Malicious Image Detection Using Machine Learning Algorithm

Dr. M. DuraiPandian [1], Mohamed Rashik S [2], Praveen P [3], Karan P [4], Kamalesh K [5]

[1]Head of the Department of IT, Hindusthan Institute of Technology, Coimbatore, Tamil Nadu
[2]*Final year students of Department of IT, Hindusthan Institute of Technology, Coimbatore, Tamil Nadu*

ABSTRACT

Cyber assaults on people, businesses, and organisations have risen in recent years. Cybercriminals are continuously searching for efficient routes to start attacks and spread malware to victims. Images are used every day by millions of people around the globe, and most people believe that using them is secure. However, some picture kinds have malicious payloads that can carry out evil deeds. Due in large part to its lossy compression, JPEG is the most widely used picture file. They are utilised by almost everyone, from small businesses to big corporations, and are present on almost every type of gadget. (digital camera, smartphone, website, social media, etc.). Because of their widespread use, low risk of misuse, and status as being safe, JPEG images are frequently used by cybercriminals as attack vectors. However, to our understanding, machine learning techniques have not been applied to identify malicious JPEG pictures. Machine learning techniques have demonstrated effectiveness in identifying both known and undiscovered malware in a variety of disciplines. No specific study method was ever employed. Here, we introduce MalJPEG, the first machine learning-based tool designed with the goal of quickly identifying undocumented malicious JPEG pictures. MalJPEG uses a machine learning classifier to automatically extract 10 easily recognisable characteristics from the JPEG file structure and uses them to differentiate between legitimate and malicious JPEG pictures.Utilizing a comprehensive sample of 156,818 real-world pictures, including 155,013 (98.85%) innocuous and 1,805 (1.15%) malignant images, we thoroughly assessed MalJPEG. The findings indicate that when MalJPEG is combined with the LightGBM classifier, the true positive rate (TPR) is 0.951 and the area under the receiver operating characteristic curve (AUC) is 0.997, the greatest with a very low false positive rate. (FPR) 0.004.

INDEX TERMS **JPEG, image, malware, detection, machine learning, features.**

I. INTRODUCTION

In recent years, there has been a rise in cyberattacks on organizations, companies, and people. Cyberattacks, according to Infosecurity Magazine, increased in 2017.1 Cyber assaults typically involve detrimental behaviours like stealing private information, spying, or tracking and hurt the target (sometimes severely). Ideology, criminal purpose, a wish for notoriety, etc. are all possible motives for attackers.

Attackers are always looking for new and effective methods to initiate assaults and send malware to victims. Files sent over the Internet have frequently been used to achieve this. Because executable files (.exe) are known to be risky, attackers are increasingly using non-executable files (e.g.,.pdf,.docx, etc.) that most users mistakenly believe are secure to use. When some non-executable files are viewed, an attacker can execute random malicious code on the intended victim computer.

JPEG (Joint Photographic Experts Group) is the most widely used picture format2, owing to its lossy encoding. JPEG images are used by nearly everyone, from people to big corporations, and on a variety of platforms. JPEG pictures can be found on PCs (personal images, papers), devices, and other electronic devices. (smartphones, digital cameras, etc.).

space (emails, social media, websites, etc.). JPEG pictures are used as an attack vector by cyber thieves to send their malicious payload to the target device due to their innocuous reputation, widespread use, and high potential for misuse.

Saumil Shah demonstrated3 how to make malicious JPEG pictures that can be opened in a browser to run malicious code automatically at the 2015 Black Hat conference. It was revealed in November 2016 that assailants used Facebook Messenger to disseminate the notorious Locky ransomware via JPEG images.6 The malware writers found security flaws in Facebook and LinkedIn that enable them to download a malicious picture onto the victim's machine without their knowledge. It was claimed in August 2017 that the SyncCrypt ransomware was spreading via JPEG pictures. Trend Micro8, a corporate cyber security firm, revealed in December 2018 that cyber criminals used memes on Twitter (JPEG images) to communicate instructions to malware.9 In December 2019, Sophos security experts released a thorough report10 on the MyKings cryptomining botnet, which hides behind an apparently innocuous JPEG of Taylor Swift.

We thoroughly test MalJPEG on a real-world sample collection of benign and malicious JPEG pictures. We also compare MalJPEG features to features retrieved using various generic feature extraction techniques that are currently available.

The following are the paper's contributions:

1) MalJPEG is a machine learning-based method for detecting malicious JPEG images, both known and undiscovered.

2) MalJPEG features - a condensed collection of ten basic yet distinguishing features for the efficient detection of malicious JPEG pictures using machine learning methods.

3) The development of a big and representative labelled collection of innocuous and malicious JPEG pictures for further scientific study.

Section II provides introductory material on the JPEG file format, and Section III discusses connected work. Section IV discusses the techniques used in this study as well as the MalJPEG characteristics. In Section V, we evaluate our approach and show the findings. In Section VI, we examine the findings and different aspects of security, and we offer our conclusions.

II. BACKGROUND

This part contains basic information about our study as well as technical information about the structure of a JPEG picture. Because the JPEG file structure is complex, we only present the fundamental information required for the reader to grasp the paper and the suggested MalJPEG solution provided in this research. The JPEG File Interchange Format (JFIF) standard describes JPEG picture format in detail.

A. JPEG FILE STRUCTURE

JPEG is an abbreviation for Joint Photographic Experts Group, and it is the most common picture format on the Internet. JPEG became a worldwide standard for compressing digital still pictures in 1992. JPEG images typically have the file∗extensio∗n.jpg or.jpeg.

A JPEG picture file is a binary file which comprises of a series o segments. Hierarchically, segments can be enclosed within other segments. Each section starts with a two-byte sign known as a "marker." The marks aid in the segmentation of the file. The first bit of a marke is 0xFF (hexadecimal representation); the second byte can be any number except 0x00 and 0xFF. The sort of data saved in the section is indicated by the marker. Segment types are given labels depending on their meaning or function; for example, 0xFFD9 is called OI, and 0xFFFE is called COM. Segment types 0xFF01 and 0xFF0@hyphe0xFF9 are completely comprised of the two-byte marker; all other markers are followed by a two-byte number showing the segment's size, followed by the payload data stored in the segment. Table 1 lists the potential identifiers, along with their hexadecimal number and definition/purpose.

JPEG CLASSIFICATION:

	Code	
APP$_n$		app
COM	0xFFFE	Comment
DAC	0xFFCC	Define arithmetic conditioning table(s)
DHP	0xFFDE	Define hierarchical progression
DHT	0xFFC4	Define Huffman table(s)
DNL	0xFFDC	Define number of lines
DQT	0xFFDB	Define quantization table(s)
DRI	0xFFDD	Define restart interval
EXP	0xFFDF	Expand reference image(s)
JPG	0xFFC8	Reserved for JPEG extensions
JPG$_n$	0xFFF0-0xFFF13	Reserved for JPEG extensions
RES	0xFF02-0xFFBF	Reserved
RST$_n$	0xFFD0-0xFFD7	Restart with modulo 8 counter n
SOF$_n$	0xFFC0-3, 5-7, 9-B, D-F	Start of *Frame*
SOS	0xFFDA	Start of *Scan*
TEM	0xFF01	For temporary use in arithmetic coding
SOI	0xFFD8	Start of image
EOI	0xFFD9	End of image

EMBEDDING MALICIOUS PAYLOAD IN JPEG IMAGES

Vulnerability Exploitation - No software is ever fully secure, and preventing the existence of flaws during the creation of a large-scale software project is nearly impossible. When abused, such flaws can enable an attacker to gain elevated rights or redirect the normal execution flow to random malicious code.

Furthermore, in order to view/parse a JPEG image, Steganography (steganos - covered, graphie - writing) - Steganography,16 a technique used for concealing information (e.g., text or malicious code) within the image without affecting its appearance (invisible to the human eye) is extremely difficult to detect. Steganography can be used to exfiltrate confidential information from the victim's host or network using JPEG images, and it can also be used to send code into the victim's host or network using a basic benign JPEG image. As a result, we distinguish between JPEG pictures that contain concealed information via steganography and JPEG images that contain a harmful payload.

It is essential to note that malicious JPEG images do not always use steganography techniques to hide the aviewer/parser programme is needed, and these programmes may be vulnerable. Since JPEG images were first published, many vulnerabilities have been discovered, and there are currently 303 known vulnerabilities13 (CVE - Common Vulnerabilities and Exposures), and 5,520 known related security issues14 associated with JPEG (CVE-2018-6612) may allow a remote attacker to cause a denial-of-service when the victim processes a malicious JPEG file.

III. METHODS

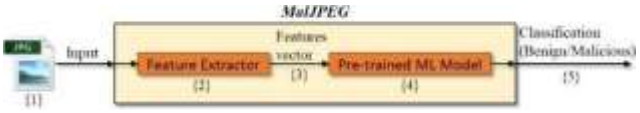
This part describes the techniques used in this study. We begin by discussing the characteristics of MalJPEG as well as the current generic feature extraction techniques. The features extracted by the MalJPEG feature extractor are then compared to those derived by the current generic feature extraction techniques. Finally, we explain the machine learning methods that we employed in this study.

A. MalJPEG SOLUTION

In this part, we show the paper's main addition, the MalJPEG machine learning-based method for detecting malicious JPEG images. As input 1, MalJPEG gets a JPEG picture. The MalJPEG feature extractor 2 converts the suggested features into a feature vector 3. The MalJPEG feature extractor inspects the file statically, without actually examining the picture (which needs running image viewer software, which may be vulnerable), and traverses the JPEG image file structure to extract the features. The features are then fed into a pretrained machine learning-based model 4, which generates a categorization (benign/malicious) 5 for the incoming picture. In the Java computer language, we built MalJPEG and its inner components, the feature extractor 3 and machine learning model 4. The following part contains a comprehensive description of how dehat are extracted using MalJPEG.

1) MalJPEG FEATURES

In this part, we show MalJPEG's compact collection of discriminative features. We created these characteristics after carefully inspecting the structure of a large number of innocuous and malicious JPEG pictures. We learned about how attackers use JPEG pictures to initiate attacks and how this impacts the JPEG file format. In terms of file structure, we discovered how malicious JPEG pictures vary from normal benign JPEG images.



B. MACHINE LEARNING ALGORITHMS

On the datasets outlined in the preceding part, we used machine learning categorization methods. We used the following widely used, high-performing traditional and nonlinear machine learning models in our experiments: Decision Tree, Random Forest, and Gradient Boosting on Decision Trees. (XGBoost and LightGBM). These models were chosen because they work well on extremely imbalanced datasets. It is worth noting that in our early tests, we looked at models from families other than the decision tree family, such as Logistic Regression and Nave Bayes, but they did not produce satisfactory results.; As a result, we did not include them in our assessment.Furthermore, on Min-Hash datasets, we used the K-Nearest Neighbors classifier (K = 5) because it is the only predictor that can match Min-Hash signatures using the Hamming distance function. We decided to u. We used Python to apply the aforementioned machine learning classifiers, including scikit- learn, XGBoost, and LightGBM. For all classifiers, we used the preset setup.

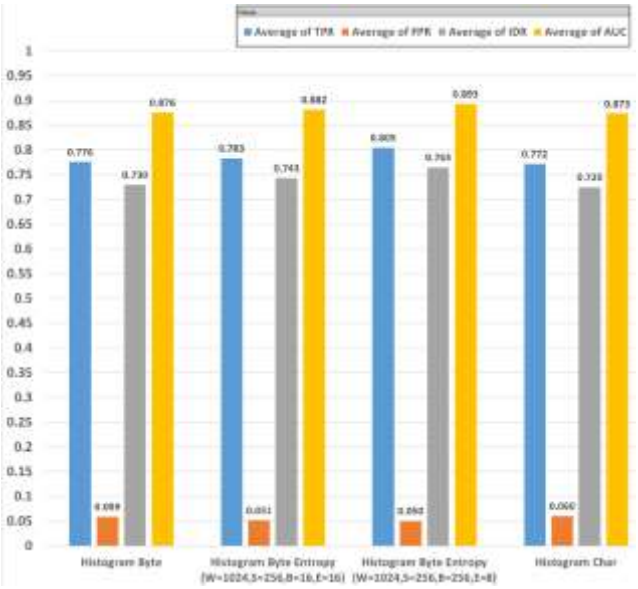
IV. EVALUATION

This part evaluates MalJPEG. We begin by presenting our data gathering for assessment, followed by a description of the dataset generation method. Then we show our study topics, metrics for assessment experimental methodology, and findings.

A. Output

1) EXPERIMENT 1

Figure 7 compares the detection results of the Random Forest classifier to those of the other classifiers used in our experiments on datasets created using the histogram methods presented in Table 3; we only provide the detection results of the Random Forest classifier because it outperforms all of the other classifiers used in our experiments on all of the datasets created using the histogram methods. To obtain the best outcomes, we increased the Random Forest threshold from 0.5 to 0.05. According to the AUC metric, the findings are sorted from greatest to lowest. As can be seen, the finest outcomes were obtained.



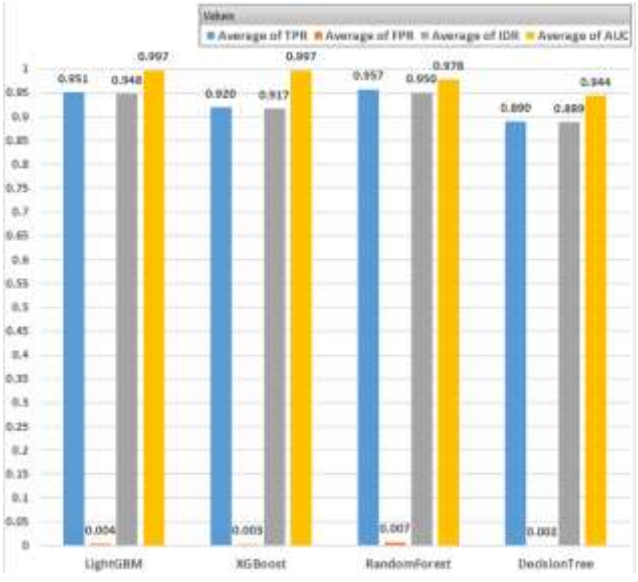


FIGURE 7. Detection results for the *Random Forest* classifier on datasets created using different histogram feature extraction methods.

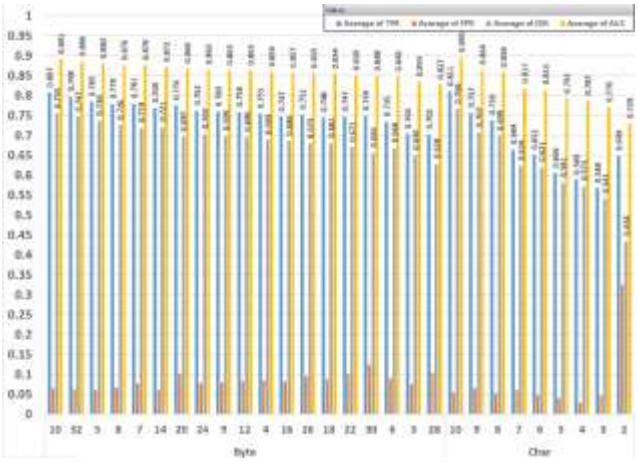


FIGURE 8. Detection results for the *K-Nearest Neighbors* classifier on datasets created using *Min-Hash* feature extraction methods with different configurations.

using the byte entropy histogram: $TPR = 0.805$, $FPR = 0.05$, $IDR = 0.765$, and $AUC = 0.893$.

Figure 8 compares the identification findings of the K-Nearest Neighbors classifier on datasets generated using the Min-Hash techniques. It is important to note that the K-Nearest Neighbors classifier is the only classifier that can actually compare Min-Hash signatures using a distance function (see Section IV-B.2; there is no actual order between the Min-Hash signature's numbers, so regular machine learning algorithms are ineffective on it). We used the K-Nearest Neighbors algorithm with K 5 and the Hamming distance function. To obtain the best outcomes, we increased the K-Nearest Neighbors classifier cutoff from 0.5 to 0.05. According to, the findings are sorted from greatest to lowest according to the AUC metric.

FIGURE 9. Detection results of the machine learning classifiers on a dataset containing *MalJPEG* features.

TABLE 4. Summary of the configurations that provide the best results for both histogram and Min-Hash methods.

Feature Extraction Method	Configuration	Number of features	Classifier	TPR	FPR	IDR	AUC
Entropy Histogram	Basic Unit = Byte, Window Size = 1024, Stride = 256, Bytes Axis Size = 256, Entropy Axis Size = 8	2048	*Random Forest*	0.805	0.050	0.765	0.893
Min-Hash	Basic Unit = Byte, Hash Functions = 200, Window Size = 10, Stride = 1	200	*Random Forest*	0.810	0.054	0.766	0.895
MalJPEG	N/A	10	*LightGBM*	**0.951**	**0.004**	**0.948**	**0.997**

V. CONCLUSION

We introduce MalJPEG, a machine learning-based method for detecting undocumented malicious JPEG pictures, in this article. To the best of our knowledge, we are the first to demonstrate a machine learning-based system specially designed for detecting malicious JPEG pictures. MalJPEG extracts ten basic but discriminative characteristics from the JPEG file format and uses them in conjunction with a machine learning classifier to distinguish between innocuous and malicious JPEG pictures.

The structure of the JPEG picture is used to derive MalJPEG characteristics. MalJPEG characteristics were specified based on a knowledge of how attackers use JPEG images to initiate assaults and how it effects the JPEG file structure when compared to normal innocuous JPEG images. MalJPEG is tested in four trials. We used a very big collection of 156,818 JPEG pictures for our evaluation: 155,013 (98.9%) innocuous and 1,805 (1.15%) malicious, collected between 2016 and 2018 from social media (benign images) and VirusTotal.(malicious images). It is worth noting that the proportion of malicious pictures in our database is exceedingly low (1.15%). We designed our collection in such a way that it represents, as much as possible, the low proportion of malicious pictures (compared to innocuous images) in the actual world. It's also worth noting that the collection's incredibly low proportion of fraudulent occurrences (positive) makes detecting malicious pictures in our tests extremely challenging.Their signatures are continually and rapidly changed. In contrast, MalJPEG, which is built on machine learning, can identify both known and undiscovered malicious JPEG pictures.

Furthermore, MalJPEG can be simply parallelized and scaled to deal with vast amounts of pictures in enterprise-scale systems. Based on our findings, it would be beneficial to adopt MalJPEG in order to safeguard businesses, online services (such as Microsoft Office 365 and Google Drive), social media platforms (such as Facebook and Instagram), and their users from malicious JPEG pictures.

REFERENCES

[1] A. Cohen, N. Nissim, L. Rokach, and Y. Elovici, ''SFEM: Structural feature extraction methodology for the detection of malicious office documents using machine learning methods,'' *Expert Syst. Appl.*, vol. 63, pp. 324–343, Nov. 2016.

[2] N. Nissim, A. Cohen, and Y. Elovici, ''ALDOCX: Detection of unknown malicious microsoft office documents using designated active learning methods based on new structural feature extraction methodology,'' *IEEE Trans. Inf. Forensics Security*, vol. 12, no. 3, pp. 631–646, Mar. 2017.

[3] A. Cohen, N. Nissim, and Y. Elovici, ''Novel set of general descriptive features for enhanced detection of malicious emails using machine learning methods,'' *Expert Syst. Appl.*, vol. 110, pp. 143–169, Nov. 2018.

[4] N. Nissim, A. Cohen, C. Glezer, and Y. Elovici, ''Detection of malicious PDF files and directions for enhancements: A state-of–The art survey,'' *Comput. Secur.*, vol. 48, pp. 246–266, Feb. 2015.

[5] N. Nissim, Y. Lapidot, A. Cohen, and Y. Elovici, ''Trusted system-calls analysis methodology aimed at detection of compromised virtual machines using sequential mining,'' *Knowl.-Based Syst.*, vol. 153, pp. 147–175, Aug. 2018.

[6] A. Cohen and N. Nissim, ''Trusted detection of ransomware in a private cloud using machine learning methods leveraging meta-features from volatile memory,'' *Expert Syst. Appl.*, vol. 102, pp. 158–178, Jul. 2018.

[7] N. Nissim, A. Cohen, R. Moskovitch, A. Shabtai, M. Edri, O. Bar-Ad, and Y. Elovici, ''Keeping pace with the creation of new malicious PDF files using an active-learning based detection framework,'' *Secur. Inform.*, vol. 5, p. 1, Dec. 2016.

[8] N. Nissim, R. Moskovitch, L. Rokach, and Y. Elovici, ''Novel active learning methods for enhanced PC malware detection in windows OS,'' *Expert Syst. Appl.*, vol. 41, no. 13, pp. 5843–5857, Oct. 2014.

[9] N. Nissim, R. Moskovitch, L. Rokach, and Y. Elovici, ''Detecting unknown computer worm activity via support vector machines and active learning,'' *Pattern Anal. Appl.*, vol. 15, no. 4, pp. 459–475, Nov. 2012.

[10] *proach for Malware Detection*. Accessed: 2019. [Online]. Available: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3383953