Spiteful JPEG : Machine Learning Based Solution for the Detection of Malware in JPEG Images And Image Forgery

Shruti Dhumal[1], Gayatri Gutte[2], Shreya Kondalkar[3], Tanuja Kale[4], Shobha Raskar[5]

Department of Computer Engineering
Modern Education Society's Wadia College of Engineering,
Pune
19, Bund Garden, V.K. Joag Path, Pune – 411001.
Savitribai Phule Pune University

***Abstract*— Malicious JPEG images are crucial for both individuals and businesses, as they are widely used. These images can contain malicious payloads and perform harmful actions. However, machine learning methods have proven to be effective at detecting both known and unknown malware in various domains. In this paper, MalJPEG is presented as the first machine learning-based solution tailored specifically for the detection of unknown malicious JPEC images. First, a machine learning classifier is used to discriminate between benign and malicious images. Then, the features extracted from the original image are then transferred to a pretrained machine learning model, which outputs a classification (benign/malicious) for the input image. The results show that the detection area under the receiver operating characteristic curve (AUC) is 0.997, which demonstrates the highest detection capabilities. The use of image steganography is becoming more and more significant because it matches human visual habits and offers more information than other types of information. In this paper, we describe the evaluation metrics that are frequently used, the experimental set-ups that were taken into consideration, and the steganalysis model that was used to potentially find and extract the images from malware binaries. This paper provides a Convolutional Neural Network (CNN) that categorizes pictures collected from malware images using a convolution neural network, a tool for machine learning detection.**

***Keywords*— classifier, steganography, detection, benign**

I. Introduction

The authenticity and integrity of images have become critical in a time when digital photography and visual communication rule the day. The widespread use of photographs in forensics, law enforcement, journalism, and social media highlights the need for effective methods to identify and stop image alteration and forgeries. Using malicious JPEG images is a common and sophisticated technique for image-based attacks, where adversaries take advantage of flaws in the JPEG format to hide harmful payloads or alter the visual content. Due to their potential for use in a multitude of harmful actions such as information theft, malware propagation, and the spread of misleading information, malicious JPEG images represent a serious concern. to provide cutting-edge tools and algorithms for detecting and reducing harmful JPEG image threats. The goal is to improve the ability to distinguish real photos from fake ones by combining deep learning, machine learning, and image processing techniques.

II. Literature Survey

. In recent years, cyber-attacks have increased, with cyber criminals seeking effective vectors to deliver malware to victims. JPEG images are the most popular image format due to their lossy compression and widespread use. These images can contain malicious payloads and perform harmful actions. Machine learning methods have been shown to be effective at detecting known and unknown malware in various domains, but they have not been used particularly for the detection of malicious JPEG images.In this paper, MalJPEG is presented as the first machine learning-based solution tailored specifically for the efficient detection of unknown malicious JPEG images. MalJPEG statically extracts 10 simple yet discriminative features from the JPEG file structure and leverages them with a machine learning classifier to discriminate between benign and malicious JPEG images. The results show that MalJPEG, when used with the LightGBM classifier, demonstrates the highest detection capabilities, with an area under the receiver operating characteristic curve (AUC) of 0.997, true positive rate (TPR) of 0.951, and a very low false positive rate (FPR) of 0.004.Cyber attacks targeting individuals, businesses, and organizations have increased in recent years, with Infosecurity magazine stating that cyber attacks doubled in 2017. Cyber attackers are constantly searching for new and effective ways to launch attacks and deliver a malicious payload to victims. The results show that MalJPEG, when used with the LightGBM classifier, demonstrates the highest detection capabilities, with an area under the receiver operating characteristic curve (AUC) of 0.997, true positive rate (TPR) of 0.951, and a very low false positive rate (FPR) of 0.004.Cyber attacks targeting individuals, businesses, and organizations have increased in recent years, with Infosecurity magazine stating that cyber attacks doubled in 2017. Cyber attackers are constantly searching for new and effective ways to launch attacks and deliver a malicious payload to victims. JPEG images are used by cyber criminals due to their harmless reputation, massive use, and high potential for misuse. Malicious JPEG images are crucial for both individuals and businesses, as they are widely used. Current endpoint defense solutions rely on signatures to detect known malware, leaving clients vulnerable to new malware variants. However, machine learning (ML) algorithms have shown the ability to detect both known and unknown malware in various domains, particularly for the detection of malware in various types of files.The number of cyberattacks has surged in recent years as hackers look for efficient ways to infect victims with malware.

The process of concealing text, images, or videos inside a cover image is known as image steganography. The confidential data is concealed so that it is not visible to the public. Recently, there has been a greater focus on deep learning technology, which has shown guarantee as a potent tool in a variety of applications, including image steganography. This paper's primary objective is to examine and talk about the different deep learning techniques used in the field of image steganography. Convolutional neural network-based, general adversarial networkbased, and traditional methods comprise the three broad categories of deep learning techniques used for image steganography. This paper describes the evaluation metrics that are frequently used, the experimental set-ups that were taken into consideration, and a detailed summary of the datasets that were used. The LSB technique is the foundation of many conventional techniques. For the purpose of embedding and extracting the secret messages, CNN-based methods rely on deep convolutional neural networks, while GAN-based methods make use of some GAN variants. CNN or GAN-based models can be used as DL models. The steganography block creates the stego image, which is then used as input by the steganalysis model to potentially find and extract the secret information. In certain techniques, the output is the probability score indicating whether the input image is a stego image or a normal image.

3In order to classify malware images, this research provides a Convolutional Neural Network model that tests at 98 percent accuracy. The model classifies pictures collected from malware binaries using a convolution neural network, a tool for machine learning. The model is reliable; testing yields 98 percent accuracy. The researchers provide a model that significantly outperforms more established detection methods including behavior-based, heuristic, and signature-based detection by using a convolution neural network to categorize photos taken from malware binaries. With hundreds of new instances of malware being created every day, malicious software, or malware, poses a serious concern in today's informationtechnology society. Conventional detection methods, such as behavior-based, heuristic, and signaturebased methods, have drawbacks, including increased space requirements and system performance impacts. The authors provide a framework that employs.

This paper introduces the min-max hash method, which significantly reduces the hashing time by half while having a slightly smaller variance in estimating pairwise Jaccard similarity. The estimator of the min-max hash only contains pairwise equality checking, making it especially suitable for approximate nearest neighbour search. Since the min-max hash is equally simple as the min-wise hash, many extensions based on it can be easily adapted to the min-max hash. Experiments show that with the same length of hash code, the min-max hash reduces the hashing time to half as much as the min-wise hash, while achieving smaller mean squared error (MSE) in estimating pairwise Jaccard similarity and better best approximate ratio (BAR) in approximate nearest neighbour search. Min-wise hash is a widely-used hashing method for scalable similarity search in terms of Jaccard similarity, but in practice, it is necessary to compute many such hash functions for certain precision, leading to expensive computational cost. In this paper, the authors introduce an effective method, the min-max hash method, which significantly reduces the hashing time by half while achieving smaller mean squared error (MSE) in estimating pairwise Jaccard similarity and better best approximate ratio (BAR) in approximate nearest neighbor search. This paper focuses on the hashing method for approximate nearest neighbor search in terms of Jaccard similarity on set data. The authors distinguish between sketching methods and hashing methods, which enable scalable similarity search. For data represented by finite sets, they can index min-wise 4 sketches efficiently via hash tables, resulting in an LSH method that enables efficient Jaccard similarity search. Min-wise hash is widely used in real applications, such as near-duplicate detection and clustering. However, it is often necessary to compute many minwise hash functions to guarantee certain precision. The authors propose a min-max hash method for scalable approximate nearest neighbor search in terms of Jaccard similarity, which reduces the hashing time of min-wise hash by half and has a slightly smaller variance in estimating pairwise Jaccard similarity. The estimator of min-max hash is equally simple as min-wise hash, i.e., it only contains pairwise equality checking, enabling scalable Jaccard similarity search via hash table. Experiments in Section IV verify the computational cost analysis in Section III.A and the variance analysis in Section III.C, showing that min-max hash reduces the hashing time to half as much as that of min-wise hash, while achieving slightly smaller MSE than min-wise hash in estimating pairwise Jaccard similarity and better best approximate ratio in approximate nearest neighbor search. Min-wise sketch is a popular sketching method for Jaccard similarity, providing an unbiased estimate of pairwise Jaccard similarity. For data represented by finite sets, a random permutation: I→I is generated. Denotemin((S)) = min iS(i), then for any two nonempty sets A and B, it is proven that Pr[min((A)) =min((B))] = —A B— —A B— =J(A, B). In conclusion, the min-max hash method offers a more efficient and accurate approach to estimating pairwise Jaccard similarity on set data. [5] Pattern recognition and computer vision researchers find image segmentation to be a difficult research topic. Support vector machine (SVM) techniques are currently popular, but performance in classification is impacted by manual selection, which decreases adaptability.

Pattern recognition and computer vision researchers find image segmentation to be a difficult research topic. Support vector machine (SVM) techniques are currently popular, but performance in classification is impacted by manual selection, which decreases adaptability. A novel support vector machine (SVM) technique integrates mean clustering, extracts texture and color features from images, and segments images using trained classifiers. Image segmentation, which separates areas of an image with special significance, is an essential component of image analysis and processing. Color image segmentation is becoming more and more significant because it matches human visual habits and offers more information than grayscale images. Color images can be segmented using the same techniques used for grayscale images, such as edge detection, artificial neural networks, region-based approaches, clustering, and histogram threshold. Combining the results yields the final segmentation result. Techniques for segmenting images Scholars have suggested using support vector machines (SVM)-based methods to train classifiers with features like gray level. SVM is unable to automatically obtain training samples, which can be tedious and haphazard. These issues can be resolved by the suggested SVM image segmentation technique, which automatically chooses training samples to increase the SVM model's self-adaptability. This paper suggests an automatic training sample selection technique for color image segmentation using the FCM clustering algorithm. While FCM does not take spatial information into account, it does retain more image information. Similarity measures include Euclidean distance and local spatial and gray information. The image is segmented after an SVM classifier trained on randomly chosen pixels. A classifier called Support Vector Machine (SVM) uses quadratic programming to solve problems and reduce structural risk. Using kernel functions, it converts non-linear problems into linear ones and determines the best hyperplane for each type of data. When segmenting images with different backgrounds and foregrounds, SVM works well. Due to the numerous iterations required to solve non-linear quadratic programming solutions, the classical SVM method is time-consuming even though it is intended for small sample sizes. The suggested SVM image segmentation model separates pixels into foreground and background by automatically choosing training samples. By using a membership function that takes into account categories, gray values, Euclidean distance, weighted indices, and local similarity, the FCM algorithm establishes the pixel class.

The model is split into two phases, the first of which is dedicated to the selection of FCM training samples. The model is split into two phases, the first of which is dedicated to the selection of FCM training samples. The algorithm expresses the local, spatial, and gray information of pixels using formulas. In order to provide more pre5 cise local spatial information, it takes into account the Euclidean distance between two pixels. The algorithm takes into account the distance between eight neighborhoods as well, which improves the validity and accuracy of the local similarity measurement. This method gets around the drawbacks of FCM algorithms that segment images without taking spatial information into account. The first pixel's coordinates and G's scale factor have an impact on the algorithm's accuracy. The method for converting a color image to a grayscale image, initialization parameter setting, local spatial and grayscale information calculation, and similarity measures are all covered in the text. It also determines the Images are used to extract texture features, which represent the scene's surface characteristics and the grayscale distribution in space. Mathematical or informatics techniques can be used to quantify texture features, which are crucial for human visual systems. The gabor wavelet is a popular technique in computer vision, pattern recognition, and content-based image retrieval because it efficiently extracts texture features and removes unnecessary information. Gabor filters are used to compute local energy information for color channels and filter band-pass images. They are easily tunable to match the characteristics of the human visual system and take into account signal resolution in both the spatial and frequency domains. G0 and G filters are used by the four-scale Gabor filter in order to preserve energy information and enhance texture reflection.

The procedure entails converting the RGB color space to the Lab color space and analyzing the image signal using 16 distinct Gabor filter banks. The original image, the amplitude matrix, and the filtering window are all utilized. Good training samples are used to train the SVM model, and the trained classifier is used to segment the data. Four images in natural color are used in the experiment, which is run in a Windows 8 environment. An unprocessed 256x256 simple image and a polynomial kernel function for SVM image segmentation are compared in the study. The method effectively represents and separates objects and backgrounds in color images, offering significant advantages in color image segmentation. Subjective segmentation and the visual effect are in line. However, data statistics and visual effects are still awaiting confirmation of the method's efficiency. [6] In order to increase the accuracy of incorrect data identification in the widely-used Chi-Squares test, a modified measure is proposed in this study. Although the Chi-Squares test is a thorough and quick technique for estimating the health of a power system, its reliability against incorrect data is not great. The study suggests an easy alteration that will enhance the ability of current state estimators to identify incorrect data: it will require the computation of a residual covariance matrix. While it can be done with low computational expense, this method is computationally expensive. The accuracy of detecting incorrect measurements and computational performance are evaluated between the suggested method and the traditional Chi-Squares approach. The redesigned metric is predicated on the idea that the objective function has a chi-squares distribution. The suggested approach is contrasted with the traditional Chi-squared test in terms of computational performance and bad measurement detection accuracy

III. PROPOSED SYSTEM AND ALGORITHM

The goal of this framework is to detect the presence of any malicious code hidden inside the JPEG image. Then extract and analyse the content which could be potentially be a malicious program. Phases of Proposed Framework

Phase 1: Steganalysis of Cover Image:

• The Input image is passes through steganography algorithm.

• This algorithm detects weather any type of the content is embedded within file or not.

• If it is found that malicious content is present in the file then the file is passes to the next Phase.

• Threat level is set to 1 and Threat is set to 0.

• If there is no malicious content in image it will directly be forwarded to the phase

Phase 2: Malicious code detection using CNN:

2.1. Extract File header using min-max hashing

• Min Max Hashing: Min-Hash is a probabilistic data structure used to estimate the Jaccard similarity between two sets. The Jaccard similarity is defined as the size of the intersection of two sets divided by the size of their union. Min-Hash works by creating a signature (or sketch) for a set of elements, which is a compact representation of the set. To compute the Jaccard similarity between two sets, you compare their Min-Hash.

Here's a simplified explanation of the process:

1. Generate multiple hash functions.

2. For each hash function, compute the minimum hash value for each element in the set.

3. Combine these minimum hash values to form a signature for the set.

2.2. Extracting features from the file header

• After completion of min max hashing features are extracted from the file header.

• Extracted Features are added into the vector named feature vector.

2.3. Passing feature vector as input to the model

• Classification:

Classification is a fundamental task in machine learning (ML) and is used to categorize data into predefined classes or categories. The goal of classification is to build a model that can automatically assign new, unseen data points to one of these categories based on the patterns and relationships learned from a labeled dataset.

• Classification model is Used to detect whether malicious code is present or not.

• If the malicious code is threat level is set to be 2 and threat set to 1 .

• Else threat level is 1 and threat is 0.

Reverse sign language recognition using StackGAN involves generating images or signs from text descriptions. StackGAN is a type of generative adversarial network (GAN) that can be trained to produce realistic images from textual descriptions. In this context, the aim is to create a system that can understand text descriptions of sign language gestures and generate corresponding images of those gestures.

Phase 3: Image Forgery Detection using chi square test

3.1 Data Partitioning

• Data partitioning, in the context of data management and analysis, refers to the process of dividing a dataset into multiple subsets or partitions for various purposes such as training machine learning models, testing, validation, and managing data efficiently. Proper data partitioning is crucial for ensuring the integrity of your analysis and model evaluation.

• Image is Partitioned into small pixels using LBP technique.

• Features from the images are extracted by Using LBP.

• Calculate Observed and expected Frequencies 3.2Chi square static Calculation

• This test is used when you want to determine whether an observed frequency distribution fits a theoretical (expected) frequency distribution. It answers the question, "Do the observed frequencies match the expected frequencies?"

• Creating a contingency table that shows the observed and expected frequencies.

• Calculating the chi-squared statistic based on the differences between observed and expected frequencies. •Comparing the chi-squared statistic to a critical value from the chi-squared distribution to assess the goodness of fit.

Phase 4: Submitting the Report of Detected Image

• Report is generated on the basis of the threats and threat Levels.

IV. SYSTEM FLOW DIAGRAM

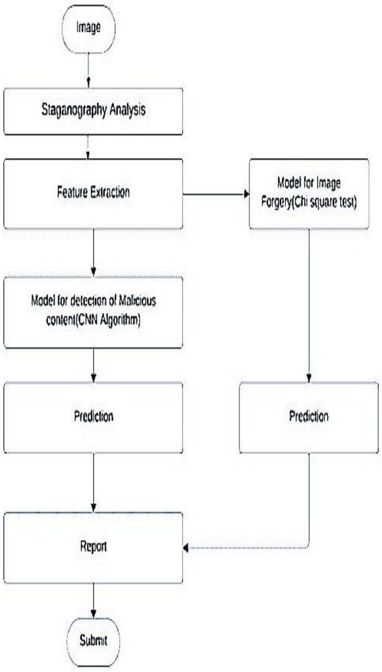
Image
Staganography Analysis
Feature Extraction
Model for Image Forgery(Chi square test)
Model for detection of Malicious content(CNN Algorithm)
Prediction
Prediction
Report
Submit

Figure 1: System Architecture

The methodology consists of several steps, including reading images as a numpy array, histogram-based thresholding to create a binary image, Canny edge detection to find edges or objects in the image, and dilating and eroding the image with vertical and horizontal kernels to merge edges if they are too close. First, the input image is read as a numpy array using the OpenCV library. Then, histogram-based thresholding is applied to convert the image to a binary format. This process involves finding the optimal threshold value by analyzing the distribution of pixel intensities in the image. The resulting binary image will have black pixels representing the background and white pixels representing the foreground.

Next, Canny edge detection is performed to identify edges or objects in the image. This process involves calculating the gradient of the image and using it to identify regions with sharp changes in intensity. The output of the Canny edge detection is a binary image where the edges are represented by white pixels and the rest of the image is black. To merge edges that are too close together, the image is dilated and eroded using vertical and horizontal kernels. This process helps to remove small gaps in the edges and connect nearby edges to form a continuous boundary.

Optical flow-based instance segmentation:

In this scholarly article, the authors introduce an innovative technique for weakly supervised instance segmentation that capitalizes on motion data, building on the foundation of BoxInst, a box-guided instance segmentation method. BoxInst employs projection and pairwise losses to direct mask learning in CondInst, a high-performing instance segmentation model that incorporates FCOS, an anchor-free object detection approach, and an instancespecific mask head with dynamic parameters tailored to each object. The projection loss directs the horizontal and vertical projections of the predicted mask using ground-truth box annotations, while the pairwise loss encourages adjacent pixel pairs with similar colors to share the same label, enabling the propagation of pseudo-mask labels. To harness motion data as supplementary signals for weak supervision, the authors present a dual-stream encoder that delivers appropriate feature maps for the detection and mask heads atop the box-guided architecture. Furthermore, they suggest a novel pairwise loss to direct the mask head. The comprehensive architecture of the proposed model is depicted, encompassing the dual-stream encoder, object detection head, and dynamic mask head. The dual-stream encoder extracts appearance and motion feature maps from input images, which are subsequently supplied to the detection and instance-specific mask heads. The object detection head generates pseudo-labels for each object, which are then em24 ployed to supervise the mask head. The proposed pairwise loss takes into account both appearance and motion data to direct the mask head, enabling superior segmentation of objects with indistinguishable appearances.

Mask R-CNN, a two-stage procedure for object detection that builds upon the success of the Region Proposal Network (RPN) introduced in [1]. While similar to RPN in the first stage, Mask R-CNN differs from recent systems that rely on mask predictions for classification. Instead, it outputs a binary mask for each Region of Interest (RoI) in parallel with predicting the class and box offset. This approach follows the methodology of Fast R-CNN [2], which uses bounding-box classification and regression in parallel, thus simplifying the multi-stage pipeline of original R-CNN [3]. During training, a multi-task loss is defined on each sampled RoI as L = Lcls + Lbox + Lmask. The classification loss Lcls and bounding-box loss Lbox are the same as those defined in [2]. The mask branch produces a Km2-dimensional output for each RoI, encoding K binary masks of resolution m x m, one for each of the K classes.

The per-pixel sigmoid is applied to the output, and Lmask is defined as the average binary cross-entropy loss. For an RoI associated with the ground-truth class k, Lmask is only defined on the k-th mask (other mask outputs do not contribute to the loss). This approach allows the network to generate masks for every class without competition among classes, decoupling mask and class prediction. In contrast, the per-pixel softmax and multinomial cross-entropy loss typically used in FCNs [30] for semantic segmentation compete among classes. Experiments show that this formulation is crucial for obtaining good instance segmentation results.

V. EXPERIMENTAL RESULTS

The user will submit a picture. Once it has been uploaded, it is checked for steganography, which finds any hidden information in the photographs. Steganalysis is used in the context of harmful image detection to determine whether a picture contains malicious payloads, sensitive data, or hidden malware. When it comes to identifying potentially harmful photos, steganalysis is generally carried out as follows Next examining variations in color histograms, pixel value distributions, steganographic embedding, and Finding anomalies in the noise patterns throughout the image by looking for deviations or inconsistencies that might point to hidden data. Finding anomalies in the statistical properties mean, variance, and higher-order moments A classifier is trained to differentiate between stego and non-stego images once features have been retrieved. The performance of the steganalysis method is then assessed using metrics like accuracy, precision, recall, and F1-score on a validation or test dataset. Common classifiers include Support Vector Machines (SVM). incorporating the steganalysis algorithm into more comprehensive systems, including antivirus software and intrusion detection systems, to identify harmful material in photos.

Take out the important aspects from the picture. These features could be statistical details like texture features or color histograms, among other things. Next, quantize the attributes into distinct groups or divisions and to find out if the distribution of characteristics in the image differs significantly from the expected distribution, use a Chi-square test. Utilize the Chi-square test findings as characteristics for a machine learning classifier. To learn patterns that differentiate real and fake photographs, the classifier can be trained on a dataset of them. Analyze if the image is a fake or not by assessing the classifier's performance using metrics like accuracy, precision, recall, or F1-score.

Following that, the data is passed via the CNN algorithm, which has multiple layers. Filters make up convolutional layers. Every filter uses local portions of a picture to execute convolutions (element-wise multiplications and summations) in order to identify particular features, such as edges, textures, or more intricate patterns. (element-wise summations and multiplications) using an image's local areas. the network is able to learn complex relationships in the data by introducing non-linearity through the use of an activation function (typically ReLU, but others like Leaky ReLU or sigmoid can also be used). Multiple filters are applied in parallel, generating a set of feature maps that represent different learned features of an image. The feature maps created by the convolutional layers are downsampled by pooling layers, which lowers their spatial dimensions while keeping the most crucialdata.

- Choosing the maximum value from a region or finding the average value are examples of common pooling processes.
- Fully connected layers use the learnt features to execute high-level reasoning using the output that has been flattened by the previous pooling layer.

- Using optimization methods like gradient descent and backpropagation, the CNN learns to reduce the discrepancy between its predictions and the true labels of the training data. This process involves adjusting the weights and biases of its layers. Through this procedure, the CNN is able to automatically pick up discriminative elements that let it distinguish between images that are malicious and ones that arenot.

CNNs can efficiently identify hazardous information in images by examining patterns at various abstraction levels, ranging from basic features like edges to intricate structures, all depending on the learned representations recorded in their layers. And in the event that a picture is not harmful or not, a report will be generated at the user's end.

VI.Conclusion

Social media has undeniably transformed the world into a hyper-connected global network. People from all walks of life readily share an abundance of information, thoughts, and experiences on these platforms. While this interconnectedness fosters communication and sharing, it also exposes our digital landscape to heightened vulnerabilities. The sheer volume of data coursing through the internet has made today's computer systems increasingly susceptible to various forms of cyber attacks. The act of clicking and sharing images has become a modern-day trend, almost a digital fashion statement. However, in the backdrop of this trend, a new reality has emerged – the need to fortify our defences against potential threats hidden within JPEG images. These threats may manifest as malicious code embedded within innocent-looking images. It is in response to this emerging challenge that the proposed framework gains significance. This framework, when fully realized, will serve as a formidable shield against attacks stemming from seemingly harmless.

VII.References

[1] Aviad Cohen, Nir Nissim, and Yuval Elovici. Maljpeg: Machine learning based solution for the detection of malicious jpeg images. IEEE Access, 8:19997–20011, 2020.

[2] Rakesh Singh Kunwar and Priyanka Sharma. Framework to detect malicious codes embedded with jpeg images over social networking sites. In 2017 International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS), pages 1–4, 2017.

[3] Mahmoud Kalash, Mrigank Rochan, Noman Mohammed, Neil D. B. Bruce, Yang Wang, and Farkhund Iqbal. Malware classification with deep convolutional neural networks. In 2018 9th IFIP International Conference on New Technologies, Mobility and Security (NTMS), pages 1–5, 2018.

[4] Jianqiu Ji, Jianmin Li, Shuicheng Yan, Qi Tian, and Bo Zhang. Min-max hash for jaccard similarity. In 2013 IEEE 13th International Conference on Data Mining, pages 301–309, 2013.

[5] Yinlong Wang, Yao Lu, and Yan Li. A new image segmentation method based on support vector machine. In 2019 IEEE 4th International Conference on Image, Vision and Computing (ICIVC), pages 177–181, 2019.

[6] Murat G¨ol and Ali Abur. A modified chi-squares test for improved bad data detection. In 2015 IEEE Eindhoven PowerTech, pages 1–5, 2015.

[7] Feng Liu. Copy-move forgery detection based on quaternion and lbp. In 2023 8th International Conference on Intelligent Computing and Signal Processing (ICSP), pages 1507–1510, 2023.

[8] Jun-Dong Chang, Bo-Hung Chen, and ChweiShyong Tsai. Lbp-based fragile watermarking scheme for image tamper detection and recovery. In 2013 International Symposium on NextGeneration Electronics, pages 173–176, 2013.

[9] Jijina M.T, Litty Koshy, and Gayathry.S. Warrier. Detection of recoloring and copy-move forgery in digital images. In 2020 Fifth International Conference on Research in Computational Intelligence and Communication Networks (ICRCICN), pages 49–53, 2020