic by embedding HTTP requests using an unsupervised linguistic representation model, training the RoBERTa model using only legitimate traffic, and ultimately clustering and visualizing patterns in HTTP traffic. Although anomaly detection models can detect unknown attacks, existing models' detection performance needs to be improved. Therefore, there is a need to develop practical anomaly detection models while improving detection performance.

1.2. **Motivation and contribution.** To resolve the aforementioned deficiencies, we propose the Seq2Seq network model of attention mechanism based on LSTM structure from the perspective of anomaly detection in this paper. Firstly, compared to existing supervised learning models, the unsupervised anomaly detector is trained only on normal request data, no longer relying on pre-defined attack categories. This allows the model to have the ability to recognize unknown attacks and overcomes the problem of the limited availability of anomalous request data. Secondly, compared to existing machine learning models, the LSTM's memory units can store long-term sequential information and effectively process long sequences through gate mechanisms that control what information needs to be retained or forgotten. Again, the Seq2Seq architecture supports variable-length input and output, enabling dynamic encoding and decoding operations on the requested sequences. Finally, the proposed LSTM deep structure incorporates an attention mechanism that automatically extracts more relevant and representative features from the request data. This eliminates the need for manual feature extraction and enhances the overall performance of the model by effectively capturing the requested data.

The main innovations and contributions of this work include:

(1) This paper investigates the Seq2Seq method based on anomaly detection with the LSTM structure of the attention mechanism. In anomaly detection, only normal data is used for training. Respectively, the model is trained without the attention mecha-nism and with the attention mechanism. Comparing the two experiments reveals that the Seq2Seq model based on the attention LSTM probes more deeply into the character of Web request time series data, resulting in more valuable model benefits.

(2) We compare the performance of the proposed LSTM-based attention mechanism Seq2Seq structure with various supervised and unsupervised anomaly models, in-cluding shallow supervised classifiers (KNN, SVM, logistic regression, and others), deep supervised classifiers (CNN, LGBM, GRU, and others), and shallow anomaly detectors (RNN, SAE, SOM, and others). Experiments demonstrate that the Web threat recognition model based on anomaly detection and attention mechanisms in the Seq2Seq network achieves the highest performance across all metrics, including an accuracy of over 99%, which is an improvement of 1–14

(3) In this paper, we validate the feasibility of the model on three different datasets: CSIC 2010 [37], FWAF [38], and HttpParamsDataset [39]. The accuracy of all three datasets is above 96%, demonstrating that using the Seq2Seq structure with attention can detect long requests and responses. The model generalization is outstanding and will not reduce the model's performance in a real-world dataset.

2. Model design.

Typically, Web servers respond to client requests, and these responses vary with the state of the request and the time. In many Web attacks, the attacker sends anomalous requests that cause the server to generate anomalous responses. This paper aims to design a feasible model to detect anomalous requests and identify Web threats, and the model framework is depicted in Figure 1. The model is comprised of data preparation, model construction, and abnormal request output. During data preparation, the model preprocesses the original HTTP request samples, builds the corresponding vocabulary, and generates a series of sequence vectors to satisfy the model's input requirements. In the model construction part, firstly, only normal data is trained based on anomaly detection; secondly, the threat recognition model of the Seq2Seq network containing multi-layer LSTM structure encoder and multi-layer Attention mechanism LSTM structure decoder is constructed, and finally, the probabilistic sequence corresponding to the request is outputted. In the anomalous request output section, the decision of whether a given request is anomalous is made by classifying it based on the probability sequence output by the Seq2Seq network through the fully connected layer.

2.1. Data preparation.

The data preparation part first reads the original Web request file. Each request corresponds to a data sample, each data sample is a sequence of characters, and the read request data is returned to the DataFrame for storage.The next step is to build an ASCII-based vocabulary and perform character embedding on the read data to construct the sequence vector. The steps are as follows:

(1) Store all characters in a vocabulary table using the read-traversed data. Each character's index number corresponds to its ASCII code, and it builds different vocabu-laries according to each data set.

(2) Process the sequence by designating $< SOS >$ with index number 1 at the beginning of the sequence and $< EOS >$ with index number 2 at the conclusion of the sequence.

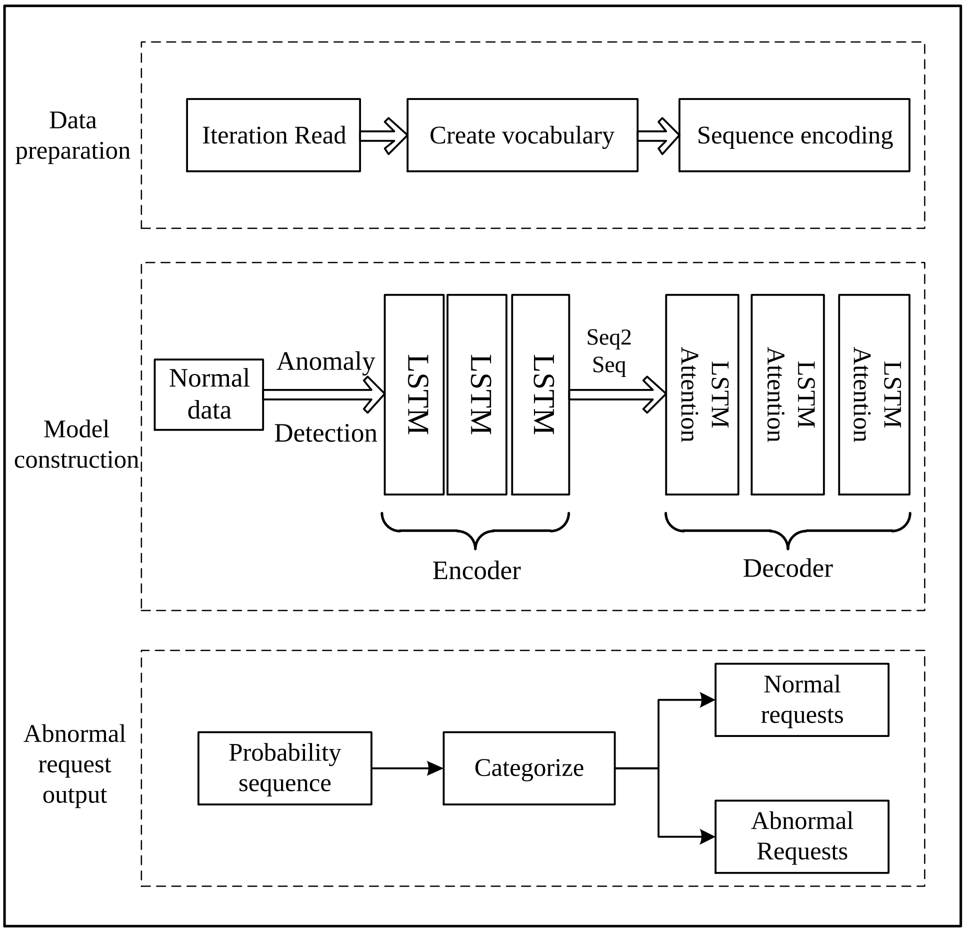


FIGURE 1. Model framework

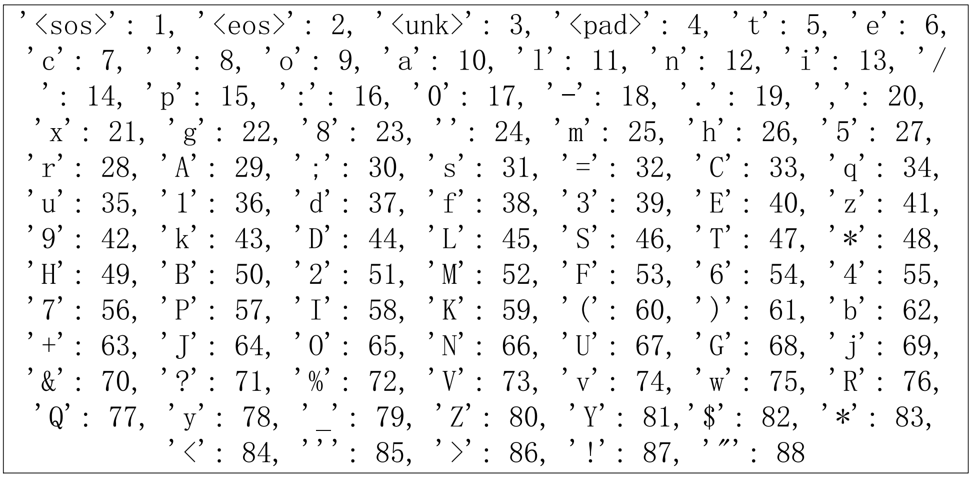
(3) Set characters not in the vocabulary list or cannot be recognized as $< UNK >$ with index number 3.

(4) Fill the sequence vectors with the same length according to $< PAD >$ with index number 4. For the CSIC2010 dataset, Figure 2 depicts the completed vocabulary list, whereas Figure 3 depicts the sequence vectors after encoding a training set of samples.

2.2. **Model construction.** The primary objective of the model construction section is to build and train a Seq2Seq threat recognition model using the LSTM structure of the Attention mechanism, which includes three parts: the Encoder construction, the Decoder construction, and the Attention mechanism introduction. The model architecture is shown in Figure 4. The Encoder part outputs the encoded input request sequence as a background vector; the vector is sent to the Decoder component to extract and decode valuable information; and the Attention part improves the limitation of the traditional Seq2Seq model that only focuses on the background vectors of the last moment so that each moment of the decoder uses different background vectors at each moment of the decoder.

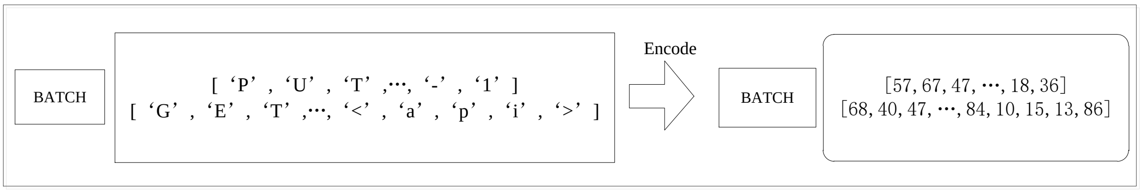
2.2.1. *Encoder.* The encoder is responsible for transforming the request input sequence of undetermined length into a background word vector c containing information about the input sequence. Let the LSTM network cell be f and the input at time t be $x_t, t = 1, \dots, T$. The variables in the encoder's concealed layer are depicted in Equation (1):

$$h_t = f(x_t, h_{t-1}) \tag{1}$$



’<sos>’ : 1, ’<eos>’ : 2, ’<unk>’ : 3, ’<pad>’ : 4, ’t’ : 5, ’e’ : 6,
’c’ : 7, ’ ’ : 8, ’o’ : 9, ’a’ : 10, ’l’ : 11, ’n’ : 12, ’i’ : 13, ’/
’ : 14, ’p’ : 15, ’:’ : 16, ’0’ : 17, ’–’ : 18, ’.’ : 19, ’,’ : 20,
’x’ : 21, ’g’ : 22, ’8’ : 23, ’’ : 24, ’m’ : 25, ’h’ : 26, ’5’ : 27,
’r’ : 28, ’A’ : 29, ’;’ : 30, ’s’ : 31, ’=’ : 32, ’C’ : 33, ’q’ : 34,
’u’ : 35, ’1’ : 36, ’d’ : 37, ’f’ : 38, ’3’ : 39, ’E’ : 40, ’z’ : 41,
’9’ : 42, ’k’ : 43, ’D’ : 44, ’L’ : 45, ’S’ : 46, ’T’ : 47, ’*’ : 48,
’H’ : 49, ’B’ : 50, ’2’ : 51, ’M’ : 52, ’F’ : 53, ’6’ : 54, ’4’ : 55,
’7’ : 56, ’P’ : 57, ’I’ : 58, ’K’ : 59, ’(’ : 60, ’)’ : 61, ’b’ : 62,
’+’ : 63, ’J’ : 64, ’O’ : 65, ’N’ : 66, ’U’ : 67, ’G’ : 68, ’j’ : 69,
’&’ : 70, ’?’ : 71, ’%’ : 72, ’V’ : 73, ’v’ : 74, ’w’ : 75, ’R’ : 76,
’Q’ : 77, ’y’ : 78, ’_’ : 79, ’Z’ : 80, ’Y’ : 81,’$’ : 82, ’*’ : 83,
’<’ : 84, ’’’ : 85, ’>’ : 86, ’!’ : 87, ’″’ : 88

Figure 2. Vocabulary list



[‘P’ , ‘U’ , ‘T’ ,···, ‘-’ , ‘1’]
[‘G’ , ‘E’ , ‘T’ ,···, ‘<’ , ‘a’ , ‘p’ , ‘i’ , ‘>’]

[57, 67, 47, ···, 18, 36]
[68, 40, 47, ···, 84, 10, 15, 13, 86]

Figure 3. Coding example

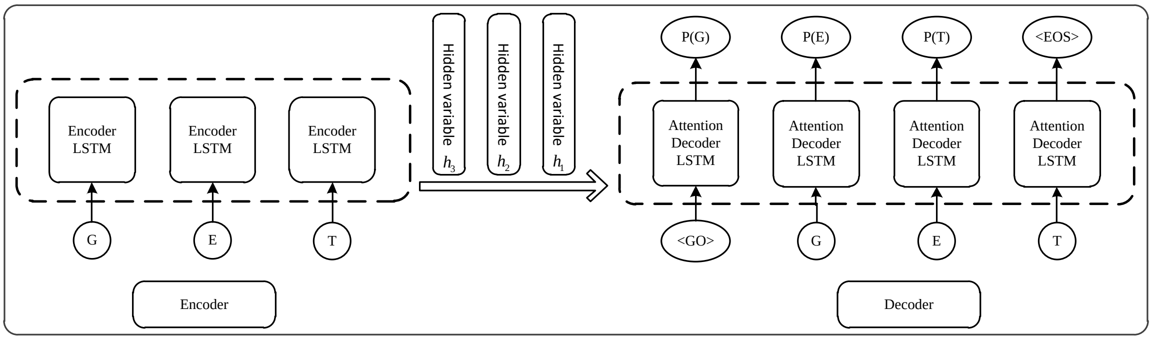


Figure 4. Architecture of Seq2Seq model of attention mechanism based on LSTM structure

The background vector of the encoder is shown in Equation (2):

$$c = q(h_1, ..., h_T) \tag{2}$$

The encoder finally outputs a background vector c that integrates the input sequence $\{x_1, x_2, ..., x_T\}$. The original Web request is fed into the Seq2Seq network through the encoder, which undergoes an encoding operation into a vector that can be used for model training.

2.2.2. *Decoder.* The role of the decoder is to extract useful information from the encoded vectors and decode them to output the probability sequence corresponding to the Web request. Let $\{y_1, y_2, ..., y_T\}$ epresent the output sequence. For each moment t', the output vector depends on the previous output and the background vector c, therefore maximizing the joint probability of the output sequence is shown in Equation (3):

$$P\left(y_1, y_2, \ldots, y_{T'}\right) = \prod_{t'=1}^{T'} P\left(y_{t'} \mid y_1, \ldots, y_{t'-1}, c\right) \tag{3}$$

The loss function of the output sequence is $-\log P\left(y_1, \ldots, y_{T'}\right)$. The LSTM is utilized as a decoder, which uses a function p to represent the probability of a single output $y_{t'}$ as shown in Equation (4):

$$P\left(y_{t'} \mid y_1, \ldots, y_{t'-1}, c\right) = p\left(y_{t'-1}, s_{t'}, c\right) \tag{4}$$

Where $s_{t'}$ is the hidden layer variable of the decoder at the moment t', as shown in Equation (5):

$$s_{t'} = g\left(y_{t'-1}, c, s_{t'-1}\right) \tag{5}$$

Where the function g is the LSTM recurrent neural unit, the above is the design of the LSTM-based Seq2Seq network without introducing the Attention mechanism, where the same background vector c is applied at each moment; however, given a pair of input sequences, due to the dynamic nature of the inputs, the inputs are not the same at each moment, and the information that affects the result is not the same at each moment, the corresponding background vectors are different at each moment, and this is manifested in the decoder by allocating varying amounts of focus to various points in the input sequence at each instant.

2.2.3. *Attention mechanism.* The Attention mechanism is applied on the decoder so that different background vectors are used for each moment of the decoder, the steps introduced by the Attention mechanism to generate the background vectors are shown in Algorithm 1, and the process diagram is shown in Figure 5, where each background vector corresponds to assigning a different amount of attention to a different part of the input sequence.

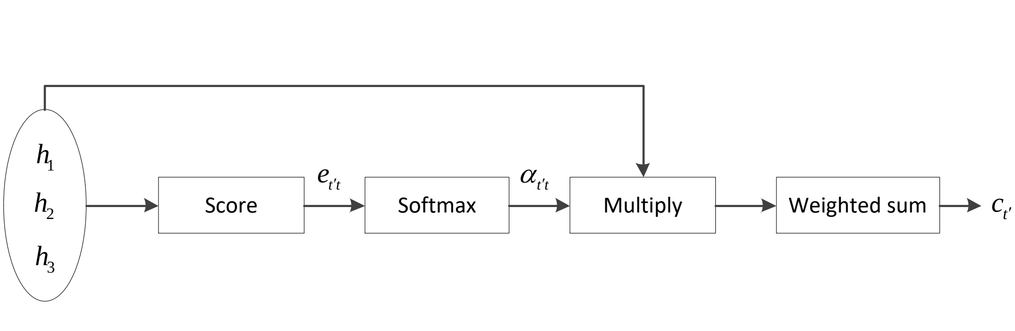


FIGURE 5. Attention layer process diagram

```
Algorithm 1: Attention background vector generation algorithm
  Input  : hidden layer variables $h_t, t = 1, \ldots, T$
  Output: background vector $c_{t'}$
1 Preparing the hidden layer variables $h_t$;
2 Perform additive attention contrast score on $h_t$ to get
    $e_{t't} = v^\top \tanh(W_s s_{t'-1} + W_h h_t)$;
3 Use Softmax to normalize to get the attention weight
    $\alpha_{t't} = Soft\max(e_{t't}) = \frac{\exp(e_{t't})}{\sum_{k=1}^{T} \exp(e_{t'k})}$;
4 Multiply $h_t$ at each moment by its corresponding weight to get $\alpha_{t't} h_t$;
5 Weighted sum to get attention background vector $c_{t'} = \sum_{t=1}^{T} \alpha_{t't} h_t$;
```

Part 2.2.2 of the decoder is enhanced by presuming that the background vector at time t' is $c_{t'}$. The Equation for the hidden layer variable of the decoder at time t' is:

$$s_{t'} = g\left(y_{t'-1}, c_{t'}, s_{t'-1}\right) \tag{6}$$

h_t is the encoder's hidden layer variable at the present moment t. The Equation for the background vector of the decoder at the moment t is:

$$c_{t'} = \sum_{t=1}^{T} \alpha_{t't} h_t \tag{7}$$

In the decoder, we must calculate the weighted average of the hidden layer variables at different times, which is referred to as the attention weight, as shown in Equation (8).

$$\alpha_{t't} = \frac{\exp\left(e_{t't}\right)}{\sum_{k=1}^{T} \exp\left(e_{t'k}\right)} \tag{8}$$

Where $e_{t't} = a(s_{t'-1}, h_t)$. a is a correlation operator using an additive attention algorithm, and $e_{t't}$ is computed as shown in Equation (9), where v, W_s, and W_h are model training parameters.

$$e_{t't} = v^\top \tanh\left(W_s s_{t'-1} + W_h h_t\right) \tag{9}$$

After adding the Attention mechanism, the output of the Decoder component is denoted by the Equation (10). By introducing the Attention mechanism, the decoder overcomes the restriction that it can only utilize the final single-vector result of the encoder, allowing the model to concentrate on all the input information that is crucial for the next target character, thereby substantially enhancing the model's performance.

$$P\left(y_{t'} \mid y_1, \ldots, y_{t'-1}, c_{t'}\right) = p\left(y_{t'-1}, s_{t'}, \mathrm{c}_{t'}\right) \tag{10}$$

2.3. **Abnormal request output.** The probability sequence generated by the Seq2Seq network still cannot directly distinguish whether it is normal or not, and the abnormal request output part is mainly supervised classification of the probability sequence generated by the Seq2Seq network and designing a fully-connected layer neural network to learn how to distinguish between normal and abnormal request occurrence probability sequence. Let the sequence output from the encoder-decoder network be $\overrightarrow{prob}_i$, and the output z is obtained after the forward propagation calculation of the fully connected layer, as shown in Equation (11):

$$z = W\overrightarrow{prob}_i + b \tag{11}$$

Where b is the bias vector and w is the weight matrix. The ReLU activation function, which is generated as given in Equation (12), is applied to the matrix as a nonlinear activation operation on the output z of the fully connected layer.

$$f = \max(0, pr\overrightarrow{o}b_i) \tag{12}$$

The Softmax function is used to compress the value of each element into the range [0, 1], ensuring that the total of all the elements of the probability sequence is equal to 1. The definition of the Softmax function is seen in Equation (13):

$$\text{softmax}\,(pr\overrightarrow{o}b_i) = pr\overrightarrow{o}b_i' = \frac{e^{pr\overrightarrow{o}b_i}}{\sum_{j=1}^{n} e^{pr\overrightarrow{o}b_i}} \tag{13}$$

Where $\sum_{\mathrm{j=1}}^{\mathrm{n}} pr\overrightarrow{o}b_i' = 1$. The fully connected layer converts the output of the encoder network's supervised classification of the coded sequence output into probability values, with outputs of 0 for normal and 1 for abnormal.

3. Experiment results and analysis.

3.1. Dataset.

The selection of datasets for Web threat identification is crucial, and the earlier da-taset DARPA [40] is outdated and lacks much real-world attack data. In this paper, we conduct experiments using CSIC 2010 [37], FWAF [38], and HttpParamsDataset [39], which have been utilized in a variety of studies. The first dataset used in this paper is the CSIC 2010 dataset, which contains 36,000 normal requests and more than 25,000 anoma-lous requests. To assess the model in a real-world setting, the proposed approach is simulated on the second dataset, FWAF, which is produced using HTTP traffic that the Web Attack Firewall (WAF) has observed and includes 1,290,000 regular requests and 48,000 unusual queries. The HttpParamsDataset dataset, which was created using a different method and includes more than 19,000 regular requests and more than 11,000 ab-normal requests, is the third dataset.

3.2. Evaluation indicators.

The confusion matrix shown in Table 1 will be applied to the experimental data to more precisely evaluate the model proposed in this study.

Table 1. Confusion matrix

	Anomalous requests	Normal requests
Anomalous requests	TP	FP
Normal requests	FN	TN

True positive (TP) represents the sample of anomalous requests, and model judgment also represents the sample of abnormal requests. False positive (FP) represents an actual sample of normal requests, whereas its model judgment result represents an actual sample of anomalous requests; False negative (FN) represents the sample of anomalous request, yet the model assesses the result as the sample of normal request. True Negative (TN) represents that both true and model judgments are the result of normal requests.

The experiments use accuracy, precision, sensitivity, and specificity, as well as the F1-score, as criteria for assessment. Accuracy is defined in Equation (14) as the percentage of samples that were properly recognized out of all the samples; Precision is defined as the ratio of the number of samples successfully classified as anomalous data by the model to the number of samples of all samples correctly identified as anomalous data, as shown in Equation (15); Sensitivity is calculated the same as the recall, as shown in

Equation (16), represents the percentage of samples that are in fact anomalous data and are correctly identified by the model, as well as the model's capacity to identify abnormal data; Specificity, as shown in Equation (17), shows the model's ability to identify normal data by dividing the number of samples it properly identified as normal data by the total number of samples that were normal data; The F1-Score, as shown in Equation (18), is the sum of the precision rate and the recall rate average; The AUC value, which is a curve with TPR as the vertical axis and FPR as the horizontal axis, is used to indicate the area under the ROC curve. Equations (19) and (20) are used to determine the FPR and TPR.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FN} + \text{FP}} \tag{14}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{15}$$

$$\text{Sensitivity} = \text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{16}$$

$$\text{Specificity} = \frac{\text{TN}}{\text{TN} + \text{FP}} \tag{17}$$

$$\text{F1-Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{18}$$

$$\text{FPR} = \frac{\text{FP}}{\text{FP} + \text{TN}} \tag{19}$$

$$\text{TPR} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{20}$$

3.3. **Experimental evaluation.** The loss value quantifies the difference between the predicted and actual values of the model. The change in loss value during the model training process is shown in Figure 6, the loss value is no longer exponential after 3000 epochs of model training, and the loss value stabilizes around 0.01, indicating that the model gradually converges and stabilizes to reach the optimal performance. For the CSIC2010 dataset with 5000 epochs, the lowest loss value of the model is 0.008. The results demonstrate that the model's loss value is low, the identification of Web request anomalies can achieve high accuracy, the error rate of the prediction results is low, and performance is enhanced.

The AUC indicates the degree of separability, and the closer the AUC is to 1, the lower the percentage of incorrectly identifying negative classes as positive classes and the greater the generalizability of the model. In Figure 7, the AUC for the model in the CSIC2010 dataset is 0.9998. The findings show that the model's generalization capacity is excellent, indicating that it is reasonable to develop distinct vocabularies for distinct datasets during the data preparation phase. Appropriate vocabulary coding is conducive to enhancing the model's adaptability to diverse datasets and generalization capability.

This paper contrasts the results of training the model without the attention mechanism and the model with the attention mechanism. On the dataset CSIC2010, the evaluation results of the Seq2Seq model and the model introducing the attention mechanism on the five indicators are shown in Figure 8. The model introducing the attention mechanism performs better in the five indicators, where accuracy is improved by 2.8%, precision by 3.02%, sensitivity by 3.96%, specificity by 2.93%, and F1-score by 3.5%. This is so that the model can concentrate just on the crucial portions of the input thanks to the inclusion of the attention mechanism. The attention model is able to deal with extended

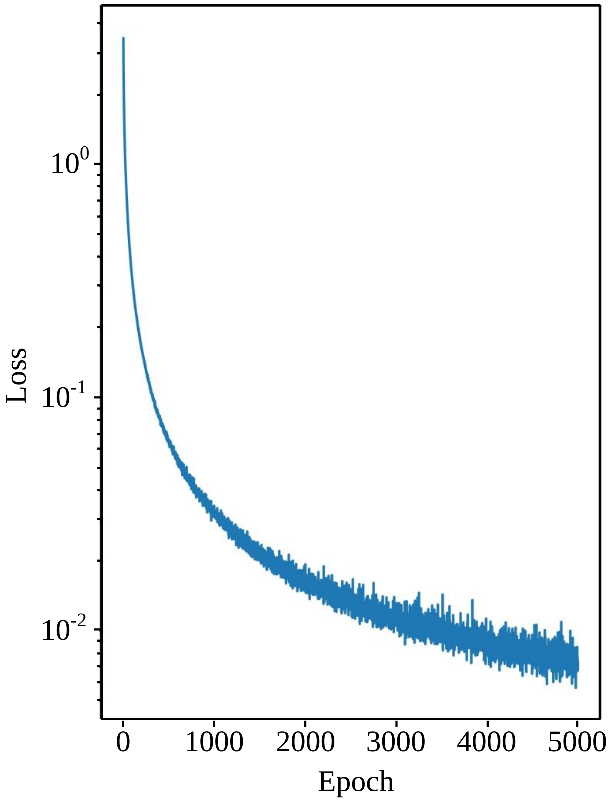


FIGURE 6. Model Loss

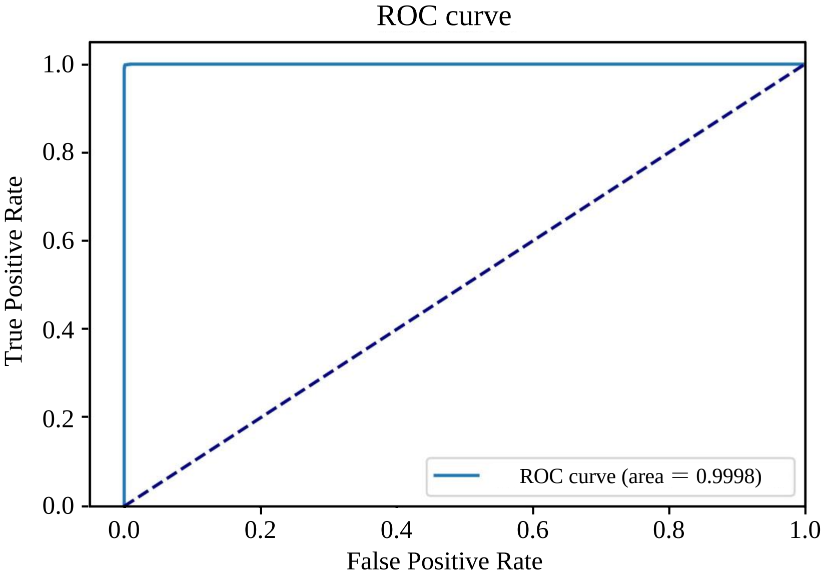


FIGURE 7. AUC-ROC curve

sequences and can keep effective information better than the Seq2Seq model that does not include the attention mechanism. However, the effectiveness of simple Seq2Seq decreases significantly as the length of the request sequence to be processed increases. The attention mechanism can focus on the context corresponding to the request part and sufficiently extract the information of the request sequence.

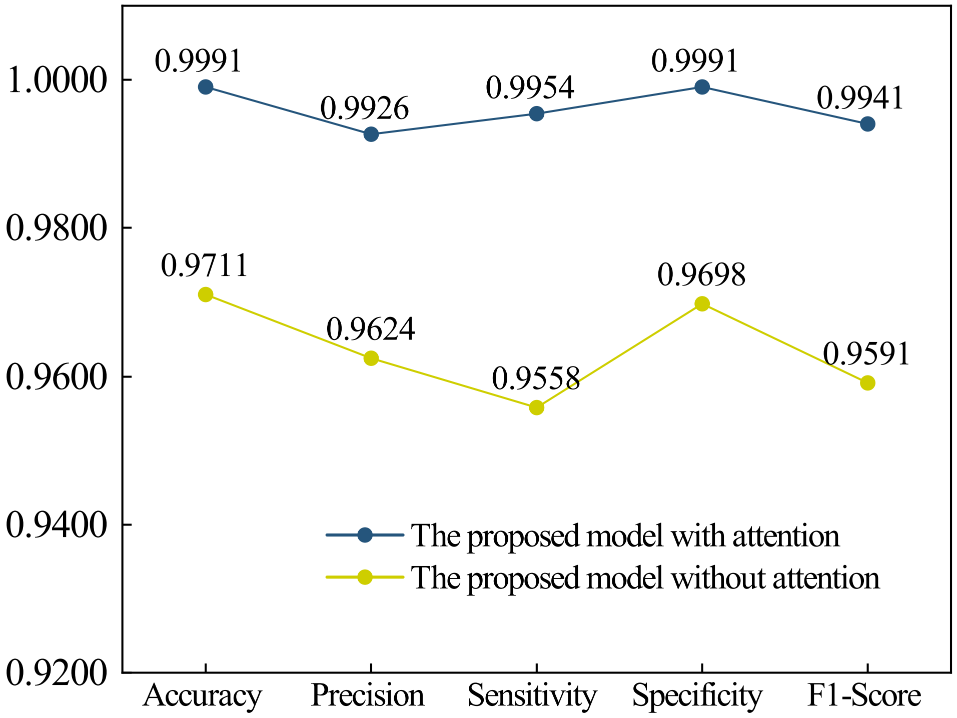


Figure 8. Evaluation of the two models on the CSIC2010 dataset

4. **Experimental comparison and generalization ability analysis.** To evaluate the efficacy of the model presented in this paper, it is compared to two other experimental models: the supervised classification model and the anomaly detection model, as shown in Sections 4.1 and 4.2, respectively. There are superficial machine learning-based classifiers and deep machine learning-based classifiers in the supervised classification. In order to evaluate the generalization ability of the experiments, experiments are carried out on three datasets as shown in Section 4.3.

4.1. **Comparison with supervised classification models.** In this paper, we train six shallow machine learning models and output the optimal results, and Table 2 presents the experimental comparison results of each shallow machine learning model with the attention model on the CSIC2010 dataset. The six machine learning models are KNN, SVM, LR, RF, DT, and XGBoost.

The machine learning models are unified to use TF-IDF for feature extraction. The KNN model, RF model, and SVM model are trained with default parameters; the LR model is trained with L2 regularization; for the DT model, the depth of the tree is set at 1-10; the XGBoost model is trained with the L2 regularization of the weight term to prevent the model from being overfitted. All models are trained iteratively according to the predefined parameters and output the optimal results.

As can be seen from Table 2, most traditional machine models can achieve a high precision of about 95%, but the model precision of this paper is 99.26% better than

Table 2. Comparison of the proposed model with the shallow machine learning models

Model	Accuracy	Precision	Sensitivity	Specificity	F1-Score
KNN	0.9233	0.9194	0.8872	0.9135	0.9030
SVM	0.9619	0.9623	0.9421	0.9596	0.9521
LR	0.9680	0.9922	0.9419	0.9660	0.9664
RF	0.9647	0.9618	0.9501	0.9628	0.9559
DT	0.9008	0.9519	0.7936	0.8750	0.8655
XG-Boost	0.9292	0.9315	0.9292	0.9238	0.9209
Proposed model	0.9991	0.9926	0.9954	0.9991	0.9941

machine learning. The classification models based on traditional machine learning are all less sensitive, so the classification performance is average, and the ability to recognize abnormal data is poor. The RF model is the most accurate machine learning model in terms of accuracy. However, the model's accuracy in this paper is 3.44% higher than that of the RF model. The machine learning model can reach about 92% in terms of specificity index, and the model in this paper is as high as 99.41%. The model has the highest recognition ability for normal samples. Compared with the above six machine learning models, the F1-score of this paper is the highest, demonstrating that the model in this research performs optimally across the board. It is because the LSTM-based Seq2seq network can automatically perform feature extraction, which can cover a more comprehensive range of features for training compared to traditional machine learning, which relies on specialized domain knowledge to select input features manually. The model can consider the sequence dependence between the output sequences, and the model is satisfactorily trained after extracting the contextual relationships of the requested sequences.

Table 3. Comparison of the proposed model with deep machine learning models

Model	Accuracy	Precision	Sensitivity	Specificity	F1-Score
GRU [30]	0.9704	0.9669	0.9691	0.9694	0.9680
CNN+ASCII Embedding [31]	0.9813	0.9483	0.9778	0.9809	0.9628
CNN [41]	0.9707	0.9743	0.9759	0.9699	0.9751
SAE+Isolation Forest [42]	0.8832	0.8029	0.8834	0.9020	0.8412
PL-RNN [43]	0.9613	0.9441	0.9779	0.9604	0.9607
LGBM [44]	0.9315	0.9390	0.9290	0.9262	0.9340
Proposed model	0.9991	0.9926	0.9954	0.9991	0.9941

Table 3 gives the results of the experimental comparison of each deep machine learning model in other literature with the attention model in this paper on the dataset CSIC2010. The six deep machine learning models are GRU [30], CNN+ASCII Embedding [31], CNN [41], SAE+Isolation Forest [42], PL-RNN [43], and LGBM [44].

As shown in Table 3, the model in this paper performs better than the improved CNN, GRU, SAE, RNN, and LGBM deep learning models, and the overall model accuracy is enhanced. The Seq2Seq detection model suggested in this research, based on an LSTM-based attention mechanism, obtains the greatest accuracy of 99.91%, which is 2.84%

higher than the CNN model, 2.87% higher than the GRU model, 11.59% higher than the SAE model, 3.78% higher than the RNN model, and 6.76% higher than the LGBM model. The suggested model outperforms existing models in the other four metrics in addition to accuracy. The accuracy, sensitivity, specificity, and F1-score are all 99.26%, 99.54%, 99.91%, and 99.41% respectively. It is particularly noteworthy that the model sensitivity is the highest, indicating that it has the highest sensitivity to anomalous data. For Web threat identification, the significance of the ability to recognize anomalous data is greater than the ability to recognize normal data. For a sample that is itself anomalous, the model fails to recognize it, often resulting in a more significant threat. The model in this paper utilizes Seq2Seq, the attention mechanism of LSTM, for threat identification, which can dynamically analyze the input request and capture the contextual relationship of the request. Moreover, the model introduces the attention mechanism for longer requests to transform the input sequences into fixed-length vectors while preserving all valid information, which can deal with long Web requests with a more desirable benefit of the model.

4.2. **Comparison with anomaly detection models.** Table 4 gives the results of the experimental comparison between other anomaly detection models in the literature and the attention model in this paper on the dataset CSIC2010. The seven anomaly detection models are RDA [33], SAE [34], RNN [35], GRU [35], LSTM [35], SOM [35], and RNN with Attention [45].

Table 4. Comparison of the proposed model with anomaly detection models

Model	Accuracy	Precision	Sensitivity	Specificity	F1-Score
RDA [33]	0.9767	0.9464	0.9462	0.9852	0.9463
SAE [34]	0.8924	0.8158	0.8948	0.8911	0.8535
RNN [35]	0.8515	0.8185	0.7403	0.9546	0.7775
GRU [35]	0.9788	0.9455	0.9722	0.9845	0.9587
LSTM [35]	0.9842	0.9684	0.9756	0.9921	0.9720
SOM [35]	0.9282	0.7761	0.9497	0.9242	0.8542
RNN with Attention [45]	0.9910	0.9755	0.9915	0.9931	0.9834
Proposed model	0.9991	0.9926	0.9954	0.9991	0.9941

As shown in Table 4, the anomaly detection-based model is generally worse in all indicators compared to the deep-supervised machine learning model. However, the deep supervised model is only for the trained attacks and cannot identify the unknown attacks. This paper proposes an anomaly detection-based attentional mechanism Seq2Seq detection model, which is trained using only normal requests to discover malicious requests other than normal. It is no longer limited to the training of the well-trained attack types. Moreover, the model in this paper outperforms other anomaly detection models proposed in the literature in all metrics, with an accuracy rate that is 2.24% higher than the RDA model, 10.67% higher than the SAE anomaly detection model, 14.76% higher than the RNN anomaly detection model, 2.03% higher than the GRU anomaly detection model, 1.49% higher than the LSTM anomaly detection model, 7.09% higher than the SOM anomaly detection model by 7.09%, and 0.81% higher than the attention mechanism RNN model. Compared with RDA, SAE, RNN, GRU, LSTM, and SOM anomaly detection models without introducing the attention mechanism, the accuracy of this paper's model is higher, which indicates that the attention mechanism is more effective for Web request extraction. For the RNN anomaly detection model with attention, this paper adopts the Seq2seq model based on LSTM that can dynamically train the data, and the LSTM's memory unit can store long-time sequence information of Web request sequences, which is more effective than RNN in processing long sequences.

4.3. **Evaluation on FWAF and HttpParamsDataset datasets.** The FWAF and HttpParamsDataset datasets are also used to assess the model's generalizability in this study, with the findings displayed in Table 5 and Figure 9. Table 5 shows that the LSTM-based Seq2seq model performs above 95% in all metrics whether or not the attention mechanism is introduced, and all the metrics are improved after introducing the attention mechanism. In the real environment of the dataset, the FWAF evaluation model has less information about the Payload data, so the results are relatively low. However, the accuracy rate in the real environment reaches 96.75%, and the F1-score reaches 96.9% on behalf of the model is of real-world significance. Evaluating the model on the HttpParamsDataset dataset, where most of the samples in the dataset are long HTTP requests, the model accuracy is up to 99.87%, sensitivity is up to 99.59%, and the F1-score is up to 99.53%. The simulation results on FWAF and HttpParamsDataset dataset show that the model is not limited to a single dataset. Different vocabularies are built for different datasets, and after training modeling with an LSTM-based Seq2seq network, the threat recognition capabilities are all better, and the model can generalize.

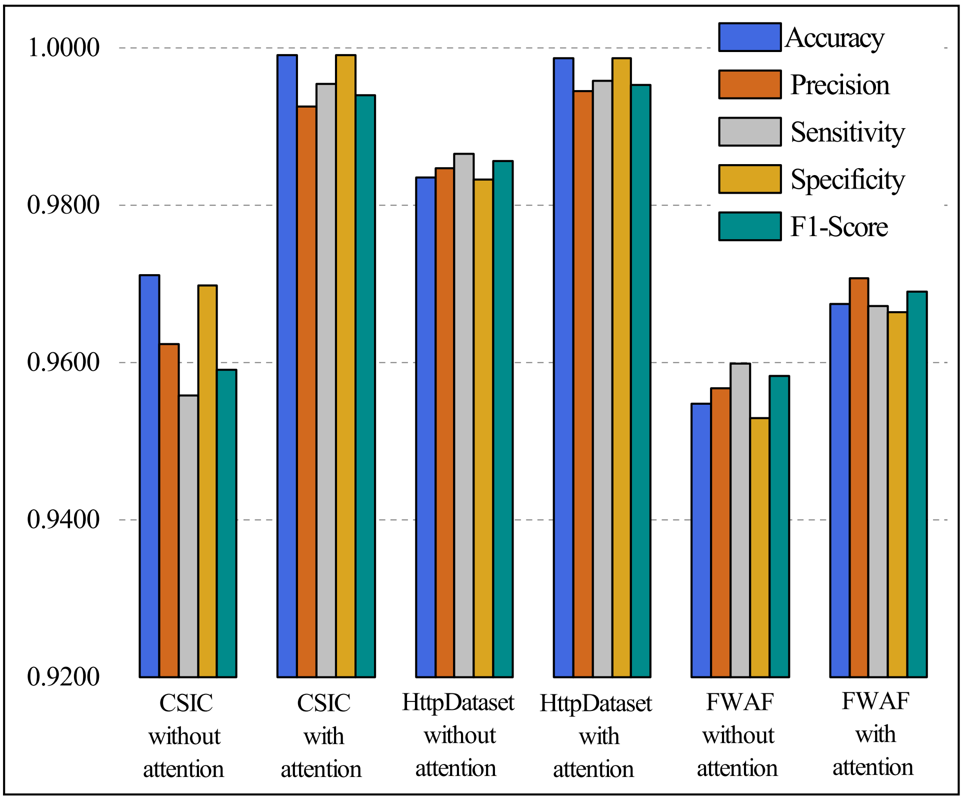


Figure 9. Evaluation on FWAF and HttpParamsDataset dataset

After introducing the attention mechanism, the effect of the model can be more intuitively seen in Figure 9, which delivers some improvement on all three datasets, CSIC 2010, FWAF, and HttpParamsDataset datasets. There are different improvements in the five metrics, in which the overall effect achieved by the model after introducing the attention mechanism on the CSIC 2010 dataset is the most significant improvement over the

Table 5. Evaluation on FWAF and HttpParamsDataset dataset

Dataset	Model	Accuracy	Precision	Sensitivity	Specificity	F1-Score
HttpParams	Without attention	0.9835	0.9847	0.9866	0.9833	0.9857
	With attention	0.9987	0.9946	0.9959	0.9987	0.9953
FWAF	Without attention	0.9548	0.9567	0.9599	0.9529	0.9583
	With attention	0.9675	0.9708	0.9672	0.9664	0.9690

unintroduced one. Following the implementation of the attention mechanism, the model's values for all the indicators in the three datasets are above 96%, and the results show that, firstly, the model's accuracy is increased in any of the datasets, with an elevated probability that the samples are recognized accurately. Second, the model's sensitivity is higher than 96%, which is practical for recognizing abnormal data.

5. **Conclusion.** The Web Threat Recognition Model based on Anomaly Detection and Attention Mechanism Seq2Seq Network proposed in this paper applies the Attention Mechanism and Seq2Seq Network architecture to Web Threat Recognition. The unsupervised model is trained in the perspective of anomaly detection and works to find out the malicious requests that are different from the benign ones. Compared with shallow machine learning supervised classifiers, deep machine learning supervised classifiers, and other anomaly detection classifiers, the model achieves the best performance on various evaluation metrics with high recognition accuracy for different datasets. It effectively solves the problems of existing methods with poor parameters, such as accuracy and F1-score, and the model's lack of generalization ability. It is no longer limited to trained malicious attacks, which improves the performance of the threat recognition model. The next steps include considering the combination of federated learning to ensure data privacy during model training and visualizing the model results to improve model interpretability.

REFERENCES

[1] J. Li, "Vulnerabilities mapping based on OWASP-SANS: a survey for static application security testing (SAST)," *arXiv preprint arXiv*, vol. 2004, 03216, 2020.

[2] S. Gupta, and B. B. Gupta, "Cross-Site Scripting (XSS) attacks and defense mechanisms: classification and state-of-the-art," *Interna-tional Journal of System Assurance Engineering and Management*, vol. 8, pp. 512–530, 2017.

[3] L. K. Shar, and H. B. K. Tan, "Defeating SQL injection," *Computer*, vol. 46, no. 3, pp. 69–77, 2012.

[4] I. Riadi, and E. I. Aristianto, "An analysis of vulnerability web against attack unrestricted image file upload," *Computer Engineering and Applications Journal*, vol. 5, no. 1, pp. 19–28, 2016.

[5] A. O. Alaswad, A. H. Montaser, and F. E. Mohamad, "Vulnerabilities of biometric authentication "threats and countermeasures," *International Journal of Information & Computation Technology*, vol. 4, no. 10, pp. 947–58, 2014.

[6] T.-Y. Wu, Q. Meng, Y.-C. Chen, S. Kumari, and C.-M. Chen, "Toward a secure smart-home IoT access control scheme based on home registration approach," *Mathematics*, vol. 11, no. 9, 2123, 2023.

[7] T.-Y. Wu, Q. Meng, Y.-C. Chen, S. Kumari, and C.-M. Chen, "Rotating Behind Security: An enhanced authentication protocol for IoT-enabled devices in distributed cloud computing architecture," *EURASIP Journal on Wireless Communications and Networking*, vol. 2023, 36, 2023.

[8] T.-Y. Wu, Q. Meng, Y.-C. Chen, S. Kumari, and C.-M. Chen, "Amassing the Security: An Enhanced Authentication Protocol for Drone Communications over 5G Networks," *Drones*, vol. 6, no. 1, 10, 2022.

[9] J.-L. Li, H. Zhang, and Z.-Q. Wei, "The weighted word2vec paragraph vectors for anomaly detection over HTTP traffic," *IEEE Access*, vol. 8, pp. 141787–141798, 2020.

[10] Y.-C. Li, R. Xiao, J.-G. Feng, and L.-J. Zhao, "The weighted word2vec paragraph vectors for anomaly detection over HTTP traffic," *IEEE Access*, vol. 8, pp. 141787–141798, 2020.

[11] J. K. Rout, A. Dalmia, K. R. Choo, S. Bakshi, and S. K. Jena, "Revisiting semi-supervised learning for online deceptive review detection," *IEEE Access*, vol. 5, pp. 1319–1327, 2017.

[12] Z. K. Maseer, R. Yusof, N. Bahaman, S. Mostafa, and C. F. M. Foozy, "Benchmarking of machine learning for anomaly based intrusion detection systems in the CICIDS2017 dataset," *IEEE Access*, vol. 9, pp. 22351–22370, 2021.

[13] Z. Yang, X.-D. Liu, T. Li, D. Wu, J.-J Wang, Y.-W Zhao, and H. Han, "A systematic literature review of methods and datasets for anomaly-based network intrusion detection," *Computers & Security*, vol. 116, 102675, 2022.

[14] G. Vigna, F. Valeur, D. Balzarotti, W. Robertson, C. Kruegel, and E. Kirda, "Reducing errors in the anomaly-based detection of web-based attacks through the combined analysis of web requests and SQL queries," *Journal of Computer Security*, vol. 17, no. 3, pp. 305–329, 2009.

[15] K. Wang, M.-J. Zhu, W. Boulila, M. Driss, T.-R. Gadekallup, C.-M. Chen, L. Wang, and S. Kumari, "SeqNovo: De Novo Peptide Sequencing Prediction in IoMT via Seq2Seq," *IEEE Journal of Biomedical and Health Informatics*, 2023. [Online]. Available : https: // doi.org /10.1109/ JBHI. 2023.3321780.

[16] S. Kumar, A. Damaraju, A. Kumar, S. Kumari, and C.-M. Chen, "LSTM Network for Transportation Mode Detection," *Journal of Internet Technology*, vol. 22, no. 4, pp. 891-902, 2021.

[17] C.-C. Luo, Z.-Y. Tan, G.-Y. Min, J. Gan, W. Shi, and Z.-H. Tian, "A novel web attack detection system for internet of things via ensemble classification," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 8, pp. 5810-5818, 2020.

[18] W.-S. Gan, L.-L. Chen, S.-C. Wan, J.-H. Chen, and C.-M. Chen, "Anomaly Rule Detection in Sequence Data" *IEEE Transactions on Knowledge and Data Engineering*, vol. 35, no. 12, pp. 12095-12108, 2023

[19] S. Naseer, Y. Saleem, S. Khalid, M. K. Bashir, J.-H. Han; M. M. Iqbal, and K.-J. Han, "Enhanced network anomaly detection based on deep neural networks," *IEEE Access*, vol. 6, pp. 48231-48246, 2018.

[20] Y.-C. Xu, L.-F. Zhang, B. Du, and L.-P. Zhang, "Hyperspectral anomaly detection based on machine learning: An overview." *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* vol. 15, pp. 3351-3364, 2022.

[21] S.-C. Zhang, X.-L. Li, M. Zong, X.-F. Zhu, and D.-B. Cheng, "Learning k for knn classification," *ACM Transactions on Intelligent Systems and Technology (TIST)* vol. 8, no. 3, pp. 1-19, 2017.

[22] S.-J. Huang, N.-G. Cai, P. P. Pacheco, S. Narrandes, Y. Wang, and W. Xu, "Applications of support vector machine (SVM) learning in cancer genomics," *Cancer Genomics & Proteomics*, vol. 15, no. 1, pp. 41-51, 2018.

[23] C. J. Peng, K. L. Lee, and G. M. Ingersoll, "An introduction to logistic regression analysis and reporting." *The Journal of Educational Research*, vol. 96, no. 1, pp. 3-14, 2022.

[24] S. Calzavara, G. Tolomei, A. Casini, M. Bugliesi, and S. Orlando, "A supervised learning approach to protect client authentication on the web," *ACM Transactions on the Web (TWEB)*, vol. 9, no. 3, pp. 1–30, 2015.

[25] T. J. Lucas, C. A. C. Tojeiro, R. G. Pires, K. A. P. d. Costa, and J. P. Papa, "Machine Learning for Web Intrusion Detection: A Comparative Analysis of Feature Selection Methods mRMR and PFI," in*Artificial Intelligence and Soft Computing: 19th International Conference*, Springer International Publishing, 2020, pp. 535–546.

[26] A. Shaheed, and M. H. D. Kurdy, "Web Application Firewall Using Machine Learning and Features Engineering," *Security and Communication Networks*, vol. 2022, 2022.

[27] T. Kattenborn, J. Leitloff, F. Schiefer, and S. Hinz, "Review on Convolutional Neural Networks (CNN) in vegetation remote sensing," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol.173, pp. 24-49, 2021.

[28] A. Sherstinsky, "Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network." *Physica D: Nonlinear Phenomena*, vol. 404, 132306, 2020.'

[29] G.-H. Zhang, Y.-S. Liu, H. Wang, and N.-W. Yu, "Smart Contract Vulnerability Detection Scheme Based on BiLSTM and Attention Mechanism," *Netinfo Security*, vol. 22, no. 9, pp. 46-54, 2022.

[30] Z.-B. Liu, W.-Q. Zhang, Y.-Y. Huang, and Q.-G. Zhou, "A Malicious Web Request Detection Technology Based on Gate Recurrent Unit," in *Frontier Computing: Proceedings of FC 2020.* Springer, 2020, pp. 103–113.

[31] I. Jemal, M. A. Haddar, O. Cheikhrouhou, and A. Mahfoudhi, "ASCII Embedding: An Efficient Deep Learning Method for Web Attacks Detection," in *Pattern Recognition and Artificial Intelligence: 4th Mediterranean Conference*, Springer International Publishing, 2020, pp. 286–297.

[32] L. Junior, D. Macêdo, A. L. I. D. Oliveira, and C. Zanchettin, "Detecting malicious HTTP requests without Log Parser using RequestBERT-BiLSTM," in *Intelligent Systems: 11th Brazilian Conference*, Springer International Publishing, 2022, pp. 328–342.

[33] H. Mac, D. Truong, L. Nguyen, H. Nguyen, H. A. Tran, and D. Tran, "Detecting attacks on web applications using autoencoder," in *the 9th International Symposium on Information and Communication Technology*, 2018, pp. 416–421.

[34] V. A. Moradi, M. Teshnehlab, and K. S. Sedighian, "Leveraging deep neural networks for anomaly-based web application firewall," *IET Information Security*, vol. 13, no. 4, pp. 352–361, 2019.

[35] J.-X. Liang, W. Zhao, and W. Ye, "Anomaly-based web attack detection: a deep learning approach," in *the 2017 VI International Conference on Network*, Communication and Computing. 2017, pp. 80–85.

[36] M. Gniewkowski, H. Maciejewski, T. R. Surmacz, and W. Walentynowicz, "HTTP2vec: Embedding of HTTP Requests for Detection of Anomalous Traffic," *ArXiv*, 2108.01763, 2021.

[37] C. T. Giménez, A. P. Villegas, and G. Á. Marañón, "HTTP data set CSIC 2010," *Information Security Institute of CSIC (Spanish Research National Council)*, vol. 64, 2010.

[38] Y. E. Seyyar, A. G. Yavuz, and H. M. Ünver, "An attack detection framework based on BERT and deep learning," *IEEE Access*, vol. 10, pp. 68633–68644, 2022.

[39] Z.-H. Tian, C.-C. Luo, J. Qiu, X.-J. Du, M. Guizani, "A distributed deep learning system for web attack detection on edge devices," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 3, pp. 1963–1971, 2019.

[40] W. B. Bonvillian, and A. R. Van, "ARPA-E and DARPA: Applying the DARPA model to energy innovation," *The Journal of Technology Transfer*, vol. 36, no. 5, pp. 469–513, 2011.

[41] A. Tekerek, "A novel architecture for web-based attack detection using convolutional neural network," *Computers & Security*, vol. 100, 102096, 2021.

[42] A. M. Vartouni, S. S. Kashi, and M. Teshnehlab, "An anomaly detection method to detect web attacks using stacked auto-encoder," in *The 2018 6th Iranian Joint Congress on Fuzzy and Intelligent Systems (CFIS)*, IEEE, 2018, pp. 131–134.

[43] H.-Y. Liu, B. Lang, M. Liu, and H.-B. Yan, "CNN and RNN based payload classification methods for attack detection," *Knowledge-Based Systems*, vol. 163, pp. 332–341, 2019.

[44] R. C. C. D. Silva, M. P. O. Camargo, M. S. Quessada, A. C. Lopes, J. D. M. Ernesto, and K. A. P. D. Costa, "An Intrusion Detection System for Web-Based Attacks Using IBM Watson," *IEEE Latin America Transactions*, vol. 20, no. 2, pp. 191–197, 2021.

[45] S. Mohammadi, and A. Namadchian, "Anomaly-based Web Attack Detection: The Application of Deep Neural Network Seq2Seq With Attention Mechanism," *IseCure*, vol. 12, 2020.