G. H. RAISONI COLLEGE OF ENGINEERING

(An Autonomous Institute affiliated to Rashtrasant Tukdoji Maharaj Nagpur University, Nagpur)
Dual Accreditations NAAC "A+" Grade & NBA (Tier-I)
CRPF Gate No. 3, Hingna Road, Digdoh Hills, Nagpur – 440 016. (INDIA)
Phone: +91 9604787184, 9689903286 E-mail: principal.ghrce@raisoni.net Web: ghrce.raisoni.net

Department of Artificial Intelligence

Malware Detection System Using Deep Learning

Team Member:- Abhiroop Sarkar(16), Prateek Dutta(47), Adnan Quraishee(22), Saurabh Barse(54) - Semester-8

Project Guide:- Prof. Achamma Thomas

Abstract:

The malware industry continues to be a well-organized, well-funded market dedicated to evading traditional security measures. Once a computer is infected by malware, criminals can hurt consumers and enterprises in many ways. Malware is one of the most serious security threats on the Internet today. In fact, most Internet problems such as spam e-mails and denial of service attacks have malware as their underlying cause. That is, computers that are compromised with malware are often networked together to form botnets, and many attacks are launched using these malicious, attacker-controlled networks.

Malware is a persistent and growing threat, and traditional security measures can struggle to keep pace with the ever-evolving techniques used by malware authors. Fortunately, machine learning and deep learning has shown promise as a means of detecting and preventing malware infections.

One of the key advantages of machine-based approaches is that they can adapt to new and previously unseen malware threats, whereas traditional signature-based approaches can only detect known malware variants. Models can be trained on large datasets of both benign and malicious software to learn the characteristics that distinguish malware from legitimate software, allowing them to identify new, unknown malware based on these learned patterns.

Deep learning models are particularly well-suited for identifying patterns and features in complex, high-dimensional data, such as the raw binary code of software applications, which can be a challenging task for traditional machine learning algorithms. These networks learn to extract relevant features from the binary code, such as opcode sequences and control flow graphs, that can be used to distinguish between malicious and legitimate software. Once trained, these models can be used to classify new, previously unseen software as either malware or benign.

<u>About:</u>

Why Deep Learning?

Nowadays deep learning has dominated the various computer vision tasks. Not only these deep learning techniques enabled rapid progress in this competition, but even surpassed human performance in many of them. One of these tasks is Image Classification. 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.

Early layers learn how to detect low-level features like edges, and subsequent layers combine features from earlier layers into a more holistic and complete representation. We can transform a malware/benign file into a grayscale image using the method described later. Then we can apply these deep learning techniques on the generated images to classify them as malware or benign.

Types of Malware

Types	**Purpose**	**Example**
Spyware	Spyware collects its victims' user activity data without their knowledge.	DarkHotel
Adware	Adwares serves unwanted advertisements. It generates revenue for its developers by automatically generating adverts on your screen, usually within a web browser.	Fireball
Trojan	Trojan disguises itself as a desirable code. It misleads users of its true intent	Emotet
Ransomware	Ransomwares purpose is to disable its victims' access to data until the ransom is paid.	RYUK
Rootkits	Rootkits gives the remote control of Victims Device.	Zacinlo
Keyloggers	Keylogger is a type of surveillance technology used to monitor and record each keystroke on a specific computer.	Olympic Vision
Wiper Malware	Wiper Malware is built with the sole purpose to erase the victims data beyond recoverability.	WhisperGate
Bots	Bots launch a broad flood of attacks. Malware	Echobot

	bots and internet bots can be programmed/hacked to break into user accounts, scan the internet for contact information, to send spam, or perform other harmful acts.	
Mobile Malware	These infect mobile devices. It targets mobile phones or wireless-enabled Personal digital assistants, by causing the collapse of the system and loss or leakage of confidential information.	Triada
Fileless Malware	It makes changes to the files native of the OS. It does not rely on files and leaves no footprint, making it challenging to detect and remove.	Astaroth
Worms	Worms spread throughout the network by replicating itself.	Stuxnet

Dataset:

Data we are approaching

We are using EMBER (Elastic Malware Benchmark for Empowering Researchers) dataset for our project and research purpose.The EMBER dataset is a collection of features from PE (Portable Executable) files that serve as a benchmark dataset for researchers. The EMBER2017 dataset contained features from 1.1 million PE files scanned in or before 2017 and the EMBER2018 dataset contains features from 1 million PE files scanned in or before 2018. This repository makes it easy to reproducibly train the benchmark models, extend the provided feature set, or classify new PE files with the benchmark models.

A labeled benchmark dataset for training machine learning models to statically detect malicious Windows portable executable files. The dataset includes features extracted from 1.1M binary files: 900K training samples (300K malicious, 300K benign, 300K unlabeled) and 200K test samples (100K malicious, 100K benign). Our aim is to do comparative analysis on this dataset using different machine learning and deep learning algorithms to derive new insights, which leads to optimized performance in an experiment.

EMBER github Source :- https://github.com/elastic/ember
EMBER paper Source :- https://arxiv.org/abs/1804.04637

DATA Layout Format:-


```
"sha256": "000185977be72c8b007ac347b73ceb1ba3e5e4dae4fe98d4f2ea92250f7f580e",
"appeared": "2017-01",
"label": -1,
"general": {
  "file_size": 33334,
  "vsize": 45056,
  "has_debug": 0,
  "exports": 0,
  "imports": 41,
  "has_relocations": 1,
  "has_resources": 0,
  "has_signature": 0,
  "has_tls": 0,
  "symbols": 0
},
"header": {
  "coff": {
    "timestamp": 1365446976,
    "machine": "I386",
    "characteristics": [ "LARGE_ADDRESS_AWARE", ..., "EXECUTABLE_IMAGE" ]
  },
  "optional": {
    "subsystem": "WINDOWS_CUI",
    "dll_characteristics": [ "DYNAMIC_BASE", ..., "TERMINAL_SERVER_AWARE" ],
    "magic": "PE32",
    "major_image_version": 1,
    "minor_image_version": 2,
    "major_linker_version": 11,
    "minor_linker_version": 0,
    "major_operating_system_version": 6,
    "minor_operating_system_version": 0,
    "major_subsystem_version": 6,
    "minor_subsystem_version": 0,
    "sizeof_code": 3584,
    "sizeof_headers": 1024,
    "sizeof_heap_commit": 4096
  }
},
```


Fig 1 :Raw features extracted from a single PE file.

The EMBER dataset consists of a collection of JSON lines files, where each line contains a single JSON object. Each object includes the following types in data:

• The sha256 hash of the original file as a unique identifier;
• Coarse time information (month resolution) that establishes an estimate of when the file was first seen;
• A label, which may be 0 for benign, 1 for malicious or -1 for unlabeled; and
• Eight groups of raw features that include both parsed values as well as format-agnostic histograms.

<u>Methodology:</u>

Malware, short for "malicious software," is a type of software that is intentionally designed to cause harm to a computer system, network, or device. Malware can take many forms, including viruses, worms, Trojans, ransomware, spyware, adware, and rootkits.

Malware can be spread through various methods, such as email attachments, malicious websites, social engineering, or exploiting vulnerabilities in software or systems. Once malware infects a system, it can cause a range of problems, including data theft, system malfunction, loss of data, unauthorized access, and financial loss. Malware can also be used to launch attacks on other systems or networks, creating further damage and disruption.

Preventing and detecting malware is an important part of cybersecurity. Antivirus software, firewalls, and other security measures can help protect against malware, while malware analysis and detection techniques such as signature-based detection, behavioral analysis, and machine learning can help identify and remove malware from infected systems.

Convolutional Neural Networks (CNNs) have been successfully used for malware detection due to their ability to extract relevant features from raw data, such as binary code or network traffic. Here are some methodologies for malware detection using CNNs:

- Raw byte sequence: In this approach, the raw byte sequence of a malware file is treated as an image and fed into a CNN for classification. The CNN learns to recognize patterns and features in the byte sequence and can detect malware with high accuracy.

- Image-based malware classification: In this approach, malware files are converted into grayscale or RGB images, and then fed into a CNN for classification. The CNN learns to identify patterns and features in the images that are characteristic of malware, such as obfuscation or encryption.

- Opcode sequence analysis: In this approach, the opcode sequence of a malware file is treated as a time-series data and fed into a CNN for classification. The CNN learns to recognize patterns and features in the opcode sequences that are characteristic of malware, such as packing or encryption.

- Network traffic analysis: In this approach, network traffic generated by malware is captured and fed into a CNN for classification. The CNN learns to recognize patterns and features in the network traffic that are characteristic of malware, such as C&C communications or data exfiltration.

In all of these methodologies, the CNN learns to identify the patterns and features that are characteristic of malware, and can distinguish between benign and malicious software with high accuracy. By using deep learning techniques, these methodologies can improve the effectiveness and efficiency of malware detection, making them increasingly popular in the field of cybersecurity.

Here is an algorithm for using a Convolutional Neural Network (CNN) for malware detection:

- Preprocessing: The malware files are preprocessed to extract features such as raw byte sequences or opcode sequences.

- Data splitting: The dataset is divided into training, validation, and test sets.

- Model architecture: The CNN model is designed with convolutional, pooling, and fully connected layers. The number of layers and their hyperparameters can vary depending on the dataset and problem.

- Model training: The CNN is trained using the training set, and the validation set is used to monitor the model's performance during training. The loss function and optimization algorithm are selected based on the problem.

- Model evaluation: Once the CNN is trained, the test set is used to evaluate its performance. Metrics such as accuracy, precision, recall, and F1 score are used to evaluate the model's performance.

- Hyperparameter tuning: The hyperparameters of the CNN, such as the learning rate, dropout rate, and number of filters, are tuned to optimize the model's performance.

- Prediction: Once the model is trained and evaluated, it can be used to predict whether new files are benign or malicious.

Overall, this algorithm involves preprocessing the data, designing and training the CNN model, evaluating its performance, and using it for prediction. The use of a CNN for malware detection can improve the accuracy and efficiency of malware detection, making it an increasingly popular approach in the field of cybersecurity.

Conclusion:

Malware detection using deep learning has several advantages over traditional methods of malware detection:

- Higher accuracy: Deep learning models are capable of detecting previously unknown malware with high accuracy. Traditional signature-based methods are limited by their ability to detect only known malware, whereas deep learning models can learn to detect new and unknown types of malware based on their behavioral patterns.

- Automated feature extraction: Deep learning models can automatically learn the features of malware and do not require human experts to manually design features. This makes the process more efficient and effective.

- Scalability: Deep learning models can handle large datasets and are highly scalable, making them well-suited for analyzing and detecting malware on a large scale.

- Adaptability: Deep learning models can adapt to changing patterns of malware behavior and are able to update themselves accordingly, whereas traditional methods require manual updates.

- Reduced false positives: Deep learning models can reduce the number of false positives by learning to distinguish between benign and malicious behavior based on their learned features and patterns.

Overall, malware detection using deep learning provides a more effective and efficient solution to the problem of malware detection, making it an increasingly popular approach in the field of cybersecurity.

GitHub Repository:- https://github.com/project2sem8ghrceAI