DeepMalware: A Deep Learning based Malware Images Classification

Mehmood Alam*, Adeel Akram†, Talha Saeed†, Sobia Arshad†
*Department of Computer Engineering UET Taxila
†Deep Packet Inspection Lab, Telecommunication Engineering Department, UET Taxila
*engr.mehmoodalam@gmail.com ,†talha.saeed@uettaxila.edu.pk,†sobia.arshad@uettaxila.edu.pk
†Correspondence: adeel.akram@uettaxila.edu.pk;

2021 International Conference on Cyber Warfare and Security (ICCWS) | 978-1-6654-1208-7/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICCWS53234.2021.9703021

***Abstract*—The rapid development in the field of communication and networks has increased the size and complexity of the network. Due to these reasons, many malwares are generated that create a challenges for systems to detect these malwares accurately. Moreover, the presence of malicious software (malware) with the aim of launching various malware files within the network cannot be ignored. Although, there are numerous efforts by the researchers to develop procedures for automatic classification of malware. The methods of manually analyzing malware file is very time-consuming. Lately, deep learning-based methods are being used for the classification of malware. In this paper, we present a rapid and accurate malware classification based on different Convolutional Neural Network (CNN) architectures—including a custom CNN as well as commodity off-the-shelf CNN architectures such as AlexNet, VGG-16, ResNet-50, Inceptionv3 models. This has been demonstrated on benchmark datasets of Malimg dataset, which is consists of malware images that were obtained after conversion of Malware binaries. The trained models allow accurate classification of malware and report a test accuracy of 98.90%.**

***Index Terms*—Machine Learning; Deep Learning; Artificial Neural Network; Cyber security; Malware classification; Ensemble Method**

I. Introduction

With the recent significance and progress in the development of communication and internet technologies during the last decade, network security has become a key research area. Network security generally include firewalls, antivirus software and intrusion detection system (IDS) [1]. These techniques ensure the safety from malware attacks. In recent years, the degree of harm caused by a number of malwares has been increased and has been affecting the computer systems around the world [2] . Many of the malware systems files are being created daily. Malware is divided into many classes according to their functionalities i.e., Viruses, Worms, Trojans, and Backdoors. These classes were divided into families according to the type of variants. Malware implements many complicated methods such as inserting dead code, rearranging a subroutine and switching code, to create variables from a file The family of malware that exists to avoid detection [3].Traditional malware detection methods that rely primarily on malware which analyze code properties. These features is used to implement machine-learning based malware detection. Although, These methods do not detect malware variants. Instead of focusing on invisible malware detection functions, [4] designed a new approach for detection of malware using visible features. The author changed the structure of the packed binary executable file to 2D Grayscale images. Later, these visual functions were used to detect malware. The results outcome that the analysis of binary texture was more precise and completed in less time. In many previous works the researchers developed many techniques to increase the rate of detection of malware and have low the false alarm rate. In this regard, researchers have shown that Technologies based on machine learning (ML) and deep learning (DL) can deliver impressive performance in the domain of malware classification.

Numerous Classical Machine learning techniques such as naive bayes [5], k-nearest neighbours [6], support vector machine [7], random forests [8], and decision tree [9], have been used for detection of malware.

The *major contributions of this paper* are as follows:

1) In this paper, we design a deep convolutional neural network (CNN) architecture for malware detection and classification, which is general type, unlike classical methods. current technologies which achieve high accuracy are frequently designed for specific dataset.
2) The proposed methodology outperformed the traditional machine learning and the past-CNN based malware classification. The trained models allow accurate classification of malware with a test accuracy of 98.90%

The rest of the paper is arranged as follows. Section II describes the related work in context of Malware classification . Section III explains our methodology that includes III-A system overview, III-B MalImg dataset, III-C Convolutional Neural Network (CNN) and III-D Transfer Learning,. Section IV provides the details of experimental results. Finally, Section V concludes the paper with a discussion.

II. Related work

To effectively detect malware variants in Windows, many researchers have worked on visualizing the malware to maximize detection and classification accuracy and decrease redundant time. This section includes related works for visualization, Techniques based on machine learning (ML) and deep learning (DL) can deliver impressive performance in the domain of malware classification.

A. Classical ML-based Approaches

Grayscale images are extricated from raw malware executable files that display the characteristics of malware [4], [10], [11]. These images allow malware to be analysed by extracting visible features. [4] They were the first to explore and use byte plots that are displayed as grayscale images of Automatic malware classification. The author used a malware

978-1-6654-1208-7/21/$31.00 © 2021 IEEE

image dataset which consists of 9,342 malware samples belonging to 25 different classes. The author of [12] using multi-category SVM malware classification with malware input as images. Use the wavelet transform to build an efficient texture malware images vector feature. The main advantage is the reducing Characteristics of vector dimensions and time complexity. The authors [13], first features were extracted from malware files and combined support vector machines, decision trees and enhancing the performance for detection of malware. The author developed a new technique static analysis based on n-grams of opcodes which is used for classification of ransomware families [14]. They extracted GIST [15] features and classification of grayscale images using KNN classification with Euclidean distance as a metric. His technique was computationally high in general sense.

B. Deep Learning Based Solutions

Numerous studies on malware classification have been conducted using CNN architectures. [16] Malware images are classified by malware conversion files in gray-scale images, the author used two different dataset Malimg [4] and Microsoft Malware [17]. The author achieves the accuracy of 98.52% and 99.97%. The author used 90% data-set for training and 10% dataset for testing. While we tested on 30% test-data. The authors of [18] design a methodology which uses weighted soft-max loss for Convolutional Neural Network for imbalance data-set and achieves a better classification results. [19] They detect malicious code variants after changing them to grayscale images and used a simple CNN model. The authors [20] created in grayscale for one channel image binary executable files into two classes and categorize them into associated classes using lightweight. CNN. The accuracies of 94.0% and 81.8% were achieved classification of malware, respectively. [21] developed a malware classification model which is based on CNN. The author performed two different experiments. In first experiments, the author classified the malware into nine different classes and obtaining the accuracies of 96.2% and 98.4%. In second experiments, the author classified the malware into 27 different families obtained 82.9% and 89% accuracies. The authors [22] proposed a methodology by a recurrent neural network (RNN) on extracted features from the process behaviour and then train a CNN to classify the characteristics extracted by a trained RNN. [23], proposed a methodology for malware classification by using DL model based on LSTM and CNN. The author achieves an accuracy of 96.3% on the Malimg data-set. [24] design a model which consists of 3 convolutional layers followed by one fully connected layer which is tested on two data-sets, Malimg data-set and Microsoft Malware Classification Challenge data-set. In [25], the authors use Malimg malware image data-set, The authors proposed a feature fusion method to combine the features extracted from pre-trained AlexNet and Inception-v3 deep neural networks with features attained using segmentation-based fractal texture analysis (SFTA) of images representing the malware code. The proposed method achieved an accuracy of 99.3% on the cubic SVM classifier

III. Methodology and Experiments

A. System Overview

Our proposed system aims for malware classification using MalImg dataset. The flow diagram shown in Figure 1 illustrates the overall methodology. First, pre-processing is applied on the dataset, after that dataset is passed through different deep learning networks for classification. We describe each step of our proposed system in detail next section.

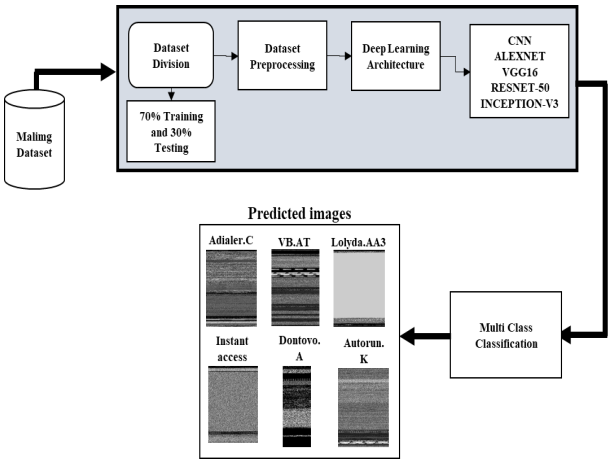


Figure 1: Proposed system workflow diagram.

B. MalImg dataset

The Malimg dataset is basically consists of 9458 malware samples which were split into 25 classes. The major feature of this dataset is that they are not providing malware samples once, but alternatively their images as they seem on disk. In a related way to the work in [26], Bytes of executable files are inconsequential assigned to floats, which will later be elucidate as pixel values of the grayscale image. Unsurprisingly, the categories in the dataset are imbalanced the largest class ('Allaple.A') contains 2949 samples, While the smallest class contains only 80 samples. In Figure 2 shows some of the random samples taken from dataset. It is clear that the images in each category have different styles that allow to distinguish between the samples of a family no matter what samples they are in another family. Dataset is illustrated in Figure 2

Dataset Division The dataset is divided into two subsets: training data and test data. From each class, 70% images are allocated for training the network while remaining 30% are allocated for testing the trained network.

C. Convolutional Neural Network (CNN)

Artificial neural network (ANN) is a group of interconnected nodes, which are inspired by a structure and functions of biological neurons in brain. The information flowing through the network affects the structure as a neural network learns input and output. Its main feature is that they can be trained through the supervised learning process. During this process, the neural networks are required for training a model using specific data containing a specific input and output

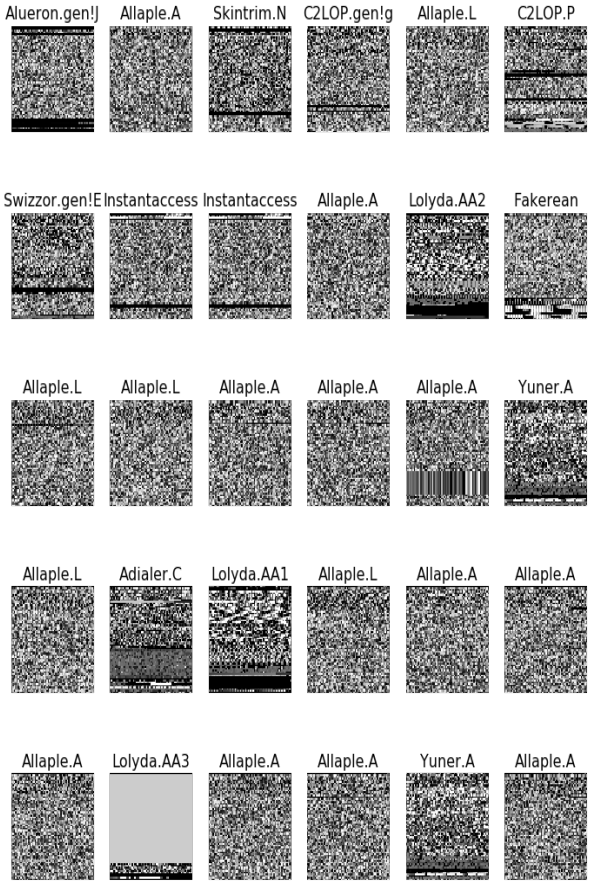


Figure 2: Samples from MalImg Dataset.

Table I: MalImg dataset

Malware Family	**Malware Kind**	**No. of Samples**
Adialer.C	Dialer	122
Agent.FYI	Backdoor	116
Allaple.A	Worm	2949
Allaple.L	Worm	1591
Alueron.gen!J	Trojan	198
Autorun.K	Worm AutoIT	106
C2LOP.gen!g	Trojan	200
C2LOP.p	Trojan	146
Dialplatform.B	Dialer	177
Donoto.A	Trojan Downloader	162
Fakerean	Rouge	381
Instaccess	Dialer	431
Lolyada.AA1	PWS	213
Lolyada.AA2	PWS	184
Lolyada.AA3	PWS	123
Lolyada.AT	PWS	159
Malex.gen!J	Trojan	136
Obfuscator.AD	Trojan Downloader	142
RBot!gen	Backdoor	158
Skintrim.N	Trojan	80
Swizzor.gen!E	Trojan Downloader	129
Swizzor.gen!I	Trojan Downloader	133
VB.AT	Worm	408
Wintrim.BX	Trojan Downloader	97
Yuner.A	Worm	800

Table II: The proposed CNN architecture.

Layer	Type	Size
0	Input layer	$224 \times 224 \times 3$
1	Convolutional+ReLu	$3 \times 3 \times 30$
2	Max-pooling	2×2
3	Convolutional+ReLu	$3 \times 3 \times 15$
4	Max-pooling	2×2
5	Dropout	25%
6	Flatten Layer	
7	Fully-Connected	128
8	Dropout	50%
9	Fully-Connected	50
10	Fully-Connected	num of classes

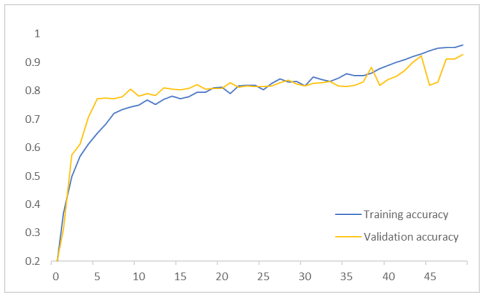
match of the system to be modeled [6]. Convolutional Neural Networks (CNN), which can be considered as a category of artificial neural networks, are especially popular these days because of its high accuracy and performance in different visual based object recognition applications [7]. CNN is made up of several convolutional layers, pooling layers, and fully connected layers. Convolution, pooling layers are consecutively integrated to build the network. The consecutive convolution and max-pooling operations create high-level characteristics in which the classification is done. The final layer is called the fully-connected layer which takes the results from convolution and pooling layer, use them for classification. In the CNN architecture, training of CNN is usually done with the back-propagation algorithm [8]. The input layer of CNN contains the pixel values of input image. The convolutional layer gives the output of neurons by product of input values and weights. The pooling layer performs down-sampling of given input image. The fully-connected layer performs classification task. The proposed CNN architecture as shown in Table II

D. Transfer Learning

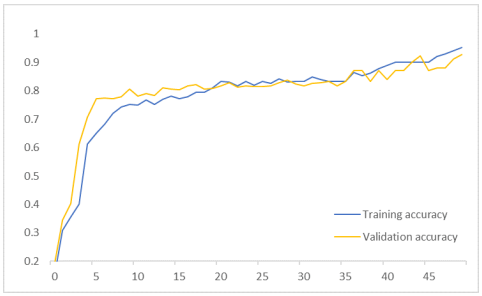
Transfer Learning is known as transfer of previously acquired knowledge from one area to another for the purpose of classification and feature extraction [9]. In the context of deep learning, transfer learning is performed using a CNN which is trained previously on a large dataset. The pre-trained CNN is used for training a new dataset that may comprise of fewer training images as compared to the earlier trained image dataset. Nowadays, the use of transfer learning gives better results because adjusting a pre-trained CNN model is generally quick and easier than building a CNN from scratch. The four architectures of CNN that we use to test our work regarding identification of malware are: (i) AlexNet [27], (ii) ResNet [28], (iii) VGG [29], and (iv) Inception V3 [30]. If transfer learning models are to be trained from scratch, they would also produce some really good results. These pre-trained models are complex networks with the concepts of branches and residual network. To get a diversity and better comparison in our results, we incorporated different experimental cases, for example, freezing the first ten pre-trained layers.

IV. RESULTS

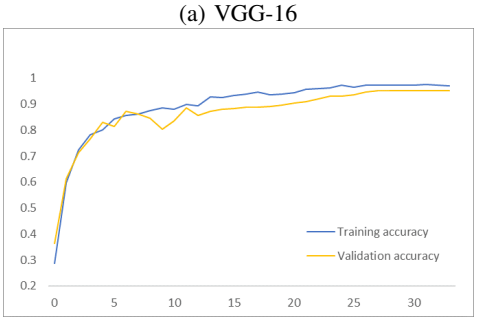
In this section, effects of the proposed techniques are analyzed. In order to obtain results, various experiments were



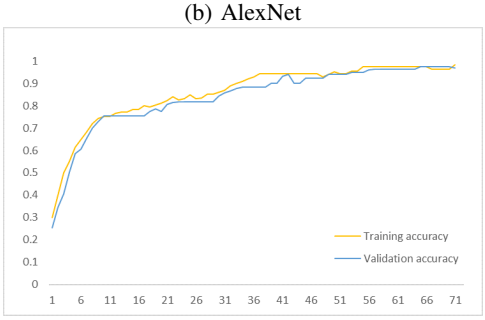
(a) VGG-16



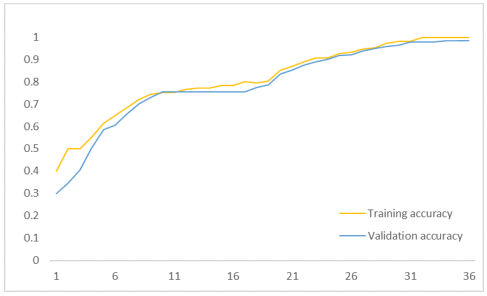
(b) AlexNet



(c) CNN



(d) ResNet-50



(e) Inception-V3

Figure 3: Training and Validation Accuracy Curves

performed. In the first experiment, training was done using all models.The data-set is divided into two subsets: training data and test data. From each class, 70% images are allocated for training the network while remaining 30% are allocated for testing the trained network. Furthermore, we can also deduce that Inception V3 performs much better than other models obtaining the accuracy of 98.90%. The performance of each classifier is evaluated in terms of accuracy. A good Malware classification should achieve high level of accuracy. Accuracy of the classifier is calculated by

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN}$$

$$\text{Precision} = \frac{TP}{TP+FP}$$

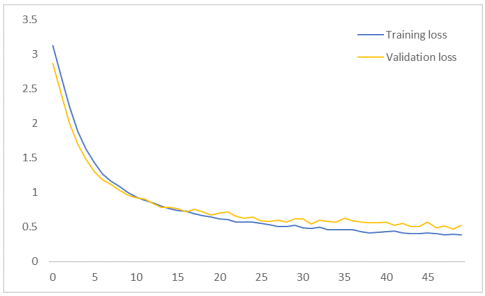
$$\text{Recall} = \frac{TP}{TP+FN}$$

$$\text{F1} = \frac{2*Precision*Recall}{Precision+Recall} = \frac{2*TP}{2*TP+FP+FN}$$

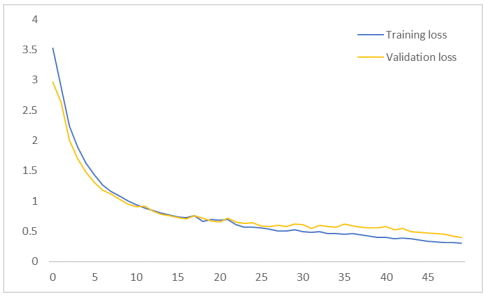
The true positive (TP) is defined as the the number of successful attack records class; The real negatives (TN) are the amount of normal traffic properly classified records; False Positives (FP) is a number Falsely categorized normal traffic logs and false negative (FN) logs The number of instances of misclassified attack logs.

A custom design CNN and transfer learning models were used for training purposes. From table III, we can see that the models trained on the Malimag dataset performed significantly better on all the networks. Furthermore, we can also deduce that Inception v3 performs much better than other models obtaining the accuracy 98.90%. Inception-v3 performed way better than the other networks because of its complex architecture and an enormous number of learnable parameters. Also due to batch-Normalization and using multiple features from multiple filters improve the performance of the network.

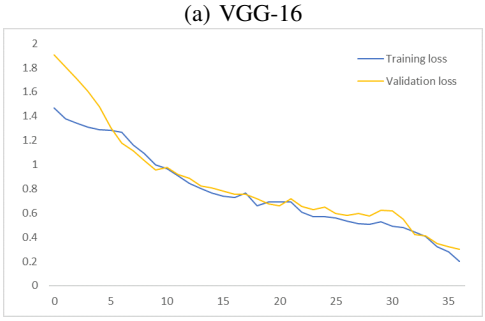
Our proposed system is compared with some Malware classification included in Refs. [31]–[35] as shown in Table IV. Results proved that our proposed system outperformed other related Malware classification with a better classification accuracy



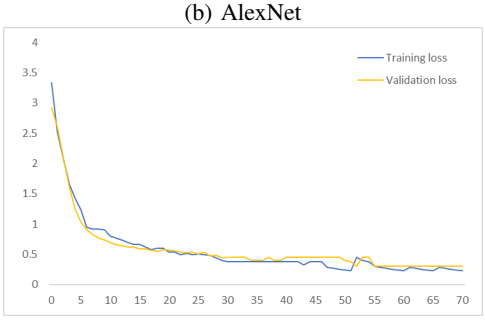
(a) VGG-16



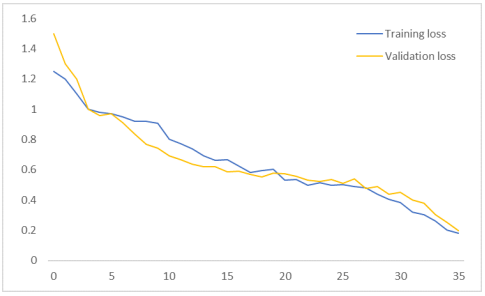
(b) AlexNet



(c) CNN



(d) ResNet-50



(e) Inception-V3

Figure 4: Training and Validation Loss Curves

Table III: Model accuracy and F1-Score on MalImg Dataset.

Model	Test Accuracy	Precision	Recall	F1-Score
CNN	95.78%	0.9571	0.9570	0.9572
ResNet-50	97.78%	0.9776	0.9772	0.9774
AlexNet	93.78%	0.9375	0.9370	0.9373
VGG-16	92.78%	0.9268	0.9270	0.9277
Inception-V3	98.90%	0.9858	0.9879	0.9889

Table IV: A Comparison of Different Malware Classification.

	Malware Classification				
Method	Proposed	[31]	[32]	[34]	[36]
Accuracy	98.90%	80.46%	95.26%	97%	97.35%

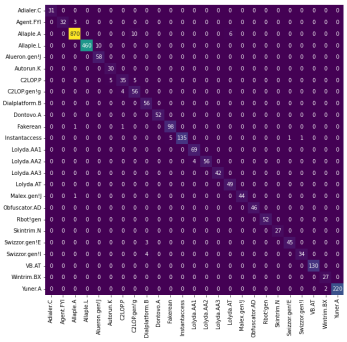
V. Conclusions

Today, many programs consist of antivirus depend on DL techniques to secure devices from malware. DL architectures have performed well in Malware detection and classification. We have presented the performance comparison among five classifiers on a malware image dataset. We used the different Convolutional Neural Network (CNN) architectures—including a custom CNN trained from scratch as well as commodity off-the-shelf CNN architectures such as AlexNet, VGG-16, ResNet-50, Inceptionv3 models to classify grayscale malware dataset. The results shows that the Inception-V3 model gives most promising result over all models. An accuracy 98.90% has been achieved. We hope that our proposed system contributes significantly to malware classification.

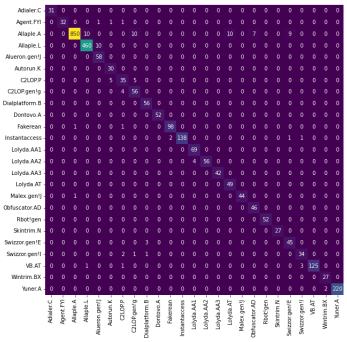
Future work will focus on achieving results using other models like Xception, Inception-ResNets and ResNeXt-50 for classifying photos with Ensemble classification algorithm for classification of malware images. We also tried to convert malware images to RGB color images before malware categorizing.

References

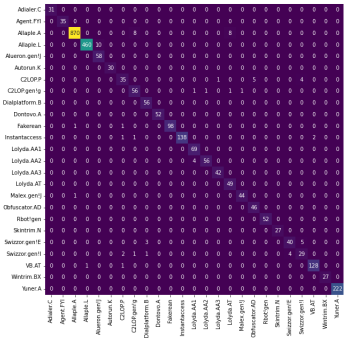
1 Debar, H., Dacier, M., and Wespi, A., "Towards a taxonomy of intrusion-detection systems," *Computer networks*, vol. 31, no. 8, pp. 805–822, 1999.
2 Kolosnjaji, B., Zarras, A., Webster, G., and Eckert, C., "Deep learning for classification of malware system call sequences," in *Australasian Joint Conference on Artificial Intelligence*. Springer, 2016, pp. 137–149.
3 Shabtai, A., Moskovitch, R., Elovici, Y., and Glezer, C., "Detection of malicious code by applying machine learning classifiers on static features: A state-of-the-art survey," *information security technical report*, vol. 14, no. 1, pp. 16–29, 2009.



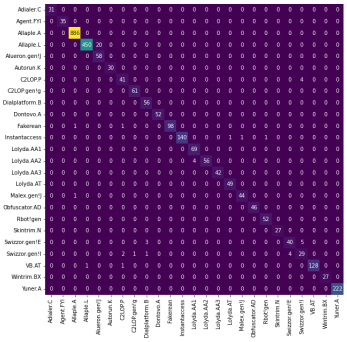
(a) VGG-16



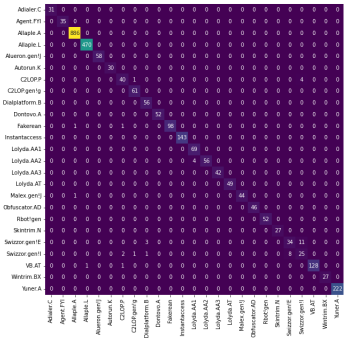
(b) AlexNet



(c) CNN



(d) ResNet-50



(e) Inception-V3

Figure 5: Confusion Matrix of Trained Models

4 Nataraj, L., Karthikeyan, S., Jacob, G., and Manjunath, B. S., "Malware images: visualization and automatic classification," in *Proceedings of the 8th international symposium on visualization for cyber security*, 2011, pp. 1–7.

5 Domingos, P. and Pazzani, M., "On the optimality of the simple bayesian classifier under zero-one loss," *Machine learning*, vol. 29, no. 2, pp. 103–130, 1997.

6 Gianfelici, F., "Nearest-neighbor methods in learning and vision (shakhnarovich, g. et al., eds.; 2006)[book review]," *IEEE Transactions on Neural Networks*, vol. 19, no. 2, pp. 377–377, 2008.

7 Keerthi, S. S. and Gilbert, E. G., "Convergence of a generalized smo algorithm for svm classifier design," *Machine Learning*, vol. 46, no. 1, pp. 351–360, 2002.

8 Liaw, A., Wiener, M. *et al.*, "Classification and regression by randomforest," *R news*, vol. 2, no. 3, pp. 18–22, 2002.

9 Quinlan, J. R., "Induction of decision trees," *Machine learning*, vol. 1, no. 1, pp. 81–106, 1986.

10 Nataraj, L., Yegneswaran, V., Porras, P., and Zhang, J., "A comparative assessment of malware classification using binary texture analysis and dynamic analysis," in *Proceedings of the 4th ACM Workshop on Security and Artificial Intelligence*, 2011, pp. 21–30.

11 Kosmidis, K. and Kalloniatis, C., "Machine learning and images for malware detection and classification," in *Proceedings of the 21st Pan-Hellenic Conference on Informatics*, 2017, pp. 1–6.

12 Makandar, A. and Patrot, A., "Malware class recognition using image processing techniques," in *2017 International Conference on Data Management, Analytics and Innovation (ICDMAI)*. IEEE, 2017, pp. 76–80.

13 Mirza, Q. K. A., Awan, I., and Younas, M., "Cloudintell: An intelligent malware detection system," *Future Generation Computer Systems*, vol. 86, pp. 1042–1053, 2018.

14 Zhang, H., Xiao, X., Mercaldo, F., Ni, S., Martinelli, F., and Sangaiah, A. K., "Classification of ransomware families with machine learning

based on n-gram of opcodes," *Future Generation Computer Systems*, vol. 90, pp. 211–221, 2019.
15 Torralba, A., Murphy, K. P., Freeman, W. T., and Rubin, M. A., "Context-based vision system for place and object recognition," in *Computer Vision, IEEE International Conference on*, vol. 2. IEEE Computer Society, 2003, pp. 273–273.
16 Kalash, M., Rochan, M., Mohammed, N., Bruce, N. D., Wang, Y., and Iqbal, F., "Malware classification with deep convolutional neural networks," in *2018 9th IFIP international conference on new technologies, mobility and security (NTMS)*. IEEE, 2018, pp. 1–5.
17 Ronen, R., Radu, M., Feuerstein, C., Yom-Tov, E., and Ahmadi, M., "Microsoft malware classification challenge," *arXiv preprint arXiv:1802.10135*, 2018.
18 Yue, S., "Imbalanced malware images classification: a cnn based approach," *arXiv preprint arXiv:1708.08042*, 2017.
19 Cui, Z., Xue, F., Cai, X., Cao, Y., Wang, G.-g., and Chen, J., "Detection of malicious code variants based on deep learning," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 7, pp. 3187–3196, 2018.
20 Su, J., Vasconcellos, D. V., Prasad, S., Sgandurra, D., Feng, Y., and Sakurai, K., "Lightweight classification of iot malware based on image recognition," in *2018 IEEE 42Nd annual computer software and applications conference (COMPSAC)*, vol. 2. IEEE, 2018, pp. 664–669.
21 Seok, S. and Kim, H., "Visualized malware classification based-on convolutional neural network," *Journal of The Korea Institute of Information Security & Cryptology*, vol. 26, no. 1, pp. 197–208, 2016.
22 Tobiyama, S., Yamaguchi, Y., Shimada, H., Ikuse, T., and Yagi, T., "Malware detection with deep neural network using process behavior," in *2016 IEEE 40th annual computer software and applications conference (COMPSAC)*, vol. 2. IEEE, 2016, pp. 577–582.
23 Vinayakumar, R., Alazab, M., Soman, K., Poornachandran, P., and Venkatraman, S., "Robust intelligent malware detection using deep learning," *IEEE Access*, vol. 7, pp. 46 717–46 738, 2019.
24 Gibert, D., Mateu, C., Planes, J., and Vicens, R., "Using convolutional neural networks for classification of malware represented as images," *Journal of Computer Virology and Hacking Techniques*, vol. 15, no. 1, pp. 15–28, 2019.
25 Nisa, M., Shah, J. H., Kanwal, S., Raza, M., Khan, M. A., Damaševičius, R., and Blažauskas, T., "Hybrid malware classification method using segmentation-based fractal texture analysis and deep convolution neural network features," *Applied Sciences*, vol. 10, no. 14, p. 4966, 2020.
26 Marastoni, N., Giacobazzi, R., and Dalla Preda, M., "Data augmentation and transfer learning to classify malware images in a deep learning context," *Journal of Computer Virology and Hacking Techniques*, pp. 1–19, 2021.
27 Krizhevsky, A., Sutskever, I., and Hinton, G. E., "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems*, vol. 25, pp. 1097–1105, 2012.
28 He, K., Zhang, X., Ren, S., and Sun, J., "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.
29 Simonyan, K. and Zisserman, A., "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.
30 Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., and Wojna, Z., "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2818–2826.
31 Agarap, A. F., "Towards building an intelligent anti-malware system: a deep learning approach using support vector machine (svm) for malware classification," *arXiv preprint arXiv:1801.00318*, 2017.
32 Garcia, F. C. C. and Muga II, F. P., "Random forest for malware classification," *arXiv preprint arXiv:1609.07770*, 2016.
33 Ghosh, P., Mandal, A. K., and Kumar, R., "An efficient cloud network intrusion detection system," in *Information systems design and intelligent applications*. Springer, 2015, pp. 91–99.
34 Kumari, M., Hsieh, G., and Okonkwo, C. A., "Deep learning approach to malware multi-class classification using image processing techniques," in *2017 International Conference on Computational Science and Computational Intelligence (CSCI)*. IEEE, 2017, pp. 13–18.
35 Singh, A., Handa, A., Kumar, N., and Shukla, S. K., "Malware classification using image representation," in *International Symposium on Cyber Security Cryptography and Machine Learning*. Springer, 2019, pp. 75–92.
36 Kim, H.-J., "Image-based malware classification using convolutional neural network," in *Advances in computer science and ubiquitous computing*. Springer, 2017, pp. 1352–1357.