Fuzzy Systems and Soft Computing
ISSN : 1819-4362

MALWIN: CLASSIFICATION AND DETECTION OF DOWNLOADED MALWARE DATASET USING DEEP LEARNING

Divya S Assistant Professor Department of Computer Science and Engineering Sree Sastha Institute of Engineering and Technology Chembarambakkam, Tamil Nadu 600123, INDIA divya.cse@ssiet.in

Abstract
As the risk of malware attacks continues to grow, creating a reliable malware detection system is crucial for identifying malware quickly. Advanced deep learning methods have been successful in categorizing complex malware from standard datasets. A significant issue with conventional deep learning classifiers is the requirement to retrain the classifier whenever a new malware family is discovered. In this paper, we propose a model for classifying malware based on a downloaded windows Malware dataset Malwin using Convolutional Neural Network (CNN) classifiers. We obtain great accuracy in malware classification using our CNN classifier. Our results demonstrate the great accuracy with which our model can categorize malware families.

Keywords : Malware Classification, Convolutional Neural Network (CNN), Malware Datasets, Detection, Algorithm

1 Introduction

Malware, short for malicious software, refers to programs designed to perform any kind of unwanted or harmful action on a computer system. These activities are intended to damage or alter a compromised system, as well as intrude on and extract important and private data. These days, networks of developers and criminal organizations are involved in the complex and expanding activity of malware. It is a worldwide sector that is expanding annually. Over 600 million harmful applications were found in the initial half of 2017, as reported by McAfee [1]. This paper tests and validates a downloaded Malwin dataset using CNN deep learning classifiers.

2 Malware Detection

Static and dynamic detection are the two primary methodologies used in malware analysis. Examining a program's code or structure without running it is known as static analysis. This type of analysis can generate a basic set of patterns, reveal functionality details, and determine whether a file is dangerous. In dynamic analysis, the software is executed and the system's behavior is observed. In contrast to static analysis, dynamic analysis enables us to see the actual operations that the program carries out. It is usually used after all other static analysis techniques have been used up or when obfuscation and wrapping have rendered static analysis ineffective. M. Egele et al.[2] provide a survey of streamlined dynamic analysis methods and resources.

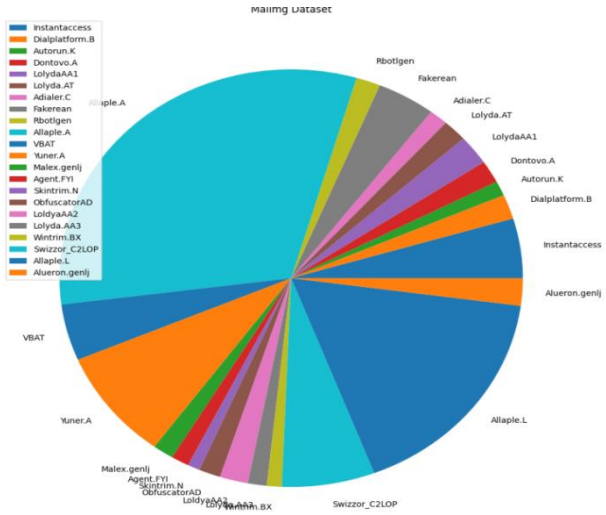
3 Dataset

The dataset consists of known malware files representing a mix of different families. Each malware file has an ID, a 20-character hash value uniquely identifying the file, and a Class, an integer representing family names to which the malware may belong. Malware Classification is done by converting PE files to byteplot images. The aim of the dataset is:

- Multiclass Classification of Malware Byteplot images.
- Managing a dataset containing both RGB and Grayscale byteplot images.

3.1. Malimg dataset (Dataset-A)

Nataraj et al.[3] introduced this dataset intending to explore signal and image processing techniques in the field of malware classification, and researchers have used it as a benchmark in various state-of-the-art research. The dataset comprises 9,339 malware samples, categorized into 22 malware families, all represented as grayscale images.



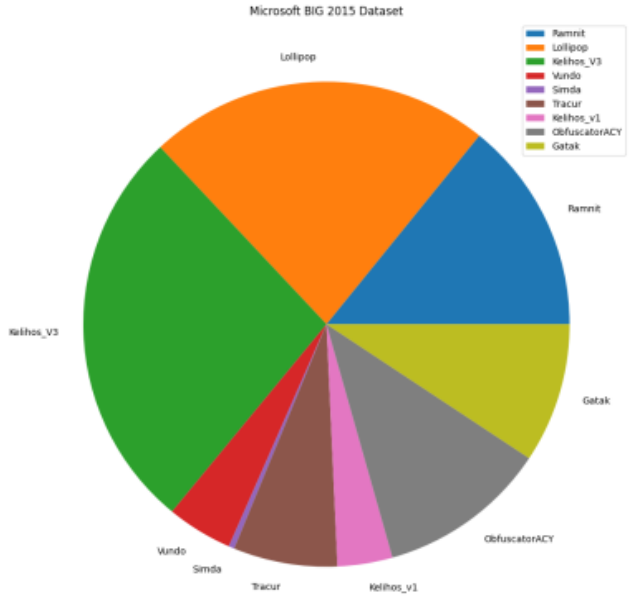
Malimg Dataset.

Family code	Family	Number of samples
A1	Instantaccess	431
A2	Dialplatform.B	177
A3	Autorun.K	106
A4	Dontovo.A	162
A5	Lolyda.AA1	213
A6	Lolyda.AT	159
A7	Adialer.C	122
A8	Fakerean	381
A9	Rbot!gen	158
A10	Allaple.A	2949
A11	VB.AT	408
A12	Yuner.A	800
A13	Malex.gen!J	136
A14	Agent.FYI	116
A15	Skintrim.N	80
A16	Obfuscator.AD	142
A17	Lolyda.AA2	184
A18	Lolyda.AA3	123
A19	Wintrim.BX	97
A20	Swizzor_C2LOP	606
A21	Allaple.L	1591
A22	Alueron.gen!J	198

Fig 1. Malicious Malware Family belonging to Mailing Dataset

3.2 Microsoft BIG 2015 dataset (Dataset-B)

This dataset was introduced by Microsoft and was part of a Kaggle competition hosted by Microsoft Ronen et al.[4] in 2015. It is a massive dataset consisting of 10868 malware samples distributed among 9 malware families, represented in the form of grayscale images. Note that in Figure 3, the images of malware belonging to the same family are similar while distinct from the images of malware from the rest of the family.



Microsoft BIG 2015 Dataset.

Family code	Family	Number of samples
B1	Ramnit	1541
B2	Lollipop	2478
B3	Kelihos_v3	2942
B4	Vundo	475
B5	Simda	42
B6	Tracur	751
B7	Kelihos_v1	398
B8	Obfuscator.ACY	1228
B9	Gatak	1013

Fig 2. Malicious Malware Family belonging to Microsoft BIG 2015 Dataset

.

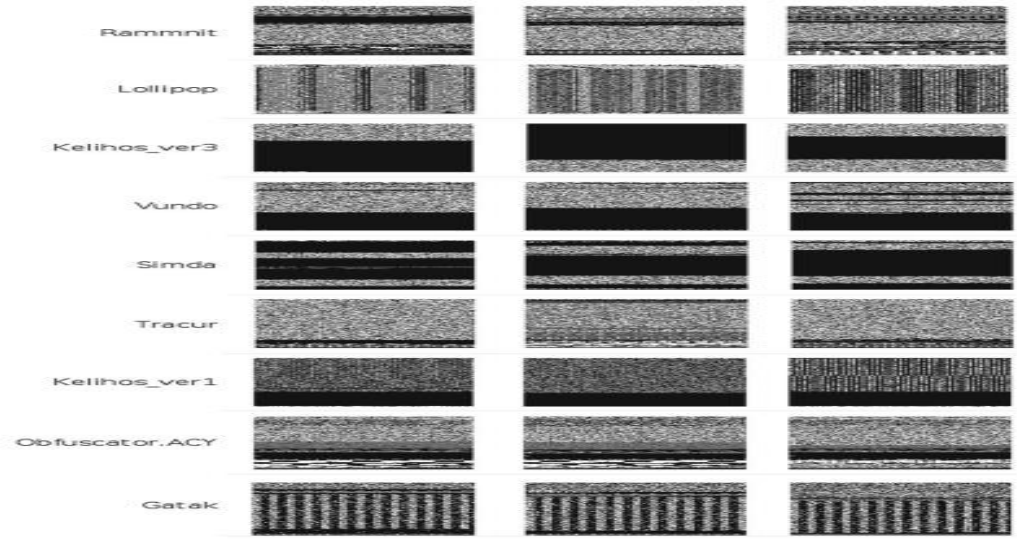
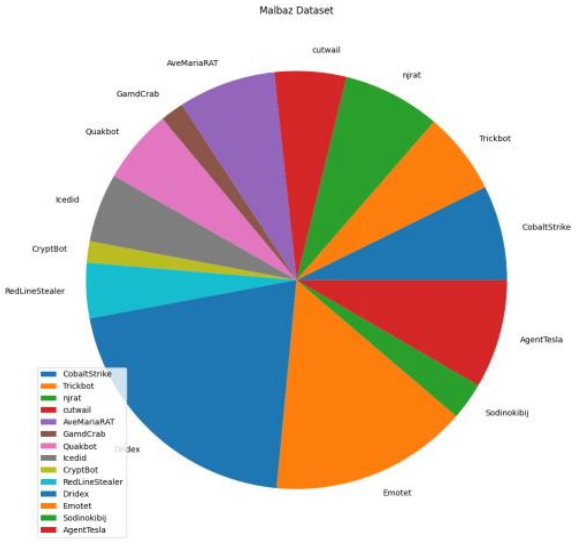


Fig 3. Gray Scale Images of Malicious Malware belonging to various families.

3.3 Malbaz dataset (Dataset-C)

Malware is frequently evolving which requires a malware classifier to possess the ability to classify recently discovered malware. We gathered malware executables from the public repository MalwareBazaar. Using the MalwareBazaar API, we compiled a dataset of 8,076 samples, distributed across 14 malware families.



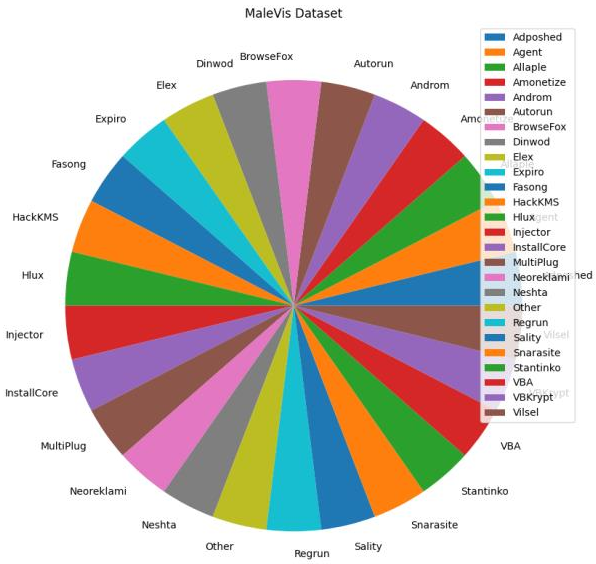
Malbaz Dataset.

Family code	Family	Number of samples
C1	CobaltStrike	592
C2	Trickbot	513
C3	njrat	606
C4	Cutwail	445
C5	AveMariaRAT	604
C6	GandCrab	146
C7	Quakbot	462
C8	IcedId	430
C9	CryptBot	133
C10	RedLineStealer	345
C11	Dridex	1659
C12	Emotet	1229
C13	Sodinokibi	233
C14	AgentTesla	679

Fig 4. Malicious Malware Family belonging to Malbaz Dataset

3.4 **Malevis dataset *(Dataset-D)***

The MaleVis (Malware Evaluation with Vision) dataset[5] was utilized to gauge the effectiveness of the proposed method. The MaleVis dataset includes 9,100 RGB byte images categorized into 26 malware classes. The Malware classes included Adposhel, Agent-fyi, and Allaple. A, Amonetize, Androm, AutoRun-PU, BrowseFox, Dinwod! rfn, Elex, Expiro-H, Fasong, HackKMS. A, Hlux! IK, Injector, InstallCore. C, MultiPlug, Neorekla-mi, Neshta, Regrun. A, Sality, Snarasite. D!tr, Stantinko, VBA/Hilium. A, VBKrypt, and Vilsel. The distribution of samples among the different malware classes contained in the dataset is 350 images total throughout all classes, which are evenly distributed. The image resolutions range between 224 × 224 and 300 × 300 pixels.



Class ID	Family	Details	
		Malware Category	Sample No.
#1	Adposhel	Adware	350
#2	Agent	Trojan	350
#3	Allaple	Worm	350
#4	Amonetize	Adware	350
#5	Androm	Backdoor	350
#6	Autorun	Worm	350
#7	BrowseFox	Adware	350
#8	Dinwod	Trojan	350
#9	Elex	Trojan	350
#10	Expiro	Virus	350
#11	Fasong	Trojan	350
#12	HackKMS	Riskware	350
#13	Hlux	Worm	350
#14	Injector	Trojan	350
#15	InstallCore	Adware	350
#16	MultiPlug	Adware	350
#17	Neoreklami	Adware	350
#18	Neshta	Virus	350
#19	Other	-	350
#20	Regrun	Trojan	350
#21	Sality	Virus	350
#22	Snarasite	Trojan	350
#23	Stantinko	Trojan	350
#24	VBA	Macro Malwares	350
#25	VBKrypt	Trojan	350
#26	Vilsel	Trojan	350
	Total	-	9100

Fig 5. Malicious Malware Family belonging to Malevis Dataset

4. Classification Models

The scarcity of malware samples for malware families is a major problem when it comes to the domain of malware classification by Wang et al.[6]. Training a conventional malware classifier necessitates a substantial amount of data. Another limitation of traditional Deep-learning models is that they can only classify an instance into one of the classes the model was trained on. For the model to classify data from a previously unknown class, it would require re-training using a significant amount of data from this new class. To address these issues, we employ the following models

- Convolutional neural networks (CNNs)
- Convolutional Siamese Neural Network (CSNN)
- Shallow Few-Shot (Shallow-FS).
- Naïve Bayes Classifier
- Random Forest Classifier
- Decision Tree Classifier
- Linear Support Vector Machines Classifier

5. Result and Discussion

A Convolutional Neural Network (CNN) is a specialized deep learning model for processing malware data. It comprises two primary types of layers: the convolutional layer, which emphasizes small details, and the pooling layer, which simplifies information to capture the overall context. Our downloaded Malwin Database consists of malware and benign classification with about 100000 rows and 35 columns of attributes in Fig 6. A heatmap is a graphical representation of data that uses colors to visualize the values within a matrix. Brighter and reddish colors are used to indicate more common values or higher activity levels, while darker colors represent less common values or lower activity levels. Heatmaps are also referred to as shading matrices. The Heatmap Matrix for malicious malware detection is shown in Fig 7. We train the model on the training data for 100 epochs, processing 8 batches of training data before advancing to the next epoch. We then demonstrate the model's predictive capabilities by evaluating it on an unseen image. The outcome was a binary classification result, indicating whether the malware was detected or not. 80% of the Malwin database is used for testing purposes and the balance 20% of the Malwin database is used for validation purposes using the CNN classifier. Our model classifier achieved a test accuracy of 99.954998% and a test loss of 0.001768 %. Table 1 shows the detailed explanation of Model Calculation for Parameters and Output Shapes such that the total params are 336,642 (1.28 MB), Trainable params are 336,642 (1.28 MB) and Non-trainable params are 0 (0.00 B).

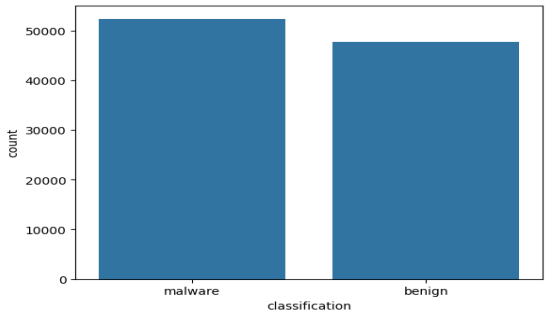


Fig 6. Malicious Malware and benign Families belonging to the Malwin Dataset

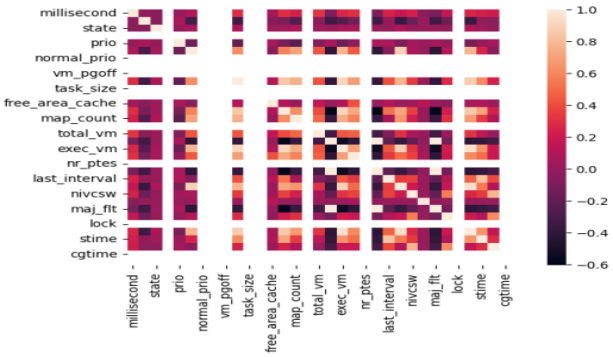


Fig 7. Heatmap Matrix of Malicious Malware and benign Families belonging to Malwin Dataset

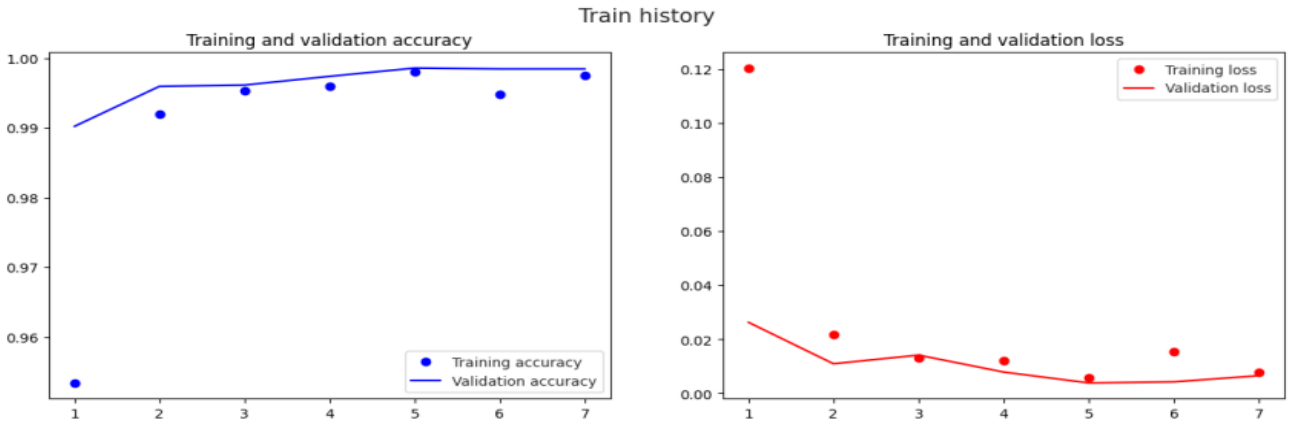


Fig 8. Training and Validation Accuracy and Loss Malicious Malware and Benign Families

Table 1: Detailed Explanation of Model Calculation for Parameters and Output Shapes

Layer (type)	Output Shape	Param #
dense_7 (Dense)	(None, 256)	7,168
dense_8 (Dense)	(None, 256)	65,792
dense_9 (Dense)	(None, 256)	65,792
dense_10 (Dense)	(None, 256)	65,792
dense_11 (Dense)	(None, 256)	65,792
dense_12 (Dense)	(None, 256)	65,792
dense_13 (Dense)	(None, 256)	514

6. Conclusion

Our model CNN classifier achieved a test accuracy of 99.954998%. We successfully built a Convolutional Neural Network (CNN) for binary image classification of malware using TensorFlow. This paper serves as a foundational guide for image classification with CNNs, leaving room for further exploration and refinement in the dynamic field of computer vision.

References

1. LLC, M.: Mcafee Labs threats report. https://www.mcafee. com/us/resources/reports/rp-quarterly-threats-jun-2017.pdf. Accessed 20 Sept (2017)

2. Egele, M., Scholte, T., Kirda, E., Kruegel, C.: A survey on automated dynamic malware-analysis techniques and tools. ACM Comput. Surv. 44(2), 6:1–6:42. https://doi.org/10.1145/2089125.2089126 (2008)
3. Nataraj, L., Karthikeyan, S., Jacob, G., Manjunath, B.S., Malware images: Visualization and automatic classification In: Proceedings of the 8th International Symposium on Visualization for Cyber Security, pp. 1-7. (2011)
4. Ronen, R., Radu, M., Feuerstein, C., Yom-Tov, E., Ahmadi, M., Microsoft malware classification challenge, abs/1802.10135 (2021)
5. Bozkir, A.S.; Cankaya, A.O.; Aydos, M. Utilization and comparison of convolutional neural networks in malware recognition. In Proceedings of the 2019 27th Signal Processing and Communications Applications Conference (SIU), Sivas, Turkey, 24–26 April 2019; pp. 1–4 (2009)
6. Wang, P., Tang, Z., Wang, J., A Novel few-shot malware classification approach for unknown family recognition with multi-prototype modeling, Comput. Secur. 106, 102273 (2021)