Malware Detection

CSIT375/975 AI and Cybersecurity

Dr Wei Zong

SCIT University of Wollongong

Disclaimer: The presentation materials come from various sources. For further information, check the references section

Outline

- Types of malware
- Malware detection strategies
- Public dataset for malware detection
- A novel deep learning-based approach for malware detection

Malware Detection

- Malware (malicious software): any intrusive software developed by cybercriminals to steal data and damage or destroy computers and computer systems
- Examples of common malware include
  - virus
  - trojan
  - botnet
  - downloader
  - rootkit
  - ransomware
  - APT
  - spyware
  - zero day

Malware Detection

- Real-world threats
  - During 2022, IBM Security studied 550 organizations across 17 countries impacted by data breaches and found the average total cost was USD 4.35 million per incident.
  - The AV Test Institute registers 450,000 new malware and potentially unwanted applications every day.
  - the total number of malware targeting Windows, Android, Mac OS, and Linux has more than doubled from 450 million in 2018 to 970 million in 2022.
  - Dark web marketplaces offer numerous and diverse hacking products and services.

Gaber, M.G., Ahmed, M. and Janicke, H., 2024. Malware detection with artificial intelligence: A systematic literature review. ACM Computing Surveys, 56(6), pp. 1-33.

Common Types of Malware

- **Virus**
  - A self-replicating program that reproduces its code by attaching copies into other executable codes, designed to disrupt a system's ability to operate
- **Trojan**
  - Executable that appears as legitimate and harmless, but once it is launched, it executes malicious instructions in the background
- **Botnet**
  - Malware that has the goal of compromising as many possible hosts of a network, in order to put their computational capacity at the service of the attacker
- **Downloader**
  - Malware that downloads malicious libraries or portions of code from the network and executes them on victim hosts
  - Malicious code that exists only to download other malicious code

Common Types of Malware

- **Rootkit**
    - Malware that compromises the hosts at the operating system level and often comes in the form of device drivers, making the various countermeasures (e.g. antiviruses installed on the endpoints) ineffective
- **Ransomware**
    - Malware that proceeds to encrypt files stored inside the host machines, asking for a ransom from the victim to obtain the decryption key which is used for recovering the original files
- **APT**
    - APT (Advanced Persistent Threat) is a form of tailored attack that exploits specific vulnerabilities on the victimized hosts

Common Types of Malware

- **Spyware**
  - Spyware is malicious software that runs secretly on a computer and reports back to a remote user. Rather than simply disrupting a device’s operations, often used to steal financial or personal information
  - A specific type of spyware is a keylogger, which records keystrokes to reveal passwords and personal information
- **Zero day**
  - Malware that exploits vulnerabilities not yet disclosed to the community of researchers and analysts, whose characteristics and impacts in terms of security are not yet known, and therefore go undetected by antivirus software

Malware Detection Strategies

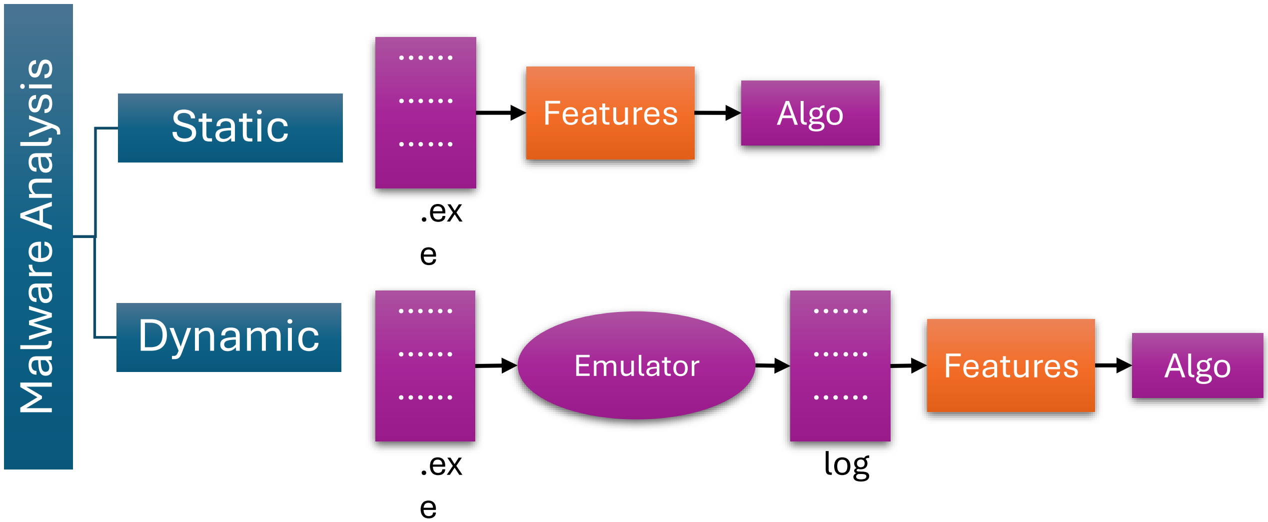
- Detection activities that can be easily automated
  - File hash calculation: To identify known threats already present in the knowledge base.
  - System monitoring: To identify anomalous behavior of both the hardware and the operating system
    - such as an unusual increase in CPU cycles
    - a particularly heavy disk writing activity
    - changes to the registry keys
    - the creation of new and unsolicited processes in the system …
  - Network monitoring: To identify anomalous connections established by host machines to remote destinations

Malware Analysis Techniques

- Two fundamental approaches to malware analysis
  - Static and dynamic
- Static (code) analysis: analyzing the code or structure of a program to determine its function without running it
- Dynamic (behavioral) analysis: running the malware and observing its behavior on the system
  - Before running the malware safely, a sandbox environment must be set up that will allow to study the running malware without damaging to the system or network.
- Usually fulfilled by security specialists

Steps for building a ML-based Malware detector

- After analyzing the malware, we can extract the features of the binary files (whether legitimate or suspect), and store them in a dataset of artifacts with which to train our algorithms



Steps for building a ML-based Malware detector

- Deploying an AI model in a production environment is challenging
  - It must be accurate and generalizable so it can detect malware it has never seen before and be practical and lightweight.
  - There are numerous types of malware where each category behaves differently and may not have any commonalities.
- Collect
  - Examples of malware and benignware
  - Used to train the model
- Extract
  - Features from each training example to represent the data
  - This step also includes research to design good features that will help the machine learning model make accurate inferences – domain experts
- Train and Test
  - Machine learning model to detect malware using the features
  - Our focus: malicious or non-malicious

Malware Sophistication

- Anti-virus and malware scanners typically use signatures.
  - A malware signature is a sequence of bytes that represent a pattern of behavior, code, or strings found in a malware file.
- The sophistication of modern malware means that signature-based and heuristic detection can be easily defeated
  - recycled malware that has been slightly changed or obfuscated bypasses signature detection engines
- Challenges
  - Evasive Malware
  - Novel Malware
  - AI-powered Malware

Gaber, M.G., Ahmed, M. and Janicke, H., 2024. Malware detection with artificial intelligence: A systematic literature review. ACM Computing Surveys, 56(6), pp. 1-33.

Malware Sophistication

- Evasive Malware
  - Seeks to hide its malicious intent.
    - Resist analysis by widely used tools, such as decompilers, debuggers, sandboxes, and etc.
  - Techniques used by adversaries
    - Obfuscation: often used to make the malware files more difficult to detect and analyze and primarily affects **static analysis**.
      - Examples
        - Code Packing compresses parts of a malicious file that are only decompressed when the program is executed.
        - Polymorphic obfuscation encrypts and mutates parts of code to change its signature. The decryption method may only decrypt parts of the file as required by execution.
      - Do not alter functionality or behavior but impact features extracted using static analysis.
      - Legitimate software may also implement evasive behaviors, to prevent analysis and reverse engineering.

Gaber, M.G., Ahmed, M. and Janicke, H., 2024. Malware detection with artificial intelligence: A systematic literature review. ACM Computing Surveys, 56(6), pp.1-33.

Malware Sophistication

- Evasive Malware
  - Techniques used by adversaries (continued)
    - In a **dynamic analysis** environment, e.g., sandboxes, evasive malware can use varied anti-analysis techniques to detect the environment in which it is running.
      - It then behaves differently and hide its malicious intent if it is under analysis.

Gaber, M.G., Ahmed, M. and Janicke, H., 2024. Malware detection with artificial intelligence: A systematic literature review. ACM Computing Surveys, 56(6), pp. 1-33.

Malware Sophistication

- Novel Malware
  - Any malware that has not been seen before and would not be detected by a malware detection engine that uses signature matching.
    - E.g., new malware that does not fit existing families, malware that employs new anti-analysis or obfuscation techniques, and zero-day malware that exploits zero-day vulnerabilities.
- AI-powered Malware
  - Exploit Neural Networks (NN) to power evasiveness and targeting.
    - NNs are not easily interpretable and lack transparency.
    - Can be immune to analysis where the malicious payload is encrypted and only decrypted when the target is detected.
    - E.g., use object detection models to recognize the Graphical User Interface (GUI).
      - Detect if the user was on a banking website and then launch an attack to transfer money to another account.

Gaber, M.G., Ahmed, M. and Janicke, H., 2024. Malware detection with artificial intelligence: A systematic literature review. ACM Computing Surveys, 56(6), pp.1-33.

Malware Repositories

- Creating a public open dataset that includes legitimate and malicious files is challenging.
  - There are legal and copyright restrictions associated with the dissemination of proprietary Windows software.
  - There are potential security liabilities where live malicious files are released to a public that may not take the correct precautions when handling them.
- Public malware datasets
  - EMBER
    - This is a dataset of features extracted from 1.1 million PE files.
    - The code, which could be used to extract the same features from other PE files, was also released.
      - Because the dataset consists of extracted features and not PE files, copyright is not an issue.
    - There are several limitations with this dataset.
      - Intact malicious and benign files are not included so that different features cannot be extracted.
      - The features were extracted using static analysis, which can be limited with obfuscated malware.
      - Even baseline classifiers can achieve excellent detection performance.
      - *Not recommended for future research.*

Gaber, M.G., Ahmed, M. and Janicke, H., 2024. Malware detection with artificial intelligence: A systematic literature review. ACM Computing Surveys, 56(6), pp.1-33.

Malware Repositories

- Public malware datasets (continued)
  - BODMAS
    - This is a dataset of recent, timestamped, and categorized malware.
    - It was collected between August 2019 and September 2020.
    - Contains features extracted from 77,142 benign samples and both the extracted features and intact PE files for 57,293 malware samples.
    - The taxonomy information covers 14 categories, such as Trojans and Ransomware.
  - SOREL-20M.
    - Contain 20 million samples.
      - 10 million extracted feature vectors and the intact but disarmed malware files.
      - Metadata and extracted feature vectors from 10 million benign samples.

Gaber, M.G., Ahmed, M. and Janicke, H., 2024. Malware detection with artificial intelligence: A systematic literature review. ACM Computing Surveys, 56(6), pp.1-33.

Malware Repositories

- Public malware datasets (continued)
  - VirusShare.
    - Provide security researchers access to live malware.
    - Contains more than 55 million live malware files with timestamps, hash, detection report, and other file information.
      - an example is shown in the figure.
    - The repository has been widely used in research
      - It allows for static and dynamic analysis to be performed on any number of intact malware files.

VirusShare.com - Because Sharing is Caring

Home • Hashes • Research • About

Please login to search and download.

System currently contains 55,030,761 malware samples.

Report for a sample recently added to the system:
401cfc7c59ca397f0b15d5c3a5fa903b6e882504c23a2418f0753b75154a841f

VirusShare info last updated 2022-12-16 00:00:00 UTC

MD5	5dab03a9ebc279457a513e20a2e6f578	
SHA1	e46f8473b592e7b0102a92ce9c47df39a2658d4f	
SHA256	401cfc7c59ca397f0b15d5c3a5fa903b6e882504c23a2418f0753b75154a841f	
SSDeep	3072:vQZosimPTDyXsLCaZg8eMxWBZBs5ZrTiLRAFIY7LIvuiTT:4RTCjBZc3Y7LIvug	
Authentihash	9f0afdb2ea9e2738ab7ace5d122d96e5961b1ce7a7e55794d85929297a408d17	
Size	187,904 bytes	
File Type	PE32 executable, for MS Windows	
Mime Type	application/x-dosexec	
Extension	exe	
TrID	OS/2 Executable (generic) (33.6%) Generic Win/DOS Executable (33.1%) DOS Executable Generic (33.1%)	
Detections (42/68)	APEX	Malicious
	Acronis	suspicious
	Ad-Aware	Trojan.Obfus.3.Gen
	AhnLab-V3	Trojan/Win32.Nabucur.C622804
	Antiy-AVL	Virus/Win32.PolyRansom.a
	Arcabit	Trojan.Obfus.3.Gen
	BitDefender	Trojan.Obfus.3.Gen
	BitDefenderTheta	AI:FileInfector.1F8DFD280F
	ClamAV	BC.Win.Virus.Ransom-9157.A
	Comodo	Packed.Win32.Graybird.B@5hgpd5
	Cybereason	malicious.9ebc27
	Cynet	Malicious (score: 100)
	Cyren	W32/S-accd10d9!Eldorado
	DrWeb	Win32.VirLock.1
	Elastic	malicious (high confidence)
	Emsisoft	Trojan.Obfus.3.Gen (B)
	FireEye	Trojan.Obfus.3.Gen
	Fortinet	W32/VirRansom.D9F1!tr
	GData	Trojan.Obfus.3.Gen
	Google	Detected

Gaber, M.G., Ahmed, M. and Janicke, H., 2024. Malware detection with artificial intelligence: A systematic literature review. ACM Computing Surveys, 56(6), pp.1-33.

Malware Repositories

- Public malware datasets (continued)
  - Malware Bazaar.
    - Also provide security researchers access to live malware.
    - Contains more than 700,000 live malware files with timestamps, hash, category and family information.
    - The repository has been used in recent research
      - Live samples are available and searchable by category and include family and variant information.
      - However, there is a file download limit of 2,000 samples per day.

Gaber, M.G., Ahmed, M. and Janicke, H., 2024. Malware detection with artificial intelligence: A systematic literature review. ACM Computing Surveys, 56(6), pp. 1-33.

Malware Repositories

- Public malware datasets (continued)
  - Microsoft malware dataset
    - Microsoft malware classification challenge was first held in 2015.
    - This dataset is almost *half a terabyte* uncompressed.
    - The dataset contains malware from nine families of malware.
  - Malimg
    - Contain 9435 real-life malware samples of 25 different families.
    - This dataset is widely used for visualization-based malware classification tasks.

Gaber, M.G., Ahmed, M. and Janicke, H., 2024. Malware detection with artificial intelligence: A systematic literature review. ACM Computing Surveys, 56(6), pp.1-33.

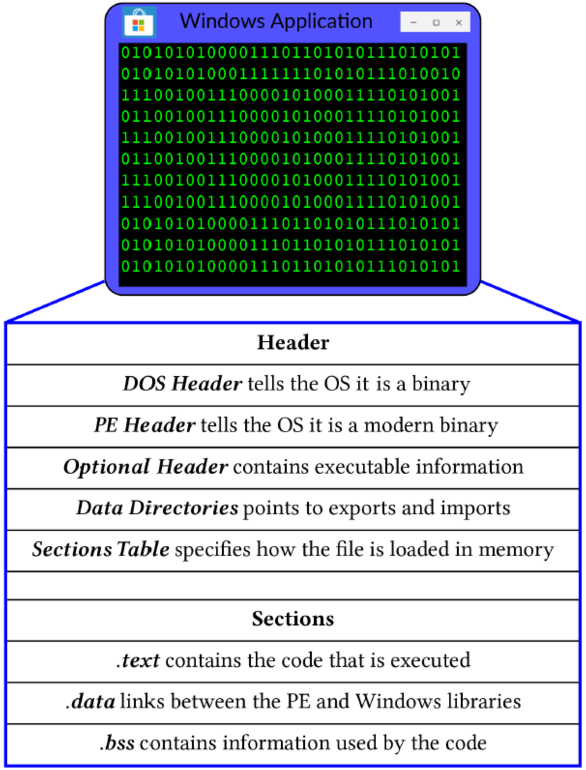
Image Recognition for Detecting Malware

- Motivation
  - Static analysis can explore all possible execution paths.
    - Dynamic analysis may be limited.
    - Covering all code paths helps provide a complete characterization of functionality of PE files.
  - Using DNN to learn static patterns can be a good idea.

Gaber, M.G., Ahmed, M. and Janicke, H., 2024. Malware detection with artificial intelligence: A systematic literature review. ACM Computing Surveys, 56(6), pp. 1-33.

Image Recognition for Detecting Malware

- Files to images
  - Malware that targets Windows is normally created using the Portable Executable (PE) file specification.
  - A PE file incorporates a Header and Sections, which includes the information necessary for Windows to run the file. In this way, a PE file specifies imports and how it is mapped to memory as well as the code that is executed.
  - PE files can be easily converted to RGB and greyscale images, where each byte in the file and each pixel in an image has a value between 0–255.
  - generated image can then be used to train varied AI models



Gaber, M.G., Ahmed, M. and Janicke, H., 2024. Malware detection with artificial intelligence: A systematic literature review. ACM Computing Surveys, 56(6), pp.1-33.

Imbalanced data

- When classifying a rare event (e.g. account hijacking)—the “good” samples can be even 99.9% of the labeled data
  - In this case the “bad” samples might not have enough weight to ‘influence’ the classifier
  - model might perform poorly when trained on this highly unbalanced data
- Some options:
  - ***Oversample*** the minority class: repeat observations in your training data to make a more balanced set
    - Synthesizing minority classes: generative models.
  - ***Undersample*** the majority class: select a random subset of the majority class to produce a more balanced set
  - ***Change the cost function*** to weight performance on the minority class more heavily, so that each minority sample has more influence on the model

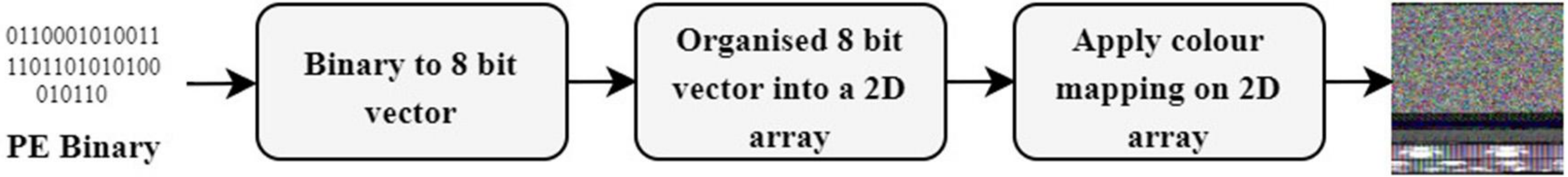
A novel deep learning-based approach for malware detection

- A two-step malware detection framework
  - The first step
    - An image-based PE dataset is generated by transforming all the malign and benign PEs into images.
    - This step also handles the imbalanced data problem.
  - The second step
    - Use deep learning and machine learning models for malware detection.
- Datasets
  - Malicious PEs are collected from three datasets
    - Microsoft malware dataset
    - Malimg (for validation)
    - VirusShare (for evaluating generalization)
  - Multiple online resources, including softpedia.com, download.com, and Internet download sites have been used to collect benign PEs.

Shaukat, K., Luo, S. and Varadharajan, V., 2023. A novel deep learning-based approach for malware detection. *Engineering Applications of Artificial Intelligence*, *122*, p.106030.

A novel deep learning-based approach for malware detection

- Transformation of malware and benign PEs into images
  - PEs are converted into colored images (red, green, blue (RGB))
    - A PE file is translated into a raw byte binary stream.
    - Image is constructed through this raw byte binary steam.



    - A raw byte binary stream of a PE file is divided into substrings of 8 bits in length.
    - Each substring of 8 bits is preserved as a pixel in an image.
      - unsigned integer ranging between 0–255.
      - Images are resized to 224*224.
    - A color map is applied to the 2D matrix to visualize the PE binary.
    - The whole image generation and later detection process **do not** require any feature engineering or domain expert.

Shaukat, K., Luo, S. and Varadharajan, V., 2023. A novel deep learning-based approach for malware detection. *Engineering Applications of Artificial Intelligence*, *122*, p.106030.

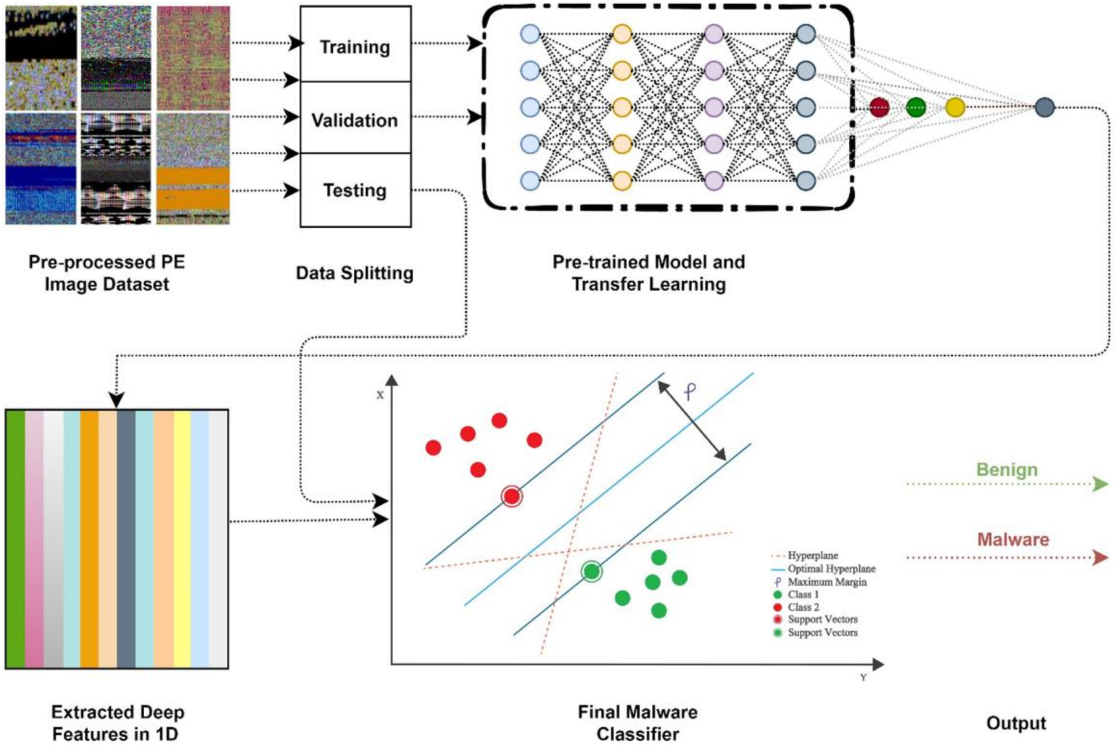
A novel deep learning-based approach for malware detection

- Data imbalance and solution
  - The malware classes in benchmark and well-known public datasets are not equally distributed.
    - The Malimg dataset comprises 9339 malware samples from 25 families and 617 benign samples.
    - Among different families of malware, the Allaple. A family has 2949 samples, whereas the Skintrim.N has only 80 samples.
    - Microsoft malware dataset is composed of 10 868 malware samples. Out of 10,868 samples, the Kelihos_ver3 malware family has 2942 samples, and Simda has only 42 samples.
  - Use state-of-the-art data augmentation techniques.
    - Include height shift, horizontal flip, brightness range, and etc.
    - Balance the training data of each class (malware and benign) into an equal number of instances.

Shaukat, K., Luo, S. and Varadharajan, V., 2023. A novel deep learning-based approach for malware detection. *Engineering Applications of Artificial Intelligence*, *122*, p.106030.

A novel deep learning-based approach for malware detection

- Hybrid malware detection
  - The data is split into training, validation, and testing.
  - A deep learning model is used to extract the deep features.
  - The extracted features are fed into a machine learning model, i.e., SVM, for final malware detection.



Shaukat, K., Luo, S. and Varadharajan, V., 2023. A novel deep learning-based approach for malware detection. *Engineering Applications of Artificial Intelligence*, *122*, p.106030.

A novel deep learning-based approach for malware detection

- Deep learning models and transfer learning
  - Fine-tune models that have been pretrained on ImageNet
    - 10+ million images with 1000 class labels.
    - 15 pretrained models are studied in experiments.

Sr#	Model	Year	Depth	Layers	Size (MB)	Parameters (millions)	Non-trainable parameters	Trainable parameters
1	VGG16	2014	16	41	515	138	137,897,600	102,400
2	ResNet50	2015	50	177	96	25.6	25,548,800	51,200
3	InceptionV3	2015	48	316	89/ 87	23.9	23,848,800	51,200
4	VGG19	2014	19	47	535	20.1	20,024,384	25,089
5	MobileNet	2018	28	55	16	3.2	3,228,864	50,177
6	Xception	2013	71	171	88	22.9	20,861,480	100,353
7	DenseNet169	2016	169	338	57	12.7	12,642,880	81,537
8	DenseNet201	2017	201	709	80	18.4	18,321,984	94,081
9	InceptionResNetV2	2017	164	825	215	54.4	54,336,736	38,401
10	MobileNetV2	2018	53	155	13/ 14	3.5	3,468,000	32,000
11	ResNet152V2	2015	307	570	232	58.4	58,331,648	100,353
12	AlexNet	2012	8	25	227	61	60,897,600	102,400
13	SqueezeNet	2017	18	68	4.5	23.5	23,084,252	48,842
14	NasNetMobile	2018	389	914	23	4.4	4,269,716	51,745
15	RegNetY320	2020	320		553	145	144,754,400	103,840

Shaukat, K., Luo, S. and Varadharajan, V., 2023. A novel deep learning-based approach for malware detection. *Engineering Applications of Artificial Intelligence*, *122*, p.106030.

A novel deep learning-based approach for malware detection

- Experimental setup
  - The collected corpus is further divided into three datasets: datasetA, datasetB, and datasetC.
    - DatasetA consists of 2000 malware and 10,000 benign PEs.
    - DatasetB consists of 20,000 malware and 10,000 benign PEs.
    - DatasetC consists of 20,000 malware and 10,000 benign PEs.
      - All the samples are the same as in datasetB.
      - Augmentation is applied to benign PEs to balance the training set.

Shaukat, K., Luo, S. and Varadharajan, V., 2023. A novel deep learning-based approach for malware detection. *Engineering Applications of Artificial Intelligence*, *122*, p.106030.

A novel deep learning-based approach for malware detection

- Results on datasetA (2000 malware and 10,000 benign PEs)
  - The samples are divided with a ratio of 60:20:20 for training, validation, and testing.
  - RegNetY320 is used in CNN-SVM to extract features.

A comparison of detection effectiveness of fine-tuned deep learning-based malware detectors with CNN-based SVM malware detector on datasetA.

Sr. No.	Model	Epochs	Batch size	Valid loss	Valid Acc.	Test loss	Test Acc.
1	VGG16	150	32	0.578	86.99%	0.4575	90.71%
2	ResNet50	150	32	30.88	69.39%	31.42	69.38%
3	InceptionV3	150	32	5.55	71.30%	3.599	72.55%
4	VGG19	150	32	0.5874	85.46%	0.4613	89.84%
5	MobileNet	150	32	10.6329	55.99%	5.44	70.25%
6	DenseNet169	150	32	4.53	77.10%	4.702	77.95%
7	Xception	150	32	3.24	65.43%	3.88	62.04%
8	DenseNet201	150	32	9.74	73.02%	10.99	75.05%
9	Inception ResNet V2	150	32	3.389	61.54%	4.63	52.80%
10	MobileNetV2	150	32	12.3141	70.34%	10.84	70.81%
11	ResNet152V2	150	32	11.49	74.87%	9.099	77.44%
12	NasNetMobile	150	32	4.307	56.17%	5.53	50.70%
13	AlexNet	150	32	9.03	66.72%	8.49	68.54%
14	SqueezeNet	150	32	10.88	71.23%	8.38	70.27%
15	RegNetY320	150	32	2.98	78.18%	2.41	78.92%
16	CNN-SVM	150	32	0.397	88.65%	0.4366	**91.17%**

Shaukat, K., Luo, S. and Varadharajan, V., 2023. A novel deep learning-based approach for malware detection. *Engineering Applications of Artificial Intelligence*, *122*, p.106030.

A novel deep learning-based approach for malware detection

- Results on datasetB (20,000 malware and 10,000 benign PEs)

A comparison of detection effectiveness of fine-tuned deep learning-based malware detectors with CNN-based SVM malware detector on datasetB.

Sr. No.	Model	Epochs	Batch size	Valid loss	Valid Acc.	Test loss	Test Acc.
1	VGG16	150	32	0.6954	88.09%	0.487	90.75%
2	ResNet50	150	32	67.29	50.00%	54.62	74.58%
3	InceptionV3	150	32	4.01	75.38%	2.72	80.23%
4	VGG19	150	32	0.5344	90.34%	0.3785	93.04%
5	MobileNet	150	32	8.503	77.42%	5.15	76.04%
6	DenseNet169	150	32	11.43	73.42%	5.94	83.59%
7	Xception	150	32	3.53	76.11%	4.29	73.93%
8	DenseNet201	150	32	15.84	68.50%	14.24	76.23%
9	Inception ResNet V2	150	32	4.27	74.41%	5.29	75.35%
10	MobileNetV2	150	32	8.61	72.05%	9.12	73.97%
11	ResNet152V2	150	32	16.02	75.86%	11.71	80.21%
12	NasNetMobile	150	32	3.77	76.17%	3.52	78.43%
13	AlexNet	150	32	10.64	72.65%	8.02	73.11%
14	SqueezeNet	150	32	9.63	71.87%	8.26	74.95%
15	RegNetY320	150	32	2.83	83.12%	1.93	82.54%
16	CNN-SVM	150	32	0.4544	92.00%	0.3373	**93.39%**

Shaukat, K., Luo, S. and Varadharajan, V., 2023. A novel deep learning-based approach for malware detection. *Engineering Applications of Artificial Intelligence*, *122*, p.106030.

A novel deep learning-based approach for malware detection

- Results on datasetC (apply augmentation to datasetB)

A comparison of detection effectiveness of fine-tuned deep learning-based malware detectors with CNN-based SVM malware detector on datasetC.

Sr. No.	Model	Epochs	Batch size	Valid loss	Valid Acc.	Test loss	Test Acc.
1	VGG16	150	32	0.4312	91.72%	0.2509	93.68%
2	ResNet50	150	32	28.88	70.97%	30.54	74.84%
3	InceptionV3	150	32	2.87	77.33%	1.9	80.46%
4	VGG19	150	32	0.4998	91.39%	0.2789	93.64%
5	MobileNet	150	32	7.58	78.24%	4.39	76.13%
6	DenseNet169	150	32	3.71	91.12%	2.66	84.15%
7	Xception	150	32	2.8	76.57%	3.41	74.21%
8	DenseNet201	150	32	9.09	77.13%	8.84	77.44%
9	Inception ResNet V2	150	32	3.38	76.69%	4.21	75.55%
10	MobileNetV2	150	32	8.51	73.04%	8.8	74.01%
11	ResNet152V2	150	32	9.5	76.62%	8.95	80.31%
12	NasNetMobile	150	32	2.33	77.86%	1.93	79.62%
13	AlexNet	150	32	8.81	76.01%	7.14	77.48%
14	SqueezeNet	150	32	9.51	73.51%	7.94	81.28%
15	RegNetY320	150	32	1.96	85.41%	1.19	88.52%
16	CNN-SVM	150	32	0.2783	93.70%	0.2303	**95.15%**

Shaukat, K., Luo, S. and Varadharajan, V., 2023. A novel deep learning-based approach for malware detection. *Engineering Applications of Artificial Intelligence*, *122*, p.106030.

Image Recognition for Detecting Malware

- Limitations
  - While converting PE files to images that are then used to train AI models is a novel approach, the technique has limitations.
    - It is not effective for detecting novel malware.
    - Can be easily defeated by inserting malicious code into benign carrier applications.
    - Further, detection accuracy significantly declines when obfuscation is used by malware samples.

- There is a need for more advanced malware visualization techniques in the future.

Gaber, M.G., Ahmed, M. and Janicke, H., 2024. Malware detection with artificial intelligence: A systematic literature review. ACM Computing Surveys, 56(6), pp. 1-33.

References

Gaber, M.G., Ahmed, M. and Janicke, H., 2024. Malware detection with artificial intelligence: A systematic literature review. ACM Computing Surveys, 56(6), pp.1-33.

Shaukat, K., Luo, S. and Varadharajan, V., 2023. A novel deep learning-based approach for malware detection. Engineering Applications of Artificial Intelligence, 122, p.106030.

Gaber, M.G., Ahmed, M. and Janicke, H., 2024. Malware detection with artificial intelligence: A systematic literature review. ACM Computing Surveys, 56(6), pp.1-33.