PROJECT REPORT

on

IoT Malware Detection Using CNN and Deep Learning: A Comparative Study

(CSE V Semester Mini project)

2023-2024

Submitted to:

Mr. Nitin Thapliyal

(CC-CSE-B-V-Sem)

Guided by:

Mr. Umang Garg

Submitted by:

Mr. Siddharth Singh Rana

Roll. No:21011885

CSE-D-V-Sem

Session: 2023-2024

DEPARTMENT OF COMPUTER SCIENCE AND INFORMATION TECHNOLOGY

GRAPHIC ERA HILL UNVERSITY, DEHRADUN

CERTIFICATE

Certified that Mr. Siddharth Singh Rana (Roll No.21011885) has developed mini project and Research paper on “IoT Malware Detection Using CNN and Deep Learning: A Comparative Study” for the CS V Semester Mini Project Lab in Graphic Era Hill University, Dehradun. The project carried out by Students is their own work to the best of my knowledge.

Date: 1/11/2023

Mr. Nitin Thapliyal

Class Coordinator

(CSE Department)

GEHU Dehradun

ACKNOWLEDGMENT

I extend my sincere appreciation to my supervisor, Umang Garg, for their invaluable guidance and support throughout this research on "IoT Malware Detection using CNN and VGG16." Their expertise has been pivotal in shaping this study.

I am thankful to Graphic Era Hill University, Dehradun for providing the necessary resources and a conducive academic environment that facilitated the completion of this research.

I appreciate the authors of datasets, research papers, and tools utilized in this study, as their contributions laid the foundation for our research.

Lastly, I express gratitude to my family and friends for their unwavering support throughout this research journey.

This research would not have been possible without the collective effort and support of all mentioned above. Thank you for being an integral part of this academic endeavor.

Mr. Siddharth Singh Rana

Roll No.- 2119247

CSE-B-V-Sem

Session: 2023-2024

GEHU, Dehradun

TABLE OF CONTENTS

Topic	Page
1. Introduction:	5
2. Dataset:	6
3. Methods:	7
4. Architecture:	8
5. Results and Interpretation:	11
6. Conclusion:	13
7. References:	15

Introduction

The escalating ubiquity of IoT-enabled devices in contemporary society has significantly transformed daily life, albeit accompanied by a parallel surge in cyber threats targeting these devices. In response to the escalating risks, this research paper delves into innovative methodologies for detecting malware attacks on IoT devices, with a particular focus on leveraging deep learning techniques. Our investigation centers around the application of Convolutional Neural Networks (CNNs), specifically exploring the efficacy of VGG16—an established CNN architecture for image classification tasks.

As part of our comparative analysis, we assess the performance of VGG16 against other noteworthy CNN architectures, including ResNet50 and ResNet101. Introduced by He et al., these architectures incorporate residual learning to address challenges associated with deep neural networks. Additionally, we explore VGG19, an extended iteration of VGG16 lauded for its exceptional performance in image classification tasks.

This research aims to offer a comprehensive evaluation of image classification models for the purpose of detecting malware in IoT environments. The subsequent sections of the paper meticulously delve into the existing literature, provide insights into the dataset, detail the methodologies employed in the study, and propose an architecture designed for optimal performance. Through rigorous analysis in the "Results and Discussion" section, we seek to identify the most effective model with optimal

parameters, ultimately contributing valuable insights to enhance IoT device security against cyber threats. The research concludes with a concise summary in the conclusive section.

Dataset

The dataset used in this study is called the "IOT Malware dataset for classification." It contains images of binary files specifically designed for classifying malware in IoT devices [24]. The dataset is organized into two folders: "benign" and "malware," with 2,486 images in the benign folder and 14,733 in the malware folder, totaling 17,219 images. The dataset is quite large, exceeding 600 megabytes.

The images in the dataset are labeled as either benign or malware, making it suitable for supervised learning methods like Convolutional Neural Networks (CNNs) and VGG16, commonly used for image classification tasks. This dataset is featured in the research paper on "IoT malware analysis using CNN and VGG16," providing an opportunity to explore how well these models can detect IoT malware. The dataset's size and quality make it an ideal choice for this purpose. Moreover, it can be used in future research on IoT malware analysis and classification using machine learning techniques.

To organize the dataset for training, testing, and validation, it was divided into three parts with a split of 10,000 images for training, 800 for testing, and 800 for validation, following a tripartite partitioning approach.

Methods

• ResNet50:

ResNet50 is a type of deep neural network designed to tackle the vanishing gradient problem in deep learning. It introduces residual connections, enabling gradients to flow directly from the output to the input of a layer. This helps the network learn features at various levels, enhancing accuracy in image recognition tasks. The architecture comprises 50 layers, including convolutional layers, batch normalization, ReLU activation, and a unique residual block. The residual connection equation is yl + 1 = F(xl) + xl, allowing for deeper and more accurate networks by addressing the vanishing gradient problem.

Other key equations in ResNet50 include the convolutional equation $y = W * x + b$, the batch normalization equation $y = (x - E[x]) / \sqrt{(Var[x] + \epsilon)} * \gamma + \beta$, and the ReLU activation function $y = \max(0, x)$.

• VGG16:

VGG16 is a deep convolutional neural network known for its success in image classification tasks. It achieves depth by stacking multiple convolutional layers with small 3x3 kernel sizes and max-pooling layers for feature map downsampling. The mathematical representation involves the output y, input x, weight matrix W, bias vector b, and activation function f(W x + b), with ReLU as the activation function.

Additional equations in VGG16 include the loss function L(y, yˆ) = - Σ yi log(ˆyi), where yi is the ground truth label, ˆyi is the predicted label, C is the number of classes, and i is the class index. The update equation for stochastic gradient descent is X = X - α ∂Y / ∂X, with α as the learning rate.

• VGG19:

VGG19, developed in 2014 at the University of Oxford, is a CNN architecture known for its performance on ImageNet classification. With 19 layers, including 16 convolutional and 3 fully connected layers, it groups convolutional layers into blocks with max-pooling. The mathematical equation involves convolution operation (a ∗ b(T) = Z ∞ -∞ a(τ)b(T − τ)dτ), ReLU activation function, and softmax function for multi-class classification.

The softmax function ensures predicted probabilities sum up to 1, generating a probability distribution over class labels (Pi = ezi / Σ ezi, where Pi is predicted probability, zi is the score, and K is the total number of classes).

Architecture

The method for designing IoT malware detection using CNN and VGG16 involves a few steps. In Figure 1, you can see images of different binary files, some with malware and some without. Figure 2 shows the architecture diagram.

First, we gather a big dataset of binary files that have both malware and non-malware samples. Then, we split the dataset into training, testing, and validation sets to check how well the model works.

Next, we use transfer learning to train three models: ResNet50, VGG16, and VGG19. Among these, VGG16 gives the best results [15] [16].

After picking the model, we carefully choose the best optimizer. In this study, the Adam optimizer works the best for its excellent performance [22].

Then, we test the system thoroughly by using the trained model to correctly identify malware and non-malware binary files [10].

In short, our proposed method for IoT malware detection with CNN and VGG16 involves gathering data, splitting it, training models, picking the best one, choosing the right optimizer, and testing it thoroughly. We found that VGG16 and the Adam optimizer work the best. This system helps improve IoT security by detecting and fighting against malware threats.

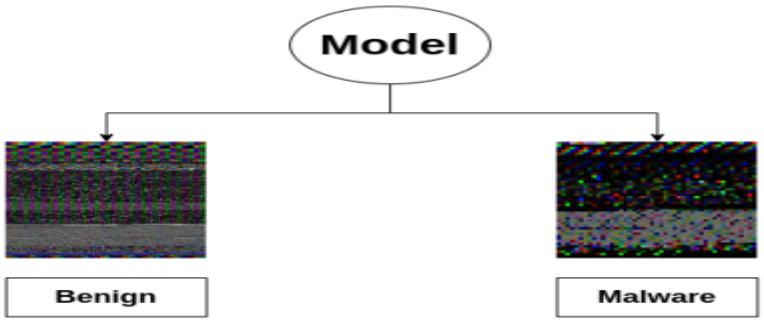

Fig. 1. Predictive Model

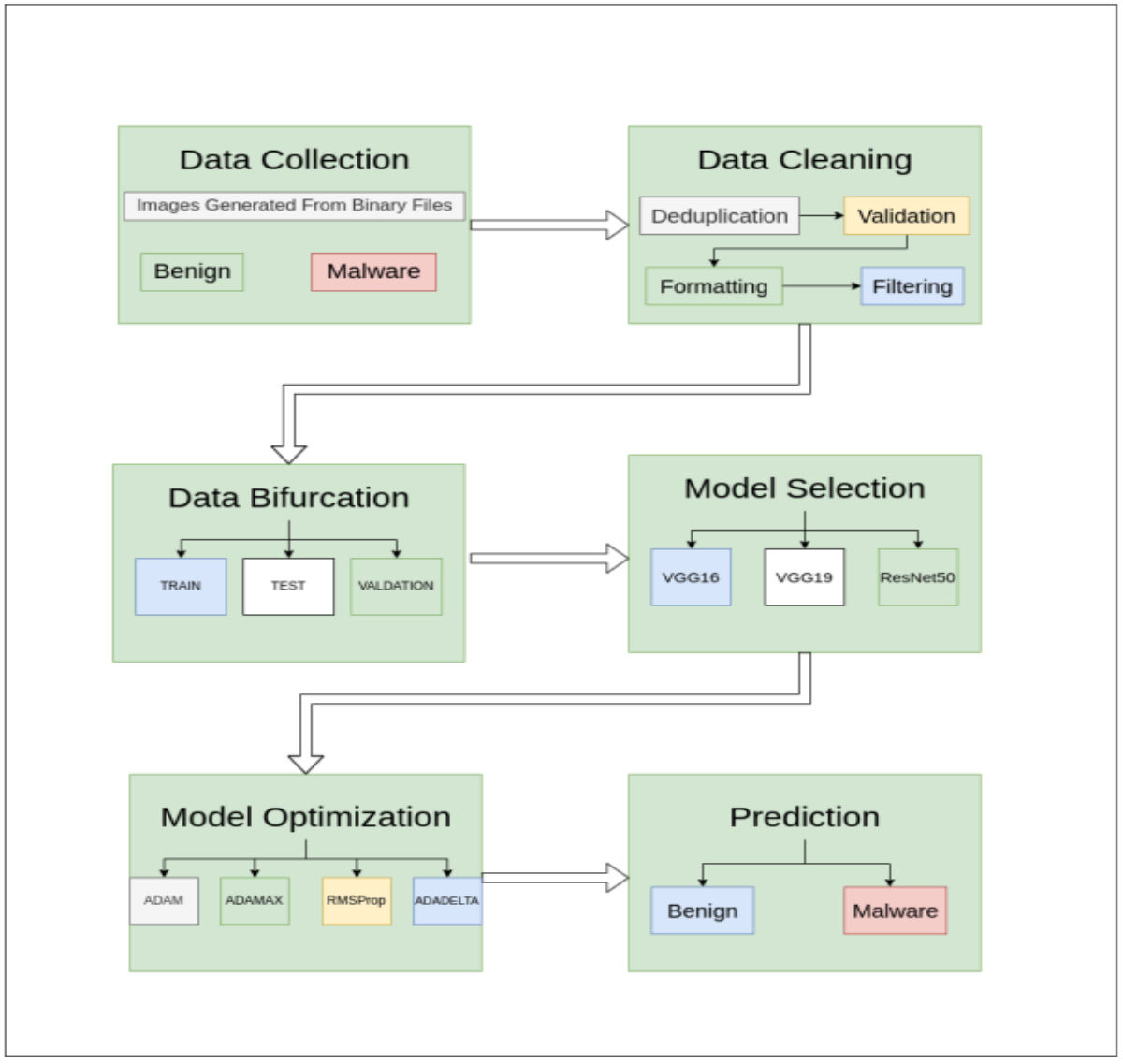

Fig. 2. Proposed Architecture

Result and Interpretation

To assess three different models, we considered accuracy, time delay, recall, standard deviation, precision, and f1 score [23]. While accuracy is commonly used, other metrics like precision and recall are crucial for binary classification. Precision measures correct positive predictions, while recall gauges true positives. The F1 score combines precision and recall. Time delay and standard deviation also matter.

Looking at the results in Table I, VGG16 outperformed other models, suggesting its potential for effective use in related research.

Experiment 1:

- Tested ResNet50 with a decent 89.4% accuracy but may not be the best for IoT malware detection due to limited layers.

Experiment 2:

- Used VGG19 with a significant 94.6% accuracy, indicating its capability in detecting IoT malware effectively.

Experiment 3:

- Tested VGG16, yielding a 94.9% accuracy, making it the most effective model among the three. Its deep architecture and efficient use of convolutional layers contribute to high accuracy.

Table II shows optimizer performance on VGG16, with the Adam optimizer producing the highest accuracy. This emphasizes the importance of optimizer selection for improved model accuracy.

The Receiver Operating Characteristic (ROC) curve in Figure 4 is a crucial tool for evaluating models. In our research, it establishes VGG16 as the top-performing model, followed by VGG19 and ResNet50 with the lowest performance. This visual aids researchers in making informed decisions about model selection, contributing significantly to deep learning research.

TABLE I
PERFORMANCE METRICS FOR DIFFERENT MODELS

Model	Accuracy	Precision	Recall	F1	Time Delay	SD
VGG16	95.6	0.85	0.83	0.84	82.15	0.32
ResNet50	94.6	0.85	0.86	0.85	84.64	0.30
VGG19	89.4	0.85	0.84	0.84	71.50	0.31

TABLE II
THE OUTCOMES OF OPTIMIZERS IMPLEMENTED ON VGG16

Model	Accuracy	Precision	Recall	F1	Time Delay	SD
Adam	95.6	0.85	0.87	0.86	60.50	0.30
Adamax	95.5	0.86	0.88	0.87	60.58	0.27
Adadelta	92.2	0.86	0.94	0.90	60.89	0.17
RMSProp	94.9	0.85	0.84	0.85	82.15	0.33

Conclusion

This study investigates the effectiveness of Convolutional Neural Network (CNN) architectures in identifying malicious software on IoT devices, using a dataset of 17,219 binary file images to evaluate ResNet50, VGG19, and VGG16 models. Results show VGG16 with superior accuracy at 95.6%, followed by VGG19 at 94.6%, and ResNet50 at 89.4%.

Figures 6-9 depict loss and accuracy curves for various optimizers applied to VGG16, with ADAM showing the best results, followed by ADAMAX, RMSProp, and ADADELTA. Detecting malware on IoT devices is crucial due to potential consequences like financial loss, privacy violations, and physical harm.

While VGG16's high accuracy suggests its reliability for malware detection, the study acknowledges the dataset's limitation to binary files. Future research should explore these models' effectiveness on other types and formats of malware.

In conclusion, the study underscores the importance of devising effective methods for detecting malware on IoT devices. The proliferation of IoT devices heightens the need for steadfast security measures. Deep learning, especially CNN models, proves highly effective in detecting malware, emphasizing their promise in addressing the growing threat. The study emphasizes the critical need for robust methods to detect and prevent malware attacks on IoT devices, contributing to the broader goal of ensuring a secure IoT ecosystem.

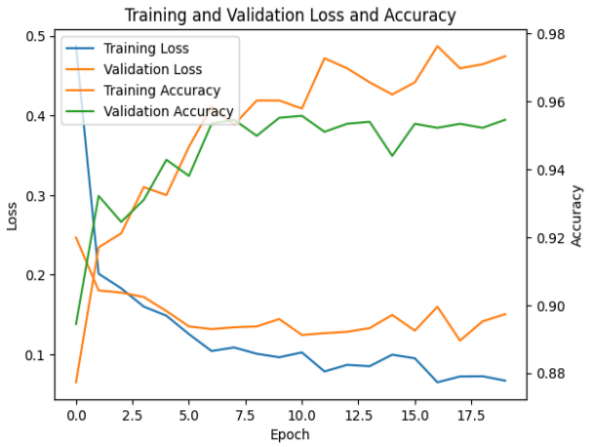

Fig. 5. Accuracy and Loss Curve for VGG16 with ADAM

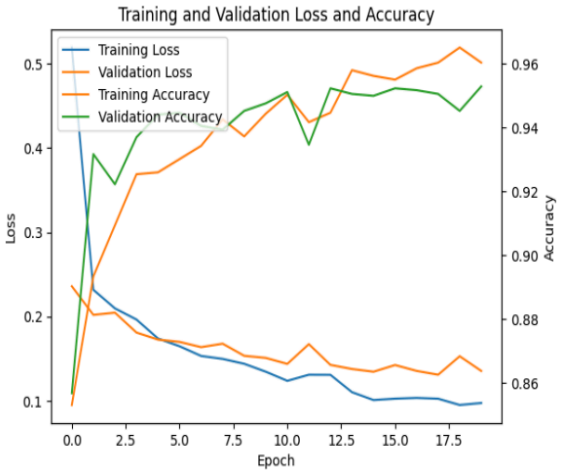

Fig. 6. Accuracy and Loss Curve for VGG16 with ADAMAX

Fig. 7. Accuracy and Loss Curve for VGG16 with RMSProp

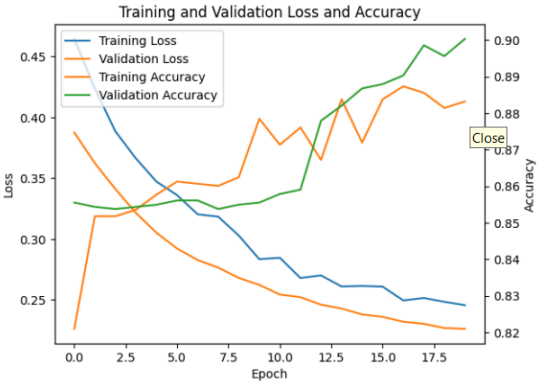

Fig. 8. Accuracy and Loss Curve for VGG16 with ADADELTA

References

[1] Kumar, S. MCFT-CNN: Malware classification with fine-tune convolution neural networks using traditional and transfer learning.

[2] vgg19 and resnet50 architecture frameworks for image classification," in 2021 International Conference on Disruptive Technologies for MultiDisciplinary Research and Applications (CENTCON), vol. 1, 2021, pp. 96–99.

[3] Chaganti, R.; Ravi, V.; Pham, T.D. Deep Learning based Cross Architecture Internet of Things malware Detection and Classification.

[4] Diro, A.A.; Chilamkurti, N. Distributed attack detection scheme using deep learning approach for Internet of Things.

[5] HaddadPajouh, H.; Dehghantanha, A.; Khayami, R.; Choo, K.K.R. A deep recurrent neural network based approach for internet.

[6] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 770–778.

[7] system using ensemble of deep belief networks. Appl. Soft Comput. 2018, 71, 66–77. [CrossRef]

[8] Ben Fredj, O.; Mihoub, A.; Krichen, M.; Cheikhrouhou, O.; Derhab, A. CyberSecurity attack prediction: a deep learning approach.

[9] Kudugunta, S.; Ferrara, E. Deep neural networks for bot detection. Inf. Sci. 2018, 467, 312–322. [CrossRef]

[10] Kumar, S. MCFT-CNN: Malware classification with fine-tune convolution neural networks using traditional and transfer learning.

[11] McDermott, C.D.; Majdani, F.; Petrovski, A.V. Botnet detection in the internet of things using deep learning approaches. In

[12] Azmoodeh, A.; Dehghantanha, A.; Choo, K.K.R. Robust malware detection for internet of (battlefield) things devices using deep

[13] Naveed, M.; Arif, F.; Usman, S.M.; Anwar, A.; Hadjouni, M.; Elmannai, H.; Hussain, S.; Ullah, S.S.; Umar, F. A Deep LearningBased Framework for Feature Extraction and Classification of Intrusion Detection in Networks. Wirel. Commun. Mob. Comput.

[14] Pekta¸s, A.; Acarman, T. Botnet detection based on network flow summary and deep learning. Int. J. Netw. Manag. 2018, 28, e2039.

[15] Proceedings of the 2018 International Joint Conference on Neural Networks (IJCNN), Rio de Janeiro, Brazil, 8–13 July 2018;

[16] Proceedings of the 2018 International Joint Conference on Neural Networks (IJCNN), Rio de Janeiro, Brazil, 8–13 July 2018;

[17] V. Rastogi, Y. Chen, and X. Jiang, “Catch me if you can: Evaluating android anti-malware against transformation attacks,” IEEE Trans. Inf. Forensics Secur., vol. 9, no. 1, pp. 99–108, 2013.

[18] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” arXiv Prepr. arXiv1409.1556, 2014.

[19] A. Azmoodeh, A. Dehghantanha, and K.-K. R. Choo, “Robust malware detection for internet of (battlefield) things devices using deep eigenspace learning,” IEEE Transactions on Sustainable Computing, vol. 4, no. 1, pp. 88–95, Feb 2018.

[20] Z. Yuan, Y. Lu, and Y. Xue, “Droiddetector: android malware characterization and detection using deep learning,” Tsinghua Sci. Technol., vol. 21, no. 1, pp. 114–123, 2016.

[21] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “Imagenet: A large-scale hierarchical image database,” in 2009 IEEE conference on computer vision and pattern recognition, 2009, pp. 248–255

[22] T. Hsien-De Huang and H.-Y. Kao, “R2-d2: Color-inspired convolutional neural network (cnn)-based android malware detections,” in 2018 IEEE International Conference on Big Data (Big Data), 2018, pp. 2633–2642.

[23] H. HaddadPajouh, A. Dehghantanha, R. Khayami, and K.-K. R. Choo, “A deep recurrent neural network based approach for internet of things malware threat hunting,” Future Generation Computer Systems, vol. 85, pp. 88– 96, Mar 2018.

[24] T. N. Phu, L. Hoang, N. N. Toan, N. Dai Tho, and N. N. Binh, “C500-cfg: A novel algorithm to extract control flow-based features for iot malware detection,” in 2019 19th International Symposium on Communications and Information Technologies (ISCIT). IEEE, Sep 2019, pp. 568–573.