Turkish Journal of Computer and Mathematics Education Vol.12 No.12 (2021), 929-*937*

Research Article

An Efficient Anti-Malware System With Multi Layer Perceptron And Discriminative Common Vector

P. Balamurugan[1], M. Sornam[2*], T. Amudha[3], J. Satheeshkumar[4]

[1, 3, 4] Department of Computer Applications, Bharathiar University, India
[2]Department of Computer Science, University of Madras, India
[1]palanisamy.balamurugan@gmail.com, [2]madasamy.sornam@gmail.com, [3]amudhaswamynathan@buc.edu.in, [4]j.satheesh@buc.edu.in

Article History: Received: 11 January 2021; Revised: 12 February 2021; Accepted: 27 March 2021; Published online: 23 May 2021

Abstract

In this current internet era security is one of the major concerns. There are lots of freely available software for download and many of them could be malicious to cause harm to the computer and network. There are new families of malwares developed and released on day-to-day basis. The existing anti-malware systems like anti-virus packages need to be updated frequently. There are chances that new malwares are not detected by the old packages. Hence there is a need for an efficient anti-malware system. This paper describes an alternate way to detect and classify malwares using Multi-Layer Perceptron (MLP). The malware are binary files. The binary files are converted into grey scale images. The converted images are classified using MLP-DCV. The result shows a classification accuracy of 93% when MLP is applied with DCV.

Keywords—ANN, Multilayer Perceptron, Support Vector Machine, Radial Basis Function, Discriminative Common Vector`

1. INTRODUCTION

In this digital era, there are lots of software packages available in the internet for free of cost. There are lot of risks involved in downloading those software packages. The chances of getting a malicious software package are very high. The malicious software which got downloaded could be a malware. The malware is software which is designed intentionally to create damage to the downloaded machine or to the network. The malicious software can be used for data stealing as well. To avoid such casualty, the software package needs to be verified for its genuineness. Currently we have anti-virus software packages available to this. But there are certain disadvantages that come with it, so it is necessary to shift the methodology used for better performance and results. Artificial Neural Network (ANN) is applied to solve many problems of classification and prediction.

ANN has an ability to learn the parameters and exhibits a generalization for classification [12]. There are lots of Neural Network models available in the literature. Multi-Layer Perceptron (MLP) is one of feed forward artificial neural network and used in classifying the data which are not linearly separable. Radial Basis Function Neural Network uses Radial Basis Function as an activation function for converting non-linear data points to linear data points. In this study both these models are used to classify different malwares belonging to different malware families.

The malware is an executable software. The problem under consideration is an image classification problem. So the malware binaries need to be converted to a visualizable format. The process comprises of two steps. The first step is to convert malware binaries into grey scale image. The second step is to use the grey scale images for classification.

The performance of the neural network model is based on the input feed into the input layers. Redundant or irrelevant data can take more epochs to train the model. There may be a possibility of learning unwanted data that lead to wrong classification. The required data from the input needs to be extracted before training the models. To exact important and relevant features Discriminative Common Vector method is used. Support Vector Machine methodology enables faster learning of network parameters

The same dataset is used for training and testing the two different network models. The performance of the network models are measured in-terms of accuracy of classification, F1 and Recall score.

The paper is organized as follows: Section-2 briefly explains about the related work done for malware visualization and classification, Section-3 contains details about the data set used in the study, Section-4 describes malware visualization technique, Section-5, explains about data processing, Section-6 provides details existing and proposed methods. Section-7 gives brief overview about DCV and MLP. Section-8 discusses the results and Section-9 concludes with the summary of the research.

2. RELATED WORK

There are various methods available in literature for analyzing the malware. One such method is to extract binary signatures from malware and comparing it with the available database of legitimate footprints. This method is inefficient because of the increase of number of malwares. The other way is to perform static and dynamic code analysis. Both of these approaches are time consuming and not efficient in determining the malicious part due to various limitations.
A novel approach was proposed by L. Nataraj et al to characterize and to analyze the malware [1]. The malware binaries are represented as a matrix and visualized as an image. During this visualization, it was observed that there are visual similarities in texture of image for the malwares belonging to same family.

There are several techniques proposed for classifying malware. Behavior analysis based on the feature set was proposed by Rieck et al [2]. The behavior of malwares was monitored in a test environment and the behavior was formulated as feature set. Support Vector Machine was used for classification.

Tian et al, used program length for Trojan classification [3]. Along with the length, the printable string information available in the malware was used for classification [4]. Park et al proposed a method to use behavioral graph for classification [5].

L. Nataraj et al [1] used a method of k-nearest neighbors with 10 cross validations using Euclidian distance over the malware images. The method was evaluated with 1713 malware images of 8 families and achieved an accuracy of 98%

With the generalization capability of deep learning, it was employed to create an intelligent anti-malware system proposed by Agarap et al [6]. The deep learning models use Support Vector Machine (SVM) as classifier.

Another very important step before classification is, features extraction from the input vectors. There are lot of approaches proposed for feature extraction. Principal Component Analysis (PCA) is a method in which the features are represented with a linear combination of weighted eigenvector [8]. PCA uses pixel wise correlation and this may be not sufficient to classify Malware images. Because malware images belonging to same family exhibit lot of similarities. Linear Discriminant Analysis (LDA) uses Fisher Liner Discriminant Model for features extraction which solves the limitations of Eigen vectors of PCA.

A new face recognition method using Discriminative Common Vector (DCV) was proposed by Cevkalp et al [9]. DCV uses within-class scatter matrix of input data and Gram-Schmidt orthogonalization procedure was used to get DCV.

Radial Basis Function network model was applied to problems of classification, time-series prediction, function approximation etc. T. Kathirvalavakumar et al [12] proposed a method to use RBF network for face recognition with reduced feature set using DCV.

Mouhammd et al, proposed a method for feature selection and classification of malware images using machine learning algorithms [11]. The features extraction was done based on the correlation factor among different malwares images.

To improve the accuracy of classifying grayscale malware images, byte-level malware classification based on Markov image and deep learning method was proposed [15]. The malware binaries into Markov images and the images are trained using deep convolutional neural network model. The model shows very good classification accuracy.

3. DATA SET

The dataset which is used in this comparison study is on Malimg dataset [1]. The dataset consists of 9,339 malware samples from 25 different malware families. Table 1. Lists the distribution of malware families and number of samples in the Malimg dataset [1]

Table I. Malware Families Available Malimg Dataset [1]

No.	Class	Family	No. of Samples
1	Worm	Allaple.L	1591
2	Worm	Allaple.A	2949
3	Worm	Yuner.A	800
4	PWS	Lolyda.AA 1	213
5	PWS	Lolyda.AA 2	184
6	PWS	Lolyda.AA 3	123
7	Trojan	C2Lp.P	146
8	Trojan	C2Lp.gen!g	200
9	Dialer	Instantaccess	431
10	Trojan Downloader	Swizzot.gen!I	132
11	Trojan Downloader	Swizzot.gen!E	128
12	Worm	VB.AT	408
13	Rogue	Fakerean	381
14	Trojan	Alueron.gen!J	198
15	Trojan	Malex.gen!J	136
16	PWS	Lolyda.AT	159
17	Dialer	Adialer.C	125
18	Trojan Downloader	Wintrim.BX	97
19	Dialer	Dialplatform.B	177
20	Trojan Downloader	Dontovo.A	162
21	Trojan Downloader	Obfuscator.AD	142
22	Backdoorx	Agent.FYI	116
23	Worm:AutoIT	Autorun.K	106
24	Backdoor	Rbot!gen	158
25	Trojan	Skintrim.N	80

4. DATA VISUALIZATION

The malware software is visualized as an image. The provided malware binary is read as a vector of 8 bit unsigned integers and then arranged into a 2D array. This arrangement can be visualized as a gray scale image in the range of 0 to 255 (255: white and 0: black). The width of the image is fixed and the height is varied based on the malware size.

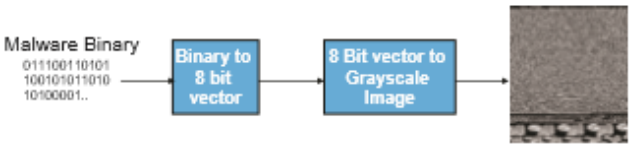

Fig. 1 Visualizing malware as a grayscale image [1]

Fig. 2 shows an example image of a common Trojan downloader, DontovoA, which downloads and executes arbitrary files [8]. From the image, the different sections or fragments are clearly seen.

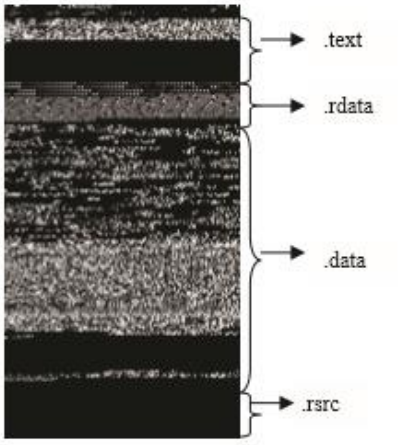

Fig. 2 Various sections of Trojan: DontavaA [1]

5. DATA PROCESSING

The malware images were resized to a 2-dimensional matrix of 32 × 32, and were compressed into 1 × 1024-size array. Each feature array was then labelled with its corresponding indexed malware family name (0 – 24). Then the features were standardized using the following equation:

$$z = \frac{X - \mu}{\sigma} \quad (1)$$

Where X is the feature to be standardized, μ is its mean value, and σ is its standard deviation.

6. METHODOLOGY

This section describes about the existing methods and the proposed method to classify the malware image dataset.

6.1. EXISTING METHOD

To classify the malware images, an efficient Antimalware System using Neural Networks with Support Vector Machine, Radial Basis Function and Discriminative Common Vector (DCV) was proposed in [13]. As a preprocessing step, standardization procedure was applied over the feature dataset. The statistical measures such as arithmetic mean and standard deviation were used to standardize the dataset. Since the dataset is not a direct image, the standardization is required. Radial Basis Function was used as classifier and DCV was used for extracting the significant features of the malware image.

The Deep Learning models along with Support Vector Machine (SVM) classifier was employed to classify the malware dataset [6]. The intelligent anti-malware system proposed was more generalized in learning the feature of the malware images and classified the malware according the class labels.

6.2. PROPOSED METHOD

The classification efficiency of RBF is depends on the dataset. It was shown that RBF classifier preformed good in recognizing the face images [7]. The dataset consider here was not a real image data. The malware executable was converted to grey scale image and RBF was applied to classify the image data [13].

This may be one of the rationales for the out performance of malware image classification. From the literate review, it was understood that MLP make good classifier algorithm when the significant features are used to train the model. Hence the method is proposed to use MLP along with DCV. The significant features are extracted using DCV and the feature set is trained using MLP for classification.

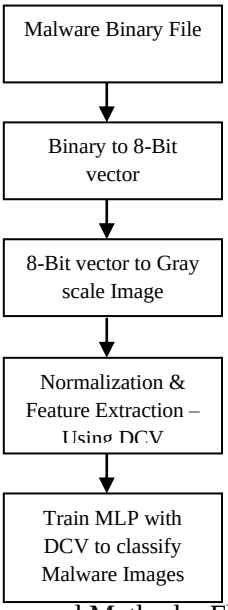

Fig. 3 Proposed Method – Flow Chart

Table II. Related Works

Reference	Methodology	Results and Accuracy
[10]	**MLP-SVM** MLP is used for training the malware feature dataset and SVM was used as a classifier to classify the malware images	The classification accuracy was achieved to be 80%. Number of False Positives were high
[14]	**RBF-DCV** DCV was used to extract vital feautres and RBF is used for classification	The accuracy was achieved to be 90%. The classification accurancy was improved because of learning relevant features.

7. ABOUT MLP AND DCV

This section describes about the terminology used in the proposed methodology.

7.1. DISCRIMINATIVE COMMON VECTOR

Discriminative Common Vector is a feature extraction mechanism for extracting the most discriminative features from the dataset. This feature extraction is necessary to avoid redundant or irrelevant data and to recover the salient features which have more discriminative power. This is very useful for malware image dataset because the images are visually looks as much as similar and need very discriminative features to classify images. Within-class scatter matrix method is applied to construct discriminative feature vector. With the extracted features of each class, a common vector is attained in the direction of eigenvectors corresponding to the non-zero Eigen values of within-class scatter matrix. The obtained new feature vector is known as Discriminative Common Vector (DCV) and shall be used for classification.

Consider, there are C classes available in the training dataset with each class containing N samples. Let x_m^i denotes m^{th} sample of i^{th} class. The within-class scatter matrix of the sample dataset constructed to get features vectors is given by [4]

$$S_W = BB^T \tag{2}$$

Where the matrix B is given by,

$$B = [\, x_1^1 - \mu_1, \ldots, x_N^1 - \mu_1, x_1^2 - \mu_2, \ldots, x_N^C - \mu_C] \tag{3}$$

Where x_i^i is ith sample of class j and μj is the mean of the samples of in the jth class.

Let Q be the set of Orthonormal eigenvectors corresponds to non-zero Eigen values of S_W. $Q = [\alpha_1 \ldots, \alpha_r]$, where r is the dimension of S_W.

The common vector is obtained by using:

$$x_{com}^i = x_m^i - Q\bar{Q}x_m^i \tag{4}$$

Where m = 1...N samples and i = 1...C classes. The discriminative component is calculated for the corresponding non-zero Eigen values using:

$$J(W_{opt}) = \text{argmax}\,[W^T S_{com} W] \tag{5}$$

$$S_{com} = B_{com} B_{com}^T \tag{6}$$

$$B_{com} = [\, x_{com}^1 - \mu_{com} \ldots . x_{com}^C - \mu_{com}] \tag{7}$$

Feature Vector for the training set is calculated using:

$$\Omega_i = W^T x_m^i \tag{8}$$

The above-mentioned steps are summarized as below:

1. Using Eq. (4) compute the value of B. Then compute the non-zero Eigen values and corresponding Eigen vectors by using the matrix BB^T
2. To obtain common vectors, select an input sample from each class and project it into the null space of S_w. Then compute the value of x_{com}^iusing the Eq. (5)
3. Using Eq. (6) and Eq. (7) compute the Eigen vectors w_k of S_{com}, corresponding to the non-zero Eigen values
4. The feature vector of the training set is obtained using Eq. (8)

7.2. MULTI LAYER PERCEPTRON (MLP)

MLP is a class of feed forward ANN (Artificial Neural Network). MLP consists of multiple layers such as Input Layer, Hidden Layer and Output Layer. MLP is mostly used in supervised learning along with backpropagation algorithm for training the network. The backpropagation algorithm computes the gradient to find the optimal value for the weight to update the network model. The algorithm consists of loss function for classification. The output layer uses ReLU as an activation function. Based on the feature set extracted using DCV, number of input layers is decided. And the number of output layers was made to be 25 to classify the malware images into one of 25 classes. The number hidden nodes were chosen based on the performance of learning, loss function and sum of squared error values. The MLP is trained with certain epochs. After training the model is used to test the images which are not part of training dataset.

8. RESULTS AND DISCUSSION

These experiments were conducted in two stages: (1) Training stage and (2) Testing stage. The Malimg dataset [1] was used in these two stages. 70% of data (6538 images out of 9339) were used for training and 30% (2801 images out of 9339) used for testing.

The experiments in this proposed work were conducted on a system with Intel (R) Core (TM) i7-6600U CPU @ 2.60GHz, 16 GB RAM with 64 bit operating system, x64 based processor. Table II summarizes the experiment results for performance metrics. The metrics includes F1 Score, Precision and Recall. The metrics shall be calculated using the following equations.

$$\text{Precision}=\frac{\text{True Positive}}{\text{True Positive} + \text{False Positive}} \tag{9}$$

$$\text{Recall}=\frac{\text{True Positive}}{\text{True Positive} + \text{False Negative}} \tag{10}$$

$$F_1 = 2.\ \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{11}$$

Before the start of the experiment, standardization procedure was applied over the feature dataset. The dataset is standardized using the Eq. (1). For normalization, the mean and the standard deviation of the feature set were used. The features used were not technically images; they were generated out from malware binary files. For general image dataset, this standardized procedure might not be required. The standardization procedure tries to normalize the feature set with respect to mean and standard deviation of binary values of the malware images.

The normalized dataset was used in both training and test phases. The Malimg dataset consists of 9,339 malware samples from 25 different malware families. The dataset were classified using Multi-Layer Perceptron with Discriminative Common Vector (MLP-DCV)

RBF network can be employed to solve image classification effectively. DCV is a feature reduction mechanism where in which it tries to build a common vector with relevant and valid data. The obtained common vector is called as DCV. DCV is passed to MLP input layer. Table 2 summarizes different network measuring parameters for both the neural network models.

Table III. Summary of Experimental Results

Measurement	**MLP-DCV**
Training Accuracy	99.12%
Testing Accuracy	93.63%
F1	0.82
Recall	0.81

Fig. 4 shows the testing performance of MLP-DCV model using Confusion Matrix. This model had a recall of 0.81 and a F1 score of 0.82

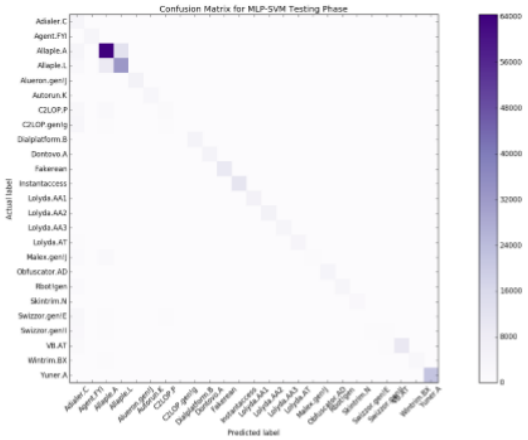

Fig. 4 Confusion Matrix for MLP-DCV

The following table compares the results of the experimental results from this study to the existing standard methods mentioned in [10] and [14]

Table IV. Experimental Results Comparison

Measurement	**MLP-SVM [10]**	**RBF-DCV [14]**	**MLP-DCV**
Training Accuracy	99.57%	98.56%	99.12%
Testing Accuracy	80.46%	89.16%	93.63%
F1	0.81	0.81	0.82
Recall	0.80	0.84	0.81

9. CONCLUSION AND FUTURE WORK

The Malimg dataset prepared by [1] consists of malware images which were used for comparison study of malware classification. MLP-DCV and RBF-SVM network models were employed for classification. As per the results, both the data models produced a classification accuracy of 92%. From the result the classification accuracy is better with 92.63% when DCV is applied with MLP.

The classification accuracy can be further improved by using any wavelet decomposition methods. The running time can also be reduced using any other dimensionality reduction techniques. This shall avoid learning unwanted features available in the dataset thereby learning the prime features.

The improvement in the network architecture design by adding more hidden layers, improving other parameter values may provide better perception for improving malware classification.

REFERENCES

CONFERENCE PROCEEDINGS

1. Lakshmanan Nataraj, S Karthikeyan, Gregoire Jacob and BS Manjunath, Malware Image: Visualization and Automatic Classification, Proceedings of the 8th international symposium on visualization of cyber security, ACM, 4, 2011

2. Rieck, K. Holz, T. Willems, C. Dussel, P. and Laskov, P, Learning and classification of malware behavior, Fifth Conference on Detection of Intrusions and Malware & Vulnerability Assessment (DIMVA'08), pp. 108-125, 2008
3. Tian, R. Batten, L.M, Verstegg. S.C, Function length as tool for malware classification, 3rd International Conference on Malicious and Unwanted Software (MALWARE), 2008
4. Tian, R. Batten, L. Islam, R. and Versteeg, S, An automated classification system based on the strings on Trojan and virus families, 4th International Conference on Malicious and Unwanted Software: MALWARE, 2009, pp.23-30
5. Park, Y. Reeves, D. Mulukutla, V. Sundravel, B, Fast malware classification by automated behavioral graph matching, Proc. Of Sixth Annual Workshop on Cyber Security and Information Intelligent Research (CSIIRW'10), 2010
6. Abien Fred M. Agarap, Francis John Hill Pepito, Towards Building an Intelligent Anti-Malware System: A Deep Learning Approach using Support Vector Machine (SVM) for Malware Classification, 2017
7. Microsoft Malware Enclyopedia, http://www.microsoft.com/security/portal/Threat/Encyclopedia/Browse.aspx
8. Turk, M. and Pentland, A., Eigenfaces for recognition, J. Cogn. Neuroscience, Vol. 3, pp.71-86, 1991
9. Cevikalp, H., Barkana, B. and Barkana, A, A comparison of the common vector and the discriminative common vector methods for face recognition, Proc. 9th World Multi-Conf. Systemics, Cybern. And Inf., Orlando, FL
10. M. Sornam, P. Balamurugan, Malware as an Image Classification – ANN approach with Discriminative Common Vector, Intl. Conf on Research Trends in Computing Technologies (ICRTCT-18), 2018, pp.384-389.
11. Al-Kasassbeh M., Mohammed S., Alauthman M., Almomani A. (2020) Feature Selection Using a Machine Learning to Classify a Malware. In: Gupta B., Perez G., Agrawal D., Gupta D. (eds) Handbook of Computer Networks and Cyber Security. Springer, Cham. https://doi.org/10.1007/978-3-030-22277-2_36
12. **JOURNAL**
13. T. Kathirvalavakumar, J. Jebakumari Beulah Vasanthi, Features Reduction using Wavelet and Discriminative Common Vector and Recognizing Faces using RBF, International Journal of Computer Applications, vol. 74, No.5, 2013
14. Swets, D.L. and Wend. J., Using Discriminant Eigen features for Image Retrieval. IEEE Trans. Pattern Analysis and Machine Intelligence, Vol. 18, No. 8, pp. 831-836, 1996
15. M. Sornam, P. Balamurugan, An Efficient Antimalware System Using Neural Networks with Support Vector Machine and Discriminative Common Vector, International Journal of Computer Application (2250-1797) Issue 8 Volume 5, Sep. – Oct. 2018
16. Baoguo Yuan, Junfeng Wang, Dong Liu, Wen Guo, Peng Wu, Xuhua Bao, Byte-level malware classification based on markov images and deep learning, Computers & Security, Volume 92, 2020