International Information and Engineering Technology Association
Advancing the World of Information and Engineering

Traitement du Signal
Vol. 37, No. 1, February, 2020, pp. 113-118
Journal homepage: http://iieta.org/journals/ts

A Malicious Webpage Detection Algorithm Based on Image Semantics

Xiangjun Li[1,2], Sifan Li[1*], Shengnan Liu[1], Lingfeng Liu[1], Daojing He[1,3]

[1] School of Software, Nanchang University, Nanchang 330047, China
[2] Department of Computer Science and Technology, Nanchang University, Nanchang 330031, China
[3] School of Software Engineering, East China Normal University, Shanghai 200241, China

Corresponding Author Email: 8003117137@email.ncu.edu.cn

https://doi.org/10.18280/ts.370115

Received: 15 July 2019
Accepted: 20 November 2019

Keywords:
deep learning, malicious attack, image semantics, backpropagation neural network (BPNN)

ABSTRACT

In the era of the Internet, malicious attacks have put user information at risk. Many malicious webpages use images as the carrier of malicious codes. If extracted accurately, the features of these images will help to improve the detection of malicious webpages. This paper aims to develop an accurate malicious webpage detection method based on the features of the said images. Since static images contain a small amount of information, semantic segmentation was performed to predict the semantics of the target attitude. Then, the final semantics of target the image were derived by the backpropagation neural network (BPNN). After that, the image semantics were fused with the other features of the malicious webpage, and sent to the classifier for recognition. Finally, the proposed algorithm was tested on an actual dataset, in comparison with other malicious webpage detection methods. The results show that our algorithm can accurately detect malicious webpages, thanks to the introduction of image semantic features.

1. INTRODUCTION

With the proliferation of the Internet, malicious attacks have put user information at risk. For example, Internet users might suffer from information leak after entering a phishing webpage. The leaked information could be highly sensitive, such as passwords and credit card number.

The malicious code of the phishing webpage cannot replicate on its own. To spread the code across the Internet, malicious attacks are often launched in three ways: sending a junk email with clickbait tile and keywords; creating a fake e-payment scenario; displaying texts and images embedded with links to malicious webpage. Once a user opens the email, completes the payment or clicks on the texts/images, his/her private information will be stolen and his/her computer will be infected with Trojan viruses.

To keep user information safe, it is necessary to identify the malicious webpages and prevent users from clicking on them. Considering their sheer number, the malicious webpages should be detected with the aid of machine intelligence. During the detection, special attention should be paid to the texts on each webpage, which are the main content on traditional webpages and the focal point of user-webpage interaction. The images, an emerging type of information carrier, should also be considered in the detection of malicious webpages.

Many malicious webpages have similar structures and visual features as the target webpages of Internet users. Hence, a possible way to identify malicious webpages is to evaluate the visual similarity between webpages. This calls for effective extraction of image features from each webpage.

This paper attempts to develop an accurate method for malicious webpage detection based on the image features on such webpages. Firstly, Mask region-convolutional neural network (Mask R-CNN) was improved to extract image features. Then, the complex semantics of the target image were predicted, using Kinect-based action matching and backpropagation neural network (BPNN). After that, the features of the target webpage were synthetized, and sent to the classifier for recognition. The accuracy of our algorithm was verified through contrastive experiments.

2. LITERATURE REVIEW

Traditionally, malicious webpages are identified by comparing each webpage against black and white list (BWL), statistical weighting and similarity judgment. For BWL comparison, the features of uniform resource locator (URL) are obtained through machine learning, and used to build a BWL database for contrastive analysis. In statistical weighting, the phishing webpages are detected by computing the term frequency–inverse document frequency (TF-IDF) of the keywords in the texts, because most malicious websites are about gambling and pornography. In similarity judgement, the phishing webpages are evaluated against normal webpages in page structure and logo image [1].

Based on machine learning, heuristic engines can recognize unknown webpages by training the page features. Using feedback supper vector machine (SVM), Barraclough et al. [2] classified and identified phishing websites through incremental sample test. Hans et al. [3] analyzed the main features of phishing websites, and relied on random forest and reinforcement learning to enhance the recognition rate of classifiers. In webpage recognition, the performance of heuristic engines depends on the selected features. Since manually extracted features are often subjective, the deep learning has been introduced to extract the features for malicious webpage detection. For instance, Sur [4] created a smart and accurate phishing webpage classifier based on the deep belief network (DBN). Rao and Pais [5] combined cost function with the feedback network to reduce the ratio of false alarms in malicious webpage detection.

The traditional detection methods for malicious webpages mainly focus on the texts. But these methods cannot adapt to the growing presence of images and videos on the Internet. The deep learning, a computer vision technique, offers a viable solution to the problem. The image/video features on webpages can be effectively extracted through deep learning, laying a good basis for malicious webpage detection. Dérian et al. [6] collected robust features like optical flow and gradient by convolutional neural network (CNN), and greatly improved the recognition accuracy of images on webpages. Ramya et al. [7] trained image features with Fourier transform and the SVM; the training reduces the complexity and enhances the accuracy of malicious webpage identification.

Search-based attitude recognition offers a top-down search strategy to capture the features needed for detecting malicious webpages [8]. Based on reinforcement learning, Ognibene et al. [9] developed a target search strategy in which each designed action is predicted through reinforcement learning, and the target is searched for according to the actions. Gosavi [10] proposed a reinforcement learning algorithm, which searches the target with only six types of actions and then represents the image layer by layer, thus reducing the search scope; the Q-learning was also adopted to narrow down the search scope for attitude detection. Zhao et al. [11] predicted the trend of visual attention through deep learning, and then identified the attitude of each attention target according to the predicted trend.

3. IMAGE FEATURE EXTRACTION BASED ON DEEP LEARNING

As mentioned before, most malicious websites are about gambling and pornography. These websites contain an increasing number of images. The image features should be extracted effectively before identifying the malicious webpages.

This paper improves the Mask R-CNN [12] to extract image features on webpages. The Mask R-CNN (Figure 1) is a combination of two classical target recognition algorithms: Fast R-CNN and fully convolutional network (FCN). To improve the accuracy, the mask is generated by adding the FCN to the end of the Faster R-CNN.

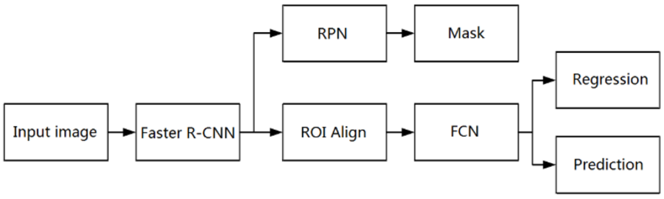


Figure 1. The structure of Mask R-CNN

The Fast R-CNN (Figure 2) maps each candidate region to the feature layer of the CNN, and directly extracts the deep features from the region of interests (ROIs) on the feature layer, eliminating the need for constant input different regions of the image. Then, the extracted features were used to predict the ROI category on SoftMax, and create a bounding box regressor. The ROI-based extraction applies to images on various scales. Through end-to-end learning, the Fast R-CNN effectively improves the efficiency of the R-CNN.

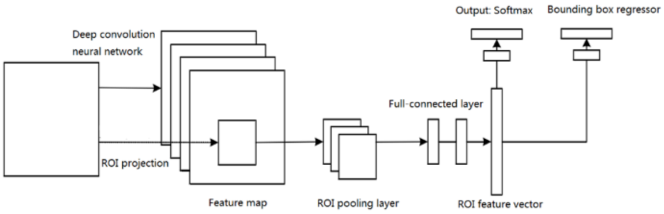


Figure 2. The structure of Fast R-CNN

The Fast R-CNN has two parallel fully-connected output layers: the Softmax calculates the probability distribution of a single eigenvector in the *k*+1 category, while the bounding box regressor calculates the parameters of the bounding box. The two output layers are trained by a joint loss function:

$$F(p,c,q^c,v) = F_{cls}(p,c) + \delta[c \geq 1]_g F_{loc}(q^c,\mathrm{v}) \quad (1)$$

where, $p = (p_0, \dots, p_k)$; $q^c = (q_x^k, q_y^k, q_w^k, q_h^k)$; $v = v_x, v_y, v_w, v_h$; k is the number of categories; $F_{cls}(p,c) = -\log{(p_c)}$ is the logarithmic cost of real category c; $F_{loc}()$ is the loss due to regression.

The actual boundary v and predicted bounding box q^c of category c can be calculated according to the definitions of the following parameters:

For $q^c = (q_x^k, q_y^k, q_w^k, q_h^k)$, each parameter can be defined as:

$$\begin{cases} q_x = \frac{(O_x - C_x)}{C_w} \\ q_y = \frac{(O_y - C_y)}{C_h} \\ q_w = \log\left(\frac{O_w}{C_w}\right) \\ q_h = \log\left(\frac{O_h}{C_h}\right) \end{cases},$$

where, (O_x, O_y, O_w, O_h) are the center coordinates, border width and border height of real target, respectively; (C_x, C_y, C_w, C_h) are the center coordinates, border width and border height of candidate area, respectively.

For the bounding regression layer, the loss can be defined as:

$$F_{loc}(q^c,\mathrm{v}) = \sum_{j \in (x,y,h,w)} Smooth(q_j^c + v_j) \quad (2)$$

where, $Smooth(y) = \begin{cases} 0.5y^2, & |y| < 1 \\ |y| - 0.5, & otherwise \end{cases}$.

After adding the mask branch, the loss function of each ROI can be computed by:

$$F = F_{cls} + F_{loc} + F_{mask} \quad (3)$$

For each ROI, the mask branch has an output of $Km * m$ dimensions, which includes K masks of $m * m$ size; each mask involves K categories.

The steps of MASK R-CNN algorithm are as follows:

Step 1. Input and preprocess the target image;

Step 2. Import the preprocessed image into a neural network for pre-training, yielding a feature map;

Step 3. Preset a ROI for each point in the feature map, producing multiple candidate ROIs;

Step 4. Send the candidate ROIs to the region proposal network (RPN) for binary classification to filter out some candidate ROIs;

Step 5. Perform the ROIAlign on the remaining ROIs;

Step 6. Classify and mask the ROIs.

4. MALICIOUS ATTACK DETECTION ALGORITHM BASED ON IMAGE SEMANTICS

Despite its excellence in detecting image objects, the Mask R-CNN is not good at recognizing image attitude. As shown in Figure 3, our malicious attack detection algorithm is implemented by segmenting the semantics of webpage images, extracting the images containing human actions, recognizing the attitude in each image, judging the sensitive features, and integrating all features to identify malicious webpage.

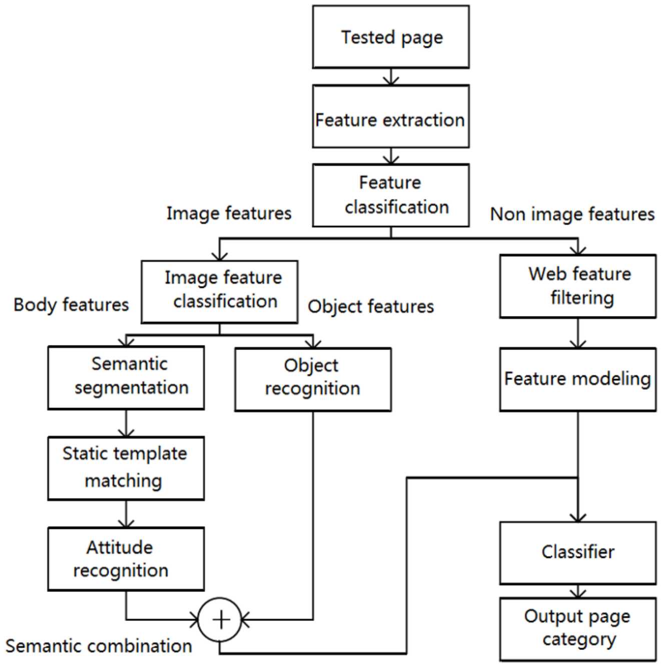


Figure 3. The workflow of our algorithm

4.1 Kinect-based action matching

To ensure the accuracy of deep learning in attitude recognition, the target image was subjected to semantic segmentation, and the human body was extracted from the image. Then, the basic attitude of the human body was recognized based on the Kinect attitude database. After that, the semantics were combined based on environmental and human body features. Finally, the complex semantics of the image were predicted.

Because most images on malicious webpages are static, the malicious webpage detection method was improved through shape learning of training samples. To eliminate the influence of background noises and non-rigid deformation of human body, the semantically segmented image was sent to Kinect recognizer [13] to extract the sub-image that contains the attitude information. Then, the points of the human joints in the sub-image were matched with the Kinect attitude database, and the description eigenvectors of the corresponding attitudes were obtained.

The description thus obtained only expresses the meaning of the attitude. Except for some explicit pornographic actions, most of the basic actions are neutral semantics, which should be judged with the aid of the semantic information of other instances of semantic segmentation. For example, the recognized action "taking" can be combined with the instances "chips", "dice" and "poker" into the semantic "gambling behavior". Then, the target image can be determined as containing sensitive information.

Let I be the attitude image of the corresponding area of mask in the input image, $\{S^{(I)}, M,\ S^{(N)}\}$ be a set of N training samples in Kinect attitude database, $\{a^{(I)}, M,\ a^{(N)}\}$ be the semantic of the corresponding action, where $a^{(I)}$ is a word for action.

First, the distance between each sample in I and K was calculated as $D_t = D(I, K^t)$. Then, the action semantic $a^{(min)}$ corresponding to the sample $K^{(min)}$ with the smallest value was selected as the semantic description of I in D_t.

Let $(x^{(i)}, y^{(i)})$ and $(x_t^{(i)}, y_t^{(i)})$ be the coordinates of the i-th joint point on I and K, out of the M joint points. Then, the steps of Kinect-based action matching can be expressed as:

Step 1. Calculate the distance $d_i^{(t)}$ between the i-th joint point on I and K:

$$d_i^{(t)} = \sqrt{(x^{(i)} - x_t^{(i)})^2 + (y^{(i)} - y_t^{(i)})^2};$$

Step 2. Compute the total distance of all joint points on I and K:

$$D((I, K^t) = \sum_{i=1}^{M} d_i^{(t)};$$

Step 3. Calculate the Kinect sample index with minimum distance:

$$\min = \arg\min_k D((I, K^t);$$

Step 4. Take a^{min} as the semantic description of I.

4.2 BPNN-based semantic inference

Most of the regions detected from a static image are basic neutral actions. Therefore, a BPNN was constructed based on feedback learning. For an image containing lots of suspected sensitive objects, the greater the degree of exposure of the human body, the larger the number of sensitive behaviors, and the more likely the image is sensitive. The BPNN algorithm can be implemented in the following steps:

Step 1. Initialize the number k of basic actions of human body, the serial number k_i of the i-th attitude, and the number N of sensitive objects.

Step 2. Calculate the length d_{touch} of the shortest edge between the hand joint and the sensitive image:

$$\vartheta_{len} = \begin{cases} 0, & d_{touch} = 0 \\ \left(\frac{2d_{touch}}{d_{width}+d_{height}}\right), & d_{touch} > 0 \end{cases};$$

where, d_{width} and d_{height} are the width and height of the image, respectively. Then, normalize ϑ_{len} to ϑ_{touch}:

$$\vartheta_{touch} = \begin{cases} 0, & \vartheta_{len} = 0 \\ 1, & 1 < \vartheta_{len} \leq 3 \\ 2, & 3 < \vartheta_{len} \leq 7 \\ 3, & \vartheta_{len} > 7 \end{cases};$$

The sensitivity of the action is positively correlated with the closeness between the hand joint and the edge of the sensitive object, and peaks at the contact between the two objects.

Step 3. Compute the number of sensitive objects n_{thing}:

$$\vartheta_{thing} = \begin{cases} 0, & n_{thing} = 0 \\ 1, & 1 < n_{thing} \leq 3 \\ 2, & 3 < n_{thing} \leq 5 \\ 3, & n_{thing} > 5 \end{cases};$$

If many suspected sensitive objects are detected in an image, the image is highly likely to contain sensitive semantics.

Step 4. Calculate the degree of exposure of human body:

$$\vartheta_{skin} = \frac{S_{skin}}{S_{body}},$$

$$\vartheta_{naked} = \begin{cases} 0, & \vartheta_{skin} = 0 \\ 1, & 0 < \vartheta_{skin} \leq 0.25 \\ 2, & 0.25 < \vartheta_{skin} \leq 0.6 \\ 3, & \vartheta_{skin} > 0.6 \end{cases};$$

where, S_{skin} is the degree of exposure; the greater the S_{skin} value, the higher the possibility of sensitive action.

Step 5. Calculate the degree of sensitivity of the image:

$$\vartheta_{image} = \begin{cases} 0, & n_{image} = 0 \\ 1, & 1 < n_{image} \leq 6 \\ 2, & 6 < n_{image} \leq 12 \\ 3, & n_{image} > 12 \end{cases};$$

where, n_{image} is the number of sensitive images in the image set of a webpage. The greater the n_{image} value, the more likely to image is sensitive.

Step 6. Perform Bayesian probability combination of semantic and sensitive objects. Suppose there are N types of sensitive semantics. Let $p(c_j|x)$ be the expected loss if sample x is identified as c.

The classifier can be obtained according to Bayesian probability:

$$h^*(x) = arg \max_{c \in C} P(c|x),$$

$$\vartheta_{action} = \begin{cases} 0, & 0 < P(c|x) \leq 0.3 \\ 1, & 0.3 < P(c|x) \leq 0.6 \\ 2, & 0.6 < P(c|x) \leq 0.8 \\ 3, & P(c|x) > 0.8 \end{cases}.$$

Then, a BPNN (Figure 4) with five input nodes can be established based on $\vartheta_{touch}, \vartheta_{thing}, \vartheta_{naked}, \vartheta_{image}$ and ϑ_{action}, and used to compute the image semantic $a^{(I)}$ of target y, creating a feature vector f_{action} of malicious attitude transform.

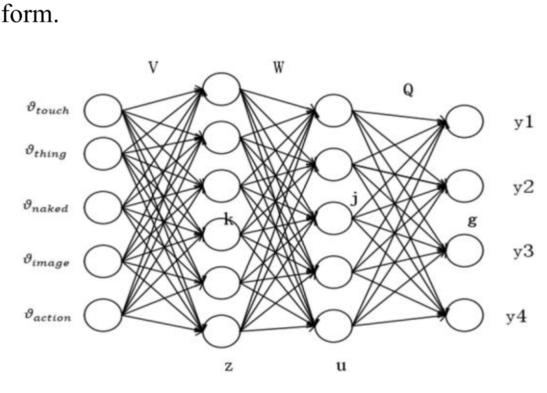


Figure 4. The structure of the BPNN

4.3 Malicious webpage identification

The description of webpage feature function is shown in Table 1.

During malicious webpage detection, the page features, e.g. URL, keywords, page structure and registration information, are usually only described by part of the page information. Thus, the features cannot be generalized to different forms of webpages. These features should be considered comprehensively to improve the recognition accuracy of webpages with more text information.

Table 1. Webpage feature functions

Functions	Description
F_1	The URL contains meaningless repeating letters.
F_2	The number of links in the webpage is greater than P_1.
F_3	The URL contains more unusual punctuations than P_2.
F_4	The URL is longer than P_3 bytes.
F_5	The URL takes the form of IP.
F_6	The URL has a fake domain name.
F_7	The URL contains sensitive words.
F_8	The location of low-level domain name is abnormal.
F_9	The webpage contains sensitive keywords.
F_{10}	The webpage contains a fake certificate number.
F_{11}	The URL contains the character @.
F_{12}	The registration is less than P_4 months.
F_{13}	The webpage ranks below P_5.
F_{14}	The webpage is updated fewer than P_6 each month.
F_{15}	The webpage has a fake document object model (DOM).
F_{16}	The URL contains more paths than P_7.

For the text information contained in the target webpage, the functions $F_1 \sim F_{16}$ were used to calculate each Boolean value in turn, and then a vector f_{text} was formed as the text feature of the webpage.

The semantic description feature of the image was obtained by the BPNN through joint point matching, and taken as the image keyword of the webpage. This feature was combined with the text feature f_{text} into the final feature $f = [f_{text}, f_{action}]$ to train the heuristic learner.

The classification and regression tree (CART) algorithm [14] were adopted to solve the detection of malicious webpage as binary classification problem. Suppose the classification problem contains K= 2 categories. For a given sample set D, let p_k be the probability that the sample points belong to the k-th category. Then, the $Gini$ value of the sample set can be calculated by:

$$Gini(D) = \sum_{k=1}^{K} \sum_{k'=1, k \neq k'}^{K} p_k p_{k'} = 1 - \sum_{k=1}^{K} p_k^2.$$

Next, the attribute with the minimum Gini value or regression variance of the child nodes was taken as the benchmark of node splitting, and the CART was constructed to classify the webpages.

5. EXPERIMENT VERIFICATION AND RESULTS ANALYSIS

To verify its performance, our algorithm was subjected to malicious webpage recognition experiments, in comparison with the FCN and Mask R-CNN. The recognition performance was mainly measured by the prediction accuracy of the classifier for malicious webpages. The experimental parameters were configured as: P_1 =40, P_2 =5, P_3=80, P_4=3, P_5=1,000, P_6=10 and P_7=5.

The COCO semantic dataset [15], which contains 91 common object categories, was selected as the training set of semantic segmentation. A total of 76,856 webpages published from January to December 2019 were randomly selected from the webpage security database PhishTank [16], and divided into a training set, a verification set and a test set at the ratio of 2:1:1.

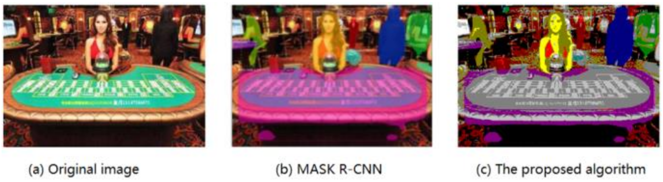
Three independent experiments were carried out to evaluate the semantic segmentation, attitude recognition and webpage recognition of the proposed algorithm and the two contrastive algorithms on different datasets.

Table 2 lists the results of the three methods for semantic segmentation on COCO dataset. It can be seen that the proposed algorithm achieved comparable segmentation accuracy to that of Mask R-CNN with 19.2% less running time. This means our algorithm can reduce the computing load without sacrificing the accuracy.

Table 2. Comparison of image semantic performance

Algorithm	Average precision (AP) (%)	Precision (%)	Running time (s)
FCN	83.6	82.7	4.43
Mask R-CNN	90.1	88.6	6.54
Proposed algorithm	90.2	88.5	5.29

Figure 5 provides an example of the recognition effects between the proposed algorithm and the Mask R-CNN.


(a) Original image

(b) MASK R-CNN

(c) The proposed algorithm

Figure 5. An example of the recognition effects of different algorithms

Table 3 compares the results of the three methods for human attitude recognition.

Table 3. Comparison of attitude recognition performance

	FCN	Mask R-CNN	Proposed algorithm
Attention	72.52%	87.2%	91.05%
Bow	68.83%	73.72%	79.91%
Raise hands	65.27%	72.96%	80.64%
Walk	67.83%	77.25%	82.76%
Handshake	64.89%	69.27%	75.92%
Take	65.73%	80.15%	85.82%
Lying	69.04%	79,51%	86.29%
Side lying	68.33%	78.71%	85.49%

As shown in Table 3, the proposed algorithm was much more accurate than the Mask R-CNN in the recognition of human attitude.

Finally, the webpage data of PhishTank were divided into three subsets, according to the proportion of texts to images on the page: pages with more text (dataset A), pages with more images (dataset B), and test pages (dataset C). The three subsets add up to the whole dataset DS_{total} . Then, the proposed algorithm was compared with two malicious webpage detection algorithms on the dataset [17, 18].

It can be seen from Table 4 that the proposed algorithm was more accurate than the two contrastive algorithms, especially for pages containing lots of texts or images. This is because the contrastive algorithms rely on the DBN for webpage recognition. The DBN can realize self-supervised training, but perform poorly in discriminating pages containing images.

Overall, the proposed algorithm achieved ideal performance in semantic segmentation, attitude recognition and webpage classification. The high accuracy of malicious webpage detection is attributable to the synthetic use of semantic features and webpage features.

Table 4. Comparison of malicious webpage detection performance

Algorithms	Dataset A	Dataset B	Dataset C	DS_{total}
Algorithm 1 [18]	81.8%	73.5%	85.8%	79.7%
Algorithm 2 [19]	83.8%	71.7%	83.9%	78.8%
The proposed algorithm	85.7%	82.6%	87.6%	85.6%

6. CONCLUSIONS

Malicious webpages pose a serious threat to the information security of Internet users. However, the traditional methods for malicious webpage detection have not made full use of image information. This paper improves the Mask R-CNN to perform semantic segmentation of images, and derives attitude and other contextual semantics. The features of the target webpage were synthetized, and sent to the classifier for recognition. Each part of our algorithm was evaluated separated through experiments. The experimental results fully demonstrate the effectiveness of our algorithm.

ACKNOWLEDGEMENT

This work was funded by the National Natural Science Foundation of China (Nos. 61862042, 61762062, 61601215, U1936120, U1636216); the National Key R&D Program of China (Nos. 2017YFB0802805, 2017YFB0801701); the Science and Technology Innovation Platform Project of Jiangxi Province (No. 20181BCD40005); the Fok Ying Tung Education Foundation of China (No. 171058); the Major Discipline Academic and Technical Leader Training Plan Project of Jiangxi Province (No. 20172BCB22030); the Primary Research & Development Plan of Jiangxi Province (Nos. 20192BBE50075, 20181ACE50033); the Jiangxi Province Natural Science Foundation of China (Nos. 20192BAB207019, 20192BAB207020); the Graduate Innovation Fund Project of Jiangxi Province (Nos. YC2019-S100, YC2019-S048); the Practice innovation training program of Jiangxi Province for college students (No. 20190403041).

REFERENCES

[1] Shreeram, V., Suban, M., Shanthi, P., Manjula, K. (2010). Anti-phishing detection of phishing attacks using genetic algorithm. International Conference on Communication Control and Computing Technologies, Ramanathapuram, pp. 447-450. https://doi.org/10.1109/ICCCCT.2010.5670593

[2] Barraclough, P.A., Hossain, M.A., Tahir, M.A., Sexton, G., Aslam, N. (2013). Intelligent phishing detection and protection scheme for online transactions. Expert Systems with Applications, 40(11): 4697-4706. https://doi.org/10.1016/j.eswa.2013.02.009

[3] Hans, K., Ahuja, L., Muttoo, S.K. (2014). Approaches for web spam detection. International Journal of Computer Applications, 101(1): 38-44. http://dx.doi.org/10.5120/17655-8467

[4] Sur, C. (2019). DeepSeq: learning browsing log data based personalized security vulnerabilities and counter intelligent measures. Journal of Ambient Intelligence and Humanized Computing, 10(9): 3573-3602. https://doi.org/10.1007/s12652-018-1084-9

[5] Rao, R.S., Pais, A.R. (2019). Detection of phishing websites using an efficient feature-based machine learning framework. Neural Computing and Applications, 31(8): 3851-3873. https://doi.org/10.1007/s00521-017-3305-0

[6] Dérian, P., Héas, P., Herzet, C., Mémin, E. (2013). Wavelets and optical flow motion estimation. Numerical Mathematics: Theory, Methods and Applications, 6(1): 116-137. https://doi.org/10.1017/S1004897900001161

[7] Ramya, M., Krishnaveni, V., Sridharan, K.S. (2017). Certain investigation on iris image recognition using hybrid approach of Fourier transform and Bernstein polynomials. Pattern Recognition Letters, 94: 154-162. https://doi.org/10.1016/j.patrec.2017.04.009

[8] Li, C., Zhang, H.Y., Yang, X.P., Wang, F., Chen, Z.P. (2010). Dual range-based space target multi-attitude angles feature fusion recognition algorithm. Control and Decision, 25(9):1374-1378.

[9] Ognibene, D., Balkenius, C., Baldassarre, G. (2008). A reinforcement-learning model of top-down attention based on a potential-action map. The Challenge of Anticipation, 161-184. https://doi.org/10.1007/978-3-540-87702-8_8

[10] Gosavi, A. (2004). A reinforcement learning algorithm based on policy iteration for average reward: Empirical results with yield management and convergence analysis. Machine Learning, 55(1): 5-29. https://doi.org/10.1023/B:MACH.0000019802.64038.6c

[11] Zhao, D., Chen, Y., Lv, L. (2016). Deep reinforcement learning with visual attention for vehicle classification. IEEE Transactions on Cognitive and Developmental Systems, 9(4): 356-367. https://doi.org/10.1109/TCDS.2016.2614675

[12] Chiao, J.Y., Chen, K.Y., Liao, K.Y.K., Hsieh, P.H., Zhang, G., Huang, T.C. (2019). Detection and classification the breast tumors using mask R-CNN on sonograms. Medicine, 98(19): e15200. http://dx.doi.org/10.1097/MD.0000000000015200

[13] Zhang, Z. (2012). Microsoft Kinect sensor and its effect. IEEE Multimedia, 19(2): 4-10. https://doi.org/10.1109/MMUL.2012.24

[14] Gutiérrez-Esparza, J.C., Gómez-Hernández, J.J. (2017). Inverse Modeling Aided by the Classification and Regression Tree (CART) Algorithm. Geostatistics Valencia 2016, 805-819. https://doi.org/10.1007/978-3-319-46819-8_55

[15] Anderson, P., Fernando, B., Johnson, M., Gould, S. (2016). Spice: Semantic propositional image caption evaluation. European Conference on Computer Vision, 382-398. https://doi.org/10.1007/978-3-319-46454-1_24

[16] Geng, G.G., Lee, X.D., Zhang, Y.M. (2015). Combating phishing attacks via brand identity and authorization features. Security and Communication Networks, 8(6): 888-898. https://doi.org/10.1002/sec.1045

[17] Lin, C.F., Lin, S.F. (2013). Efficient face detection method with eye region judgment. EURASIP Journal on Image and Video Processing, 2013(1): 34. https://doi.org/10.1186/1687-5281-2013-34

[18] Iyengar, S., Hahn, K.S. (2009). Red media, blue media: Evidence of ideological selectivity in media use. Journal of Communication, 59(1): 19-39. http://dx.doi.org/10.1111/j.1460-2466.2008.01402.x