Massive Malware Variants Detection Based on Bag-of-words Perceptual Hashing

Jian Yu
College of Computer and Information Engineering Hanshan Normal University
Chaozhou, Guangdong, China
yj@hstc.edu.cn

Abstract. Presently, most widely used malware detection methods use signature with reverse engineering to recognize malware variants. Nevertheless, this approach is problematic because the signatures simply modified by using packers on which compress and/or encrypt the executable code to evade detection. In this paper, we present a novel Bag-of-words perceptual hashing to detect variants. The proposed method visualizes malware binary code as grey-scale image, extracts the Grey-level Co-occurrence Matrix features vector and using Bag-of-words model to generate perceptual hashing for malware variants detection. Experimental results show that, the proposed method has a high accuracy and fast detection speed, and has good resilience to popular packers, which is suitable for massive malware variants detection.

1. Introduction
Malicious code or software is called malware, which refers to viruses, worms, Trojan horses, logic bombs, spyware etc. Malwares proliferate rapidly in recent years [1]. The main reason for this fast increase is that the malware variants produced easily by using packers on which compress and/or encrypt the malware executable code.

Most traditional approaches towards malware detection usually utilize static analysis. Static analysis works by disassembling the malware executable code to look for malicious patterns that called signature. This signature-based detection method needs to unpack and/or decrypt the code before static analysis, which is problematic. This approach is also hard to detect the variants of known malware or previously unknown malware.

With the rapid development of computer vision technology, some image processing techniques and pattern recognition methods proposed to malware detection. Some scholars begin to try to visualize the malicious code, and hope to find a more effective malware detection method with image processing and pattern recognition approaches.

In [2], Conti et al. used byte-plot visualization method to view binary objects and created an interactive system for exploration, and SOM neural network used to visualize and detect malicious executable code in [3]. In [4,5], L. Nataraj et al. used byte-plot method and extracted 320 dimensional GIST texture features vector of the malware image and K Nearest Neighbour (KNN) for malware classification and gained accurate results. Kancherla K et al. propose the use of lightweight features like intensity-based features and wavelet-based features instead of computationally expensive Gabor based features in [6].

However, the classification method based on high dimensional image features vector is not suitable for massive malware variant detection, for it is time-consuming in searching a malware sample in a
massive malware database. In this paper, a Bag-of-words perceptual hashing method proposed to massive malware detection, which projects a high-dimensional image feature space to a low-dimensional hashing space, and retrieves malware samples from malware database using hamming distance. Compared to computationally expensive image features, the proposed method obtained much better efficiency and similar accuracy.

2. Malware image analysis

The malware executable code can be converted into byte-plot image which treated malware binary as a vector of 8 bit unsigned integers and organized it into a 2D array which can be visualized as a grey scale image in the range [0,255].

Using the byte-plot method, some images of malware families (malware and its variants) listed as follows. In each family, only three malware variants exhibited.

Some samples of malware variants images are shown as follows:

![Fig. 1 Samples of malware belonging to the Allaple.A family](image1)

![Fig. 2 Samples of malware belonging to the Fakerean family](image2)

![Fig. 3 Samples of malware belonging to the Swizzor.gen!I family](image3)
As shown in Fig.1 to Fig.4, Allaple.A, Fakerean and Swizzor.gen! families are malware variants based on windows platform, and FakeAV family is malware variants based on Android os.

The above four malware image sets show that, in the same family of malware, malware image is quite similar, whereas the malware image is quite different between different malware families. Therefore, perceptual hashing method generated from the texture features of the malware image utilized to characterize and classify malware.

### 3. Bag-of-words Perceptual Hashing

We used GLCM to extract the texture feature vector of malware grey scale image, then treated the texture feature vector as Bag-of-words and combined visual vocabulary to generate Bag-of-words perceptual hashing.

#### 3.1. Texture Features Vector Extraction

In this paper, the texture features of the malware image are extracted by GLCM which is constructed with matrix of joint probability density between 16 grey levels of malware image. GLCM can represent spatial relationship of any two points in the image.

Let \( [P(i, j, d, \theta)]_{L \times L} \) represent the value (joint probability) of \( i \)-th line and \( j \)-th column in grey level co-occurrence matrix where \( d \) is distance, \( \theta \) is the direction and \( L \) is the number of grey level, i.e. it denotes that \( P(i, j, d, \theta) \) is the probability where the grey level \( i \) is the origination and the grey level \( j \) appearing as the destination. Let \( d=1, \theta=0^\circ, 45^\circ, 90^\circ \) and \( 135^\circ \). In actual application, some statics defined as feature value of texture analysis based on grey level co-occurrence matrix. Haralick et al. [7] proposed a texture features method with 14 statics from GLCM. In this study, we chose eight statics from the 14 statics as malware image texture features throughout our experiments. The eight statics are the following: Mean, Variance, Angular second moment, Entropy, Inverse difference moment, Homogeneity, Contrast and Correlation.

#### 3.2. Bag-of-words Perceptual Hashing

With the extraction of the above eight statics of GLCM from four different directions, then 32 dimensional texture features vector gained, which took as Bag-of-word named V. For the generated malicious code feature vectors, we used K-means algorithm to determine the number of cluster centres in the texture feature space of malware image, and then calculate the final N clustering centres by iterative algorithm.
Set a visual dictionary $D(a,b,c,d,e,...)$ with $N$ dimensional visual vocabulary. The feature vector $V$ mapped to one of the $N$ clustering centres. Each cluster centre randomly selects the words with the number of feature vectors $V$ dimension in the dictionary $D$ as the dictionary, for example, the 1 corresponding bag of the clustering points is $D_1(a, r, h, b,...)$. The clustering point 2 corresponding to the word bag is $D_2(t, b, p, e,...)$. In this way, each cluster centre corresponds to a word bag, and the feature vector $V$ maps to which cluster centre to get the corresponding word bag.

The steps of the Bag-of-word perceptual hashing are following Table 1.

### Table 1 Steps of the Bag-of-word Perceptual Hashing

| Bag-of-word perceptual hashing Algorithm |
|-----------------------------------------|
| **Input**                               |
| a visual dictionary $D$ (N dimensional visual vocabulary) |
| feature vector $V$ (L dimensional texture feature vector) |
| **Output**                               |
| Bag-of-word perceptual hashing $S$ (M bits) |
| **Steps**                               |
| For each visual vocabulary of the visual dictionary $D$, M dimension hash is produced by the traditional Hash algorithm (such as MD5). |
| for $i = 1$ to $N$                       |
| for $j = 1$ to $M$                      |
| if $b[i][j] = 1$ then                   |
| $V[j] = V[j] + L[i]$;                  |
| else $V[j] = V[j] - L[i]$;             |
| end for                                 |
| end for                                 |
| for $j = 1$ to $M$                      |
| if $V[j] > 0$ then $S[j] = 1$;          |
| else $S[j] = 0$;                        |
| end for                                 |
| return $S$                             |

### 4. Experimental Results and analysis

The experimental environment is Lenovo ThinkPad notebook computer L460-56, Intel i5-6200U 2.3G CPU, 8G RAM, Windows 7 64-bit operation system, using Anaconda Python 2.7. A large malware dataset named Malimg provided by [4,5] which is utilized in this experiment. The dataset consists of 25 malware families, consisting of 9,339 malwares in total.

Each family name, quantity and the malware variants detection accuracy on the Malimg dataset are shown in Table 2. An extremely randomized trees (ET) method is used as a machine learning classifier for the 25 malware families with 10 fold-crossover verification.

### Table 2 Accuracy on Malimg Dataset

| NO. | Class  | Family Name | Quantity | Accuracy |
|-----|--------|-------------|----------|----------|
| 1   | Dialer | Adialer.C   | 122      | 98%      |
| 2   | Backdoor | Agent.FYI  | 116      | 100%     |
| 3   | Worm   | Allaple.A   | 2949     | 99%      |
| 4   | Worm   | Allaple.L   | 1591     | 97%      |
| 5   | Trojan | Alueron.gen!J | 198   | 100%     |
| 6   | Worm:AutoIT | Autorun.K  | 106      | 100%     |
| 7   | Trojan | C2LOP.gen!g | 200      | 94%      |
Although Yuner.A, VB.AT, Malex.gen!J and Autorun.K, Rbot!gen families were packed with UPX packer tool, they are correctly classified by our proposed method. The confusion matrix of the result of the classification is shown in Figure 5.

As shown in Table 2 and Figure 5, the proposed method obtained more than 97.8% accuracy in most malware families. The classification performance compared to [5] is in Table 3.
Table 3 Performance of Comparison

| Method            | Feature               | Average Accuracy | Speed(second) |
|-------------------|-----------------------|------------------|---------------|
| Literature [5]    | GIST                  | 97.4%            | 1.88          |
| Proposed method   | Bag-of-word perceptual hashing | 97.8%            | 0.04          |

As shown in Table 3, our proposed method has a little higher average accuracy and faster computation speed than the state-of-the-art method [5] which used GIST texture features and neural networks classifier.

5. Conclusion
In this paper, we visualized malware variants as malware images, and used GLCM texture features vector and Bag-of-words model to generate perceptual hashing for malware variants detection. Experimental results show that, the proposed method outperformed the state-of-the-art method with a higher accuracy and a faster detection speed. Furthermore, it has good resilience to popular UPX packer evasive technique. Therefore, it is suitable for massive malware variants detection.

Acknowledgments
The research work described in this paper is fully supported by the Project of Innovation and Strengthening School of Higher Education in Guangdong (Project No. 2017KTSCX122).

References
[1] Symantec. 2017 Symantec Internet Security Threat Report [EB / OL], Vol 22, 2017.
[2] Conti G, Bratus S, Shubina A, et al. A Visual Study of Primitive Binary Fragment Types[J]. Black Hat USA, 2010.
[3] Yoo, I. Visualizing Windows Executable Viruses Using Self-Organizing Maps., 2004 International Workshop on Visualization for Cyber Security (VizSec).
[4] Nataraj L, Kaurhikeyan S, Jacob G, et al. Malware images: visualization and automatic classification[C]// International Symposium on Visualization for Cyber Security. ACM, 2011:1-7.
[5] Nataraj L, Manjunath B S. SPAM: Signal Processing to Analyze Malware [Applications Corner][J]. IEEE Signal Processing Magazine, 2016, 33(2):105-117.
[6] Kancherla K, Mukkamala S. Image visualization based malware detection[C]// Computational Intelligence in Cyber Security. IEEE, 2013:40-44.
[7] Haralick R M, Shanmugam K, Dickstein I H. Textural features for image classification[J]. Systems, Man and Cybernetics, IEEE Transactions on, 1973(6):610-621.
[8] Bettadapura V, Schindler G, Plotz T, et al. Augmenting bag-of-words: Data-driven discovery of temporal and structural information for activity recognition[C]// Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on. IEEE Conference on. IEEE, 2013:2619-2626.