IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received November 1, 2021, accepted November 24, 2021, date of publication November 30, 2021,
date of current version December 8, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3131713

Vision-Based Malware Detection: A Transfer Learning Approach Using Optimal ECOC-SVM Configuration

W. K. WONG[1], FILBERT H. JUWONO[1], (Senior Member, IEEE), AND CATUR APRIONO[2], (Member, IEEE)
[1]Department of Electrical and Computer Engineering, Curtin University Malaysia, Miri 98009, Malaysia
[2]Department of Electrical Engineering, Universitas Indonesia, Depok 16424, Indonesia

Corresponding author: Catur Apriono (catur@eng.ui.ac.id)

This work was supported by the Universitas Indonesia's Q1Q2 Grant, in 2019, under Contract NKB-0292/UN2.R3.1/HKP.05.00/2019.

ABSTRACT Currently, malicious software (malware) detection is becoming important due to the presence of various malware as well as ransomware in digital cyberspace. Advances in Deep Learning (DL) have attracted a lot of interests in applications of malware detection. The file binaries are fed into the DL neural networks for training and testing. However, we find that overfitting may occur despite applying some precautions, such as dropout layers. The limitations can also be attributed to the final classification layers. Furthermore, in a multiclass classification task, the performance can be improved by employing a final classifier layer that is more efficient at dealing with malware characteristics. In this paper, we apply transfer learning using ShuffleNet and DenseNet-201, which are two models trained on large dataset to recognize daily objects. Features embedded in all layers may be further exploited in a way that does not result in overfitting. In particular, the entire network is frozen to prevent overfitting and an Optimal Error Correction Output Coding (ECOC) ensemble configuration of Support Vector Machines (SVM) is applied as the final classification layer. Several ECOC coding matrices are applied, i.e., One vs. All (OVA), One vs. One (OVO), Dense Random (DR), and Sparse Random (SR). Each of these configurations represents varying complexity and ensemble size and, hence, a tradeoff between computation reduction and complex non-linear separation appears. Given that the continuous values of SVM parameters may take up high computation for acquiring the optimal parameter configuration, we apply discrete values combination using a grid search approach for parameter optimization. We test the proposed model on Malimg, MaleVis, virus-MNIST, and Dumpware10 datasets. The results show better/comparable accuracy compared with the existing work. The best/average accuracy values for each dataset over 10 trials are: Malimg (99.14%/98.87%), MaleVis (95.01%/93.91%), Virus-MNIST (86.36%/85.79%), Dumpware10 (96.62%/95.79%).

INDEX TERMS Malware, machine learning, ECOC, SVM.

I. INTRODUCTION

Malicious software (malware), in general, is software designed to harm or destroy computers and computer systems. Malware can take the form of viruses, worms, Trojan horses, spyware, adware, and ransomware. The advent of the digital era has paved the way for many unscrupulous organizations to create harmful software. According to *statista.com*, the number of malware detected as of March 2020 was 677.66 million programs. Malware's prevalence and ever-expanding malware variants offer security, ethical, and economic risks in the form of extortion and data loss. It is consequently critical to identify not only the existence of malware in computer systems, but also the varieties of malware for complete security and analysis.

Conventional malware detection may typically be accomplished using three approaches [1]. The first and second approaches, respectively, make advantage of static and dynamic features [2]. Static features are obtained without executing the program. On the contrary, dynamic features are

The associate editor coordinating the review of this manuscript and approving it for publication was Vicente Alarcon-Aquino.

obtained by executing malware at runtime [1]. Furthermore, static features are appropriate for identifying malware with minimum computing overhead whereas dynamic features can detect transformed malware [3]. The third approach is the hybrid approach which combines the static and dynamic approaches.

Another approach, which has just recently emerged, is to employ image processing methods on file binaries [4]. This concept involves the conversion of codes into image pixels. The subsequent steps involve usage of various imaging technologies. In this aspect, various methods have been investigated, including textural method [5]. The recent development in DL has also attracted the application in malware detection mainly due to its ability self-generate features for classification.

To date, there exist various attempts to improve the malware recognition using various architecture schemes of DL. Most of these attempts have been tested on a specific dataset and may not guarantee an improved separation between the multiclass malware. Many approaches have been proposed for a more extensive and even specialized network. As a result, there are a plethora of publicly available datasets that have been released and then removed from public circulation, arguing irrelevance owing to the age of the dataset and the constant change of malware types. Most datasets identify up to 10 different types of malware. In practice, there may be a finite class that must be distinguished. The direction of malware class identification development should target towards enhancing multiclass classification by improving the final classification layer. In this aspect, any classifier may be enhanced by applying Error Correction Output Coding (ECOC) ensemble principle.

We also note that there is a growing interest in using transfer learning in DL applications. There are many benefits of using transfer learning, e.g., the convenience of training method and the ability to take use of the extensive characteristics obtained in its integrated layers. Because the majority of these characteristics are primarily learned on large image sets, we need to investigate whether they might be useful in malware classification. As most of the fully connected networks are trained on relatively identical objects, global averaging layers are more logical to capture diversity. This is especially true when attempting to hybridize features derived from multiple networks.

A. RELATED WORK

In [4], the authors applied Convolutional Neural Network (CNN) with a modification on the pooling layers to deal with varying image sizes. We note that this particularly a concern as malware script (normally in .asm file) may have varying length. Regardless of the author's claim of originality, this issue can be solved using a variety of image resizing methods. Authors in [6] applied autoencoders to detect malware. When compared to existing malware detection algorithms, the proposed approach achieved 93% accuracy, obtained higher F1-score values, and required less training data. Authors in [7] applied one million instances of malware-goodware dataset spanning executables collected over one year in duration (EMBER dataset). Various machine learning models were used and compared. Note that other lesser dataset includes Malimg [8], MaleVis [9], Dumpware10 [10], and Virus-MNIST [11]. Most of the dataset have varying sizes except for Virus-MNIST which have a size of $32 \times 32\times\ 1$. In [12], authors applied CNN and LSTM which yielded a recognition of approximately 95%-96% using two selected CNN-LSTM configurations on Malimg dataset. In [13], authors applied VGG16 DL features on Support Vector Machine (SVM) and yielded 92% accuracy.

Apart from DL method, it is observed that textural analysis also remains an active approach in detecting malware through imaging approach. In [14], authors applied textural features which consisted of wavelet transform and Gabor transform to extract textural features. Classification using k-Nearest Neighbor (KNN) algorithm for detection yielded 97% Accuracy rate. Another example of this approach can be seen in [10] where Histogram of Oriented Gradients (HOG) was applied on a particular set of malware memory content datasets. The datasets did not take conventional approach by converting Portable Executable (PE) file binaries into images but applies their memory dumps as RGB images pixels. This yielded 96.39% accuracy for 10 classes of malware families.

B. MOTIVATION

One of the highly notable problems is the presence of large number of classes in malware detection. As the number of classes increases, so does the misclassification among the classes. ECOC ensemble has been proposed in many areas to improve the classification performance. For example, in [15], it was shown that 99.7% recognition can be achieved using CNN-ECOC for the case of brain tumor detection. Similarly, for skin cancer detection [16], AlexNet, a pre-trained CNN model, was used to extract the features. For classification, the ECOC-SVM classifier was used. Using ECOC-SVM, the overall accuracy achieved was 86.21%. The last example is in [17] where brain tumour was detected using CNN-ECOC. The best configuration was achieved using AlexNet, which was 99% accuracy. For malware detection, we borrow the ideas to use ECOC-based classification as explored in the above-mentioned papers.

Another motivation for our research is the exploration of the transfer learning approach in detection of malware. As we know, most of the existing deep learning models are trained on recognizing approximately 1000 objects. It would be interesting to observe how these embedded features in existing models can contribute to the recognition of malware. Extracting these features as raw numerical inputs are explored as training could possibly incur over fitting.

C. CONTRIBUTIONS

To date, most of the research work emphasizes on applying some modified DL configuration for public domain datasets. Others have resorted to extracting dump memories

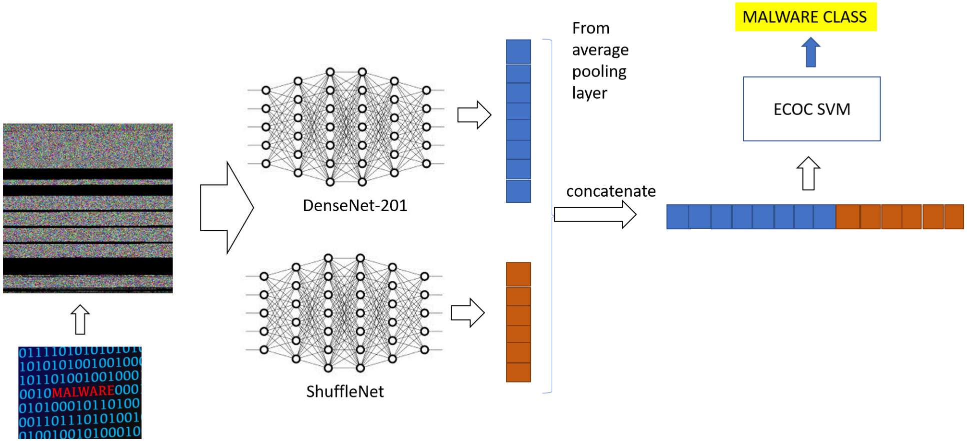


FIGURE 1. Proposed malware detection concept.

for identification, while some turn file binaries into pixels. Overall, DL remains an attractive approach for malware detection. To the best of our knowledge, the use ECOC has not been applied to the malware detection.

In this paper, we propose to concatenate features extracted from the global averaging layers of two DL networks, i.e., ShuffleNet and DenseNet-201. Note that the networks represent a mid-size and a large DL models, respectively. Concatenating these features into a single vector, we further train the data on ECOC–SVM ensemble. We take advantage of complex mapping of ECOC to separate the multiclasses. In order to optimize the SVM parameters for the ensemble, a grid-search is performed individually on the respective dataset. The overall concept is depicted in Fig. 1. Furthermore, results are tested on 20% reserved independent data with 10 trials. We compared and benchmark the results against other benchmark research and against similar deep learning model. The contributions of this paper can be summarized as follows:

- The usage of applying transfer function layers of ShuffleNet and DenseNet-201 by freezing the entire network is explored. This is because each DL network contains important features that may be useful for malware detection.
- Various ECOC configurations which contribute to the recognition are investigated. The trade-off between computation efficiency vs complexity is also discussed.
- In order to optimize the ensemble classifiers parameters, a grid search approach using discrete value combination is applied.

II. METHODOLOGY

A. DATASET

Fig. 2 shows four malware samples from Maling dataset visualized as binaries. From visual observation, they are almost indistinguishable. We consider multiple networks for the proposed four datasets, except for the virus MNIST as the generated features will exceed the original pixel size (32 × 32 × 1).

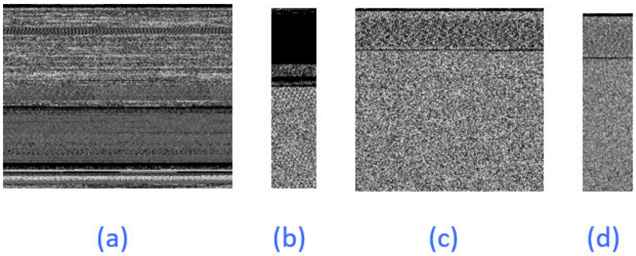


FIGURE 2. Sample images from the Malimg dataset (a) Adialer.C (b) Agent.FYI (c) Allaple.A (d) Allaple.L.

Table 1 summarizes the dataset and the details of the dataset selected for evaluation. Malimg, MaleVis, and Virus-MNIST are images extracted from file binaries whereas Dumpware10 is memory dumps converted into images. These various image extraction groups would reflect the resilience of our technique. Regardless of whether the features enclosed in deep learning models are binaries or memory dumps, it is considered that the features encapsulated in deep learning models contain adequate characteristics that may be transferable for malware detection.

We note that Virus–MNIST dataset is a compilation of several datasets [18]. The sources are acquired from *virusshare.com*, whereas the non-malicious samples are from *portableapps.com*. The limitation and challenge of this dataset lie in its small size (i.e., $32 \times 32 \times 1$). However, we may argue that this may open new possibilities for investigation in terms of maximizing computing resources. The term implies that the dataset intends to be assessed like the MNIST handwriting dataset [19].

B. FEATURE EXTRACTION

The common way to extract numerical features in a DL network is from the fully connected layer. These features are then fed to a more complex classifier for training. There is, however, a growing trend to apply global average pooling

TABLE 1. Summary of malware datasets.

Dataset	Sample no (No of classes)	Class labels
Malimg	9,339 (25 classes of malware)	1) Adialer.C, 2) Agent.FYI, 3) Allaple.A, 4) Allaple.L, 5) Alueron. gen!J, 6) Autorun.K, 7) C2Lop.P, 8) C2Lop.gen!G, 9) Dialplatform.B, 10) Dontovo.A, 11) Fakerean, 12) Instantaccess, 13) Lolyda.AA 1, 14) Lolyda.AA 2, 15) Lolyda.AA 3, 16) Lolyda.AT, 17) Malex.gen!J, 18) Obfuscator.AD, 19) Rbot!gen, 20) Skintrim.N, 21) Swizzor.gen!E, 22) Swizzor.gen!I, 23) VB.AT, 24) Wintrim.BX, 25) Yuner.A
MaleVis	13,760 (25 classes of malware + 1 class benignware)	1) Adposhel, 2) Agent-fyi, 3) Allaple.A, 4) Amonetize, 5) Androm, 6) AutoRun-PU, 7) BrowseFox, 8) Dinwod!rfn, 9) Elex, 10) Expiro-H, 11) Fasong, 12) HackKMS.A, 13) Hlux!IK, 14) Injector, 15) InstallCore.C, 16) MultiPlug, 17) Neoreklami, 18) Neshta, 19) Other, 20) RegRun.A, 21) Sality, 22) Snarasite.D!tr, 23) Stantinko, 24) Hilium.A, 25) VBKrypt, 26) Vilsel
Virus-MNIST	51,880 (9 classes of regrouped malware + 1 class benignware)	1) Beneware (non-malicious), 2) Adware, 3) Trojan (Type 1), 4) Trojan (Type 2), 5) Installer, 6) Backdoor, 7) Crypto, 8) Backdoor, 9) Downloader, 10) Heuristic
Dumpware10	4,294 (10 groups of malware + 1 group of benignware)	1) Adposhel, 2) Allaple.A, 3) Amonetize, 4) Autotun-PU, 5) BrowseFox, 6) Dinwod!rfn, 7) Installcore, 8) Multiplug, 9) Other, 10) VBA, 11) Vilsel

layer instead of the fully connected layer. The global pooling average layer is the average of all the sub-pooling layers' values. The global average layer has the benefit of not being prone to overfitting. Furthermore, in the case of our approach to performing feature extraction, global averaging makes more sense because the output is normally classified as the same object class.

The global average pooling layers from two pre-trained DL models are considered. As mentioned previously, the networks include ShuffleNet and DenseNet-201. ShuffleNet consists of 173 layers and has been train on ImageNet database. DenseNet-201, as the name suggest, contains 201 layers and, likewise, has been trained on ImageNet database. The features extracted from the global average pooling layer from each network are then concatenated to be fed into the classification layer.

C. PROPOSED ECOC-SVM CLASSIFICATION LAYER

ECOC is a state-of-the-art classifier ensemble configuration that is typically employed with high-dimensional and extremely non-linear data. In the context of malware detection, this is highly relevant. The idea uses the classifier's output for a binary string to calculate any distance metric, and the class with the shortest distance or proximity is assigned. Distance metrics like the hamming distance and the Euclidean distance are often used. Any binary classifier can be used in the ECOC setup. However, SVM, which is commonly considered as a binary classifier, is frequently used as the configuration's basic classifier. Furthermore, multi-class setup need some sort of ensemble setting, i.e., ECOC configuration.

To separate two target/outlier classes, the individual binary SVM employs a kernel projection and plane. Consider (x_i, y_i) where x_i is the feature set, $y_i \in [-1, +1]$ is the respective label values, and $i = 1, 2, \cdots, n$ for n instances. The border that separates the data vectors x_i into the label of -1 or $+1$ is denoted by $f(x) = w^T x + b$, where w is the weight and b is the bias. We could use a hyperplane vector to divide the multi-dimensional data into their appropriate labels. Essentially, The goal of plane optimization is to reduce $w^T w$ s.t. $y_i f(x) \geq 1$ to a minimum.

We also note that the optimization will not yield much practicality without considering a "soft margin" in optimization during training. The soft margin mechanism considers the slack variables as well during optimization i.e sample sets that are near to the boundary $f(x) = 0$ as follows

$$\min_{w,b}(||\langle w, w\rangle|| + c\sum_i \zeta_i), \tag{1}$$

$$\text{s.t. } y_i(\langle w, x_i\rangle - b) \geq 1 - \zeta_i, \forall i \tag{2}$$

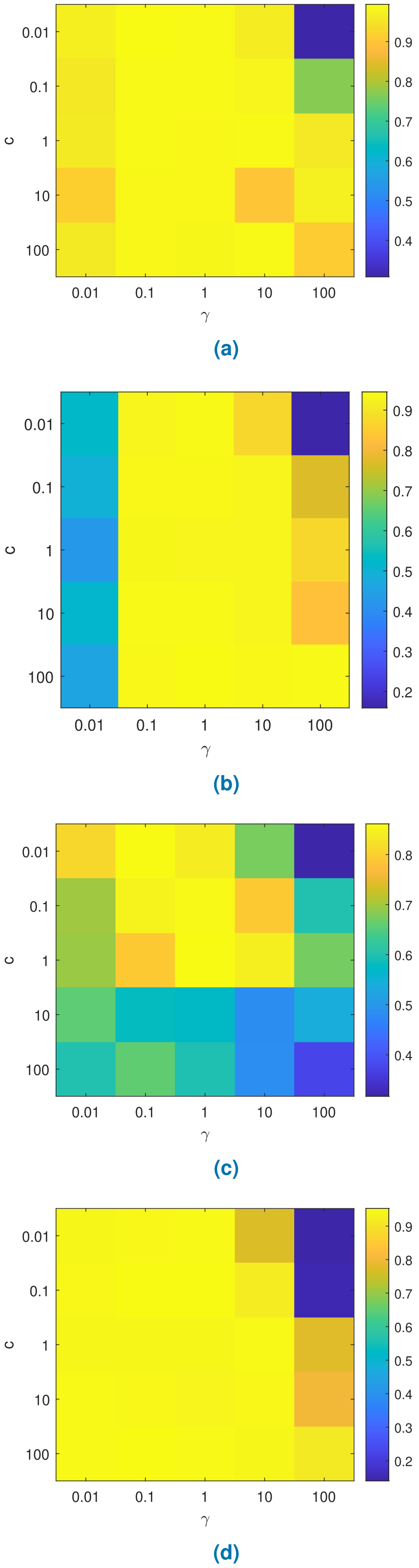
and

$$\zeta_i \geq 0, \ \forall i, \tag{3}$$

where $\langle \cdot, \cdot \rangle$ denotes the inner product, ζ_i are the slack variables, and c is the constraint parameter determines the weightage during optimization of boundary to reduce slack variables.

Another aspect of SVM that highly determines the effectiveness of hyperplane setting lies in γ value by incorporating a kernel that projects the original data into a higher dimension of $n + 1$. This way, we can further optimize non-linear separation across classes. Various transforms can be used to gain higher dimensions. We used the Radial Basis Function (RBF) given by

$$G(x, \bar{x}) = \exp(-\gamma||x - \bar{x}||)^2 \tag{4}$$



(a)

(b)

(c)

(d)

FIGURE 3. Grid search results in estimating optimal *c* and *γ* values: a) Malimg, b) Malevis, c) Virus-MNIST, d) Dumpware10.

to get the additional dimension (kernel) for projection. Assuming that x and $\bar{x}$ are two feature vectors (2-D), a higher dimension can be created with careful selection of γ.

TABLE 2. Test results (10 trials).

Config.	Malimg	MaleVis	Virus-MNIST	Dump-ware10
Accuracy (Average)				
OVA	**98.87%**	92.59%	83.70%	95.14%
OVO	98.82%	93.54%	81.80%	94.95%
DR	98.69%	93.26%	**83.70%**	**95.79%**
SR	98.76%	**93.91%**	84.10%	95.48%
F1 (Average)				
OVA	96.97%	93.40%	78.94%	93.66%
OVO	96.71%	94.63%	85.79%	93.61%
DR	96.56%	93.94%	77.50%	94.67%
SR	96.72%	95.57%	79.20%	94.28%

In order to acquire optimal values, a grid search approach is applied. We apply average of five trials for each configuration using cross validation on the training sets. The values consist of accuracy values. From the acquired results, the optimal γ and c values are acquired from further evaluation using various ECOC configurations.

Based on the optimal parameter configurations, subsequent coding matrix configurations are also evaluated for the individual dataset. As discussed earlier, ECOC configuration enables binary classifiers to take advantage a binary vector. Let $|m_{k,j}|$ be the absolute value of the (k, j)-th element of the coding matrix which is used to calculate the distance from an assigned class. The assigned class, $\hat{k}$, therefore, can be expressed as

$$\hat{k} = \min_{k} \frac{\sum_{j=1}^{L} |m_{k,j}| g(m_{k,j}, s_j)}{|m_{k,j}|}, \tag{5}$$

where $k = 1, \cdots, K$ is the index of the class, K is the number of classes, L is the length of the code, which is $\lfloor 10 \log_2 K \rfloor$. In (5), $g(\cdot, \cdot)$ is the binary loss function given by

$$g(u, v) = \frac{\max(0, 1 - uv)}{2}, \tag{6}$$

where $\max(0, a)$ returns a when $a \geq 0$ and 0 otherwise.

We explore several configurations namely One vs. All (OVA), One vs. One (OVO), Dense Random (DR), and Sparse Random (SR) coding matrix. The first two configurations represent configurations with lower binary classifiers while the subsequent two configurations represents configurations with higher number of binary classifiers. Both DR and SR codes utilize stochastic elements in the coding design of the matrix. As the name suggest, the configurations of DR and SR differ only by the sparseness of the coding matrix. In SR, a probability is generated for each cell matrix, and it randomly assigns classes as positive (+1) or negative (−1) with a probability of 0.25 for each, while ignoring classes with a probability of 0.5. In DR configuration, each cell in the matrix is assigned either positive or negative based on 0.5 probability. As a result, DR applies random assignment of +1 and −1 in its matrix entries.

Note that the lightest configuration is the OVA. In OVA, each classier will be assigned −1, +1 in which each classifies

TABLE 3. Test results (10 trials using best parameter setting configurations).

Metric		Malimg	MaleVis	Virus-MNIST	Dumpware10
Accuracy	Mean	98.87%	93.91%	85.79%	95.79%
	Best	99.14%	95.01%	86.37%	96.62%
	Std Dev	0.19%	0.52%	4.00%	5.70%
Precision	Mean	99.50%	99.75%	98.38%	99.58%
	Best	99.96%	99.80%	98.45%	99.66%
	Std Dev	0.01%	0.02%	0.04%	0.06%
Recall	Mean	97.06%	94.83%	83.05%	94.69%
	Best	98.23%	95.61%	83.88%	96.04%
	Std Dev	0.51%	0.52%	0.40%	0.85%
F1	Mean	96.97%	94.51%	81.80%	94.67%
	Best	98.03%	95.37%	82.51%	95.88%
	Std Dev	0.46%	0.46%	0.30%	0.78%

TABLE 4. Benchmarking with other research work.

Reference	Description	Accuracy
Malimg		
[1]	-	98.48% (10 fold cross validation 84.92%)
[20]	Deep Random Forest Paradigm	98.65%
[21]	CNN/ResNet-50	98.98% (CNN), 99.40% (ResNet-50)
[22]	DL with SVM classification layer	84.92%
DenseNet-201	Fine tuning with transfer function (learning rate 0.001, SGDM optimization)	97.30%
Proposed	-	99.14% (best), 98,87% (mean)
MaleVis		
[20]	Deep Random Forest Paradigm	97.53%
[21]	CNN/ResNet-50	93.00%
DenseNet-201	Fine tuning with transfer function (learning rate 0.001, SGDM optimization)	89.50%
Proposed	-	95.01% (best), 93.91% (mean)
Virus-MNIST		
[11]	MobileNetV2	80.00%
Proposed	-	85.79% (mean), 86.37% (best)
Dumpware10		
[10]	GIST and HOG	96.39%
Proposed	-	96.62% (best), 95.79% (mean)

one class against all. The exact number of classifiers in this configuration is K. In contrast, in OVO configuration, each classifier function needs to distinguish between two classes and those that are irrelevant are assigned 0 (no evaluation). In terms of ensemble complexity, OVA has the smallest ensemble. In the case of K classes, OVA requires K binary classifiers. OVO has exactly $K(K-1)/2$ binary classifiers. SR and DR have approximately $15\log_2 K$ and $10\log_2 K$, respectively. Hence, in terms of complexity, we may rank the lightest ensemble as OVA followed by OVO, DR, and SR.

D. EVALUATION METRICS

Four evaluation metrics (accuracy (Acc), precision (Pr), recall (Re), and F1-score ($F1$)) are used to evaluate the performance of our proposed classification method. The four metrics are defined as follows

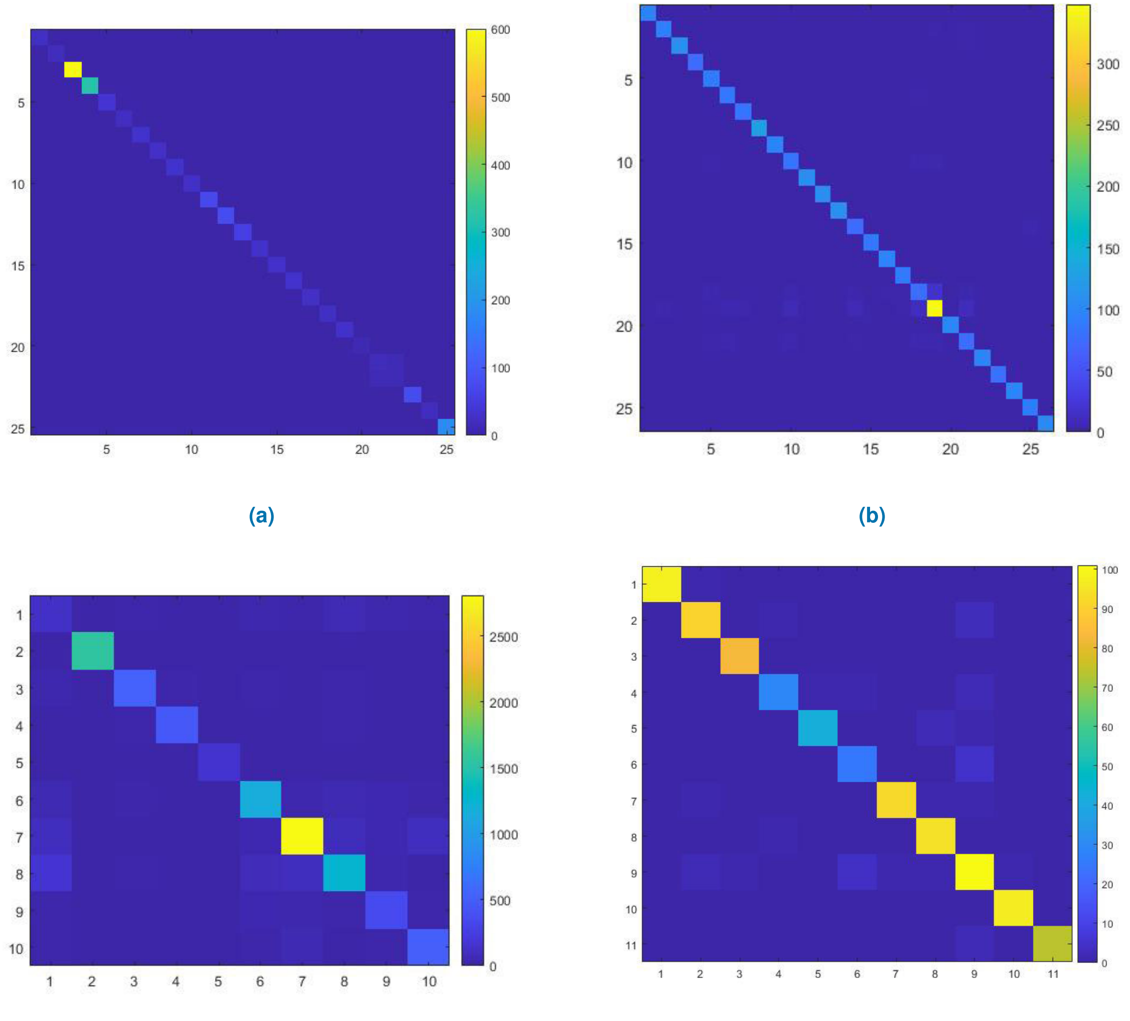
$$Acc = \frac{TP + TN}{TP + TN + FP + FN}, \tag{7}$$

$$Pr = \frac{TP}{TP + FP}, \tag{8}$$

$$Re = \frac{TP}{TP + FN}, \tag{9}$$

$$F1 = \frac{2 \times Pr \times Re}{Pr + Re}, \tag{10}$$

where TP, TN, FP, and FN are the number of true positive, true negative, false positive, and false negative samples, respectively.



(a)

(b)

(c)

(d)

FIGURE 4. Confusion matrices: a) Malimg b) MaleVis c) Virus-MNIST d) Dumpware10 (please refer to Table 1 for classes information).

III. RESULTS

The optimal c and γ values for each dataset are shown in Fig. 3. For evaluating the performance of our proposed model, we use the optimal c and γ values as well as apply 80%-20% separation for training and testing, respectively. Table 2 shows the comparison between the various coding matrix configurations stated earlier. The colours in Fig. 3 show the average accuracy criteria for each pair of c and γ. From Fig. 3 and Table 2, we summarize the optimal configurations for our proposed model:

- Malimg: OVA, $\gamma = 0.1$, $c = 0.1$,
- Malevis: OVO, $\gamma = 1$, $c = 1$,
- Virus-MNIST: OVO, $\gamma = 1$, $c = 1$,
- Dumpware10: SR, $\gamma = 0.1$, $c = 0.1$.

Furthermore, using optimal configurations, Table 3 summarizes the performance metrics of our proposed model. Note that precision, recall, and F1-score are based on macro calculation.

We also provide some comparisons with the existing work. Table 4 shows the benchmark with other references. We note that for the four datasets, the performance of our proposed model is on par with other approaches. This is an indication that features from the images encapsulated in the deep learning layers (in specific the global averaging layers) are suitable for malware detection purposes. We also find that the detection performance using the Virus-MNIST dataset is rather low when compared to others. This is due to the small size image that we fit into the ShuffleNet input layer. A larger image size may result in a greater detection rate. Regardless, the comparable performance should be taken into account. It is worth noting that the first three datasets are file binaries, but the Dumpware10 dataset is made up of memory dumps rendered as images. This result indicates that the images contained in the networks are equally appropriate for the stated purposes.

The confusion matrices are presented in Fig. 4. The legend denotes the number of samples. We note that the datasets are not balanced. The performance is mainly dictated by large confusion between one or two classes of data. Therefore, F1-score may be used as the verification metric. In general, it can be seen that the classes are well classified.

IV. CONCLUSION AND FUTURE WORK

In this paper, we have proposed a classification layer model using ECOC-SVM for detecting malware images. In addition, we have also used concatenated features from global pooling average layer of two pre-trained networks, i.e., ShuffleNet and DenseNet-201. The evaluation conducted in this research shows that our proposed method produced encouraging results across the four malware datasets. The best results are: 98.87% for Malimg, 93.91% for MaleVis, 85.79% for Virus-MNIST and 95.79% for Dumpware10. The results have indicated that optimal separation between the classes are dependent on the dataset. Therefore, it is challenging to generalize a common ECOC coding matrix for this specific application.

It is important to balance complexity of the classification algorithm with computation time. In the context of ECOC coding matrix configuration, OVA represents the lightest configuration while SR configuration represents the highest computation complexity. Although we envision a lighter capacity configuration, results have shown that some datasets may perform better with more complex configuration. There is also no clear pattern in the parameters settings. However, the output layer from global averaging pooling has shown that most numerical outputs are in the range of $[1, 2]$. We conclude that optimal range may fall between $[0.1, 1]$ for both c and γ of SVM parameters.

Further questions are arise. Can we introduce simultaneous feature selection and parameter tuning in view of the current progress? Again, this would incur high computational optimization time. Are the features optimal for transfer learning for malware detection? This can further be answered by exploring other pre-trained DL networks. Currently, it appears that features encapsulated from networks trained on large image datasets have shown promising outcomes when compared to others.

Furthermore, we have compared a number of classifiers required for this task. It is believed by observing the results from comparing with the dataset, more optimality is expected across most malware dataset when compared to their conventional DL training approaches. As shown, feature extraction by freezing the entire network has removed the overfitting issue. We have also proposed a grid search approach to optimize the parameters, namely the box constraint and the RBF parameters as an optimal approach when compared to other approaches that would incur high numbers of in-loop evaluations. However, this will incur high computational time due to the in loop fitness evaluation.

REFERENCES

[1] D. Gibert, C. Mateu, J. Planes, and R. Vicens, "Using convolutional neural networks for classification of malware represented as images," *J. Comput. Virol. Hacking Techn.*, vol. 15, no. 1, pp. 15–28, Mar. 2019.

[2] B. Xu, Y. Li, and X. Yu, "Malware detection based on static and dynamic features analysis," in *Machine Learning for Cyber Security*, X. Chen, H. Yan, Q. Yan, and X. Zhang, Eds. Cham, Switzerland: Springer, 2020, pp. 111–124.

[3] T. Kim, B. Kang, and E. G. Im, "Runtime detection framework for Android malware," *Mobile Inf. Syst.*, vol. 2018, pp. 1–15, Jan. 2018.

[4] K. He and D.-S. Kim, "Malware detection with malware images using deep learning techniques," in *Proc. 18th IEEE Int. Conf. Trust, Secur. Privacy Comput. Commun./13th IEEE Int. Conf. Big Data Sci. Eng. (TrustCom/BigDataSE)*, Aug. 2019, pp. 95–102.

[5] Y. Zhao, W. Cui, S. Geng, B. Bo, Y. Feng, and W. Zhang, "A malware detection method of code texture visualization based on an improved faster RCNN combining transfer learning," *IEEE Access*, vol. 8, pp. 166630–166641, 2020.

[6] X. Jin, X. Xing, H. Elahi, G. Wang, and H. Jiang, "A malware detection approach using malware images and autoencoders," in *Proc. IEEE 17th Int. Conf. Mobile Ad Hoc Sensor Syst. (MASS)*, Dec. 2020, pp. 1–6.

[7] C. Galen and R. Steele, "Empirical measurement of performance maintenance of gradient boosted decision tree models for malware detection," in *Proc. Int. Conf. Artif. Intell. Inf. Commun. (ICAIIC)*, Apr. 2021, pp. 193–198.

[8] L. Nataraj, S. Karthikeyan, G. Jacob, and B. S. Manjunath, "Malware images: Visualization and automatic classification," in *Proc. 8th Int. Symp. Vis. Cyber Secur. (VizSec)*. New York, NY, USA: Association for Computing Machinery, 2011, pp. 1–7.

[9] A. S. Bozkir, A. O. Cankaya, and M. Aydos, "Utilization and comparision of convolutional neural networks in malware recognition," in *Proc. 27th Signal Process. Commun. Appl. Conf. (SIU)*, Apr. 2019, pp. 1–4.

[10] A. S. Bozkir, E. Tahillioglu, M. Aydos, and I. Kara, "Catch them alive: A malware detection approach through memory forensics, manifold learning and computer vision," *Comput. Secur.*, vol. 103, Apr. 2021, Art. no. 102166.

[11] D. Noever and S. E. M. Noever, "Virus-MNIST: A benchmark malware dataset," 2021, *arXiv:2103.00602*.

[12] H. Guo, J.-T. Wu, S.-G. Huang, Z.-L. Pan, F. Shi, and Z.-H. Yan, "Research on malware variant detection based on global texture features," in *Proc. IEEE 20th Int. Conf. Commun. Technol. (ICCT)*, Oct. 2020, pp. 1443–1446.

[13] E. Rezende, G. Ruppert, T. Carvalho, A. Theophilo, F. Ramos, and P. de Geus, "Malicious software classification using VGG16 deep neural network's bottleneck features," in *Information Technology—New Generations*, S. Latifi, Ed. Cham, Switzerland: Springer, 2018.

[14] R. Vinayakumar, M. Alazab, K. Soman, P. Poornachandran, and S. Venkatraman, "Robust intelligent malware detection using deep learning," *IEEE Access*, vol. 7, pp. 46717–46738, 2019.

[15] C. Q. Lai, H. Ibrahim, J. M. Abdullah, A. Azman, and M. Z. Abdullah, "Convolutional neural network utilizing error-correcting output codes support vector machine for classification of non-severe traumatic brain injury from electroencephalogram signal," *IEEE Access*, vol. 9, pp. 24946–24964, 2021.

[16] N. Hameed, A. M. Shabut, and M. A. Hossain, "Multi-class skin diseases classification using deep convolutional neural network and support vector machine," in *Proc. 12th Int. Conf. Softw., Knowl., Inf. Manage. Appl. (SKIMA)*, Dec. 2018, pp. 1–7.

[17] M. K. Abd-Ellah, A. I. Awad, A. A. M. Khalaf, and H. F. A. Hamed, "Two-phase multi-model automatic brain tumour diagnosis system from magnetic resonance images using convolutional neural networks," *EURASIP J. Image Video Process.*, vol. 2018, no. 1, pp. 1–10, Dec. 2018.

[18] A. Oliveira, "Malware analysis datasets: Raw PE as image," IEEE Dataport, 2019. [Online]. Available: https://ieee-dataport.org/open-access/malware-analysis-datasets-raw-pe-image, doi: 10.21227/8brp-j220.

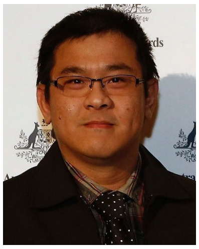
[19] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.

[20] S. A. Roseline, S. Geetha, S. Kadry, and Y. Nam, "Intelligent vision-based malware detection and classification using deep random forest paradigm," *IEEE Access*, vol. 8, pp. 206303–206324, 2020.

[21] A. Singh, A. Handa, N. Kumar, and S. K. Shukla, "Malware classification using image representation," in *Cyber Security Cryptography and Machine Learning*, S. Dolev, D. Hendler, S. Lodha, and M. Yung, Eds. Cham, Switzerland: Springer, 2019, pp. 75–92.

[22] A. F. Agarap, "Towards building an intelligent anti-malware system: A deep learning approach using support vector machine (SVM) for malware classification," 2019, *arXiv:1801.00318*.

W. K. WONG received the M.Eng. and Ph.D. degrees from Universiti Malaysia Sabah, in 2012 and 2016, respectively. He is currently working as a Senior Lecturer with the Department of Electrical and Computer Engineering, Curtin University Malaysia. Prior to joining academia, he was with the telecommunication and building services industry. His research interests include embedded system development, machine learning applications, and image processing.



FILBERT H. JUWONO (Senior Member, IEEE) received the B.Eng. degree in electrical engineering and the M.Eng. degree in telecommunication engineering from the University of Indonesia, Depok, Indonesia, in 2007 and 2009, respectively, and the Ph.D. degree in electrical and electronic engineering from The University of Western Australia, Perth, WA, Australia, in 2017. He is currently with the Department of Electrical and Computer Engineering, Curtin University Malaysia. His research interests include signal processing for communications, wireless communications, power-line communications, machine learning applications, and biomedical engineering. He was a recipient of the prestigious Australian Awards Scholarship, in 2012. He serves as an Associate Editor for IEEE Access, a Review Editor for *Frontiers in Signal Processing*, and the Editor-in-Chief for a newly established journal *Green Intelligent Systems and Applications*.

CATUR APRIONO (Member, IEEE) received the B.Eng. and M.Eng. degrees in telecommunication engineering from the Department of Electrical Engineering, Universitas Indonesia, Indonesia, in 2009 and 2011, respectively, and the Ph.D. degree in nanovision technology from Shizuoka University, Japan, in 2015. Since 2018, he has been an Assistant Professor in telecommunication engineering with Universitas Indonesia, where he is also a Lecturer with the Department of Electrical Engineering, Faculty of Engineering. His main research interests include antenna and microwave engineering, terahertz waves technology, and optical communications. He has been a member of the IEEE Antenna and Propagation Society (AP-S) and the IEEE Microwave Theory and Technique Society (MTT-S). He has had involved in the IEEE Joint Chapter MTT/AP Indonesia Section as a Secretary and a Treasurer, in 2017, 2018, and 2019, and also active in various chapter activities, such as the First Indonesia–Japan Workshop on Antennas and Wireless Technology (IJAWT) as a Secretary and the 2019 IEEE International Conference on Antenna Measurements Applications (CAMA), Bali, in October 2019, as a Treasurer.