EURASIP Journal on Wireless Communications and Networking

RESEARCH Open Access

Web intrusion detection system combined with feature analysis and SVM optimization

Chao Liu[1,2], Jing Yang[1*] and Jinqiu Wu[1]

Abstract

The current network traffic is large, and the network attacks have multiple types. Therefore, anomaly detection model combined with machine learning is developing rapidly. Frequent occurrences of Web Application Firewall (WAF) bypass attacks and the redundancy of the data characteristics in Hypertext Transfer Protocol (HTTP) protocol make it difficult to extract data characteristics. In this paper, an integrated web intrusion detection system combined with feature analysis and support vector machine (SVM) optimization is proposed. By using expert's knowledge, the characteristics of the common Web attacks are analyzed. The related data characteristics are selected by the analysis of the HTTP protocol. In the classification learning, the mature and robust support vector machine algorithm is utilized and the grid search method is used for the parameter optimization. Consequently, a better detection capability on Web attacks can be obtained. By using the HTTP DATASET CSIC 2010 data set, experiments have been carried out to compare the detection capability of different kernel functions. The results show that the proposed system performs good in the detection capability and can detect the WAF bypass attacks effectively.

Keywords: Hidden Markov model, Protocol analysis, Support vector machine, Grid search

1 Introduction

The 2017 Global Threat Intelligence Center (GTIC) [1] Q2 threat intelligence report pointed out that among all types of attacks, Web application have the highest proportion of attacks, accounting for 21%, of which Structure Query Language (SQL) injection accounts for 97%.Therefore, the prevention of Web attacks is still the most important. Although the WAF products are constantly upgraded, WAF bypass attacks always exist. Abnormal intrusion detection based on data mining and machine learning has been developed rapidly in order to better exploit intrusion characteristics. In 2014, Devaraju et al. used the neural network algorithms in the intrusion detection [2] to effectively perform feature extraction and classification. Zhao et al. applied the Markov model to IDS in conjunction with the commonly used method of reference [3]. Mukkamala et al. applied the supervised standard SVM algorithm to intrusion detection [4], which has better detection effect compared with the intrusion detection using the neural network method.

Jabbar Akhil proposes an intelligent network intrusion detection system using AODE algorithm for the detection of different types of attacks [5], and average one dependence estimator (AODE) is one of the recent enhancements of naive Bayes algorithm. AODE solves the problem of independence by averaging all models generated by traditional one dependence estimator and is well suited for incremental learning. Through experiments, it got a good detection result. Longzhi Yang proposes a data-driven network intrusion detection system [6], in particular, the developed system equipped with a sparse rule base not only guarantees the online performance of intrusion detection, but also allows the generation of security alerts from situations which are not directly covered by the existing knowledge base.

In the machine learning-based anomaly detection model, there are some applications of hidden Markov, SVM, and neural networks [7–9] in which SVM compensates the over-fitting and generalization better compared to other algorithms and has the advantages of small-scale and high-dimensional Web intrusion detection [10]. Experiments show that the performance of the SVM algorithm is directly affected by its parameters.

* Correspondence: 545361998@qq.com
[1] Harbin Engineering University, Harbin, China
Full list of author information is available at the end of the article

In 2014, Xuefeng Li proposed a network intrusion detection model based on genetic algorithm to synchronously select features and support vector machine parameters [11], aiming at the problem of high-dimensional data generated by intrusion detection system and parameter optimization of support vector machine, which can improve the accuracy of network intrusion detection and meet the real-time requirements of network intrusion detection. In order to solve the shortcomings of slow convergence and easy to fall into local optimum in the process of parameter optimization of support vector, Zhang et al. proposed particle swarm optimization algorithm [12]. Some researchers utilize the ant colony algorithm [13] to find the SVM parameters with the highest detection rate, thereby improving the detection rate. Later, PRK Varma [14] proposes a set of network traffic features that can be extracted for real-time intrusion detection and also proposes fuzzy entropy-based heuristic for ant colony optimization (ACO) in order to search for global best smallest set of network traffic features for real-time intrusion detection data set. However, in practice, the algorithm is relatively complicated, time-consuming, and does not necessarily achieve an optimal solution. Relatively, the grid search algorithm has the advantage of easy to implement and the appropriate solution can be found in a short time, which is more suitable for the complex and variable network environment. In this paper, an optimized Web intrusion detection system based on feature analysis and SVM algorithm is proposed. The hidden Markov model is used to identify the parameter types, and the grid search algorithm is applied to optimize the parameters to improve the intrusion detection rate.

The remainder of this paper is organized as follows: Starting from the analysis of Web attack characteristics, Section 2 studies the attack techniques of common Web attacks and summarizes the attack characteristics of the bypass attack techniques for the protection detection. Section 3 studies the characterization of data detection, using SVM algorithm to establish the model and optimize the parameters. The overall design of the intrusion detection system is given in Section 4, and the CSIC datasets were used to carry out experiments in the model establishing. Conclusions are drawn in Section 5.

2 Related work

When visiting a website, attackers often analyze the vulnerability of the website first and then attempt to access the sensitive resources of the website by manually designing the uniform resource locator (URL) request. After that, the website resources are modified or injected the Trojan by attackers and the corresponding network data packet will change. Common Web attacks include SQL injection, cross-site script (XSS) attack, command execution, remote code execution, and directory traversal attacks. Among them, SQL injection usually uses the SQL blind annotation to obtain the hidden information of the database and then achieve the purpose of obtaining database information and Web shell.

At present, most WAF products are based on the blacklist of expert knowledge for detection and interception. However, hackers often bypass the detection rules to achieve attack. The following is a common bypass Web attack analysis, as shown in Table 1.

Besides, websites often have vulnerabilities in code detection and buffer overflow protection. Therefore, attackers always try to attack websites by encode bypassing or making the length of packet length large. Among them, for the detection of sensitive characters, the commonly used bypass methods are used including comments, equivalent functions or equivalent commands, special symbols, and encoding bypass of sensitive characters. These bypass laws are traceable, and the corresponding parameter features can be extracted from the attack bypassing features to mark the potential attack risk. The summary feature detection points are shown in Table 2:

According to the summarized detection points, key parameter features such as access request, parameter length, number of parameters, parameter type, parameter encoding, and parameter sensitive character are extracted. The extracted data features are used as the attribute values of the model samples, which summarize the data features of the Web attacks and solve the limitations of large network data traffic, high dimensionality, and difficulty in data feature extraction.

3 Methods

3.1 Proposed model establishment

According to the characteristics of the Web attack summarized in the previous section, especially for the common methods of bypass attacking, the parameters such as URL identifier, parameter length, sensitive character detection identifier, abnormal code detection identifier, and parameter type identifier are extracted. Then, the nonlinear SVM algorithm is applied to find a kernel function that is more suitable for this scenario. Finally, combined with cross-validation, a grid search algorithm is used to find the best combination of parameters. Because of the differences between access requests, a separate model needs to be established for each type of access request. The model establishing process is shown in Fig. 1.

Table 1 Feature analysis of bypass Web attack

Common Web protection measures	Corresponding bypassing techniques
Case sensitive	Bypass by case conversion
URL encoding detection	Bypass by the multiple URL encoding
Sensitive character detection	Bypassing by comments

3.2 Data characterization

After summarizing the characteristics of the Web attack in the previous section, the key parameter features are extracted. The characteristics of each part are as follows:

(1) Access request identifier: A HTTP request access consists of a domain name, a file path and a commit parameter. The HTTP access request is identified by the URL and parameter name and then use the md5 encryption for desensitized.

(2) Parameter length: Get the length of the character according to the extracted “post” or “get” parameters. Transform sequence $x_1, x_2, \quad, x_n$: $y_i = \frac{x_i - \overline{x}}{s}$. In which$\overline{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$, $s = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n} (x_i - \overline{x})^2}$. Then the mean value of new sequence $y_1, y_2, \quad, y_n$ is 0 and the variance is 1.

(3) Number of the parameters: Separating the parameters to obtain each parameter name and its corresponding value. Normalizing the parameters using the Min-max method, in which v_i represents the number of parameters. Performing Min-max transformation: $y_i = \frac{v_i - v_{\min}}{v_{\max} - v_{\min}}, i = 1, 2, \cdots, n$, $v_{\min} = \min\limits_{1 \le i \le n}\{v_i\}$, and $v_{\max} = \max\limits_{1 \le i \le n}\{v_i\}$. Then, the normalized attribute value data $y_1, y_2, \quad, y_n$ falls in the interval [0 1].

(4) Abnormal Encoding Detection Identification: Define 9 encoding types: urldecode, md5, sha1, sha256, base64, unicode, utf8, html entity encoding, and undefine (normal). Among them, the lengths of md5, sha1, and sha256 are different while the modes are the same. Therefore, the min-max method is used for the normalization processing.

(5) Sensitive character detection identifier: A sensitive character library is built for general attack, SQL injection, and sensitive directory scanning. Then, the Min-max method is used for the normalization processing.

(6) Parameter type identification: According to the different parameters of different URL requests, the hidden Markov model (HMM) algorithm is trained separately to obtain the score of the HMM algorithm. The type of the parameter is identified by a numerical value and the feature is digitized. Then, the *z*-score method is used for normalization, which will be expanded in the next section.

Table 2 Summary of feature detection points

Attack techniques	Feature detection points
Case sensitive, keyword replacement	Detect sensitive characters
Multiple URL encoding	Detecting the form of %25%
Comment the bypass, empty byte bypass, special symbol bypass	Detect the specific character of characteristics
Weak coding detection	Detect the hex encoding, char encoding, html entity encoding, unicode encoding
Overflow	Detect packet length

3.3 The identification of parameter type by the HMM algorithm

The HMM algorithm [15] describes how to convert a hidden Markov chain into a state sequence and how to obtain an observation sequence from a state sequence.

It can be concluded that the parameter value of the HTTP access request consists of letters, numbers, connectors (-_\), other special characters (ASCII codes with the number 32-47), and other characters such as Chinese characters. The hidden state is recorded as: S1, S2, S3, S4, S5, and the generalization of the parameter values is as follows: [a-zA-Z] is generalized to A, [0-9] is generalized to B, [-_\] generalized to C, characters with ASCII code 32-47 are generalized to D, and Chinese and other characters are generalized to N, as shown in Fig. 2.

Since the value range of the parameter values in each URL is different, it is necessary to establish HMM model according to different parameters in different URLs. Furthermore, the HMM model parameters of each parameter in each URL are obtained, and the calculation of the model parameters is performed using the Baum-Welch algorithm.

After the HMM model is obtained, all the access requests are traversed: matching the corresponding HMM models and then using the forward algorithm to calculate the probability of different parameters in different URLs. The probability of different parameters in the same URL is summed up as the tag value of this URL parameter type.

3.4 SVM algorithm modeling

SVM is a binary classification model, which is the classifier defined in the feature space with the largest interval. The learning strategy is to maximize the interval, which can be formalized into a problem of solving convex quadratic programming. The learning algorithm of SVM is an optimization algorithm for solving convex quadratic programming.

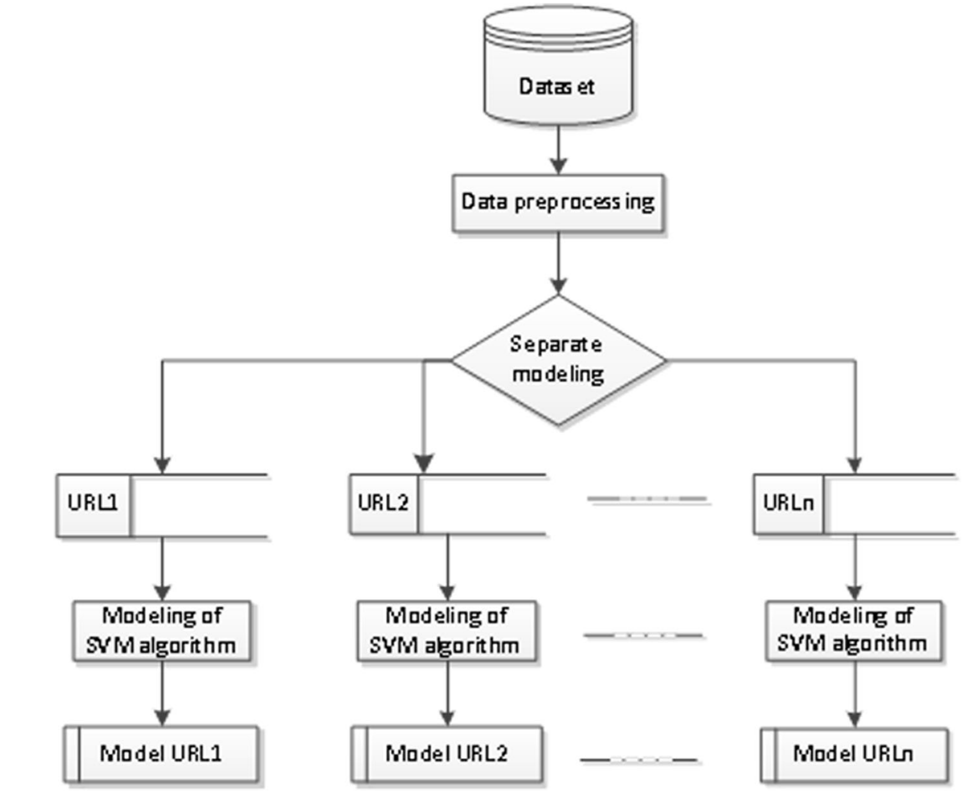


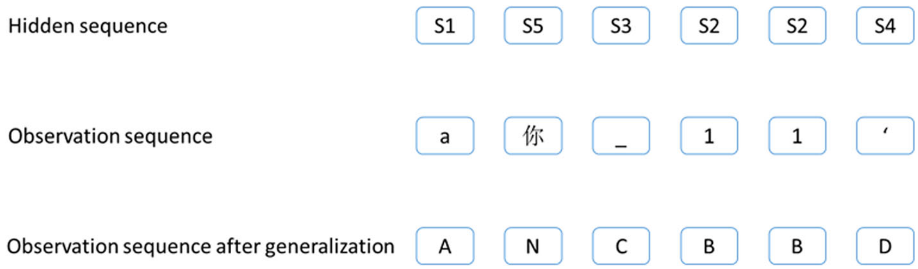
Fig. 1 Model establishing process

Assume that the training set sample is $T = \{(x_1, y_1), (x_2, y_2), \cdots, (x_n, y_n)\}$ among them, $x_i \in \mathcal{X} = R^m, y_i \in \mathcal{Y} = \{-1, +1\}, i = 1, 2, \cdots, n$. The feature vector x_i of the i-th sample, that is, the parameter vector participating in the operation, y_i is the flag of the i-th sample, and when $y_i = -1$, the sample is an attack sample, and when $y_i = +1$, the sample is a normal sample.

When the training data is linearly inseparable, the soft-interval SVM algorithm is applied, and a slack variable $\xi_i \geq 0$ is introduced for each sample, and a corresponding penalty parameter $C \geq 0$ is added to the corresponding objective function to obtain the following convex quadratic programming problem:

$$\begin{aligned} &\min_{\omega,b} \frac{1}{2}\omega^2 + C\sum_{i=0}^{n}\xi_i \\ &s.t.\;\; y_i(\omega \cdot x_i + b) \geq 1 - \xi_i, i = 1, \cdots, n \\ &\qquad \xi_i \geq 0, i = 1, 2, \cdots, n \end{aligned} \tag{0.1}$$

$$\begin{aligned} &\min_{\omega,b} \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j \mathrm{K}(x_i \cdot x_j) - \sum_{i=1}^{n}\alpha_i \\ &s.t.\;\; \sum_{i=1}^{n}\alpha_i y_i = 0 \\ &\qquad 0 \leq \alpha_i \leq C, i = 1, 2, \cdots, n \end{aligned} \tag{0.2}$$



Hidden sequence	S1	S5	S3	S2	S2	S4
Observation sequence	a	你	_	1	1	‘
Observation sequence after generalization	A	N	C	B	B	D

Fig. 2 Character generalization

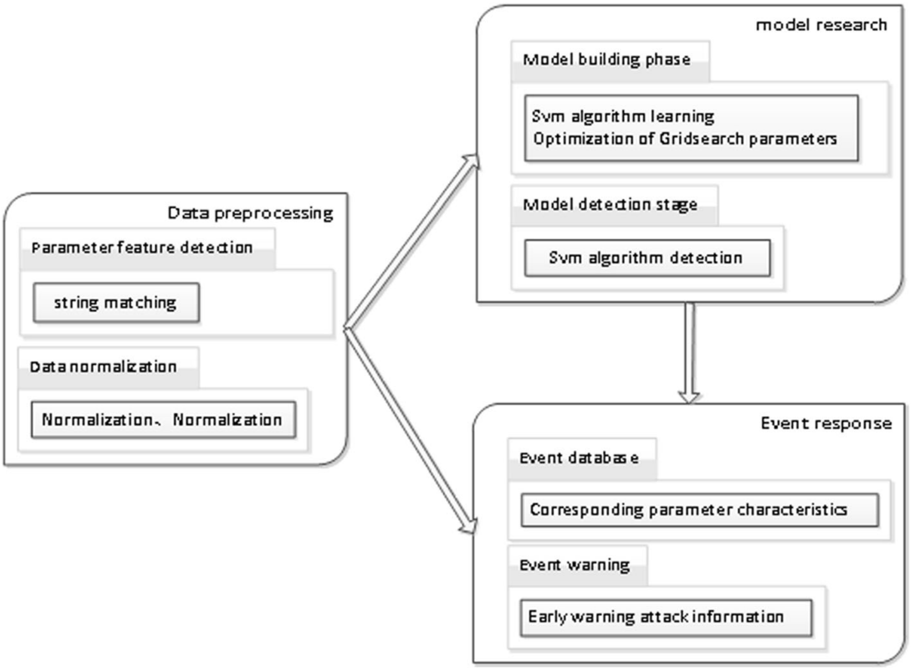


Fig. 3 System structure diagram

Among them, $C \geq 0$ is the penalty parameter, and $K(x_i \cdot x_j)$ is the kernel function. The parameter characteristics of different access requests are independent of each other, and each access request is modeled to more accurately represent the characteristics of an access request. First, get a list of URLs, sort the dataset by URL, and get the dataset for each URL. The model establishment process of the i-th URL request is as follows:

In the first step, the HMM algorithm is used to calculate the tag value of each parameter in the request, and the model parameters of the HMM algorithm are stored for data preprocessing during detection.

In the second step, the data is implemented in vectorization for the calculation of the algorithm, and the SVM algorithm is used for training and learning to obtain the model.

In the third step, the established model is stored for the use in the matching model for intrusion detection. A corresponding model is created for each URL request in the list for application detection.

3.5 Parameter optimization

Cross-validation (CV) [16, 17] groups the original data into training sets and test sets. First training with the training set, and then using the test set for verification, which is a good statistical analysis method for classifier performance testing.

Grid search method refers to the parameter dividing and traversing all points in the grid according to the network within a given range. For the selected parameters, the CV method is used to calculate the classification accuracy rate under this value. And the set of parameters with the highest classification accuracy is taken as the optimal parameter. In order to avoid the occurrence of over-learning state, when the penalty parameter C has multiple values, the minimum value is selected as the optimal parameter.

4 Results and discussion

4.1 Design of the intrusion detection system

The feature analysis and SVM algorithm-optimized Web intrusion detection system proposed in this paper is mainly composed of data preprocessing, model research, and event response. The system structure diagram is shown in Fig. 3.

Among them, the data preprocessing stage is mainly divided into two parts: parameter feature detection and data normalization processing:

1) Parameter feature detection: According to the detection points summarized in the Web attack feature, the data is characterized, and the parameter matching value of the data packet is obtained by using the string matching algorithm which is taken as the tag value of the attribute.
2) Data normalization processing: Data normalization processing is mainly performed for the abovementioned detected feature values, so as to facilitate statistical analysis of subsequent data and satisfy the data requirements of the SVM algorithm.

Table 3 HMM calculation score value

Parameter	Scores
admin	19
admin123	20
root	18
zw-123	15
Test'%3Cscript%3Ealert (1)%3C/script%3E	− 6389

The model analysis part is mainly divided into two parts: model establishment and model detection:

1) Model establishment: Firstly, the data set is classified according to the URL to obtain the data sets of each URL. Then the data is preprocessed to obtain normalized data.
   Using the HMM algorithm to identify the parameter type and the python module LIBSVM to implement the SVM algorithm. The parameter optimization is achieved by the grid search algorithm to find an appropriate parameter to ensure a better classification effect. In the simulation stage, different kernel functions can be adopted by modifying the input parameters of the functions in the LIBSVM module to seek for an optimal model, which can ensure each model has a better detection result. The model is established for each URL identifier, and the classification model function for each URL identifier is obtained for the application detection of the model.
2) Model detection: Firstly, the network accesses the network packet of the HTTP application layer and then performs protocol parsing to extract the parameter features. The normalized data is obtained according to the data preprocessing method, and the application detection is performed by matching the model data to detect whether there is an attack. If there is an abnormal attack, the detection point of the exception is extracted for the response of the event.

Table 4 Classification accuracy of different kernel functions

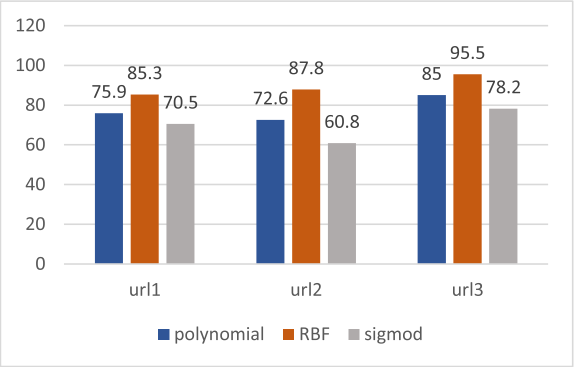


Table 5 Detection rate of different values of parameters

C (σ)	0.8	1	2	4	10	100
1	0.800	0.786	0.728	0.700	0.600	0.557
2	0.786	0.771	0.714	0.686	0.614	0.557
3	0.771	0.757	0.700	0.700	0.614	0.557
4	0.757	0.757	0.700	0.700	0.614	0.557

The event response part is composed of two parts: event database establishment and event warning.

1) Event database establishment: The corresponding event database is generated according to the parameter features in the Web attack feature analysis, and then, the event database is matched basing on the detected abnormality detection points to obtain a more likely attack event.
2) Event warning: Using the detection information to make an early warning prompt and give a possible attack mode.

4.2 Simulation

4.2.1 Dataset

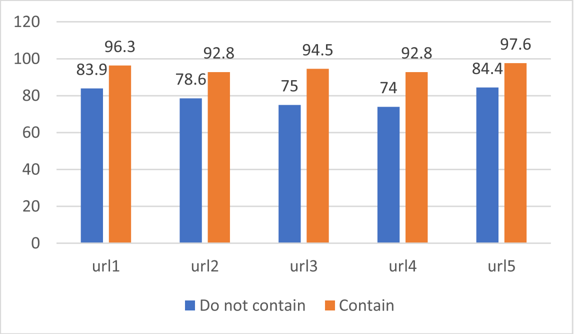
The dataset is derived from the HTTP dataset Canadian Society Of Immigration Consultants (CSIC) 2010 developed by the Information Security Institute of CSIC (National Committee for Spanish Studies), which generates traffic for e-commerce Web applications, including 36,000 normal requests and more than 25,000 exception requests. HTTP requests are marked as normal or abnormal. The data set includes SQL injection, buffer overflow, information collection, file disclosure, Carriage Return/line Feed (CRLF) injection, XSS, server-side inclusion, parameter tampering, and other attacks. It is divided into three different subsets. In the training phase, a subset with normal flow is selected. In the testing phase, two subsets are selected, one of which is normal traffic and the other one is malicious traffic.

4.2.2 Data pre-processing

The data set is consist of HTTP protocol request. Firstly, the access path and access parameters are extracted from them and then detecting the summarized parameter features to obtain the attribute values of the model.

Normalized data is obtained through data preprocessing. The training set data is classified according to the URL identifier, and the training sets with different URL identifiers are obtained and the training sets of each group are established respectively.

Table 6 Classification accuracy of different attributes



4.3 Experiment

The specific system information is as follows:

1) System: linux (ubuntu16.04)
2) Software environment: hadoop2.7.3, openjdk-8-jdk, anaconda, libsvm3.22, hmmlearn.

Hadoop Distributed File System (HDFS) is used to store data sets, anaconda is used to complete data preprocessing, the hmmlearn library is used to implement HMM model, the libsvm library is used to implement SVM algorithm, and then, the model system is established.

Because of the difference between access requests, it is necessary to separately establish models for each type of access request. First, the data set is classified according to the URL, and the data set of each URL is obtained. Data preprocessing is then performed to obtain normalized data.

The parameter type tagging process using the HMM learn library is divided into two steps:

In the first step, the data is classified according to different parameters of different URLs, and then, the preprocessing is performed to obtain the observation set of the parameter values to calculate the model parameters. The Gaussian HMM is used to calculate the model and the resulting model parameters are saved.

In the second step, the obtained model is used for calculating the probability of each observation sequence, which is used to identify the parameter type of the URL. The different parameter values of the same URL are preprocessed and then written into the dataset.

Use the score function to calculate the score value of each parameter value which is shown in Table 3. A larger value indicates a higher probability that the value of the parameter will appear and vice versa. Taking the username parameter of a certain URL as example:

This section uses classification accuracy (ac) to evaluate the classification effect of the model. Through data preprocessing, the data set of the SVM algorithm is obtained. The calculation of the SVM algorithm is implemented using the LIBSVM module C-Support Vector Classification (C-SVC) class problem. The time complexity of the SVM algorithm is $O(m^2n^2)$ which is closely related to the number of samples (m) and the dimension of the feature (n). Meanwhile, the optimization problem directly affects the computation time of the algorithm. The Sequential Minimal Optimization (SMO) algorithm optimization process selects two samples per operation, which can greatly shorten the calculation time.

First, select the kernel function. After experiments, the radial basis function kernel (RBF) kernel, the polynomial kernel, and the sigmod kernel are selected as the kernel functions to build the model. The classification accuracy of the model is shown in Table 4, taking three random selected URL as example:

The experimental data shows that the classification accuracy of RBF core is better than the polynomial kernel, and the polynomial kernel is better than the Sigmod core. Therefore, this paper adopts the RBF kernel function to establish the model which can guarantee the accuracy of the classification.

The SVM algorithm with the RBF kernel is implemented after which the grid search parameter optimization is realized. The detection rate of the SVM algorithm with different C, σ is shown in Table 5:

Experiments show that the value of the parameter greatly affects the detection effect. It is especially important to select an appropriate parameter. This paper is devoted to finding the appropriate parameters to ensure that the established model has a good detection rate in a short time. Therefore, the grid search algorithm is

Table 7 Detection result of the model

Inde Group	Number of access	Number of attacks	Normal number	Detection rate	False alarm rate
1	2128	864	1264	0.982	0.014
2	2136	868	1268	0.952	0.036
3	2120	850	1270	0.985	0.020
4	2172	904	1268	0.938	0.034
5	2116	860	1256	0.988	0.012
average	2134	869	1265	0.958	0.0232

Table 8 Detection result of the other existing model

Papers	Paper 1 [18]	Paper 2 [19]	Paper 3 [20]	Paper 4 [21]	Paper 5 [22]
Detection rate	91.81	91.53	96.63	94.6	89.71
False alarm rate	0.55	0.58	9.1	3.0	0.1

selected. In the LIBSVM module, a grid.py is provided to implement the grid search. The parameters are continuously changed according to the fixed step size, and then, the detection rate is calculated. The set of parameters with the highest detection rate is taken as the optimal parameter. Although this may not be the best one, it guarantees that the models established for each type of URL has an appropriate parameter to ensure the detection capabilities.

In this paper, the HMM algorithm is used to identify the parameter type with different parameters, through experiments, building the model with and without parameter type identification respectively, and selecting the most representative five URLs. The classification accuracy of the model is shown in Table 6.

Experiments show that the HMM algorithm can distinguish abnormal parameters from normal parameters, which can improve the detection effect of the model..

Finally, five sets of data were randomly selected and tested with the established model and the detection result is shown in Table 7.

Experiments show that the model can achieve 95.8% detection rate and 2.32% false alarm rate, which is a very good effect.

In addition, in the existing research of hybrid intrusion detection model, the detection effect obtained by using KDD CUP99 training set is as shown in Table 8.

It is shown that in the existing research of hybrid intrusion detection model, the detection rate has mostly reached 90%, and the false alarm rate is less than 10%, so that the intrusion detection model studied in this paper has a relatively good detection effect.

5 Conclusion

This paper summarizes the characteristics of common Web attacks, combines with the analysis of HTTP protocol, and selects relevant data features, which solves the limitations of large network data traffic, high dimension, and difficulty in data feature extraction. Because there are large differences in the parameter characteristics of different access requests, separate models are established for each type of access requests. Data set are sorted by URL and then preprocess the data to obtain normalized data. Then, the parameter type is identified by the HMM algorithm. SVM algorithm is used for the learning classification and finally using the grid search method for parameter optimization. The simulation and experimental results show that the SVM algorithm with RBF kernel function has better detection effect. At the same time, the using of grid search optimization can accelerate the parameter optimization process and ensure each model has a better detection capability in a short time, which is especially important in the dynamic and complex network environment.

Abbreviations

WAF: Web application firewall; HTTP: Hypertext Transfer Protocol; GTIC: Global Threat Intelligence Center; SQL: Structure Query Language; HMM: Hidden Markov Model; URL: Uniform Resource Locator; XSS: Cross-site script; CV: Cross-validation; CSIC: Canadian Society Of Immigration Consultants; CRLF: Carriage Return Line Feed; C-SVC: C-Support Vector Classification; RBF: Radial Basis Function kernel; SMO: Sequential Minimal Optimization

Authors' contributions

JY contributed to the conceptualization, software, manuscript reviewing, editing, and funding acquisitions. CL contributed in the software development, investigation, methodology, simulations, result analysis, and draft manuscript writing. All authors read and approved the final manuscript

Funding

The authors thank the project of the National Natural Science Foundation of China No.61672179, No.61370083, No.61402126, and No. 61901134; the Natural Science Foundation of Heilongjiang Province (Grant No. F2015030, 201336); the Fundamental Research Funds in Heilongjiang Provincial Department of Education No. 135109247; and the Science Foundation for Young Scientists of Qiqihar University No. 2014k-M09. The authors also thank the Technical Bureau of Qiqihar GYGG-201516 and GYGG-201622.

Availability of data and materials

The datasets used and/or analyzed during the current study are available from the corresponding author on a reasonable request

Competing interests

The authors declare that they have no competing interest.

Author details

[1]Harbin Engineering University, Harbin, China. [2]Qiqihar University, Qiqihar, China.

Received: 15 July 2019 Accepted: 30 October 2019
Published online: 03 February 2020

References

1. NNT Security. 2017 Global Threat Intelligence Center (GTIC) Quarterly Threat Intelligence report Q2. http://www.nttcomsecurity.com/uploads/documentdatabase/NTT%20Security%20Q2%202017_FINAL.pdf
2. S. Devaraju, Performance comparison for intrusion detection system using neural network with KDD Dataset [J]. Ictact J. Soft Comput. **4**(3), 743–752 (2014)
3. Y. Zhao, Network intrusion detection system model based on data mining [C]// Ieee/acis International Conference on Software Engineering, Artificial Intelligence, NETWORKING and Parallel/distributed Computing. IEEE **1**, 155–160 (2016)
4. Mukkamala S, Janoski G, Sung A. Intrusion detection using neural networks and support vector machines[C]// International Joint Conference on Neural Networks. IEEE,(2002), pp.1702–1707
5. Yang L , Li J , Fehringer G, et al. Intrusion detection system by fuzzy interpolation [C]// IEEE International Conference on Fuzzy Systems. IEEE, 2017.

6. Akhil J , Sultana A. Intelligent network intrusion detection system using data mining techniques [C]// ICATCCT16. IEEE, 2016.
7. D.P. Muni, N.R. Pal, J. Das, Genetic programming for simultaneous feature selection and classifier design [J]. IEEE Trans. Syst. Man. Cybern. B. Cybern. **36**(1), 106–117 (2006)
8. J. Kennedy, R.C. Eberhart, *Particle swarm optimization [C]// Proceedings of IEEE International Conference On Neural Networks* (IEEE, Perth, 1995), pp. 1942–1948
9. C.-H. Chen, H.-Y. Kung, Feng-Jang Hwang. Deep learning techniques for agronomy applications [J]. Agronomy **03**(9), 1 (2019)
10. S.X. Tang, W.J. Cai, Intrusion detection based on unsupervised clustering and hybrid genetic algorithm [J]. J. Comput Appl. **28**(2), 409–411 (2008)
11. X. Li, Network intrusion detection using genetic algorithms for synchronized selection of features and support vector machine parameters [J]. Comput. Appl. Softw. **3**, 301–303 (2014)
12. T. Zhang, J. Wang, Network intrusion detection model based on CQPSO-LSSVM [J]. Comput. Eng. Appl. **51**(2), 113–116 (2015)
13. L.I. Zhengang, Q. Gan, University T C, Network intrusion detection model based on MACO-SVM[J].J.Journal of Chongqing University of Posts and Telecommunications. **26**(6), 785–789 (2014)
14. P.R.K. Varma, V.V. Kumari, S.S. Kumar, Feature selection using relative fuzzy entropy and ant colony optimization applied to real-time intrusion detection system [J]. Procedia Comput. Sci. **85**, 503–510 (2016)
15. L. Denoyer, H. Zaragoza, P. Gallinari, *HMM-based passage models for document classification and ranking [J]*. Ecir (2016), pp. 126–135
16. Y. Zhang, B. Li, H. Lu, et al., *Sample-specific SVM learning for person re-identification [C]// IEEE Conference on Computer Vision and Pattern Recognition*. IEEE (2016), pp. 1278–1287
17. J.I. Changming, T. Zhou, T. Xiang, et al., Application of support vector machine based on grid search and cross validation in implicit stochastic dispatch of cascaded hydropower stations [J]. Electric Power Automation Equip. **34**(3), 125–131 (2014)
18. B. Pfahringer, Winning the KDD99 classification cup: bagged boosting. ACM SIGKDD Explorations Newsletter **1**(2), 65–66 (2000)
19. I. Levin, KDD-99 classifier learning contest: LLSoft's results overview. ACM SIGKDD Explorations Newsletter **1**(2), 67–75 (2000)
20. C. Xiang, S.M. Lim, in *IEEE Workshop on Machine Learning for Signal Processing*. IEEE. Design of multiple-level hybrid classifier for intrusion detection system (2005), pp. 117–122
21. L. Kuang, M. Zulkemine, *An anomaly intrusion detection method using the CSI-KNN algorithm. In; Proceedings of the 2008 ACM symposium on Applied computing*. ACM (2008), pp. 921–926
22. Z. Ma, *Application of hybrid soft computing technology in intrusion detection [Doctoral Dissertation]* (Chongqing University, Chongqing, 2010)

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.