MINISTRY OF EDUCATION AND TRAINING

VIETNAM ACADEMY OF SIENCE AND TECHNOLOGY

GRADUATE UNIVERSITY OF SIENCE AND TECHNOLOGY

NGUYEN TRONG HUNG

RESEARCH ON WEB ATTACK DETECTION SOLUTIONS USING WEB LOGS AND WEB PAGE SCREENSHOTS

SUMMARY OF DISSERTATION ON INFORMATION SYSTEM
Code: 9 48 01 04

Ha Noi - 2024

The dissertation is completed at: Graduate University of Science and Technology, Vietnam Academy of Science and Technology

Supervisors:
Supervisor 1: Assoc.Prof, Dr Hoang Xuan Dau, Posts and Telecommunications Institute of Technology
Supervisor 2: Assoc.Prof, Dr Nguyen Duc Dung, Institute of Information Technology, Viet Nam Academy of Sciences and Technology

Referee 1: Assoc.Prof, Dr Bui Thu Lam
Referee 2: Assoc.Prof, Dr Nguyen Linh Giang
Referee 3: Assoc.Prof, Dr Luong The Dung

The dissertation will be examined by Examination Board of Graduate University of Science and Technology, Vietnam Academy of Science and Technology at…….. (time, date, year…)

This dissertation can be found at:
1) Graduate University of Science and Technology Library
2) National Library of Vietnam

LIST OF THE PUBLICATIONS RELATED TO THE DISSERTATION

1. Hoang Xuan Dau, Ninh Thi Thu Trang, **Nguyen Trong Hung**,"*A Survey of Tools and Techniques for Web Attack Detection"*. Journal of Science and Technology on Information security, Special Issue CS (15) 2022, pp. 109-118.
2. **Trong Hung Nguyen**, Xuan Dau Hoang, Duc Dung Nguyen, *"Detecting Website Defacement Attacks using Web-page Text and Image Features"*, Article Published in International Journal of Advanced Computer Science and Applications(IJACSA), Volume 12 Issue 7, 2021, Scopus Q3.
3. Hoang Xuan Dau, **Nguyen Trong Hung**, *"Phát hiện tấn công web thường gặp dựa trên học máy sử dụng web log",* Hội nghị khoa học quốc gia về "Nghiên cứu cơ bản và ứng dụng Công nghệ thông tin" FAIR 2020.8.
4. **Trong Hung Nguyen**, Dau Hoang, Nguyen Duc Dung, Vu Xuan Hanh, *"Phát hiện tấn công thay đổi giao diện trang web sử dụng đặc trưng văn bản",* Hội nghị KHCN Quốc gia lần thứ XVII về Nghiên cứu cơ bản và ứng dụng Công nghệ thông tin(FAIR), Hà Nội, 8/2024.
5. Xuan Dau Hoang, **Trong Hung Nguyen**, Hoang Duy Pham, *"A Novel Model for Detecting Web Defacement Attacks Using Plain Text Features"* Indonesian Journal of Electrical Engineering and Computer Science (IJEECS), 2024, Scopus Q3.

INTRODUCTION

1.1. The urgency of the thesis

Due to the dangerous nature of web attacks to agencies, organizations and individuals, many solutions have been researched, developed and deployed to detect, prevent, and prevent web attack, such as the use of a web firewall (WAF), Web IDS, and intrusion testing [40][72][86]. In general, there are currently two main approaches to web attack detection: (1) signature-based detection, and (2) abnormal-based Detection [40] [49] [80].

In the approach (2), a thesis on the use of abnormal-based web attack detection techniques, More specifically, the thesis focuses on two main directions: (i) the detection of basic forms of web attacks, including SQLi, XSS, path browsing, CMDi, and (ii) the detecting of web interface-change attacks. In direction (i), there are not many surveys using datasets from web logs and these studies usually only detect one form of attack on a specific experimental data set. *Therefore, this thesis continues to investigate the simultaneous detection of common forms of web attacks, including SQLi, XSS, path browsing, CMDi based on web log data using supervised machine learning models.* In direction (ii), through surveys, the evaluation of most studies has focused on using only one type of characteristic related to website content without a combination of typical characteristics, including content and images of the attacked site changing the interface. *Therefore, the thesis focuses on detection methods for changing website interface attacks using deep learning algorithms and combining text/content characteristics and forms of expression - web screenshots to improve accuracy, speed, and time of computing.*

2. Research objectives of the thesis

- Research, evaluation, methods, techniques, solutions, tools to detect web attacks.

- Research proposes a model for detecting common forms of web attacks based on monitored machine learning techniques using web log data, to improve accuracy, reduce false warnings, and allow for detection of various types of Web attacks.

- Research proposed an attack detection model that modifies the site interface based on deep learning techniques and combines two types of text and visual characteristics of the site, to improve accuracy and reduce false warnings.

- Install, test and evaluate proposed web attack detection models using published datasets and actual data collection.

3. The main research content of the thesis

Chapter 1. Overview of Web Attack Detection An overview of the web and web services, web security vulnerabilities under OWASP, common types of web attacks, a number of solutions and tools for web attack detection. Next, this chapter gives an overview of machine learning, deep learning and describes some supervised machine learning and deep learning methods used in the web attack detection models proposed in chapters 2 and 3. The end of the chapter indicates two issues that will be addressed in the thesis.

Chapter 2. Machine learning based web attack detection using web log A general introduction to web logs, some proposals to detect web attacks using machine learning, an assessment of the advantages of the proposals. The final part of this chapter carries out the construction, installation, testing and evaluation of common web attack detection models based on machine learning using web logs.

Chapter 3. Detection of site interface change attacks An overview of interface change attacks, methods of detection of interface changes, comparison of methods to detect interface change using web screenshot characteristics. The final part of the chapter carries out the construction, installation, testing and evaluation of a deep learning-based attack detection model that uses a combination of screenshot characteristics and site text characteristics.

CHAPTER 1: OVERVIEW OF WEB ATTACK DETECTION

1.1. Overview of web and web services

Web service: World Wide Web Consortium (W3C) A web service is a software system that allows different machines to interact with each other over a network. Web services accomplish this task with the help of open standards, including XML, SOAP, WSDL and UDDI [33]. *A web application* is a software application that runs on the web [102]. Web applications are also run based on the HTTP client model (Client/Sever). *A website* is a set of websites installed and hosted on a web server. A Web page is a part of a website that provides a content header or a specific feature of the website. The common language used to create web pages is HTML.

1.2. Overview of web attacks

Web attack, or web application attack, is the exploitation of weaknesses, vulnerabilities that exist on the website system, web application to perform exploitative behaviour, steal sensitive data that exists on the system [68]. Also according to [68], up to 75% of cyber attacks have recently been carried out at the web application level.

These include various types of attacks, common intrusion into websites, web applications (abbreviated as web attacks), including SQL injection, XSS (Cross-Site Scripting), Cross-site Request

Forgery (CSRF), Command Injection (CMDi), path browsing, DoS/DDoS and interface change attacks [21] [37] [41].

1.3. Detected Web Attack

In general, there are three defensive approaches to these attacks, including (1) checking, authenticating all input data, (2) reducing attack surfaces, and (3) using the "defense in depth" strategy [8] [41] [111]. Specifically, the approach (1) requires all input data for web applications to be thoroughly tested using input data filters and only the legitimate input is transferred to the next steps for processing. On the other hand, the approach (2) requires splitting the web application into several parts and then applying appropriate access controls to restrict user access. With regard to the approach (3), a number of defences are deployed in successive layers to protect websites, web applications, and web users.

Web attack detection solutions and tools: There are many solutions, web attack detection tools developed and practical applications deployed, such as [6][25][61][77][78][103][104][109]. Web Attack Detection Techniques: There are many proposed and applied web attack detection techniques over the years. However, there are two commonly used types of technical detection of web attacks, including (1) detection based on signature, template, or set of rules [1] and (2) anomaly-based detection [32].

1.4. The research direction of the thesis

The research direction of the thesis is to detect common web attacks and attacks that change the web interface on an abnormal basis because this method is capable of detecting new forms of web attack, while also being able to automate the development of detection models. On the basis of surveys, analysing the advantages and limitations of existing proposals, the research-focused thesis

addressed the following issues: (1) Proposing a commonly found web attack detection model based on machine learning using web logs and (2) Proposed a web interface change detection module based on in-depth learning using a combination of web site content text data and web site screenshots. The reason (1) is because some abnormal-based detection techniques detect only one type of attack on a specific data set, without simultaneously detecting multiple types of web attacks, such as: XSS, SQLi, path browsing, CMDi. In addition, some abnormal-based detection proposals have low and low correct detection rates and high false warning rates. Similarly, the implementation (2) aims to increase the correct detection rate and reduce the false warning rate for an interface-change attack detection model using input data combined between the text data of the site content and the screenshot of the website.

CHAPTER 2: MACHINE LEARNING-BASED WEB ATTACK DETECTION USING WEB LOGS

2.1. Machine Learning Based Web Attack Detection

Research and survey finds that proposed solutions to detect web attacks based on web log data are an effective direction. In particular, the research direction using machine learning is promising due to a simple detection model, which can be constructed automatically from the training data set. This is also the research branch of the dissertation.

Some of the issues that need further research are: (1) some proposals, though using simple mechanisms, but only for high detection accuracy with specific datasets or with a specific type of web attack, and too few or too many typical characteristics such as the studies of Sharma and colleagues [91], Saleem and associates [85]; (2) some proposal to use deep learning models or to use a server

monitoring toolkit should require large computational costs for model building, as well as detection monitoring and this reduces the ability to deploy applications on real systems[58][76]; and (3) some suggestions to use in-depth modeling, which requires a lot of computing resources, but does not detect many forms of web attacks (SQLi, XSS, CM, Browse path), such as [40][58].

2.2. Building and testing a machine learning-based web attack detection model using web logs

2.2.1. Description of the detection model

2.2.1.1. Model introduction

The proposed web attack detection model is deployed in two phases: (a) the training phase and (b) the detection phase. The training phase is as shown in Figure 2.4.

During the training phase the attack and normal URI data is collected, followed by the preliminary data processing to extract the characteristics for the training process. In the training step, supervised machine learning algorithms, such as Naïve Bayes, SVM, Decision Tree, Random Forest are applied to learn the classification set, the algority for the best results will be used for the detection model. During the Detection phase, the URI queries will be filtered from the weblog data, through a preprocessing process such as the Training phase and to the usage classification phase. The classifier from the Training phase to determine the Normal or Attack query.

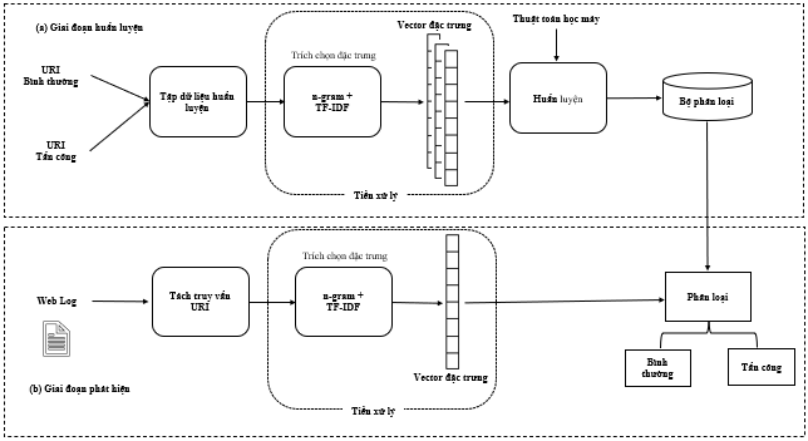


Figure 2.4. Web attack detection model based on weblog data

2.2.1.2. Data preprocessing, training and detection

Preprocessing web log data based on n-gram, TF-IDF and reduction techniques is done in the following steps:

Step 1: Separate queries **?query_string** in URI queries

Step 2: Separate n-gram characteristics from these queries

Step 3: Calculate values for n-gram characteristics using the TF-IDF method [105].

Step 4: Reduce data using the correlation coefficient method, the Information Gain method, or the PCA method.

Machine learning algorithms used include naive bayes, SVMs, decision trees, and random forests. For each algorithm, randomize 80% of the data used for the training process to construct the detection model, then use the 20% data to test for the results of the evaluation measurements..

2.2.2. Test dataset

HTTP Param Dataset [88] has normal queries filtered from the CISC 2010 CTTP dataset [5] and SQLi, XSS, CMDi attack queries, path browsing queries made from the SQLmap attack environments XSSya, Vega Scaner, FuzzDB repository. This data set consists of

31,067?query_string query strings in the URI of web requests, including the length and tag of the query. There are two types of query labels: Norm (Normal) and Anom (Attack). Anom tags include four specific types of attacks: SQLi, XSS, CMDi, and path browsing.

2.2.3. Test and results

2.2.3.1. Test scenario

Scenario 1: Assess the impact of the 2-gram, 3-gram, 4-gram, 5-gram parameters on the proposed model with the Random Forest machine learning algorithm from which to select the n-gram parameter for the best result. In this scenario, the thesis retains the original characteristics and does not use the data reduction method.

Scenario 2: Assess the impact of the three methods of data reduction: PCA, Information Gain, and the characteristic correlation coefficient obtained from Script 1 (the Random Forest algorithm used with n-grams for best results). From there, the data reduction method is chosen for the best results.

Scenario 3: Evaluation of the results of the training model using supervised machine learning algorithms Navie Bayes and SVM, Decision Tree, Random Forest (10, 30, 50, 60 trees) with 3-gram and PCA data reduction methods from Scenarios 1 and 2 results, from which the selection of the best results will be used for the detection process.

Scenario 4: Evaluate the proposed model with monitored machine learning algorithms for the best results from Scenary 3 with related studies.

2.2.3.2. Test results

Table 2.4. Results of simulation of Scenario 1

Algorithm	n-gram	PPV	TPR	FPR	FNR	ACC	F1	Time(s)
Random Forest	2-gram	98,94	99,32	0,64	0,68	99,34	99,13	17,90

Algorithm	n-gram	PPV	TPR	FPR	FNR	ACC	F1	Time(s)
	3-gram	100	99,14	0	0,86	99,68	99,57	92,99
	4-gram	99,91	99,1	0,05	0,9	99,63	99,51	132,56
	5-gram	100	98,80	0	1,20	99,55	99,40	135,23

Results from Table 2.4 showed with the Random Forest algorithm using 3-gram characteristics for the highest general ACC accuracy and F1 measurement compared to using 2-gram, 4-gram and 5-gram characterizations.

Table 2.5. Results of Scenario 2

Algorithm	Dimensionality Reduction	PPV	TPR	FPR	FNR	ACC	F1
Random Forest	PCA	98,97	98,72	0,62	1,28	99,13	98,84
	Information Gain	99,28	94,53	0,41	5,47	97,68	96,85
	Correlation Coefficient	99,59	92,77	0,23	7,23	97,14	96,06

Results from Table 2.5 show that after implementing data reduction, the reduction method with PCA for ACC, F1, Recall results is the highest compared to the method with Information Gain and Correlation Factor.

Table 2.6. Scenario 3 Results

Algorithm	PPV	TPR	FPR	FNR	ACC	F1
NavieBayes	89,48	96,41	6,84	3,59	94,38	92,82
SVM	99,87	98,50	0,08	1,50	99,09	99,18
Decision tree	96,48	98,42	2,17	1,58	98,05	97,44
Random Forest- 10	98,13	98,85	1,14	1,15	98,86	98,49
Random Forest- 30	98,68	98,80	0,80	1,20	99,05	98,80
Random Forest- 50	**98,97**	**98,72**	**0,62**	**1,28**	**99,13**	**98,84**
Random Forest- 60	98,80	98,76	0,72	1,24	99,08	98,78

The results in Table 2.6 show that when using the Random Forest algorithm (50 trees) with a 3-gram characteristic, combining

the PCA data reduction method for the best ACC and F1 measurement results, the NavieBayes algoritm gives the lowest result..

Table 2.7. Scenario 4 Results

Thuật toán	PPV	TPR	FPR	FNR	ACC	F1	Thời Gian huấn luyện	Thời gian phát hiện
Đề xuất - Rừng ngẫu nhiên (50 cây)	**98,97**	**98,72**	**0,62**	**1,28**	**99,13**	**98,84**	**27,52**	**1.49**
Liang và cộng sự[58]	99,04	96,88	1,13	3,12	97,78	97,95	1177,20	5,67
Ming Zhang và cộng sự [111]	98,59	93,35	1,37	6,65	96,49	95,92	151,00	4,18
Saiyu Hao cùng cộng sự [40]	98,77	93,71	0,62	6,29	97,41	96.17	13063,56	15,05
Pan và cộng sự [76]	90,60	92,80				91,80		
S. Sharma và cộng sự[91]	99,60	91,52	0,20	8,48	96,91	95,39		

Table 2.7 shows the Random Forest algorithm (50 trees) used for the suggested model for results with measurements of ACC, F1, Recall better than the proposed [40][58][76][91][111].

Table 2.8. Detection Rate (DR) for Web Attacks on Machine Learning Algorithms

Algorithms	SQLi(%)	XSS(%)	CMDi(%)	Path(%)	Average (%)
Random Forest	99,90	98,68	82,02	98,62	99,67

Table 2.8 shows the results of the proposed model in the detection of specific types of attacks: SQLi, XSS, CMDi, path browsing and average detection rates As you can see, the model for the highest SQLi attack detection rate and the lowest CMDi attack.

2.3. Comment

The thesis will evaluate the detection performance of the proposed model on the basis of the following aspects: (1) the impact of the distribution of the number of web attacks on detection rates, (2) the detection performance of a model based on different machine

learning algorithms, and (3) comparisons between the suggested model and previous proposals. The number of specific types of web attacks in the data set is not evenly distributed and therefore affects the detection performance with each type of web attack. This results in the highest detection rate for SQLi attacks and the lowest for CMDi attacks. In terms of detection performance, the proposed model has the highest accuracy of ACC, F1, Recall compared to [40][58][76][91][111] and the training time of the suggested model is also much faster than that of the studies [40][58][91].

CHAPTER 3: DETECTING WEBSITE DEMOLITION ATTACKS

3.1. Interface change attack, interface change attack detection methods

A site interface change attack is the exploitation of vulnerabilities on a website or web server to execute mining code that changes the interface or deletes, modifies the content of a website through text, images, or both [7].

From the evaluation studies surveyed in section 1.3.3.2. of a type.You can see the suggested solutions for the detection of the site's existing problems:

- Solutions like checksum testing, DIFF comparison, and DOM tree analysis can only work well with web sites..

- Some proposals require more computing resources because they use highly complex detection models, typical of [13][24].

- Some other proposals have high levels of false warning, while detection performance depends on the selection of detection thresholds, typically research work [54]

- Many suggestions can only handle the text of the pages web. Other important web site components, such as JavaScript code, CSS,

embedded image files that are not processed or are only processed by simple techniques, like integrity testing based on hash functions, typically studies [38][43][44].

- Studies [13][38][43][44] used small or very small datasets with about 300 to more than 1,000 site data attacked and barreledg. Small set of test data affects the reliability of the results.

- Most studies focus only on the characteristics of HTML files and check the hash code of images embedded in web content, no studies focus on pure text content in HTML files combined using screenshots of we pages.

According to the study [45] the combination of attack data and text data, visual data is a rich source of data for large-scale detection of cross-cutting attack activity.n. On the other hand, according to Mao and his colleagues [66] and the statistics of attacks that altered the site's interface from [12] showed that after the interface was changed, the new site appeared only in one color (black-white, red-black,...) or contained messages, embedded images or logos and videos that were not related to the title of the page web. Therefore, the model proposal is based on the combination of two characteristics of the website screenshot and the text content of the page web.

3.2. Collect test data

The studies [13] [38] [43] [44] [106] used relatively small datasets ranging from 100 to about 4,000 normal and hacked web templates. Such small datasets don't really fit the deep math algorithms used in training and detection models. At the same time, these studies gathered attack-labelled data from open-source, hosted interface-change websites, such as zone-h.org, zone-xsec.com, and did not publish usage data. Therefore, the dissertation will proceed to collect attack data with the label “Defaced” from the source zone-h.org - this

is a repository of data on site interface change attacks established since 2002 with many parameters stored, including screenshots of the site at the time of the attack. Data from regular websites labeled “Normal” is collected from the one million pages set provided by Alexa [99].

3.3. Detect interface changes using website screenshots

3.3.1. Description of the detection model

The model that uses input data is a screenshot of the normal site and the attacked site changes the entire interface. The model is built on the basis of an analysis of the identity characteristics of an attack that changes the site's interface, when the site is attacked changes the interface, the entire site content is changed, and the site interface is changed. The proposed model is deployed in two phases: (1) the training phase and (2) the detection phase..

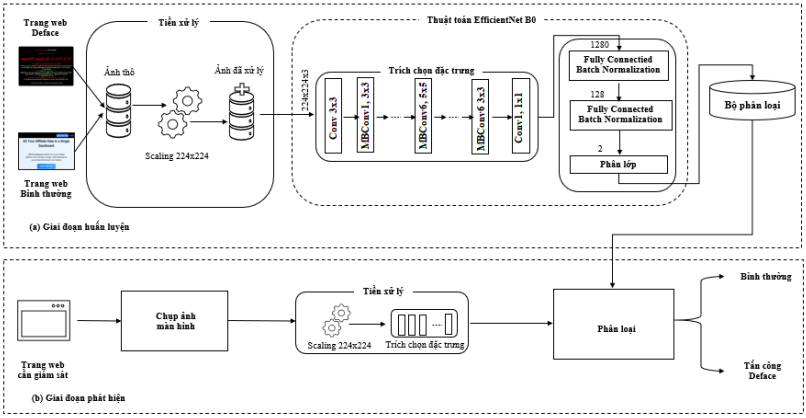


Figure 3.9. Attack detection model changes the site interface using a screenshot of the site

3.3.2. Data Preprocessing and Training

From the data collected from normal and affected sites, the program performs screenshots of each site, preprocessing, standardizing as input data for the EfficientNet(B0) algorithm,

performing characteristic selection and training. Specifically, the process is as follows:

Step 1: Use Scaling to standardize the original raw image to the correct size of 224x224 - the standard input size of the EfficientNet(B0) algorithm [98].

Step 2: Using the EfficientNet(B0) model with the basic structure as in Table 3.2 (in the thesis contents), through each layer of the algorithm obtained 1280 characteristic vector.

Step 3: After obtaining the 1280 character set from the EfficientNet(B0) model, use a BatchNormalization layer to standardize the data, avoiding interference with the characteristics. Then there's a Dense layer with 128 nodes using the activation function softmax combined with a BatchNormalization layer to standardize soon afterwards, the output value of this layer will be included as the input value to calculate the final output with two nodes.

3.3.3. Test data set

Test data used as described in section 3.2. (in the content of the thesis).

The data set is randomly divided into three parts: a training set, a validation set, and a test set in the following proportions:

- 60% training is used to enter the model and refine the model parameters;

- A 20% authentication set is used to test the accuracy of the model during the model training to adjust the model parameters to avoid overmatches during the training;

- 20% of the tests are used to evaluate the model after the model has been trained.

3.3.4. Tests and results

Table 3.4. Performance of the detection model with deep mathematical algorithms

Deep learning	**ACC(%)**	**PPV(%)**	**TPR(%)**	**F1(%)**	**FPR(%)**	**FNR(%)**
EfficientNet(B0)	94.12	94.60	90.71	92.62	3.55	9.29
Xception	94.01	93.98	91.21	92.58	4.05	8.79
Inception	89.91	89.37	84.78	87.02	6.69	15.22
Bi-LSTM	89.18	87.73	85.22	86.46	8.13	14.78

Table 3.4 provides performance of the proposed model based on EfficientNet(B0) and the models based on Xception, Bi-LSTM and Inception. The results show that the proposed model based on EfficientNet(B0) for the best ACC and F1 measurements is followed by the Xception, Inceptio-based model.

Table 3.6. Predictive model performance with deep learning algorithms and previous models

Detection model	Features	ACC(%)	PPV(%)	TPR(%)	F1(%)	FPR(%)	FNR(%)
Naïve Bayes Hoang [38]	Text	82,54	78,12	79,26	78,69	15,21	20,74
Decision Tree Hoang [38]	Text	87,33	84,4	84,4	84,4	10,67	15,6
Random Forest Hoang [44]	Text	93,88	93,81	90,76	92,26	4,03	9,24
Xception	Image	94,01	93,98	91,21	92,58	4,05	8,79
Inception	Image	89,91	89,37	84,78	87,02	6,69	15,22
Bi-LSTM	Image	89,18	87,73	85,22	86,46	8,13	14,78
EfficientNet(B0)	Image	94,12	94,60	90,71	92,62	3,55	9,29

Table 3.6. provides synthesized data comparing the performance of the proposed model with deep mathematical algorithms and detection models based on the Naïve Bayes, Decision Tree and Random Forest proposals in the Desert [38] and Desert[44]. It can be seen that the model proposed for performance is better than

the proposed by Hoang and his colleagues [44] and much better than that of Hoang & his associates [38].

Limitations: Although the overall accuracy of the proposed model is significantly higher than that of the existing models, the false warning rate, including FPR and FNR remains above 10% is relatively high and with poorly detected model image characteristics with images with little color variation such as Figure 3.5. Therefore, the next part of the thesis will provide an attack detection model that changes the web interface that can solve the problems that still exist above by using a text-specific-detection model in HTML files.

3.4. Detect changes to the interface using text content

3.4.1. Model Introduction

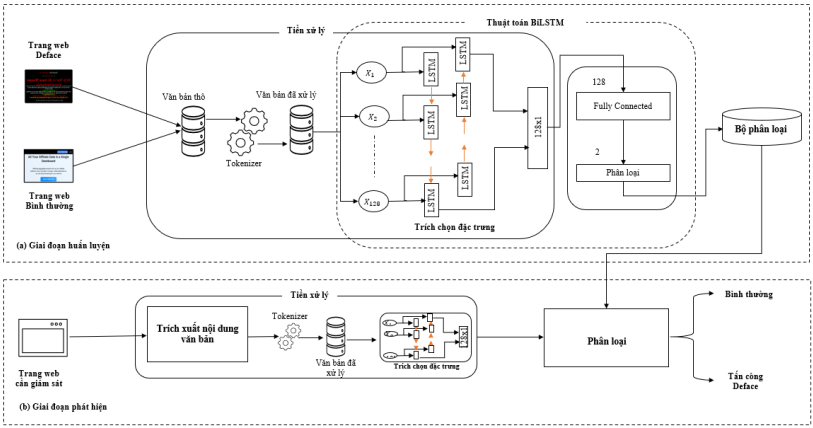


Figure 3.16. Training model, attack detection changes interface with text characteristics

Figure 3.16 shows the proposed model for detection of site interface change attacks using text characteristics through two phases: (a) training phase and (b) detection phase. During the training phase, the training data set is collected from extracting text content in the attacked and normal web pages, then separated into words with the

Tokenizer technique, then trained with the BiLSTM deep learning algorithm to generate the classifier. During the detection phase, the monitoring site is extracted from the text, through the data preprocessing process such as the training phase and to the classification phase using the classifier from the Training phase to determine whether the status is normal or the interface has changed.

3.4.2. Data Preprocessing and Model Training

Step 1: From normal web pages and changing web pages, use a Python handwritten program to extract the text as data for the training process.

Step 2: From the text data set obtained, the Tokenizer technique [95] is used to separate the words in the text and each word is mapped into a positive integer.Then select the first 128 words in a row as input to the BiLSTM algorithm.

Step 3: Use the Embedding layer to help the model understand the semantic relationship of words through the input vector of the model. The result is a 128x128 vector that shows the characteristics of the words and the relationships between the words in the data set, increasing the ability to understand the text of the model.

Step 4: Use the GlobalMaxPooling class to reduce the size of the data to 128.

Step 5: The complete connection layer ultimately metabolizes 128 characteristics of the model's classification value, using the softmax activation function to calculate the probability of detection of an attack or normal.

3.4.3. Test data set

Test data used as described in section 3.2. (in the content of the thesis).

The data set is randomly divided into three parts: Training set, Validation set and Test set as follows:

- 60% training is used for model input and model parameters refinement;

- A 20% authentication set is used to test the accuracy of the model during the model training to adjust the model parameters to avoid overmatches during the training;

- 20% of the tests are used to evaluate the model after the model has been trained.

3.4.4. Tests and results

The selection test proposed detection model is based on the Bi-LSTM algorithm and the detection models proposed by [38] (Naive Bayes, Decision Tree) and [44] (Random Forest) only uses text data extracted from the website for comparison, evaluation.

Table 3.8. Test results of detection models based on machine learning algorithms using only text characteristics

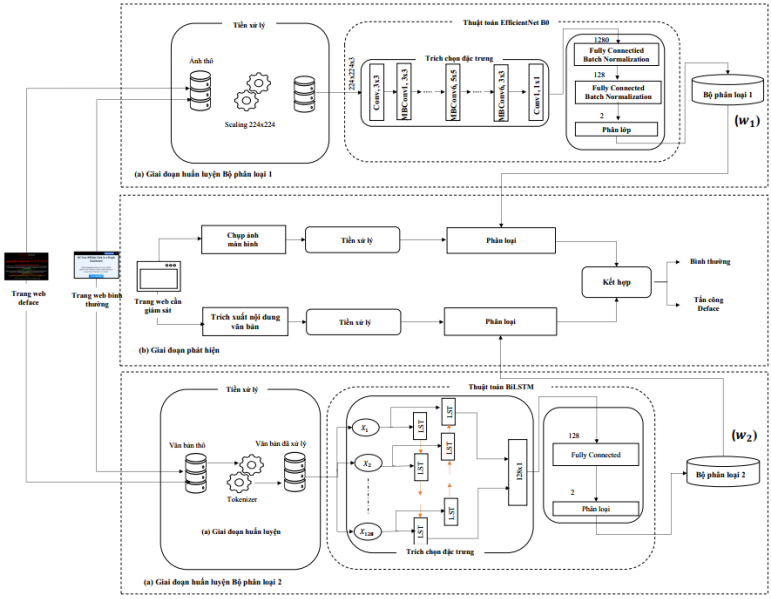
Detection Models	**Đặc trưng**	**ACC(%)**	**PPV(%)**	**TPR(%)**	**F1(%)**	**FPR(%)**	**FNR(%)**
Naïve Bayes Hoang [38]	Text	82,54	78,12	79,26	78,69	15,21	20,74
Decision Tree Hoang [38]	Text	87,33	84,4	84,4	84,4	10,67	15,6
Random Forest Hoang Hoang [44]	Text	93,88	93,81	90,76	92,26	4,03	9,24
BiLSTM	Text	96,54	96,93	94,43	95,66	2,03	5,57

Test results for on Table 3.8. The highest ACC accuracy and F1 measurement of the BiLSTM-based detection model can be seen compared to the models proposed by [38] (Naive Bayes, Decision Tree) and [44] (Random Forest).

3.5. Detect changes to the interface using a combination of text content and screenshots

3.5.1. Detection model description

The proposed attack detection model changes the site's interface using a combination of visual and text characteristics including: (a) the training phase and (b) the detection phase. The training phase was carried out at the branch models in section 3.3. Detecting interface changes using a website screenshot with the resulting classifier 1 and item 3.4. Attack detection changes the interface using text content with the result being Classifiers 2. Both classifiers are used for the detection phase as described in Figure 3.20. During the detection phase, the site needs to be monitored separately, with screenshots and text data, in turn performed with the respective branch models, the final results of each branch after being classified will be combined using a methodology that weighs the results from the branch models from which there is a final classifier that detects the normal state or the change in the appearance of the site under attack. close.



Hình 3. 20. Mô hình phát hiện tấn công thay đổi giao diện kết hợp đặc trưng văn bản và hình ảnh trang web

3.5.2. Data Preprocessing, Training and Detection

Preprocessing datasets are trained to build component detection models. Simple text data trained using BiLSTM deep learning algorithm and screenshot data trailed using EfficientNet deep learning.

3.5.3. Test data set

Test data used as described in section 3.2.(in the content of the thesis). The test data set is also randomly divided into three sub-sets: 60% for the training set, 20% for the validation set, and 20% for testing. (Testing Set).

3.5.4. Tests and results

Table 3.11. Experimental results of matching models

Deep and Combined Engineering	**Đặc trưng**	**ACC(%)**	**PPV(%)**	**TPR(%)**	**F1(%)**	**FPR(%)**	**FNR(%)**
Naïve Bayes Hoang[38]	Văn bản	82,54	78,12	79,26	78,69	15,21	20,74

Deep and Combined Engineering	Đặc trưng	ACC(%)	PPV(%)	TPR(%)	F1(%)	FPR(%)	FNR(%)
Decision Tree Hoang [38]	Văn bản	87,33	84,4	84,4	84,40	10,67	15,6
Random ForestHoang [44]	Văn bản	93,88	93,81	90,76	92,26	4,03	9,24
SVM Siyan Wu[106]	Văn bản	95,34	95,37	95,34	95,32		
EfficientNet(B0)	Ảnh	94,12	94,60	90,71	92,62	3,55	9,29
BiLSTM	Văn bản	96,54	96,93	94,43	95,66	2,03	5,57
BiLSTM+ EfficientNet (Kết hợp)	Văn bản và Ảnh	98,12	98,83	96,49	97,65	0,78	3,51

Results on Table 3.11. shows that the combination of two branches of the model with the BiLSTM and EfficientNet(B0) algorithms yields better results than by the independent branches; and also better models have Naïve Bayes[38], Decision Tree[38], Random Forest[44], SVM Siyan Wu [106].

CONCLUSION

This thesis focuses on two issues: (1) research, proposing a monitored machine learning-based web attack detection model using web log data, aimed at increasing accurate detection rates and reducing false warning rates, and a model that is capable of detecting four types of hazardous web attacks including SQLi, XSS, CMDi and path browsing; and (2) research, suggesting characteristics and options for using in-depth learning methods in line with the specific characteristics of building a web interface change attack detecting model, with the aim of building detection models that enable effective detection of site interface change attacks. The problem (1) is solved by the first contribution of the thesis, and the problem (2) is resolved by a second contribution to the dissertation.

THE CONDITIONS OF THE TRIBUNAL

The first contribution of the thesis was to propose a model for detecting forms of machine learning-based web attacks using character character characteristics in URI query data extracted from web logs. Monitored machine learning algorithms used include Random Forest, Decision Tree, Navie Bayes, and SVM. The proposed model is capable of effectively detecting the four most commonly dangerous types of web attacks, including SQLi, XSS, CMDi and path browsing. Tests on labelled sample data sets and real web log data sets confirmed the suggested model based on Random Forest algorithms for better performance than deep learning-based detection models[58][76]. In addition to high detection performance, the proposed model has a number of advantages compared to previous proposals: (i) the suggested model is constructed using traditional monitored machine learning algorithms with low computational costs but still achieving high results, which is important for practical deployment because the web attack detection system often has to handle a huge amount of web logs, and (ii) the recommended model can be constructed automatically from training data and does not require regular updates..

The second contribution of the thesis was to propose three deep learning-based attack detection models that use the site's screenshot characteristics, the text characteristics extracted from the site, and the combination of the text features extracted by the site in conjunction with the website's screen characteristics. The test results showed that the EfficientNet-based branch detection model, the BiLSTM-based and the combined model both yielded higher detection performance than the models proposed by previous studies and models based on other deep mathematics algorithms. In particular, the detection model

based on the combination of two visual and text characteristics of the site has superior detection performance compared to the results suggested by Hoang and colleagues[44] and Hoang[38], as well as models based on deep learning algorithms Xception, Inception and Bi-LSTM and EfficientNet only on the site's screenshot characteristics.

The problems of the existence of the proposals in the thesis are also open directions for further research, supplementation. Specifically:

- The first problem is that URI queries can only be extracted from web logs if the HTTP method used is GET. If the method used is POST, the data sent from client to server is not stored in the log. One way to solve this problem is to deploy a detection model in the form of a web application firewall (WAF) to capture and process all user access requests.

- The BiLSTM and EfficientNet-based interface-change attack detection model requires high computational costs for training due to the processing of large amounts of screenshots and website text content using in-depth learning algorithms. One way to solve this problem is to train the model that can be done offline so it doesn't have much to do with the detection time. In addition, the model could combine the use of signature-based detection with known forms of attack to reduce detection time.