National College of Ireland

An Approach to Detect Stegomalware in an Image using Machine Learning

MSc Research Project
Programme Name

Harsha Vardhan Masina
Student ID: 20196075

School of Computing
National College of Ireland

Supervisor: Michael Pantridge

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Harsha Vardhan Masina		
Student ID:	20196075		
Programme:	MSc in Cybersecurity	**Year:**	2022
Module:	Research project		
Supervisor:	Michael Pantridge		
Submission Due Date:	19/09/2022		
Project Title:	An Approach to Detect Stegomalware in an Image using Machine Learning		
Word Count:	6179	**Page Count**	18

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: Harsha Vardhan

Date: 19/09/2022

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

An Approach to Detect Stegomalware in an Image Using Machine Learning

Harsha Vardhan Masina
X20196075

Abstract

Now a days the technology had increasing very rapidly at the same time wide range of new methods are increasing the development of malicious activity by the cybercriminals with help these techniques they are gaining lots of sensitive data and credentials. There is continuous growth in cybersecurity technology same as a variety of techniques are implementing by the attacker. In that one of the important techniques is stegomalware. In this process the malware is encrypt or hide in the different forms like images, documents, and videos etc is known as stegomalware by using this kind of method they steal the critical data. This is the reason detection of malware is important it helps to detect the malware at early stage and prevent the devices from getting attacked. There are many detection approaches are established but to detect accurately the techniques of machine learning can be used because it gives very accurate results. In this paper I will using techniques of machine learning of K-NN, DT, and MLP algorithms used to detect malware with help of the datasets. In this we will evaluating the results of the model performance based on this will be able to implement confusion matrix.

1 Introduction

Today's world the technology is upgrading day by day and increasing various versions in each update which have great impact on our life. Everyone is surrounded by the technology which making our life much easier. This is where cybercriminals come into the picture these hackers have many opportunities to spread malware in any forms whereas emails, links, documents, and images etc. Mainly there are using social media platforms to spread the malware and it is one of the easiest ways to spreading the malware. This is happened because of IT sector are developed and focus on design of software, securing organization network and able to prevent any malicious approach but most of the organizations are investing billions of moneys to secure the critical data of the organizations. The researchers and cybersecurity expertise are working hard and doing excellent work to prevent cyber-attacks which comes from the outside of organization network.

In terms of malware, it is piece of code that was developed by the cybercriminals are used to damage any kind of device or any technology to steal critical information etc. They are various types of malwares are viruses, worms, spywares, and ransomware etc. The malware is developed by the hacker, and it is used to make money and gain money by selling critical information in the dark web. In which the advantage of the malware are some organizations can also buy the malware to test their security features that they are developed. Malware are provided great loss to the reputation organizations and steals critical data from the company. According to the McAfee company said that any new type of malware can easily detect for every four seconds. Moreover, that any anti-virus can't be able to detect the types of malwares for example according to the research there are 68,557,897 different viruses which are undetected till now. Every time malware doesn't need to develop in the same way it can be able to change and can go undetected. Now a days attackers are new methods to access the data or steal the data from that the important method is stegomalware. The stegomalware is one type of malware in which that uses steganography to hide the detection. The practice of

steganography is hiding any piece of code in image, video, and file etc. This kind of malware which operates to build a steganographic device to hide infected code in the device and it will extract and executes them dynamically.
How can we detect image based stegomalware using machine learning technique

To mitigate various kinds of problem a better approach would be required in which we are used machine learning approaches and aggregation of different algorithms methods because of that the rate of detection of the malware will increase. The model of the software that will created and it will run on the machine through the instructions it was created with python and gives the result accurately. In this we will download the datasets from the official datasets platform these datasets should have to train with machine learning algorithms with help of this it will give accuracy of the malware detection.
In this researcher paper is divided into the following sections in section 2 the literature review which focuses on the various research paper of a malware detection in which these are proposed by other researchers. In section 3 will consists of research methodology and in section 4 will be design sections of the thesis. In section 5 will consists of implementation in these details of model is presented. In section 6 it consists evaluation in which it contains results of the model.

2 Related Work

2.1 Malware detection

The malware detection alerts a warning signal for your systems or device, and it informs you that these devices are on a secure platform. The malware detection keeps attackers away from the user's systems and prevent critical information from getting leaked. All users must know about the malware detection and types of techniques to detect the malwares.

The paper (Choi et al., 2017) and (Krithika and Vijaya, 2020) they focused on deep learning approach using convolutional neural network (CNN) method and binary classification converted executable files into grayscale images in they used deep learning technique along with Gist feature vector. As a result, they achieved 88% of accuracy which had smaller datasets. Moreover, they are focused on faster detection malware on the images. The limitation of this paper is author is used only CNN method which is theoretically proved and there is possibility to get better optimized result. In our model we are using machine learning approach using KNN, DT, and MLP it helps to achieve high percentage of accuracy of a faster detection the datasets will be used in this model are larger datasets and it will be proved by practically.

In this article (Jin et al., 2020) and (Pinto, Duarte and Sant'Ana, 2020) the authors have proposed a method on deep learning approach using autoencoders. They have focused on various malware families of datasets. The limitation of this paper is author is focused on the malware families and failed to get optimized result and F1-score. In our model, we are using different algorithms approaches of KNN, DT, and MLP machine learning methods this includes large number of image datasets, and it helps to achieve high percentage of optimized result and F1-score.

In this article (Roseline et al., 2019) and (Mathew and Kurian, 2020) they focused on the method multi-layered random forest ensembling along with deep learning approach and used steganalysis along with logistic classification which is used to detect malware on images. They used smaller datasets in this approach. The demerit of this paper they failed to propose the result of the outcome is mentioned theoretically without any paperwork and there is possibility to get optimize result. In this paper they mainly focused on the malware families of datasets which there are used in this model. In In our model, we are using different

algorithms of unique approaches of KNN, DT, and MLP methods which includes the all kinds of metrics like accuracy, F1-score, and recall and it helps to achieve high percentage of optimized result.

In this research papers (Chen et al., 2017) and (Samuel et al., 2022) the authors have proposed a method on deep learning using convolutional neural network (CNN) and catalyst kernel which is used to boost detection rate these are approaches to JPEG images. In which this JEPEG segments is converted into the Gray-scale image then the CNN will carry out these images of a smaller number of datasets. As a result, they got 96 % of accuracy of the JEPEG image datasets. The limitation of this research paper they are failed to give unique approach and used existing approach to give the optimized result. In our model, we are using different algorithms of unique approaches of KNN, DT, and MLP methods which includes high image datasets, and it helps to achieve high percentage of optimized result.

2.2 Machine Learning

This branch focused on using data and algorithms which used to allowed machines to improve the over time which helps to increase accurate decision this is based on the making predictions or classifications. Mainly it works based on three ways it starts with by using combinations of data and algorithms to predict various patterns and classify number of datasets then an error function which helps to evaluate result of accuracy and the process of optimization will fit the datapoints into their according models.

In this paper (Ozkan, Isik and Kartal, 2018) and (Kumar et al., 2018) the authors on focused on deep learning approach using convolutional neural network (CNN) features method. The algorithms which are used KNN, and SVM on the large number of datasets which including 36 families of malware. The accuracy results achieved is 85% of this paper. The other paper also which focused on the results of accuracy achieved is 98% along with large number of datasets of 25 different malware families in this they have used CNN method. The limitation of this paper the author didn't focused on deeper of the other metrics like F1-score, recall, and time taken. In our model we are using different kinds of algorithms like KNN, DT, and MLP to give high optimize accuracy result along with other metrics like F1-score and recall.

In this research paper (Cakir and Dogdu, 2018) and (Shao Yang, 2019) the authors have proposed a method on deep learning method of CNN approach. The results of accuracy are achieved 94% and they used high number of datasets in this paper. The limitation of this paper they failed to be proposed new methodology this paper is already contain with existing work. In our model we are using different kinds of algorithms like KNN, DT, and MLP along with new approach of methodology to give high optimize result.

In this article (Poonguzhali et al., 2019) and (Chen, 2020) the authors have proposed a method on deep learning method on RGB images of an CNN approach. The feature extraction has done from the Gray-scale image by using CNN approach along with support vector machine classifier. The accuracy which achieved is 94 % with help of the CNN approach. The demerit of this paper the accuracy which are proved by theoretically and there is no evidence for the optimize result. There is no set of datasets has used in this paper. In our model we are using different kinds of algorithms like KNN, DT, and MLP to give better result along with 3200 datasets.

In this paper (He and Kim, 2019) and (Iqbal *et al.*, 2021) they focused on the method on deep learning method of CNN approach. They have implemented SPP and RGB along with CNN along with grayscale images of an JPEG images which is used to measure the performance of the model. They have used smaller set of datasets in this paper. The limitation of this paper the author didn't focused on any metrics and there is no result for the optimized

outcome. In our model we are using different kinds of algorithms like KNN, DT, and MLP to give optimize result along with different metrics like F1-score and recall.

2.3 K-Nearest Neighbors (KNN), Decision Tree (DT), and Multi-layer Perceptron (MLP) Classifiers

In this we have used machine learning algorithms K-NN, DT, and MLP in which it gives high detection accuracy rate of malware detection. The large number of datasets can be able to be applied to this model. We have applied 80% for training data and 20% testing data based on the working model it will give optimised result.

In this research paper (Kosmidis and Kalloniatis, 2017) the author proposed a method machine learning approach using K-nearest neighbors, decision tree, multi-layer perceptron, stochastic gradient, and nearest centroid approaches. They have used high 9342 image datasets of various large number of 25 malware families. The limitation of this research paper is author is failed to get optimized result with help of 3 different algorithms. In our model, we are using machine learning methods using KNN, DT, and MLP of 3000 image datasets and it helps to achieve high percentage of optimized result.

In this paper (Liu and Wang, 2016) and (Gupta, Bansal and Kumar, 2018) the authors have proposed a method uniform local binary pattern (ULBP-RF) and three different features like registry activity, libraries which are imported, and functions of API along with DT and KNN algorithms. In this paper they achieved result up to 96 %. The demerit of this paper they failed to propose the other metrics and there is no evidence of an optimize outcome. In our model we are using different kinds of algorithms like KNN, DT, and MLP along with high number of datasets to give optimize results of other metrics.

In this study (Mosli et al., 2016) and (Narayanan, Djaneye-Boundjou and Kebede, 2016) the authors have proposed a method to convert files into Gray-scale images which increase efficiency this are mapped into vectors features and principal component analysis (PCA) used for feature extraction on the images along with RF, SVM, DT, ANN, K-NN, K-means, and LT algorithms. They have used large number of datasets of 20 malware families. The results which they achieved n-gram and API values by using these algorithms and they failed to achieve results of metrics like accuracy. In our model we are using different kinds of algorithms like KNN, DT, and MLP which helps to give optimize results along with other metrics.

In this article (Zhou, Pang and Liang, 2017) and (Xu et al., 2021) they proposed a method on Gray scale image used Gabor filter and multi-feature fusion on N-value opcode N-gram of images to extract features along with machine learning algorithms like GBDT, KNN and RF. As a result, author achieved 96.19 % by using these classifiers. The drawback in this paper author failed to give unique approach and this method of approach is already in existing which helps to find the results in Gray scale images and they didn't focused on accuracy. In our model we are using different kinds of algorithms like KNN, DT, and MLP which helps to give optimize results along with other metrics and we are unique approach to get better results.

In this research paper (Darus, Salleh and Mohd Ariffin, 2018) and (Duan, 2018) the authors have proposed method on grayscale images used to extract features from the images and automatic recognition method have used in the first step have to train the model then the second step is compared recognition images with images which are trained along with authors are used algorithms like DT, KNN and RF. They are used smaller dataset in this model and results of accuracy they achieved 84.14% in this model. The limitations of this paper they failed to achieve optimised results by using aggregation of three algorithms. In our model we are using different kinds of algorithms like KNN, DT, and MLP which helps to give optimize results along with large number of 3200 datasets.

In conclusion most of the authors are focused on the metrics of accuracy they are getting good rate accuracy percentage, but they are focused on the accuracy percentage. At some point they are lagging, and they focused on only one metrics. There is possibility to increase the performance of a metrics.

2.4 Related Worktable

Table 1: A table

Related Work	Strengths	Limitations
Mohd Ariffin et al., 2018	Here they have used three different algorithms	By using three different machine learning algorithms they failed to get high percentage of accuracy
Jin et al., 2020	They have used smaller number of datasets for deep learning approach	They have used deep learning technique with CNN method, but the author is lagging to find good result
Iqbal *et al.*, 2021	They have achieved good number of percentages of datasets by using deep learning approach	They have used deep learning technique with CNN method they achieved good percentage but they failed to focus on other metrics.

Figure:1 Related Worktable

3 Research Methodology

Stegomalware

The stegomalware is a kind of a malware in which we can be able to encrypt any malicious code into the image this process is known as stegomalware. The process of hiding malicious code in the image is known as steganography. Here in this model, we used the datasets images files which are encrypted with malware. This image files which is applied to this model, and it contains machine learning algorithms in this model.

Data processing

The steps involved in pre-processing of a data is important in algorithms of machine learning and it is necessary to the model that we are using it. Then it is important to compare with

other algorithms in which we have implemented and compared it. The machine leaning algorithms are K-Nearest Neighbors Algorithm (KNN), Decision Tree (DT) and Multi-layer perceptron (MLP) which are used in this model. The feature extraction of the datasets has implemented in our model. The steps involved in feature extraction has explained below in detail.

Datasets

The image files datasets has collected from the James Z.Wang website from this website we have collected 3,200 image files. In these datasets of image files 1,077 which are malicious files and 2,154 which are benign files then the malicious image files are converted with the help of the matlab functions[1]. From these datasets we are divided 80% for the training set and 20% for the testing set in our model.

Training data

The final dataset that we are using are 3,200 image datasets out of in which 2,000 image dataset that we are using in training model. In these datasets 1,077 which are malicious files and 2,154 which are benign files. These 2,000-image dataset is used to train the training model and load into the training set. In which we are providing 80% of dataset image to the training model which is used to the train the model with help of the machine leaning algorithms is K-Nearest Neighbors Algorithm (KNN), Decision Tree (DT) and Multi-layer perceptron (MLP)

Testing data

In this final dataset there are total 3,200 image datasets in which 2,000 image files that are used to train the model of a machine learning algorithms. In which we are providing 20% of dataset image files to the testing model in which is used to the test the model with help of the machine leaning algorithms is K-Nearest Neighbors Algorithm (KNN), Decision Tree (DT) and Multi-layer perceptron (MLP)

Python programming

Python is a one of the programming languages which is used to build websites and software's along with various tasks and analysis of data. Python can be used general purpose language which means it can be used to create variety of various programs and it is not specialized for the problem. Then install the packages of python libraries on the tool which is relate to the source code. The aggregation of model is developed by python language and whole source code are developed through this language.

Machine learning algorithms

The machine learning approaches enable system to operate freely without doubt of programming. The applications of machine learning are fed with new data based on that it can independently learn, grow, develop, and adapt. The machine learning algorithm performance which can improve adaptively with help of the number of samples which are available during the process of learning. For an example deep learning is method of sub-domain of ML in which those trains systems to imitate natural human traits like learning and it offers high performance. There are two types of machine learning supervised machine learning and unsupervised machine learning. In learning supervised machine learning where

[1] http://wang.ist.psu.edu/docs/related/

the machines are trained on the dataset, and which enable to predict the outputs based on the training are provided. These datasets which particularly specifies some input and output based on the parameters which are mapped. Finally, this machine is trained with corresponding input and output. Then the device is used to predict output using these datasets. In unsupervised machine learning which refers to learning method that lacking supervision. Then the machine is trained by using a dataset and which is enable to predict the outcome by using without any supervision. The unsupervised machine learning algorithm which aims to the group of different datasets which is based on similarities of input like differences and patterns.

The machine learning algorithms used for the development of this thesis:

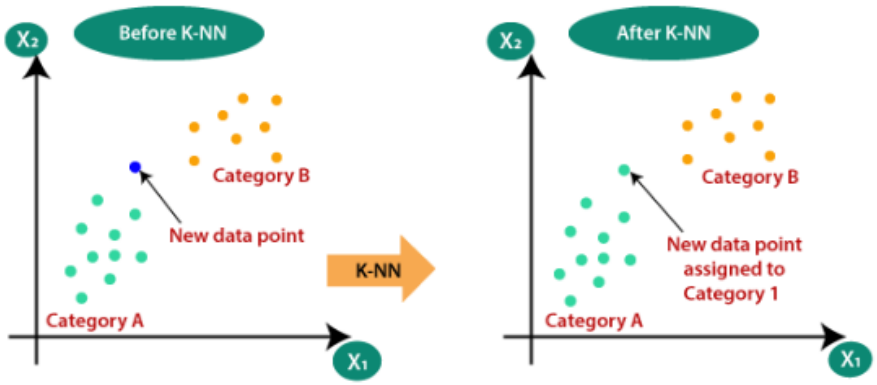
1. K-Nearest Neighbors Algorithm (KNN)
2. Decision Tree
3. Multi-layer perceptron (MLP)

K-Nearest Neighbors Algorithm (KNN)

The K-Nearest Neighbour algorithm is one of the simplest algorithms of machine learning based on the technique of supervised learning. This algorithm assumes that similarity between the new case or new data and the cases which are available it will put the new cases into the list of categories which is most relatable to the available categories. The algorithm which stores the data which are available and classifies a new data which is based on the similarity. If in case any new data appears it will easily be classified into category using this algorithm. This algorithm which can be used for classification or regression this is widely used for classification.

The steps involved in working of KNN algorithm

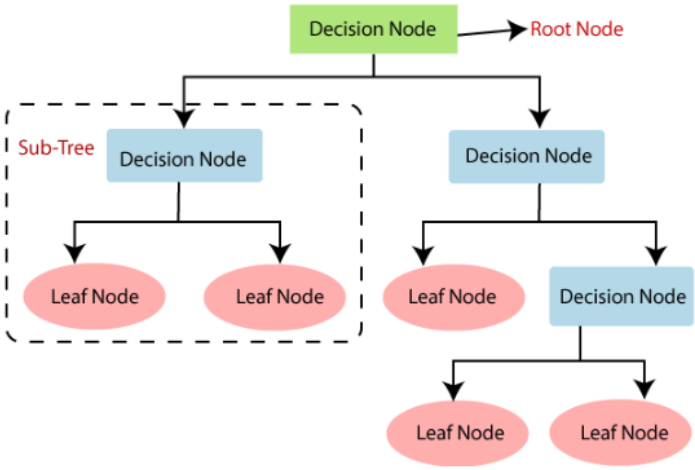
The datasets which are divided into training and testing dataset and these datasets should be load to the training set and testing set. This training dataset which divided into various clusters based on the type. Then the K value which is selected for the data point in which the classification should be done. The data points which are selected from all clusters which are classified then the distance function is used to finding closest neighbors. Then the new data point will be assigned to the cluster in which the number of neighbors is maximum.



Data point classification

Decision Tree

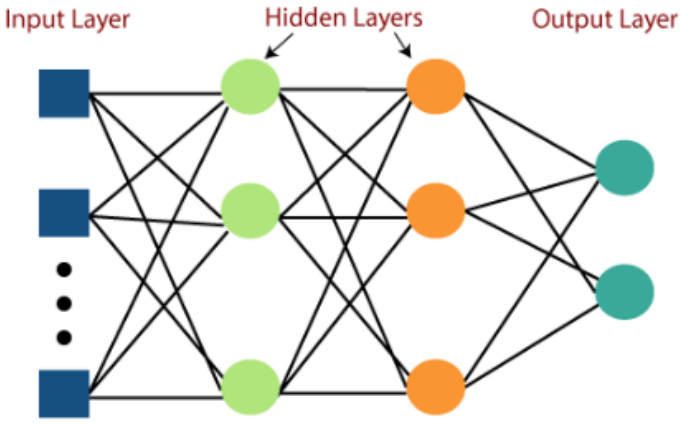
The decision tree algorithm is one of the supervised learning techniques which can be used for problems of both classification and regression but its mainly used for problem solving of classification. This algorithm is like tree-structured classifier which includes internal nodes that represents datasets features, branches represent the rules of decision and every leaf node which represents the outcome. In decision tree algorithm have two nodes decision node and leaf node. The decision nodes which are used to make any decision and it have number branches. The leaf nodes are the outcomes of those decision, and it doesn't contain any branches. The test of the decision tree algorithm which is performed based on give datasets.



Decision tree classification

Multi-layer perceptron (MLP)

The multi-layer perceptron is one of the supervised learning techniques and it also known as back propagation's algorithm. Which is one of the most complex architectures of an artificial neural network. It is formed with help of the multiple layers of a perceptron. The working of multi-layer perceptron algorithm is feed into artificial neural network which generates the set of outputs from the set of inputs. The structure of MLP algorithm which have a multiple layer's nodes of input which is connected between the input and output layers, and it uses technique of back propagation for the training set.

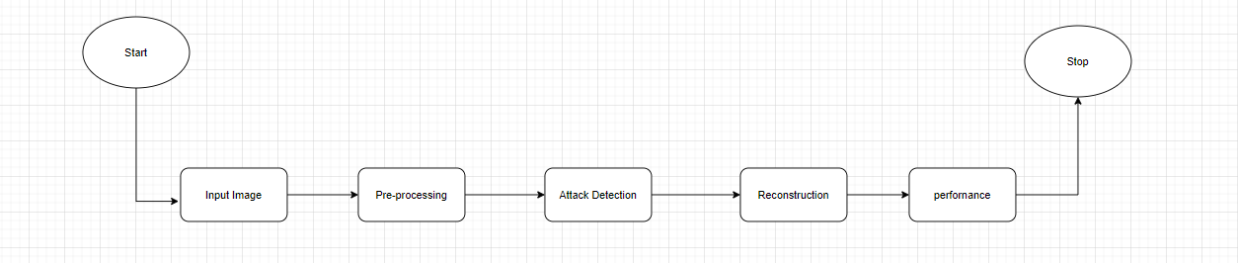


Mult-layer perceptron classification

4 Design Specification

In this section will consist of overall discussion of different methods which is used for developing this thesis. In this section will explain about machine learning algorithms along with other functional and non-functional then the tools which are required for the development of this thesis.

In this figure shown that list of datasets images that we had given as an input in this it contains malicious and benign files. Then the datasets went through the process of data pre-processing in this the image files will divide into training section and testing section with help of these datasets we had train the model. In this model it contains aggregation of K-NN, DT, and MLP. After this model had trained and we test the model with help of the datasets. After this model had tested and it identified malicious and benign image files based on that performance of the model of an metrics had found.



Step1: Importing Packages // Installing packages in anaconda prompt

Step2: Get the Input data // Getting list of directories malicious and Benign files

Step3: Generating FOR loop // Reading images one by one and split into malicious and Benign files

Step4: Initialize dataset // Splitting datasets for training 80% and testing 20%

Step5: Layer assignment // Features of images which includes filters, pool_size, dense, conv2D, and dropout

Step6: Extracting feature values// Getting mean value, standard deviation, and variance values

Step7: Loaded train features // Collected overall features from the datasets and load the features which is used for the algorithm classifier

Step8: Applied algorithms classifier (KNN, DT and MLP) // Importing packages of the classifier and create the syntax.

Step9: Fitting algorithms classifier (KNN, DT and MLP) // In fitting train features and labels from the images have used.

Step10: Predict algorithms classifier (KNN, DT and MLP) // In prediction features of input image have used.

Step11: Generating conditions // Framing conditions of attack and non-attack using FOR loop.

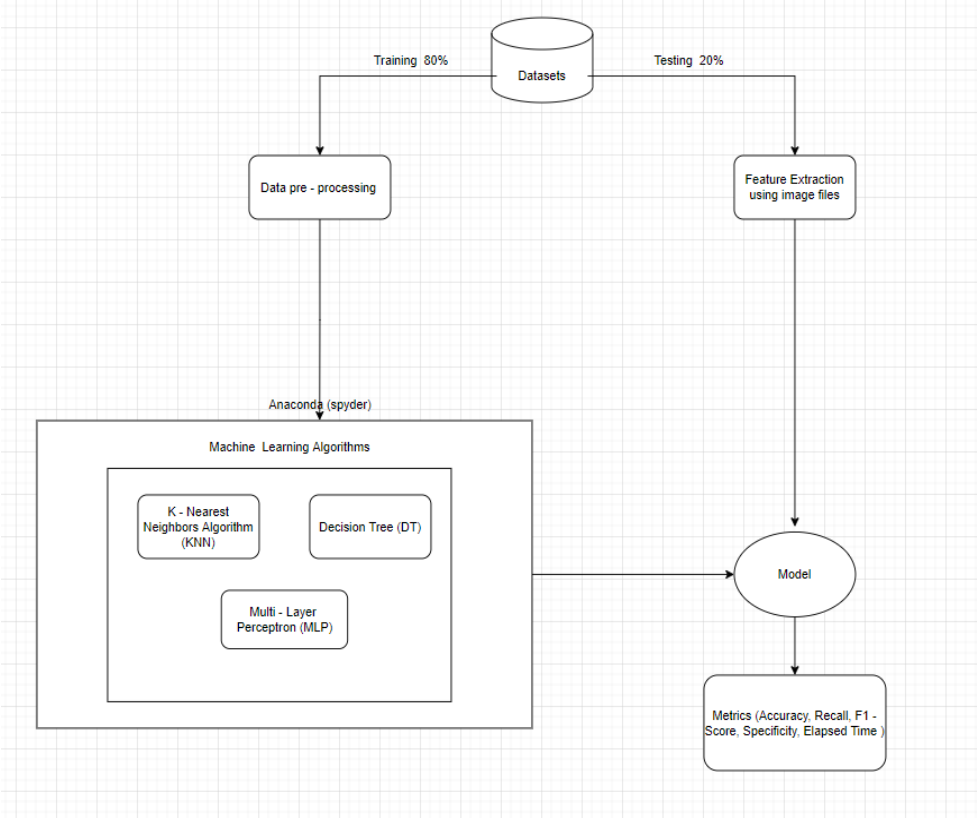
Step12: Generating Accuracy for algorithm classifier (KNN, DT, MLP) // Framing and pass training label of all datasets image and results of prediction into the model.

Step13: Generating F1-Score for algorithm classifier (KNN, DT, MLP) // Framing and pass training label of all datasets image and results of prediction into the model.

Step14: Print the statement of accuracy and F1-Score of an algorithm (KNN, DT, MLP) // Getting the results of induvial and overall results of accuracy and F1-Score of an algorithm (KNN, DT, MLP).
Step14: Print the statement of elapsed time of an algorithm (KNN, DT, MLP) // Getting the results of induvial and overall results of elapsed time of an algorithm (KNN, DT, MLP).
Step15: Generating confusion matrix // Generating confusion matrix by using training label and results of prediction then it will return four values true positive (TP) true negative (TN) false positive (FP) false negative (FN).
Step16: Generating sensitivity or recall and specificity of an algorithm (KNN, DT, MLP) // Generating sensitivity or recall and specificity by using values of values true positive (TP) true negative (TN) false positive (FP) false negative (FN).
Step17: Print the statement of sensitivity or recall and specificity of an algorithm (KNN, DT, MLP) // Getting the results of induvial and overall results of sensitivity or recall and specificity of an algorithm (KNN, DT, MLP).
Step15: Initialize Time taken//

5 Implementation

In this figure depicts that working architecture of this thesis. The datasets are collected from the James z. wang research group website. The collected datasets which contain malicious and benign image files, and these datasets is divided into two sections. The whole datasets are divided into 80% is used for training data and 20% is used for testing data. In this data pre-processing feature extracting of values had done it get the values of mean value, standard deviation, and variance values of an images. The we loaded train features these are Collected from overall features of the datasets and load the features which is used for the machine learning algorithm classifier. In this model we had used aggregation of three machine learning algorithm classifiers K-NN, DT, MLP and this machine learning algorithms used in Spyder tool of anaconda. The model which is developed and used for testing purpose and feature extraction is done on testing data. Then these datasets are applied to the model which is developed during the training data. Based on the training data the model of K-NN, DT, MLP are developed and predicts the results of the metrics like accuracy, f1-score, recall or sensitivity, specificity, and elapsed time



Hardware details
OS Name: windows 11 home single language
System type: x64-based PC
processor 11th Gen core(TM) i7 malicious
RAM:16.0 GB

Software details

Technology used	Version	Illustration
Python	3.10	Python is the programming language which is used create the model and performace of the model.
Anaconda navigator	2.1.1	Anaconda is one of the open source in this many of IDE which is related to the data science are avaiable on open source.
Spyder	2.57	Spyder tool is used because of it gives live output and better perfornce of work and also visualization.

Figure 2: Software details Tabel

File Structure

S.No	Name	Type	purpose
1	Malicious	Folder	In this folder it contains 1,077 malicious image files
2	Benign	Folder	In this folder it contains 2,154 benign image files
3	Mainfile_updated	File	In this file it contain source

			code of the model.
4	Trainfea.pickel	File	In this file it contain source code of the traning along with feature extraction.
5	Training	File	In this file it contain source code of the traning file.

Figure 3: File Structure Tabel

Training Phase

In this training phase the datasets are used and applied to the machine learning algorithms. The 70% datasets are used to train the model. During pre-processing these datasets is used to train the model with malicious and benign files.

```
x_train, x_test, y_train, y_test = train_test_split(dot,labels,test_size = 0.2, random_state = 0)

x_train1=np.zeros((len(x_train),50,50,3))

for i in range(0,len(x_train)):
        x_train1[i,:,:]=x_train[i]
```

Figure 4: Training data

Testing Phase

In this testing phase the datasets are used and applied to the machine learning algorithms. The 20% datasets are used to test the model. These feature extraction values are used and sent into the model which developed in training phase then the model make prediction whether image is malicious or not.

```
x_test1=np.zeros((len(x_test),50,50,3))

for i in range(0,len(x_test)):
        x_test1[i,:,:]=x_test[i]
```

Figure 5: Testing data

K-NN algorithm

In this figure shows implementation of K-NN algorithm model. The dataset as a input is divide into traning and testing phase as Train_features, Y_trains then the model is fitted to the train dataset of input (Class_KNN = neigh.predict (features)). In this predict holds results of K-NN classifier

```
# -- KNN Classifier

from sklearn.neighbors import KNeighborsClassifier
start = timer()

neigh = KNeighborsClassifier(n_neighbors=3)

# Fitting the model
neigh.fit(Train_features, y_trains)

# Predict the model
Class_KNN = neigh.predict(Features)

end1 = timer()
print(end1 - start)
```

Figure 6: K-NN Classifier

DT algorithm

In this figure shows implementation of DT algorithm model. The dataset as a input is divide into traning and testing phase as Train_features, Y_trains then the model is fitted to the train dataset of input (Class_DT = clf1.predict (features)). In this predict holds results of DT classifier

```
#-- Decision Tree Classifier

from sklearn import tree
start = timer()

clf1 = tree.DecisionTreeClassifier()

# Fitting the model
clf1 = clf1.fit(Train_features, y_trains)

# Predict the model
Class_DT = clf1.predict(Features)

end2 = timer()
print(end2 - start)
```

Figure 7: DT Classifier

MLP algorithm

In this figure shows implementation of MLP algorithm model. The dataset as a input is divide into traning and testing phase as Train_features, Y_trains then the model is fitted to the train dataset of input (Class_MLP = clf.predict (features)). In this predict holds results of MLP classifier

```
# -- Multi Layer Perceptron Classifier

from sklearn.neural_network import MLPClassifier

start = timer()

# Fitting the model
clf = MLPClassifier(random_state=1, max_iter=77).fit(Train_features, y_trains)

# Predict the model
Class_MLP = clf.predict(Features)


end3 = timer()
print(end3 - start)
```

Figure 8: MLP Classifier

6 Evaluation

In this section will discuss the results achieved by the implementation of machine learning algorithms of K-NN, DT, and MLP and will compare results of each algorithm classifiers in terms of accuracy, F1-score, sensitivity or recall, specificity, and elapsed time.

Confusion matrix

It is a matrix is developed based on the result achieved by the machine learning algorithms classifier. This confusion matrix which is used to describe the overall performance of the machine learning algorithms classifiers. It also provides easy classification between various classes of result in this section. In this confusion matrix the four boxes describe that the counts of correct and incorrect data which are classified data, and this represents the classifier of actual state while making prediction.

	Class 1 Predicted	Class 2 Predicted
Class 1 Actual	TP	FN
Class 2 Actual	FP	TN

Figure 9: Confusion Metric Tabel

TP (True Positive): The actual data is positive, and it is predicted positive by our machine learning algorithm classifiers.
TN (True Negative): The actual data is negative, and it is predicted negative by our machine learning algorithm classifiers.
FP (False positive): The actual data is negative, and it is predicted positive by our machine learning algorithm classifiers.
FN (False Negative): The actual data is positive, and it is predicted negative by our machine learning algorithm classifiers.

6.1 Accuracy

The accuracy is measured by the number of correct predictions is done by the classifiers and divided by the total prediction done by the classifiers

Accuracy = TP+TN/TP+TN+FP+FN

```
Accuracy of MLP =  95.94552770040235  %
Accuracy of KNN =  95.140823274528  %
Accuracy of DT  =  99.93809965954813  %

Overall Accuracy =  97.00815021149283  %
```

Figure 10: Accuracy results of indivual and overall machine learning algorithms

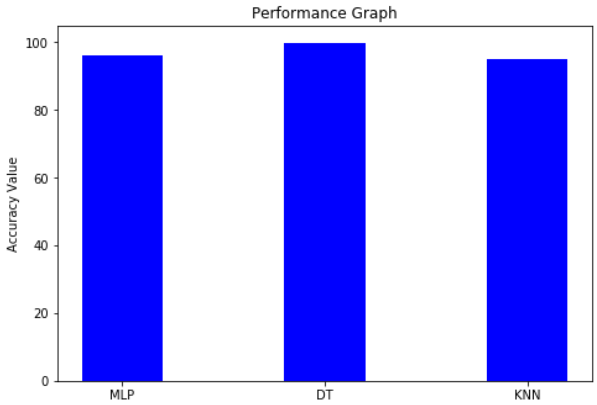


Figure 11: Graphical represntation of accuracy of an indivual and overall machine learning algorithms

6.2 F1-Score

The F1-score is measured, and the value is achieved by precision and recall, or sensitivity and it is a harmonic mean.
F1=2*Precision * recall / Precision + recall

```
F-Score of MLP =  95.91504294627003  %
F-Score of KNN =  93.82028267822139  %
F-Score of DT  =  99.91746621273084  %

Overall F-Score =  96.55093061240741  %
```

Figure 12: F1-Score results of indivual and overall machine learning algorithms

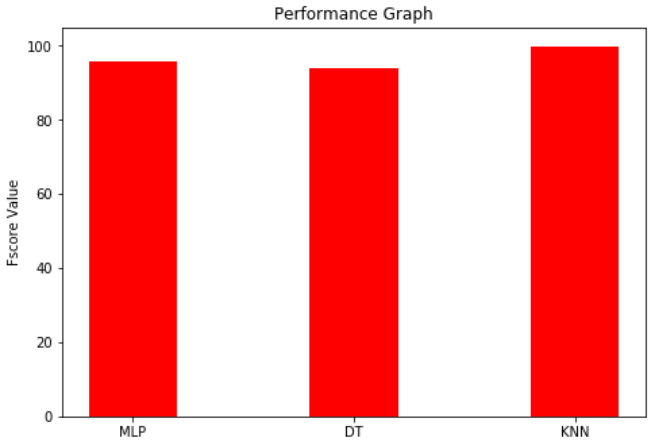


Figure 11: Graphical represntation of F1-Score of an indivual and overall machine learning algorithms

6.3 Sensitivity or Recall

Sensitivity is also known as recall it is measured by the data points this are predicted as positive from the total predictions which is known as positives by the classifiers.
Recall = True positive / True positive + False Negative

```
Sensitivity MLP =  50.0  %
Sensitivity KNN =  50.0  %
Sensitivity DT =  50.0  %
-----------------------------------------

Overall Sensitivity =  50.0  %
```

Figure 12: Sensitivity or recall results of indivual and overall machine learning algorithms

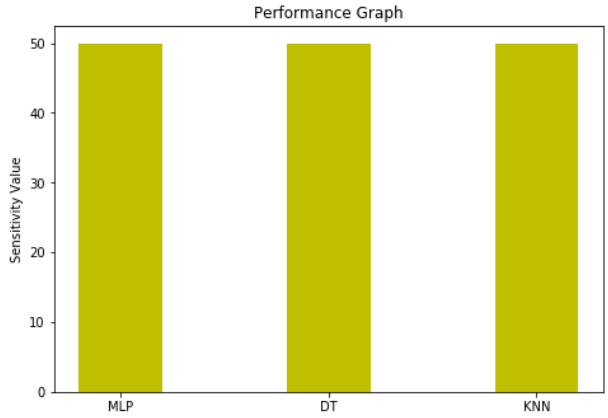


Figure 13: Graphical represntation of sensitivity or recall of an indivual and overall machine learning algorithms

6.4 Specificity

The specificity is measured and predict values of true negative in which that are correctly identified by the model.
Specificity = True Positives / True Positive + False Negative

```
Specificity MLP =  100.0  %
Specificity KNN =  100.0  %
Specificity DT =  50.0  %

Overall Specificity =  83.33333333333334  %
```

Figure 14: Specificty results of indivual and overall machine learning algorithms

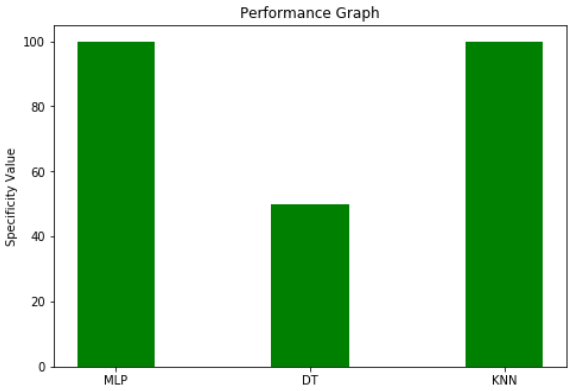


Figure 15: Graphical represntation of specificity of an indivual and overall machine learning algorithms

6.5 Elapsed Time

The elapsed time is measured the amount of actual time from the start of an event up to its finish.
Elapsed time = End Time – Start Time

```
Elapsed Time of MLP =  0.00127348521  Seconds
Elapsed Time of DT  =  0.004271025548333334  Seconds
Elapsed Time of KNN =  0.005340847463333334  Seconds

Overall Time Taken For Thee Algorithms =  21770.716443333335  Seconds
```

Figure 16: Specificty results of indivual and overall machine learning algorithms

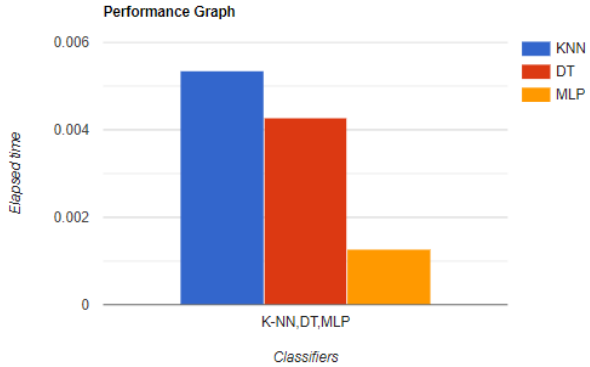


Figure 17: Graphical represntation of elapsed time of an indivual and overall machine learning algorithms

Test Cases	Classifiers Used	Result	Reasons
1	Random Forest Algorithm	unsuccessful	The model didn't recognize malicious or not for a singular image and accuracy is

			80%
2	RF and DT Algorithms	successful	The model recognizes malicious or not for a singular image and accuracy is 90%
3	RF, DT and KNN	unsuccessful	The model didn't recognize malicious or not for an all-image files and accuracy is 80%
4	RF, DT and Nive Bayes	successful	The model recognizes malicious or not for an all-image files and accuracy is 93%
5	k-NN, DT, MLP Algorithms	Successful	The model recognizes malicious or not for an all-image files and accuracy is 97%

Figure 18: Different times of test conducted

6.6 Discussion

In this thesis, we developed the model by using K-Nearest Neighbors, Decision Tree, and Multi-layer Perceptron as described in the table the results of classifiers had high detection rate. The aggregation of the K-Nearest Neighbors, Decision Tree, and Multi-layer Perceptron algorithms had improvised the performance in the model by training data and the accuracy of the model is 97%. In our thesis we had tried aggregation of two algorithms for each images files, but the result of the model was unsuccessful. After done few of the test cases we got optimized results which include all kinds of metrics like f1-score is 96.5%, sensitivity or recall is 50%, specificity is 83.3% and time taken by the model is 24121.7 sec. We can that machine learning algorithms which are used in this model tends to perform well and developed with this kind of problem in which detecting malicious codes in the images.

S.No	Existing Work	Algorithms Used	Accuracy	F1-Score	Sensitivity (or) Recall	specificity	Elapsed Time
1	Mohd Ariffin et al., 2018	K-NN, RF, and DT	84.14%	N/A	N/A	N/A	N/A
2	Jin et al., 2020	Autoencoders	96%	N/A	N/A	N/A	N/A
3	Iqbal *et al.*, 2021	CNN	93%	N/A	N/A	N/A	N/A
4	Proposed work	K-NN, DT, and MLP	97%%	96.5%	50%	83.3%	24121.7 sec

Figure 19: Validation Tabel

7 Conclusion and Future Work

For this thesis we haves used 3200 datasets image files out of it 1,077 are malicious and 2,154 are benign files. These datasets are divided and applied to the training data and testing data. In training data 80% datasets are used and to train model. This model is developed by using machine learning algorithms K-NN, DT and MLP. Then 20% datasets are used and test the model. After that we found the aggregation of this model predict high accuracy 97% which includes other metrics like F1-score, recall or sensitivity, specificity, and elapsed time. Overall, we found that the aggregation of three machine learning classifiers K-NN, DT, and

MLP are performed well to get malicious and benign files. By performing various study of algorithms in this thesis, this literature is successful in that finding various types of machine learning algorithms which can boost the accuracy for malware detection on images. This model developed and produced best accuracy as compared to the other literature were discussed according to the result in evaluation section of this thesis.

In this model showed that machine learning algorithms K-NN, DT, and MLP are performed well while detecting the malware in the image. In future a hybrid technique of learning method can be used to increase the detection rate of malicious code in image files. The technique of hybrid learning method would be able to use deep learning or stacking approach in which it consists of combination of any machine learning algorithms.

References

Cakir, B. and Dogdu, E. (2018) "Malware classification using deep learning methods," in Proceedings of the ACMSE 2018 Conference. New York, NY, USA, 29 March 2018, pp. 1–5, ACM. doi: 10.1145/3190645.3190692.

Chen, J. (2020) "A malware detection method based on rgb image," in Proceedings of the 2020 6th International Conference on Computing and Artificial Intelligence. New York, NY, USA: 20 August 2020, pp. 283-290, ACM. doi: 10.1145/3404555.3404622.

Chen, M. et al. (2017) "JPEG-Phase-Aware Convolutional Neural Network for Steganalysis of JPEG Images," in Proceedings of the 5th ACM Workshop on Information Hiding and Multimedia Security. New York, NY, USA, 20 June 2017, pp. 75-84, ACM. doi: 10.1145/3082031.3083248.

Choi, S. et al. (2017) "Malware detection using malware image and deep learning," in 2017 International Conference on Information and Communication Technology Convergence (ICTC). Jeju, Korea (South), 18-20 October 2017, pp. 1193–1195, IEEE Xplore. doi: 10.1109/ICTC.2017.8190895.

Darus, F. M., Salleh, N. A. A. and Mohd Ariffin, A. F. (2018) "Android malware detection using machine learning on image patterns," in 2018 Cyber Resilience Conference (CRC). Putrajaya, Malaysia, 13-15 November 2018, pp. 1–2, IEEE Xplore, doi: 10.1109/CR.2018.8626828.

Duan, Z. (2018) "Characters Recognition of Binary Image using KNN," in Proceedings of the 4th International Conference on Virtual Reality - ICVR 2018. New York, New York, USA, 27-24 February 2018, pp. 116–118, ACM Press. doi: 10.1145/3198910.3234651.

Gupta, S., Bansal, P. and Kumar, S. (2018) "ULBP-RF: A hybrid approach for malware image classification," in 2018 Fifth International Conference on Parallel, Distributed and Grid Computing (PDGC). Solan, India, 20-22 December 2018, pp. 115–119, IEEE Xplore. doi: 10.1109/PDGC.2018.8745989.

He, K. and Kim, D.-S. (2019) "Malware detection with malware images using deep learning techniques," in 2019 18th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/13th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE). Rotorua, New Zealand, 05-08 August 2019, pp. 95–102, IEEE Xplore. doi: 10.1109/TrustCom/BigDataSE.2019.00022.

Iqbal, A. et al. (2021) "Malicious image detection using convolutional neural network," in 2021 International Conference on Artificial Intelligence and Mechatronics Systems (AIMS). Bandung, Indonesia, 28-30 April 2021, pp. 1–6, IEEE Xplore. doi: 10.1109/AIMS52415.2021.9466042

Jin, X. et al. (2020) "A Malware Detection Approach Using Malware Images and Autoencoders," in 2020 IEEE 17th International Conference on Mobile Ad Hoc and Sensor

Systems (MASS). Delhi, India, 10-13 December 2020, pp. 1–6, IEEE Xplore. doi: 10.1109/MASS50613.2020.00009.
Kosmidis, K. and Kalloniatis, C. (2017) "Machine Learning and Images for Malware Detection and Classification," in Proceedings of the 21st Pan-Hellenic Conference on Informatics. New York, NY, USA, 28 September 2017, pp. 1-6, ACM. doi: 10.1145/3139367.3139400.
Krithika, V. and Vijaya, M. S. (2020) "Malware detection using gist features and deep neural network," in 2020 6th International Conference on Advanced Computing and Communication Systems (ICACCS). Coimbatore, India, 06-07 March 2020, pp. 800–805, IEEE Xplore. doi: 10.1109/ICACCS48705.2020.9074325.
Kumar, R. et al. (2018) "Malicious code detection based on image processing using deep learning," in Proceedings of the 2018 International Conference on Computing and Artificial Intelligence - ICCAI 2018. New York, New York, USA, 12 March 2018, pp. 81–85, ACM Press. doi: 10.1145/3194452.3194459.
Liu, L. and Wang, B. (2016) 'Malware classification using gray-scale images and ensemble learning,' in 2016 3rd International Conference on Systems and Informatics (ICSAI). Shanghai, China, 19-21 November 2016, pp. 1018–1022, IEEE Xplore. doi: 10.1109/ICSAI.2016.7811100.

Mathew, A. B. and Kurian, S. (2020) "Identification of malicious code variants using spp-net model and color images," in 2020 IEEE 15th International Conference on Industrial and Information Systems (ICIIS). RUPNAGAR, India, 26-28 November 2020, pp. 581–585, IEEE Xplore. doi: 10.1109/ICIIS51140.2020.9342648.
Mosli, R. *et al.* (2016) 'Automated malware detection using artifacts in forensic memory images', in *2016 IEEE Symposium on Technologies for Homeland Security (HST)*. Waltham, MA, USA, 10-11 May 2016, pp. 1–6, IEEE Xplore. doi: 10.1109/THS.2016.7568881.
Narayanan, B. N., Djaneye-Boundjou, O. and Kebede, T. M. (2016) 'Performance analysis of machine learning and pattern recognition algorithms for Malware classification,' in 2016 IEEE National Aerospace and Electronics Conference (NAECON) and Ohio Innovation Summit (OIS). Dayton, OH, USA, 25-29 July 2016, pp. 338–342, IEEE Xplore. doi: 10.1109/NAECON.2016.7856826.
Ozkan, K., Isik, S. and Kartal, Y. (2018) "Evaluation of convolutional neural network features for malware detection," in 2018 6th International Symposium on Digital Forensic and Security (ISDFS). Antalya, Turkey, 22-25 March 2018, pp. 1–5, IEEE Xplore. doi: 10.1109/ISDFS.2018.8355390.
Pinto, D. R., Duarte, J. C. and Sant'Ana, R. (2019) "A deep learning approach to the malware classification problem using autoencoders," in Proceedings of the XV Brazilian Symposium on Information Systems - SBSI'19. New York, New York, USA: 20 May 2020, pp. 1–8, ACM Press. Doi: 10.1145/3330204.3330229
Poonguzhali, N. P. et al. (2019) "Identification of malware using CNN and bio-inspired technique," in 2019 IEEE International Conference on System, Computation, Automation and Networking (ICSCAN). Pondicherry, India, 29-30 March 2019, pp. 1–5, IEEE Xplore. doi: 10.1109/ICSCAN.2019.8878696.
Roseline, S. A. et al. (2019) "Towards efficient malware detection and classification using multilayered random forest ensemble technique," in 2019 International Carnahan Conference on Security Technology (ICCST). Chennai, India, 01-03 October 2019, pp. 1–6, IEEE Xplore. doi: 10.1109/CCST.2019.8888406.
Samuel, H. D. et al. (2022) "Automation detection of malware and stenographical content using machine learning," in 2022 6th International Conference on Computing Methodologies

and Communication (ICCMC). Erode, India, 29-31 March 2022, pp. 889–894, IEEE Xplore. doi: 10.1109/ICCMC53470.2022.9754063.
Shao Yang. (2019) “An image-inspired and CNN-based android malware detection approach,” in 2019 34th IEEE/ACM International Conference on Automated Software Engineering (ASE). 7 February 2019, pp. 1259–1261, ACM. doi: 10.1109/ASE.2019.00155
Xu, M. et al. (2021) “Malicious code detection method based on multiple features,” in 2021 IEEE 4th International Conference on Electronics and Communication Engineering (ICECE). Xi'an, China, 17-19 December 2021, pp. 8–15, IEEE Xplore. doi: 10.1109/ICECE54449.2021.9674573.
Zhou, X., Pang, J. and Liang, G. (2017) “Image classification for malware detection using extremely randomized trees,” in 2017 11th IEEE International Conference on Anti-counterfeiting, Security, and Identification (ASID). Xiamen, China, 27-29 October 2017, pp. 54–59, IEEE Xplore. doi: 10.1109/ICASID.2017.8285743.