
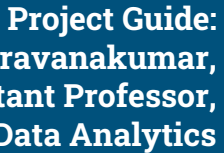


# Malware Detection System using Convolutional Neural Networks



Minor Project By  
Sachin S - 20BDA036,  
III-BDA

Project Guide:  
Mr.S.M.Saravanakumar,  
Assistant Professor,  
Computer Science with Data Analytics



# About the project

In recent years, the amount of malware spreading through the internet and infecting computers and other communication devices has tremendously increased. My proposal is to develop an automated framework that can identify malware by leveraging current neural network techniques. This has a significant and immediate value for the security field, as current antivirus software is typically able to recognize the malware type only after its infection, and preventive measures are limited.

---

# Hardware and Software Specifications

---

Processor: AMD Ryzen 7 5800H with 4.1Ghz Clock Speed.

Graphics Card: Nvidia RTX 3060 Max Q.

RAM: 16GB DDR4.

Operating System: Ubuntu Linux 22.04.1 LTS

Language: Python with Jupyter Notebook.

Frameworks used: TensorFlow, Flask.

Dataset: 25760 malware samples



# Feasibility Analysis

## Operational Feasibility

---

This project aims to present a model that is the simplest solution for image-based malware detection which requires no special transformation of the images from binaries, no data augmentation or feature engineering, and no complex architecture.

## Technical Feasibility

---

I propose a convolutional neural network (CNN) based on deep learning framework for image-based identification of malwares. This can be achieved by conversion of the malware sample into images.

# Feasibility Analysis

## Economic Feasibility

The project involves analysing malware samples by converting it into images. Processing the model once using a high specifications computer and we can use it any other normal computer systems. Hence, the project is economically possible.

# Existing System



## Existing System

Existing malware classifiers use Signatures to identify malwares. Malware within a family shares similar properties that can be used to create signatures for detection and classification. Signatures can be categorized as static, or dynamic based on how they are extracted. A static signature can be based on a byte-code sequence, binary assembly instruction, or an imported Dynamic Link Library (DLL).



## Proposed System

Unlike more traditional methods of machine learning techniques, deep learning classifiers are trained through feature learning rather than task-specific algorithms. What this means is that the machine will learn patterns in the images that it is presented with rather than requiring the human operator to define the patterns that the machine should look for in the image. In short, it can automatically extract features and classify data into various classes.

# Modules

1.Data Preprocessing

2.Model Training

3.Web Interface using Python



# Data Preprocessing

---

We obtain malware samples from Kaggle which is executable files, and these data should be converted into image files to involve it into the Convolutional layers and further classification. For each file, the raw data contains the hexadecimal representation of the file's binary content. The goal is to convert those files into PNG images and use them as the input of our CNN.





# Model Training

---

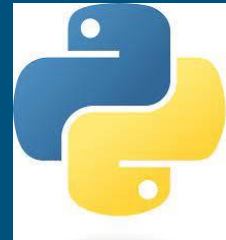
This section deals with training of the model. Now after preprocessing and converting our dataset, we can build our model using Keras Framework in python.



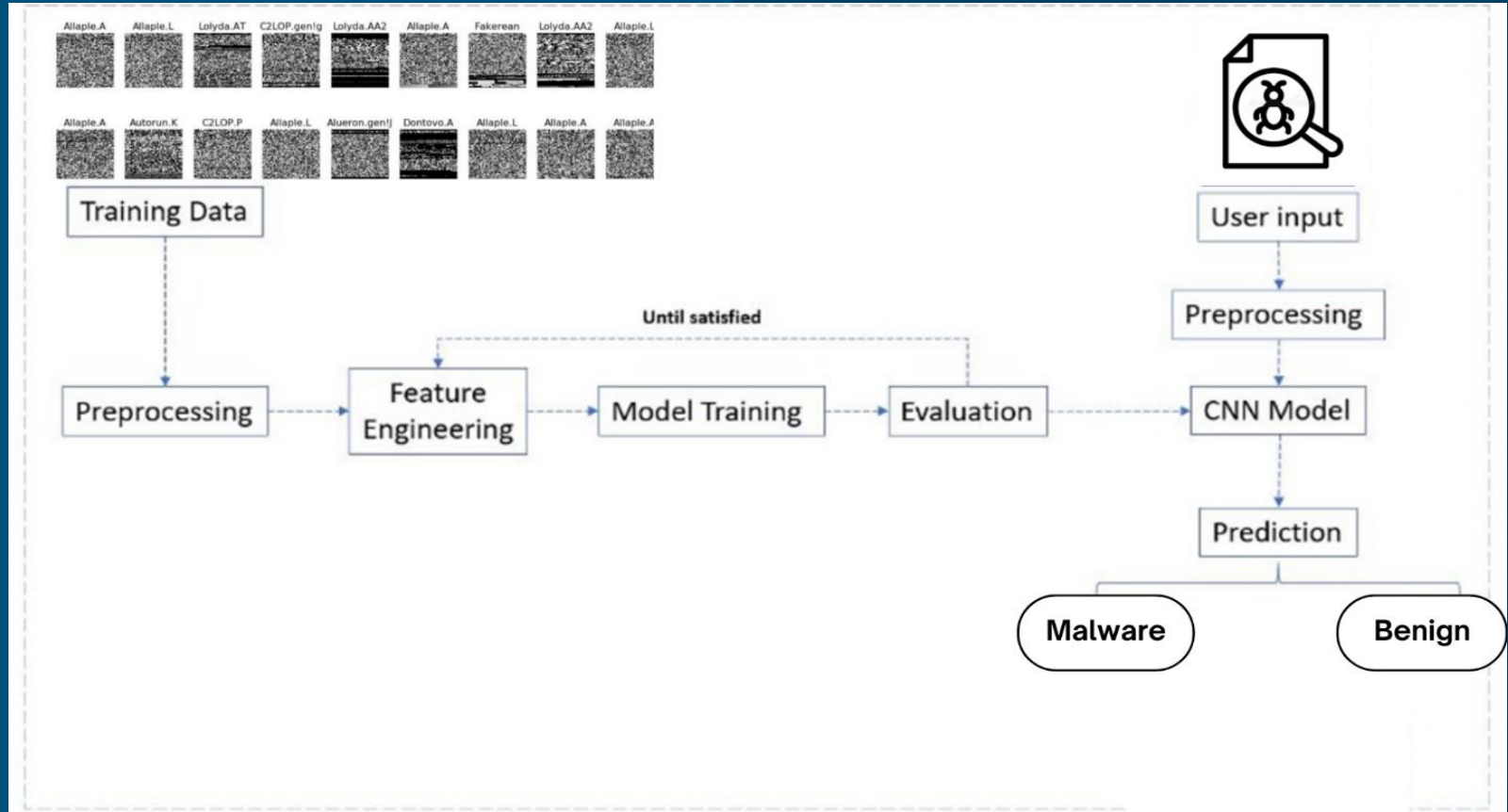
# Web Interface using python

---

Flask is a way for web servers to pass requests to web applications or frameworks. Flask relies on the WSGI external library to function, as well as the Jinja2 template engine. In this module, the processed model is connected to Flask API and a web interface is created to insert an executable file which is the test data to test whether the file is a malware or a normal executable file.



# System Architecture



# Sample Input Page

Malware Detector 

Analysis board >

## Malware Detection System

Analyze Files

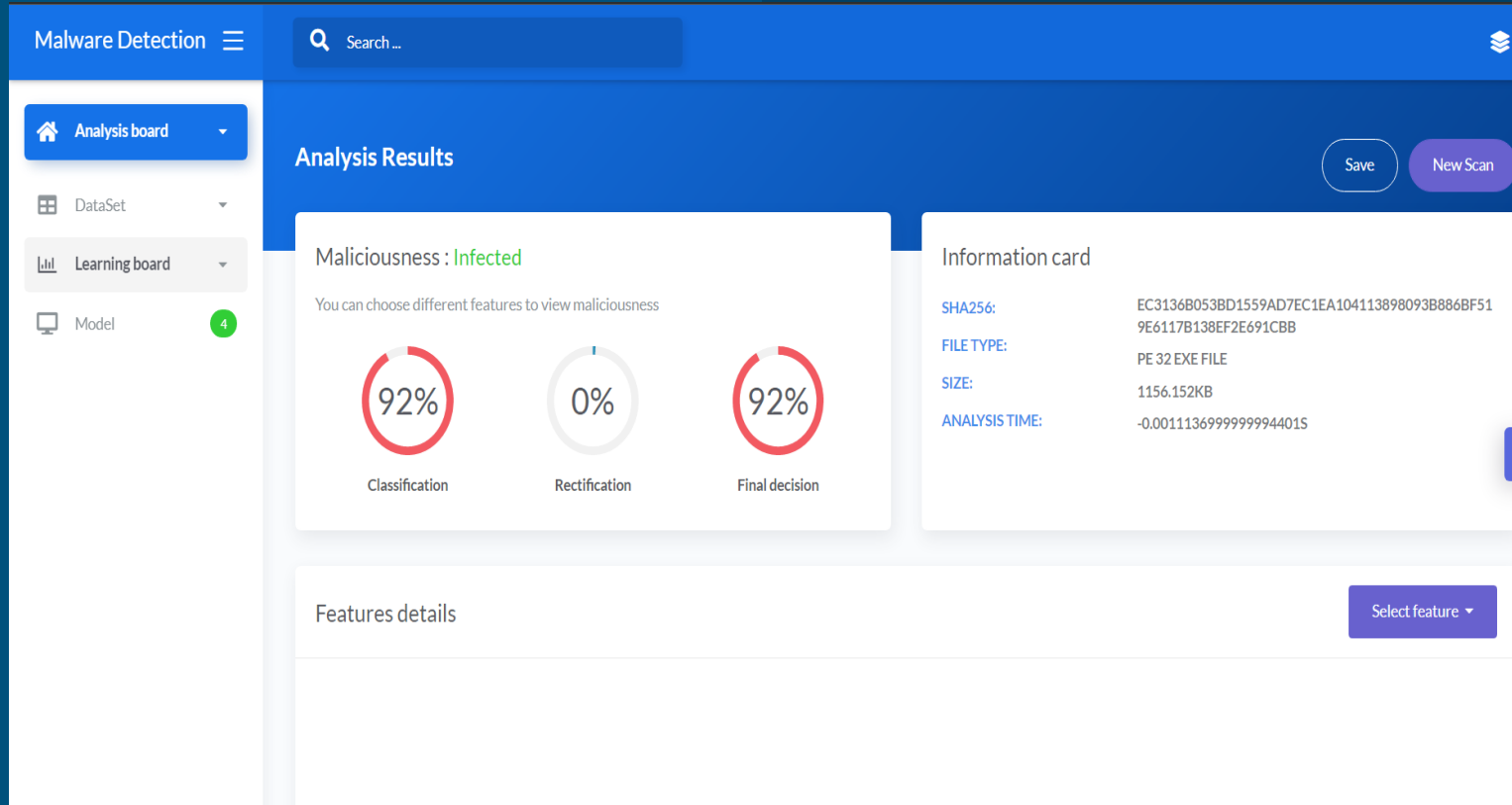
This feature supports only valid Windows EXE Files



Choose file

No file chosen

# Output Page



Thank You

---