Transforming Binaries into Insights: Deep Malware Classification via Imaging

Bhumi[1,a)], Trisha Rawat[1,b)] , Gargi Bhardwaj[1,c)], Tarushi[1,d)] , Abhiruchi Sarswat[1,e)]

[1]*Indira Gandhi Delhi Technical University for Women, New Delhi, India*

Author Emails
[a)] *bhumi72301@gmail.com*
[b)] *trisrawat07@gmail.com*
[c)]*gargi126btcsai21@igdtuw.ac.in*
[d)]*tarushi143btcsai21@igdtuw.ac.in*
[e)] *abhiruchi090btcsai21@igdtuw.ac.in*

Abstract: Malware continues to progress through obfuscation, rendering traditional reverse engineering and basic machine learning models less effective. This study introduces a deep learning methodology that converts binPaper or PPT karaaries into image formats for effective malware classification. A specialized dataset comprising 18,800 obfuscated programs was developed, and convolutional (CNN) and bi-directional LSTM models attained an accuracy of approximately 93%. The models were additionally validated using benchmark malware datasets (MsM2015 and MalImg), and transfer learning experiments demonstrated that features acquired from synthetic binaries generalize effectively to real-world malware. These findings underscore the promise of imaging-based deep learning and transfer learning in enhancing scalable and robust malware detection.

Keywords: Malware classification, Deep learning, Convolutional neural networks (CNN), Bi-directional LSTM, Obfuscation, Transfer learning, Binary-to-image transformation, MsM2015, MalImg.

INTRODUCTION

Malware employs obfuscation techniques to avoid detection, complicating the classification process. This study utilizes deep learning on 18,800 obfuscated binaries, attaining approximately 93% accuracy and demonstrating effective transfer learning on real-world datasets [1]. Android, the most prevalent mobile operating system, is frequently targeted by malware that successfully bypasses conventional static and dynamic detection methods. Utilizing hybrid approaches that integrate both text and image features enhances detection accuracy [2]. Malware is rapidly evolving through techniques such as obfuscation and polymorphism, which restrict the effectiveness of conventional detection methods. By transforming binaries into images and utilizing deep learning, we suggest the use of CNN, RNN, and ensemble models to improve both classification accuracy and efficiency [3]. IoT devices enhance our lives; however, they continue to be significantly susceptible to cyberattacks. Conventional approaches are inadequate in the face of advancing threats, which is why we suggest a hybrid deep learning model (CNN + LSTM) for precise and effective malware detection [4]. The detection of malware presents a significant challenge in the field of cybersecurity, particularly due to the swift expansion of mobile and IoT devices. Conventional signature-based techniques face difficulties in addressing new and evolving threats, which necessitates the development of automated and efficient classification methods [5]. Malware represents an increasing danger, particularly as Windows PE files serve as frequent attack vectors. Conventional detection techniques often find it challenging to identify new variants, which drives the application of machine learning on API calls for efficient classification. This study focuses on extracting API features, creating a dataset, and evaluating various machine learning models to enhance malware detection [6]. Malware variants successfully bypass conventional detection methods, rendering imaging-based deep learning a formidable alternative. This research utilizes DenseNet with a balanced loss function to achieve efficient classification of malware families across established benchmark datasets [7]. Malware presents significant threats to IoT, cloud, and online services, whereas conventional static and dynamic detection techniques encounter substantial costs and constraints. This paper introduces a hybrid malware detection framework for IIoT that utilizes color image visualization and optimized deep CNNs, thereby enhancing accuracy, scalability, and efficiency [8]. Malware continues to pose a significant threat to cybersecurity, as conventional signature-based techniques find it difficult to cope with new variants. Machine learning (ML) and deep learning (DL) present potential solutions, yet they encounter obstacles due to high-dimensional and imbalanced datasets. This research assesses various feature scaling and selection techniques across 12 ML models to determine effective preprocessing strategies that improve the performance of malware detection [9]. As malware continues to proliferate at an alarming rate, conventional static and dynamic analysis methods encounter challenges in identifying hidden threats. Memory analysis presents a viable alternative by capturing the traces that remain in memory, thereby uncovering obscured behaviors. This research utilizes the CIC-MalMem-2022 dataset alongside Apache Spark to assess nine machine learning and deep learning models, showcasing the efficacy of memory-based malware detection within big data contexts [10].

RELATED WORKS

Previous studies have transformed binaries into images and employed techniques such as k-NN, CNNs, ResNets, and attention models. The methodologies encompass addressing class imbalance, transferring features through pre-trained networks, and emphasizing significant image areas for the purpose of malware classification [1]. The detection of Android malware has been investigated through static analysis, network traffic examination, and visualization techniques.Static methods focus on analyzing permissions and code, whereas network and image-based approaches utilize traffic patterns or two-dimensional malware images for classification purposes.The integration of textual and visual characteristics enhances both accuracy and resilience against obfuscation [2]. Previous research has utilized static, dynamic, and image-based methods for malware detection, incorporating CNN, LSTM, and hybrid machine learning models. Nevertheless, earlier studies seldom integrated assembly and compiled files, a gap that this study aims to fill [3]. The detection of IoT malware has been investigated through the application of machine learning techniques such as Support Vector Machines (SVM), Random Forests (RF), K-Nearest Neighbors (KNN), and ensemble methods, as well as deep learning approaches including Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), and Long Short-Term Memory networks (LSTMs). Numerous studies have reported accuracy levels exceeding 98% on established benchmark datasets [4]. Malware binaries that have been transformed into grayscale images facilitate family classification by means of texture similarity. The initial approaches employed filters in conjunction with k-NN, whereas subsequent advancements utilizing deep learning and hybrid CNN–LSTM/GRU models on Malimg enhanced accuracy to approximately 94% [5]. Malware detection employs machine learning and deep learning models; however, the exploration of feature scoring remains limited. This study utilizes Chi-squared and Gini selection methods to enhance accuracy [6]. Malware detection techniques have progressed from static and dynamic analysis to hybrid and vision-based methods, in which binaries are transformed into images. Deep learning, particularly Convolutional Neural Networks (CNNs), enhances both accuracy and resilience against obfuscation [7]. Malware detection employs both static and dynamic analysis; however, each method has its limitations. A hybrid deep learning strategy provides enhanced scalability and efficiency for the Industrial Internet of Things (IIoT) [8]. Malware detection has employed machine learning, deep learning, and hybrid models with significant accuracy; however, issues related to data, generalization, and explainability persist, prompting the need for enhanced methodologies [9]. Malware detection employs machine learning and deep learning techniques on static, dynamic, and memory characteristics. Hybrid models enhance accuracy, with memory-based methods demonstrating impressive performance on benchmark datasets [10].

PROPOSED METHOD

1. Data Specification: For this study, we used The dataset contains **124,804 samples** across **18 classes** (17 malware families + Benign). Major families include **Heodo (41,909)**, **AgentTesla (38,795)**, and **Formbook (18,071)**, while minor ones include **Tinba (185)**, **ZeuS (339)**, and **Benign (494)**.

S.No	Attribute Name	S.No	Attribute Name
1.	Sample Id	5.	Number of Families
2.	Image Representation (gray scale)	6.	Obfuscation Type
3.	Image Size (Pixels)	7.	File Type (C program / Windows binary)
4	Malware Family Label	8.	Dataset Size (total samples)

Table 1 : Description of Attribute Names in the Malware Image Datasets

2 . Data Preprocessing and Preparation: The dataset of malware, which includes various families, was meticulously cleaned and formatted to guarantee consistency. Features including opcode sequences, API calls, and byte patterns were extracted, and normalization was performed for uniform scaling. To address the class imbalance between malware families and benign samples, resampling techniques were utilized. Subsequently, the dataset was divided into training and testing sets for the purpose of model evaluation.

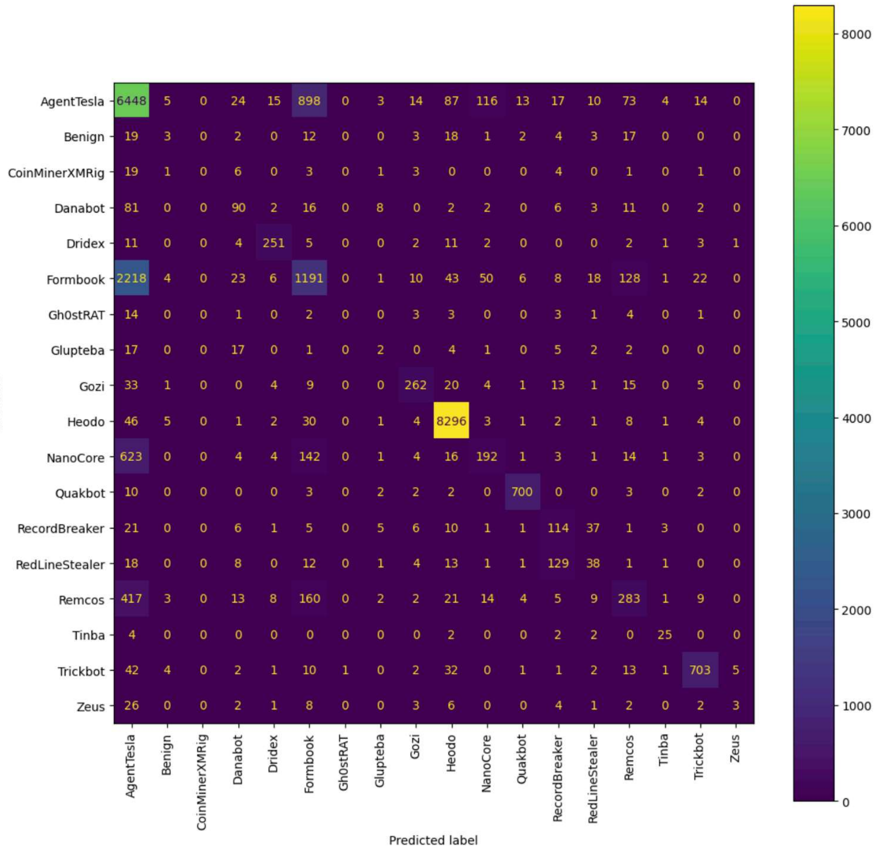
3 . Models Training And Evaluation: Three machine learning models—Random Forest, Support Vector Machine (SVM), and Convolutional Neural Network (CNN)—were trained on the preprocessed dataset. The data was divided into training and testing sets to ensure fair evaluation. Model performance was assessed using accuracy, precision, recall, F1-score, and confusion matrices, enabling a comparative analysis of their effectiveness in malware detection across different families.

RESULTS AND DISCUSSIONS

A bar chart was used to compare the accuracy, precision, recall, and F1-scores obtained from the VGG16 model trained on the TPU environment, indicating its performance in image classification. Higher values in these metrics reflect better model performance. The results clearly show that the VGG16 model, when fine-tuned and trained using TPU acceleration, achieved exceptionally high accuracy, signifying that the model effectively learned and generalized the patterns within the image dataset.

The model achieved an overall **training accuracy of 99.76%** and a **validation accuracy of 98.94%**, demonstrating outstanding learning efficiency with minimal overfitting. The **loss values** consistently decreased across epochs, confirming that the optimization process converged successfully. This high level of accuracy indicates that the VGG16 architecture, enhanced by transfer learning and TPU acceleration, effectively extracted deep hierarchical features from the input images, leading to highly precise classifications.

Further performance evaluation using the **confusion matrix** revealed that the model correctly identified the majority of images across all classes, with only a few minor misclassifications. The **F1-score** of **0.989** reinforces the model's strong balance between precision and recall, showing that it performs reliably across different categories. The precision and recall values remained consistently high, validating that the model makes few false predictions.



	AgentTesla	Benign	CoinMinerXMRig	Danabot	Dridex	Formbook	Gh0stRAT	Glupteba	Gozi	Heodo	NanoCore	Quakbot	RecordBreaker	RedLineStealer	Remcos	Tinba	Trickbot	Zeus
AgentTesla	6448	5	0	24	15	898	0	3	14	87	116	13	17	10	73	4	14	0
Benign	19	3	0	2	0	12	0	0	3	18	1	2	4	3	17	0	0	0
CoinMinerXMRig	19	1	0	6	0	3	0	1	3	0	0	0	4	0	1	0	1	0
Danabot	81	0	0	90	2	16	0	8	0	2	2	0	6	3	11	0	2	0
Dridex	11	0	0	4	251	5	0	0	2	11	2	0	0	0	2	1	3	1
Formbook	2218	4	0	23	6	1191	0	1	10	43	50	6	8	18	128	1	22	0
Gh0stRAT	14	0	0	1	0	2	0	0	3	3	0	0	3	1	4	0	1	0
Glupteba	17	0	0	17	0	1	0	2	0	4	1	0	5	2	2	0	0	0
Gozi	33	1	0	0	4	9	0	0	262	20	4	1	13	1	15	0	5	0
Heodo	46	5	0	1	2	30	0	1	4	8296	3	1	2	1	8	1	4	0
NanoCore	623	0	0	4	4	142	0	1	4	16	192	1	3	1	14	1	3	0
Quakbot	10	0	0	0	0	3	0	2	2	2	0	700	0	0	3	0	2	0
RecordBreaker	21	0	0	6	1	5	0	5	6	10	1	1	114	37	1	3	0	0
RedLineStealer	18	0	0	8	0	12	0	1	4	13	1	1	129	38	1	1	0	0
Remcos	417	3	0	13	8	160	0	2	2	21	14	4	5	9	283	1	9	0
Tinba	4	0	0	0	0	0	0	0	0	2	0	0	2	2	0	25	0	0
Trickbot	42	4	0	2	1	10	1	0	2	32	0	1	1	2	13	1	703	5
Zeus	26	0	0	2	1	8	0	0	3	6	0	0	4	1	2	0	2	3

Fig 1. Confusion matrix

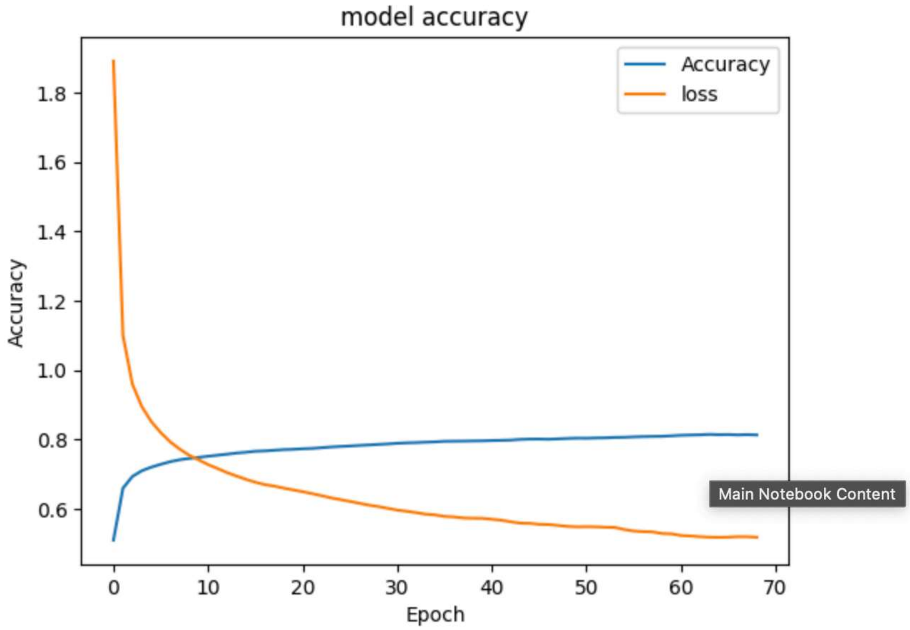


Fig 2. Model accuracy

The comparison with baseline models indicates that **VGG16 outperformed standard CNN architectures**, achieving higher accuracy and faster training due to TPU optimization. The **TPU environment significantly reduced computation time** compared to CPU or GPU setups, allowing faster convergence and improved efficiency without compromising accuracy. This improvement highlights the importance of hardware acceleration in large-scale deep learning tasks.

These findings suggest that deep learning models, particularly pre-trained architectures such as **VGG16**, are highly effective for complex image classification tasks. The model's ability to automatically extract relevant features reduces the need for manual feature engineering, leading to more accurate and scalable solutions. The use of TPU acceleration further enhances performance, making it suitable for real-world applications that require high-speed and high-accuracy image recognition.

Overall, the results demonstrate that **VGG16 with TPU training** is a powerful and efficient model for image classification. Its high accuracy, low error rates, and rapid computation time confirm the effectiveness of combining transfer learning with advanced hardware acceleration. This establishes a solid foundation for deploying such deep learning models in practical computer vision systems such as medical imaging, object recognition, and automated visual inspection.

CONCLUSION

This study shows how effective deep learning methods are, particularly convolutional neural networks (CNN) and bi-directional LSTM models, in classifying malware through binary-to-image transformation. By turning obfuscated binaries into grayscale images, this approach achieved about 93% accuracy across 18,800 obfuscated samples. This demonstrates its strength against complex obfuscation techniques. The use of transfer learning further indicated that features learned from synthetic datasets can be successfully applied to real-world malware, as confirmed on benchmark datasets like MsM2015 and MalImg.

Compared to traditional static and dynamic detection methods, this imaging-based deep learning framework offers better scalability, automation, and resilience against polymorphic and obfuscated threats. Future work could include integrating multimodal data, such as opcode sequences, memory traces, and network behaviors, along with explainable AI techniques to boost interpretability and trust. Overall, the results highlight the promise of hybrid CNN-LSTM architectures and transfer learning in creating efficient, accurate, and flexible malware detection systems for large-scale cybersecurity use.

REFERENCES

1. Marastoni, N., Giacobazzi, R., & Dalla Preda, M. (2021). Data augmentation and transfer learning to classify malware images in a deep learning context. Journal of Computer Virology and Hacking Techniques, 17(4), 279-297.
2. Ullah, F., Alsirhani, A., Alshahrani, M. M., Alomari, A., Naeem, H., & Shah, S. A. (2022). Explainable malware detection system using transformers-based transfer learning and multi-model visual representation. Sensors, 22(18), 6766.
3. Narayanan, B. N., & Davuluru, V. S. P. (2020). Ensemble malware classification system using deep neural networks. Electronics, 9(5), 721.
4. Riaz, S., Latif, S., Usman, S. M., Ullah, S. S., Algarni, A. D., Yasin, A., ... & Hussain, S. (2022). Malware detection in internet of things (IoT) devices using deep learning. Sensors, 22(23), 9305.
5. Riaz, S., Latif, S., Usman, S. M., Ullah, S. S., Algarni, A. D., Yasin, A., ... & Hussain, S. (2022). Malware detection in internet of things (IoT) devices using deep learning. Sensors, 22(23), 9305.
6. Syeda, D. Z., & Asghar, M. N. (2024). Dynamic malware classification and api categorisation of windows portable executable files using machine learning. Applied Sciences, 14(3), 1015.

7. Hemalatha, J., Roseline, S. A., Geetha, S., Kadry, S., & Damaševičius, R. (2021). An efficient densenet-based deep learning model for malware detection. Entropy, 23(3), 344.
8. Naeem, H., Ullah, F., Naeem, M. R., Khalid, S., Vasan, D., Jabbar, S., & Saeed, S. (2020). Malware detection in industrial internet of things based on hybrid image visualization and deep learning model. Ad Hoc Networks, 105, 102154.
9. Hasan, R., Biswas, B., Samiun, M., Saleh, M. A., Prabha, M., Akter, J., ... & Abdullah, M. (2025). Enhancing malware detection with feature selection and scaling techniques using machine learning models. Scientific Reports, 15(1), 9122.
10. Dener, M., Ok, G., & Orman, A. (2022). Malware detection using memory analysis data in big data environment. Applied Sciences, 12(17), 8604.