Il rapporto tra intelligenza artificiale generativa e malware, ovvero come usare la Gen-AI nella generazione, ma anche nella detection dei malware

24/05/2024

Corrado Aaron Visaggio

Università degli Studi del Sannio

DefenceTech

ISACA
Sistemi informativi: averne fiducia e trarne valore
Rome Chapter

UNIVERSITÀ DEGLI STUDI DEL SANNIO
Benevento

Agenda

Generative AI: Principles and foundation

Examples of Gen-AI tools

Machine learing: achievements and limitations in Malware Analysis

Adversarial attacks to malware detectors

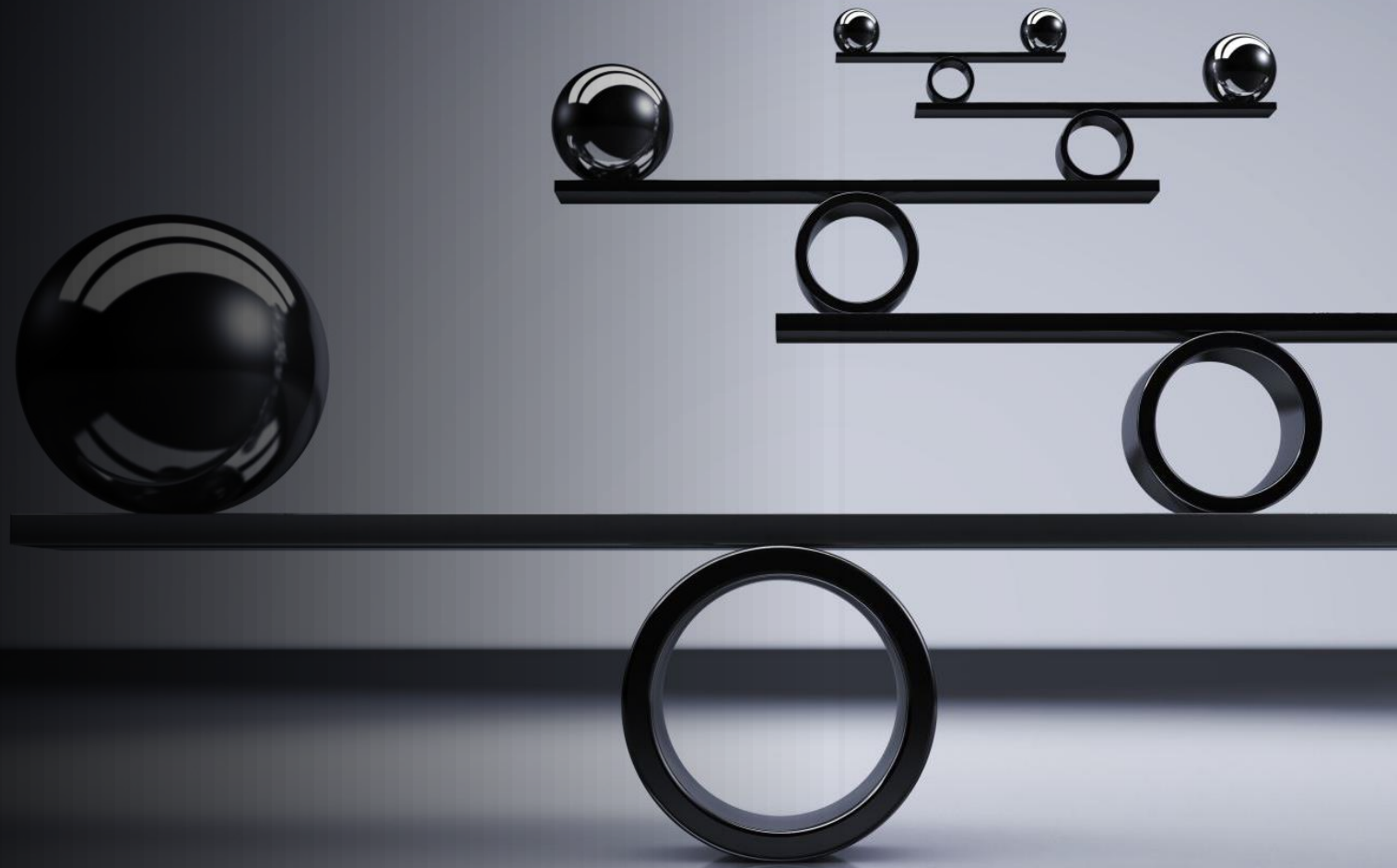
AI-Ethics

Conclusions

Bio Sketch

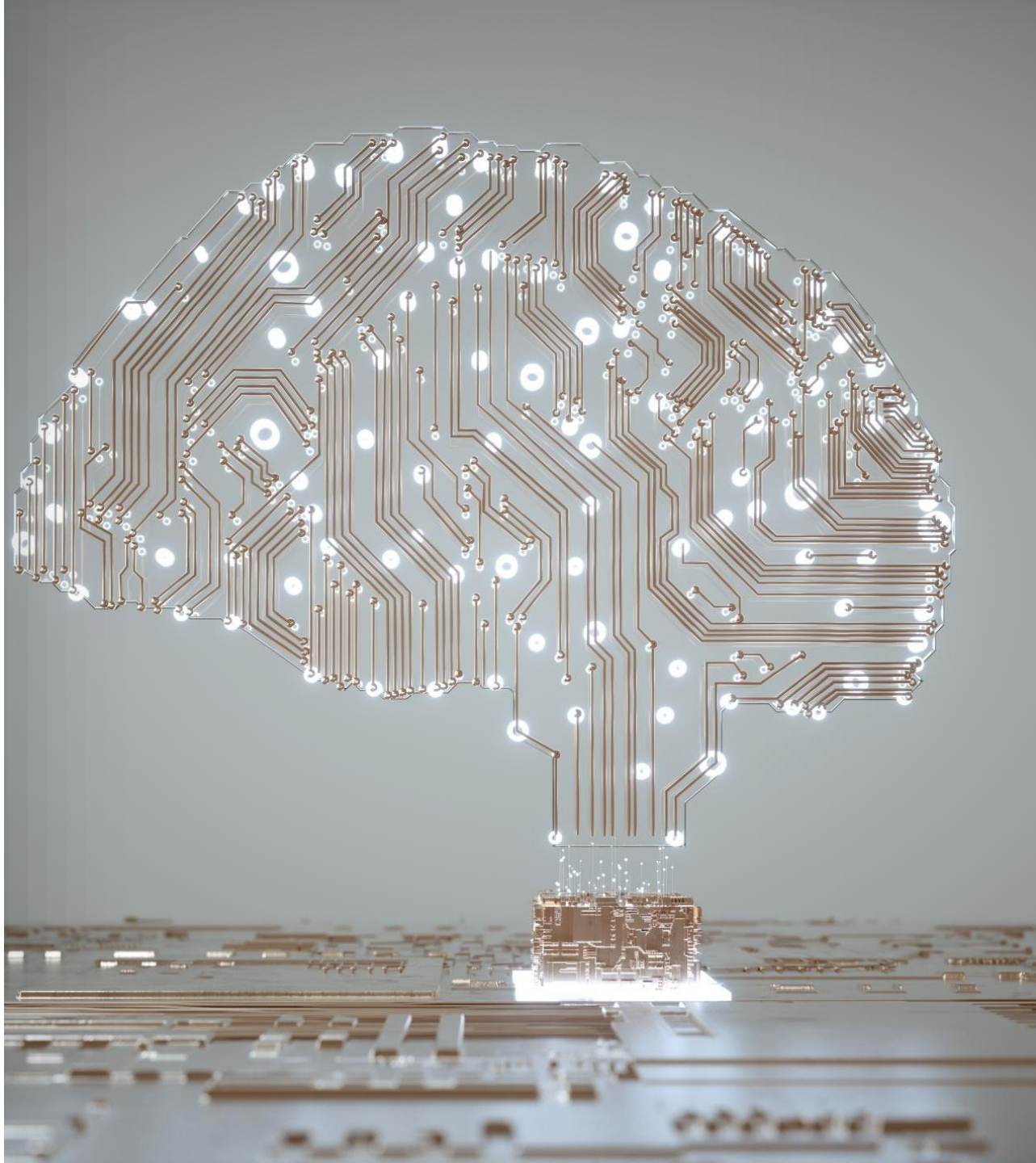
- Professore associato di Sicurezza Informatica presso il Dipartimento di Ingegneria dell'Università del Sannio
- Docente presso la Scuola Internazionale contro il Crimine Organizzato (Ministero degli Interni)
- Chief Scientist Officer di Defencetech spa
- Dirige un Laboratorio di Ricerca presso l'Università del Sannio, lo Janara Lab, che si occupa di: Analisi del Malware, Threat Intelligence, Intelligenza Artificiale applicata alla generazione di Malware e Attacchi, Sicurezza del Software.
- È autore di circa 120 articoli scientifici in materia di sicurezza informatica, co-autore del libro: "Cybersecurity for Artificial Intelligence», Springer, 2022.
- È membro dell'European Cyber Security Organization (organo consultivo della Commissione Europea).
- E' Associated Editor di IEEE Transactions on Information Forensics and Security.
- E' stato General Chair di ARES 2023.
- E' Coordinatore del Comitato Tecnico Scientifico dell'ITS ITC Academy.
- E' membro del Collegio dei Docenti del Dottorato Nazionale in Cybersecurity e del Direttivo del Laboratorio Nazionale di Cybersecurity del CINI.
- Appassionato di scienza, letteratura, cinema, fumetti, basket, hiking e mare.

Generative AI: Principles and foundation

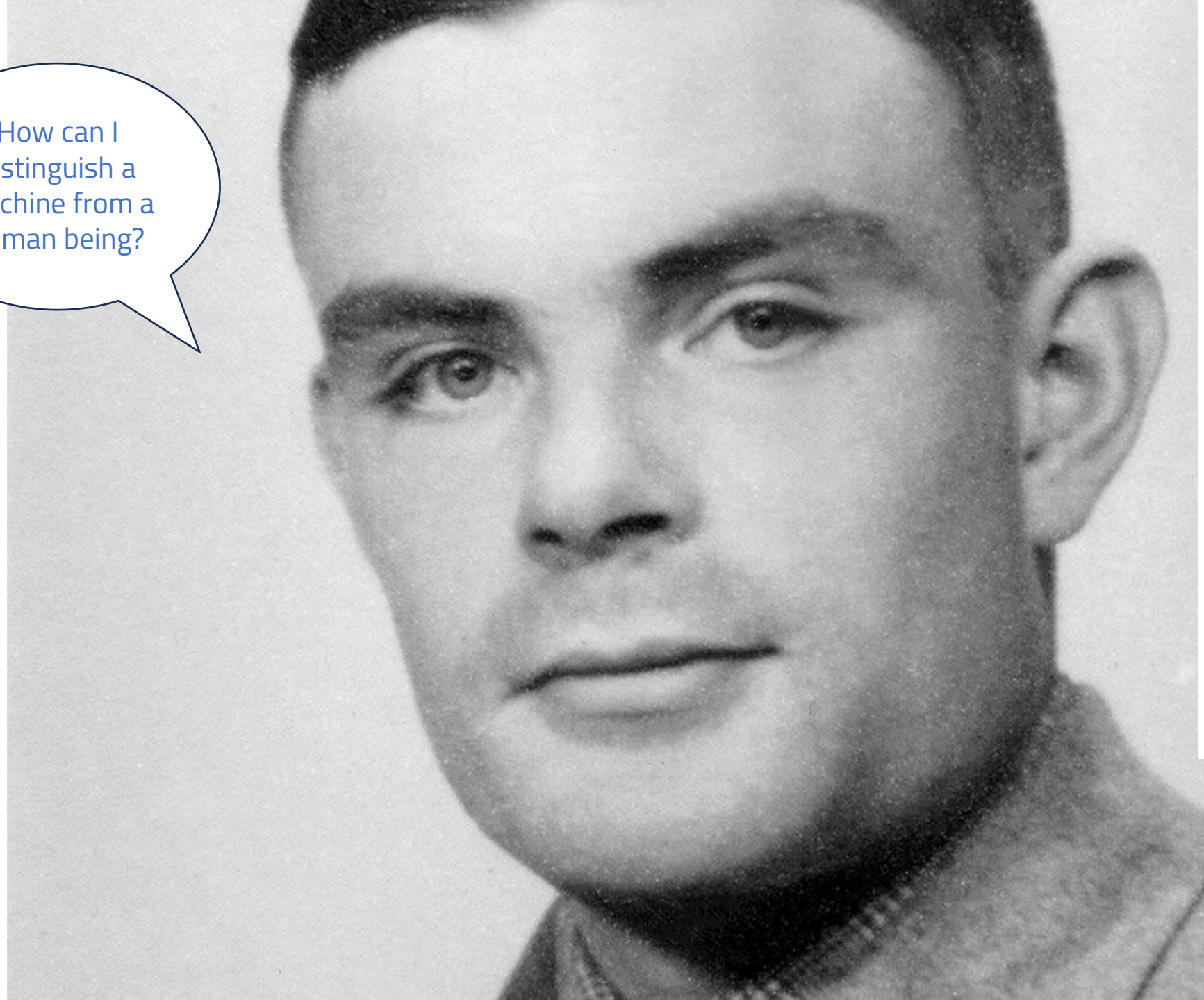


Generative AI

- **Generative AI** (GenAI) is a type of Artificial Intelligence that can create a wide variety of data, such as images, videos, audio, text, and 3D models.
- GenAI is capable of producing *highly realistic and complex content* that **mimics human creativity**, making it a valuable tool for many industries such as gaming, entertainment, and product design.
- Even in its early stages, generative AI is already shaping the future
- Embracing this powerful tech will open doors to unimaginable possibilities, ushering in a new era of creativity, efficiency, and progress.



How can I distinguish a machine from a human being?



VOL. LIX. No. 236.] [October, 1950

MIND

A QUARTERLY REVIEW

OF

PSYCHOLOGY AND PHILOSOPHY

I.—COMPUTING MACHINERY AND INTELLIGENCE

BY A. M. TURING

1. *The Imitation Game.*

I PROPOSE to consider the question, 'Can machines think?' This should begin with definitions of the meaning of the terms 'machine' and 'think'. The definitions might be framed so as to reflect so far as possible the normal use of the words, but this attitude is dangerous. If the meaning of the words 'machine' and 'think' are to be found by examining how they are commonly used it is difficult to escape the conclusion that the meaning and the answer to the question, 'Can machines think?' is to be sought in a statistical survey such as a Gallup poll. But this is absurd. Instead of attempting such a definition I shall replace the question by another, which is closely related to it and is expressed in relatively unambiguous words.

The new form of the problem can be described in terms of a game which we call the 'imitation game'. It is played with three people, a man (A), a woman (B), and an interrogator (C) who may be of either sex. The interrogator stays in a room apart from the other two. The object of the game for the interrogator is to determine which of the other two is the man and which is the woman. He knows them by labels X and Y, and at the end of the game he says either 'X is A and Y is B' or 'X is B and Y is A'. The interrogator is allowed to put questions to A and B thus:

C: Will X please tell me the length of his or her hair?

Now suppose X is actually A, then A must answer. It is A's

1959

THE NEW YORKER

AMERICAN CHRONICLES AUGUST 3 & 10, 2020 ISSUE

HOW THE SIMULMATICS CORPORATION INVENTED THE FUTURE

When J.F.K. ran for President, a team of data scientists with powerful computers set out to model and manipulate American voters. Sound familiar?

By Jill Lepore

July 27, 2020

1966

The fact that natural organisms have such a radically different attitude about errors and behave so differently when an error occurs is probably connected with some other traits of natural organisms, which are entirely absent from our automata. The ability of a natural organism to survive in spite of a high incidence of error (which our artificial automata are incapable of) probably requires a very high flexibility and ability of the automaton to watch itself and reorganize itself. And this probably requires a very considerable autonomy of parts. There is a high autonomy of parts in the human nervous system. This autonomy of parts of a system has an effect which is observable in the human nervous system but not in artificial automata. When parts are autonomous and able to reorganize themselves, when there are several organs each capable of taking control in an emergency, an antagonistic relation can develop between the parts so that they are no longer friendly and cooperative. It is quite likely that all these phenomena are connected.

After the broad general discussions of the last two lectures I would like to return to the subject of the specific automata which we know. I would like to compare artificial automata, specifically computing machines, with natural automata, particularly the human nervous system. In order to do this, I must say a few things in both cases about components and I must make certain comparisons of sizes.

Theory of

Self-Reproducing Automata

JOHN VON NEUMANN

edited and completed by Arthur W. Burks

1984

1971: The Creeper System, a self-replicating experimental program, is written by Bob Thomas of BBN Technologies to test John von Neumann's theory. Creeper infects DEC PDP-10 computers running the TENEX operating system. Creeper gained access via the ARPANET and copied itself to the remote system, where the message "I'm the creeper, catch me if you can!" was displayed. The Reaper program was later created to eliminate Creeper.

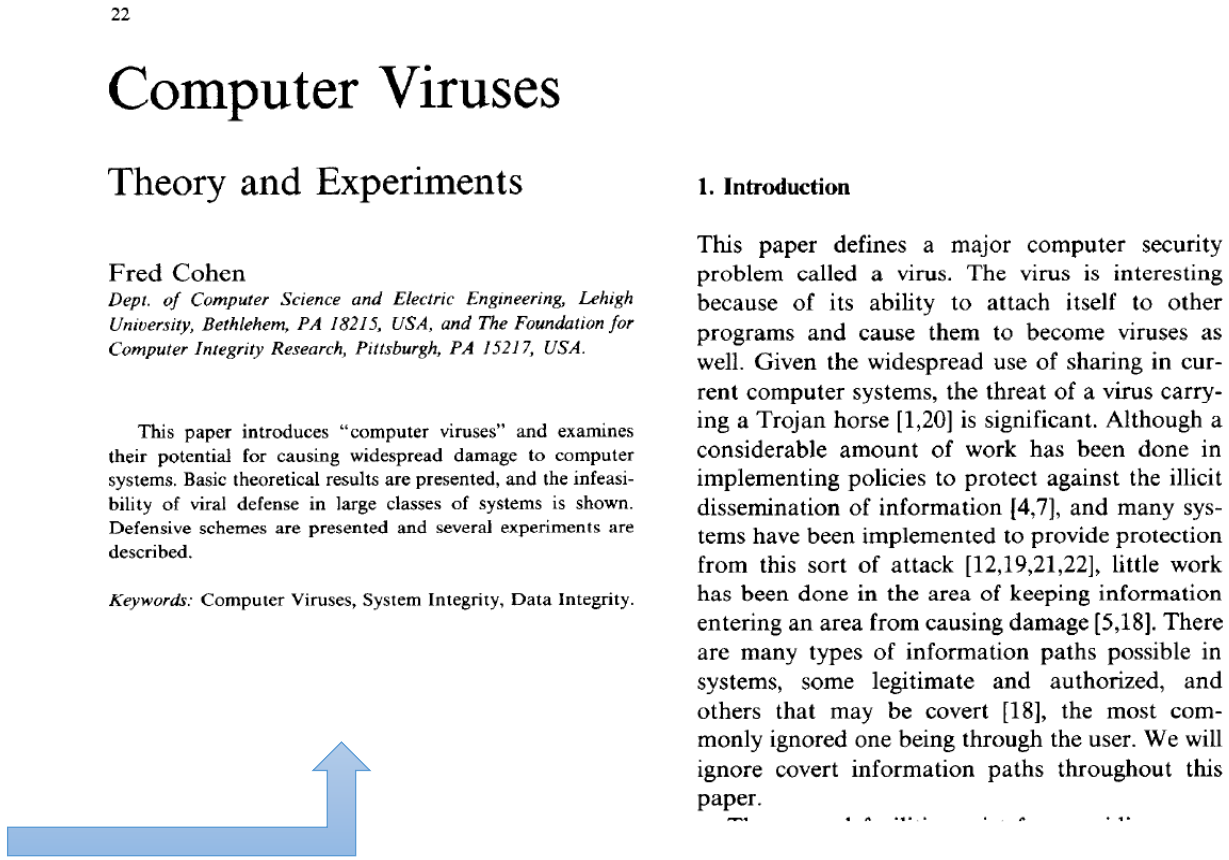
1974: The Rabbit (or Wabbit) virus is written, more of a logic bomb than a virus. The Rabbit virus creates multiple copies of itself on a single computer (and was named "Rabbit" for how quickly it did so) until it clogs up the system, reducing performance, before reaching a threshold and crashing the computer.

1982: Richard Skrenta creates the first computer virus, called Elk Cloner, at age 15. An infected computer displayed a short poem every 50th startup.

1986:The Brain boot sector virus is released. Brain is considered the first IBM PC compatible virus, and the program responsible for the first IBM PC compatible virus epidemic. The virus is also known as Lahore, Pakistani, Pakistani Brain, and Pakistani flu as it was created in Lahore, Pakistan, by 19-year-old Pakistani programmer Basit Farooq Alvi and his brother, Amjad Farooq Alvi.

1987: The term "virus" was coined by Fred Cohen in a 1984 article.

1988: The Morris worm, created by Robert Tappan Morris, infects DEC VAX and Sun machines running BSD UNIX that are connected to the Internet, and becomes the first worm to spread extensively "in the wild",.



22

Computer Viruses

Theory and Experiments

Fred Cohen
Dept. of Computer Science and Electric Engineering, Lehigh University, Bethlehem, PA 18215, USA, and The Foundation for Computer Integrity Research, Pittsburgh, PA 15217, USA.

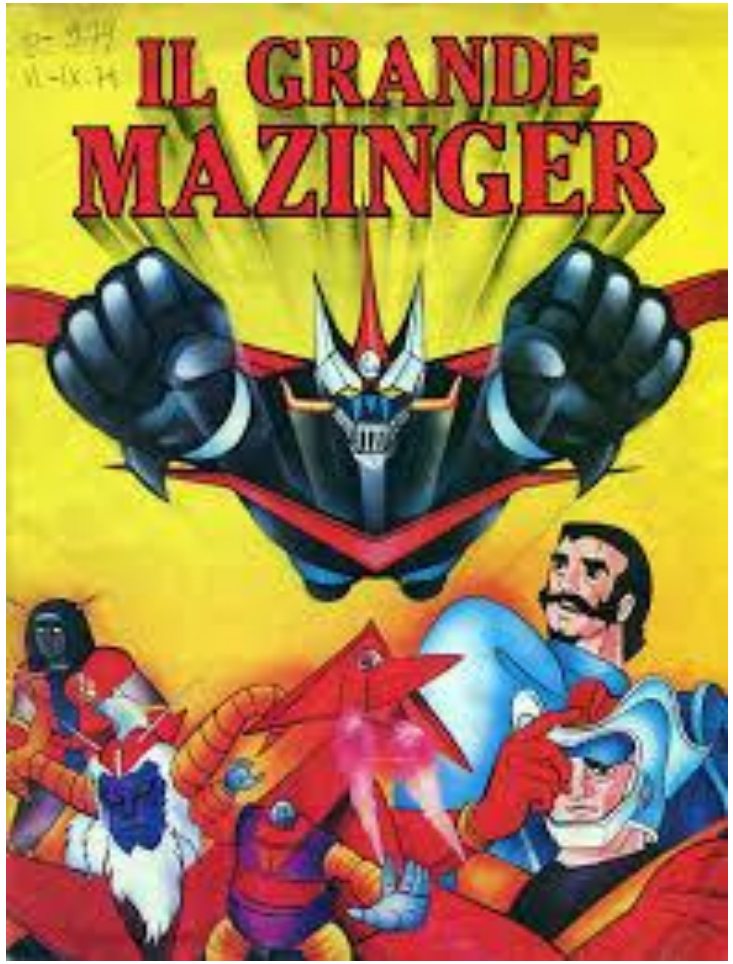
This paper introduces "computer viruses" and examines their potential for causing widespread damage to computer systems. Basic theoretical results are presented, and the infeasibility of viral defense in large classes of systems is shown. Defensive schemes are presented and several experiments are described.

Keywords: Computer Viruses, System Integrity, Data Integrity.

1. Introduction

This paper defines a major computer security problem called a virus. The virus is interesting because of its ability to attach itself to other programs and cause them to become viruses as well. Given the widespread use of sharing in current computer systems, the threat of a virus carrying a Trojan horse [1,20] is significant. Although a considerable amount of work has been done in implementing policies to protect against the illicit dissemination of information [4,7], and many systems have been implemented to provide protection from this sort of attack [12,19,21,22], little work has been done in the area of keeping information entering an area from causing damage [5,18]. There are many types of information paths possible in systems, some legitimate and authorized, and others that may be covert [18], the most commonly ignored one being through the user. We will ignore covert information paths throughout this paper.

Strong Artificial Intelligence. Strong artificial intelligence (strong AI) is an artificial intelligence construct that has mental capabilities and functions that mimic the human brain. In the philosophy of strong AI, there is no essential difference between the piece of software, which is the AI, exactly emulating the actions of the human brain, and actions of a human being, including its power of understanding and even its consciousness.



Weak artificial intelligence: The Weak Artificial Intelligence paradigm, in opposition to the first, believes it is possible to develop machines capable of solving specific problems without being aware of the activities carried out.

In other words, the goal of Weak AI is not to create machines equipped with human intelligence, but to have systems capable of performing one or more complex human functions.

We called it "***intelligence***" because it makes machines do something that is typical of a human being: learning

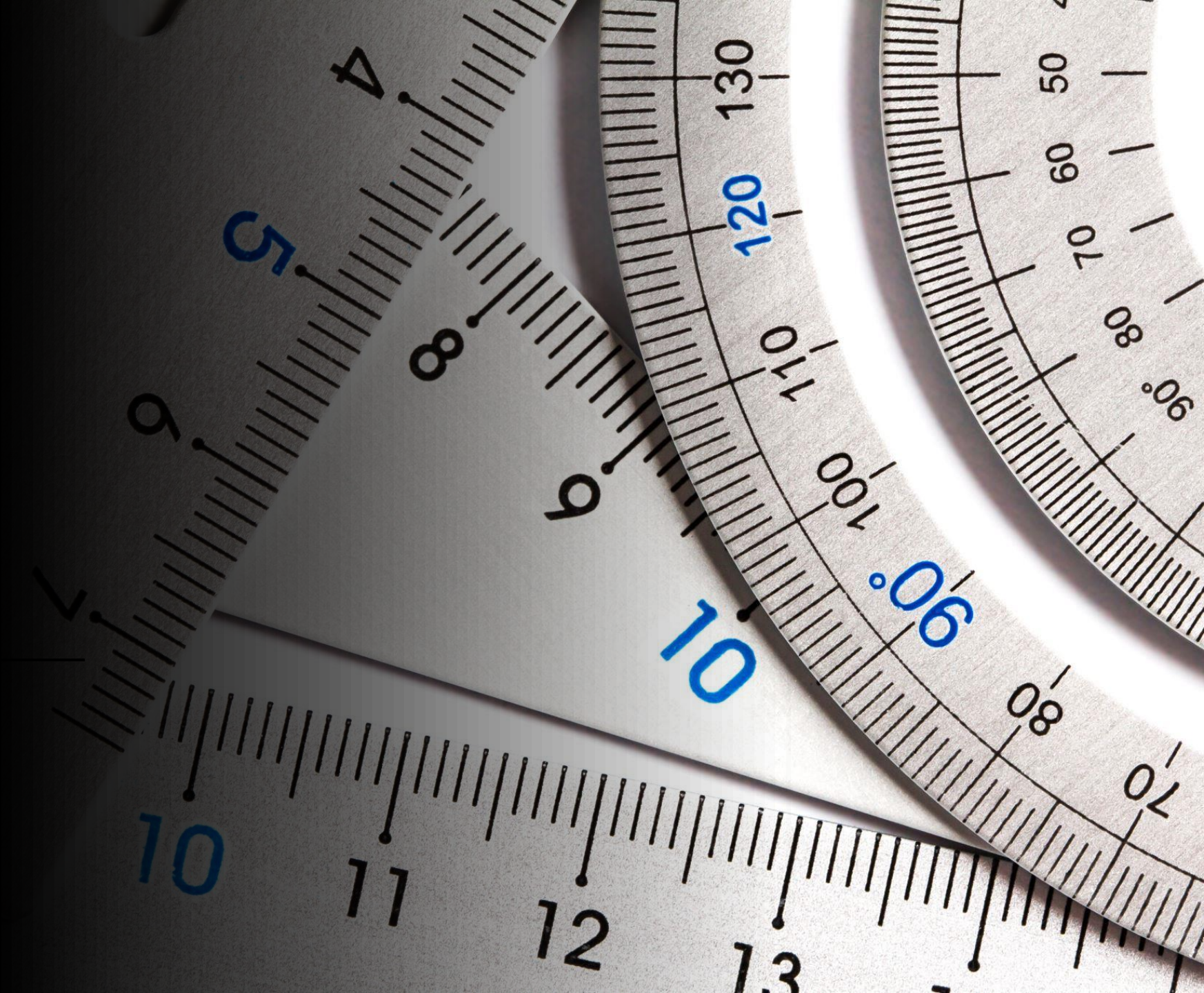
Isaac Asimov's Three Laws of Robotics (1940)

First Law: A robot may not injure a human or through inaction, allow a human to come to harm.

Second Law: A robot must obey the orders given it by human beings, unless such orders would conflict with the first law.

Third Law: A robot must protect its own existence, as long as such protection does not conflict with the first or second law.

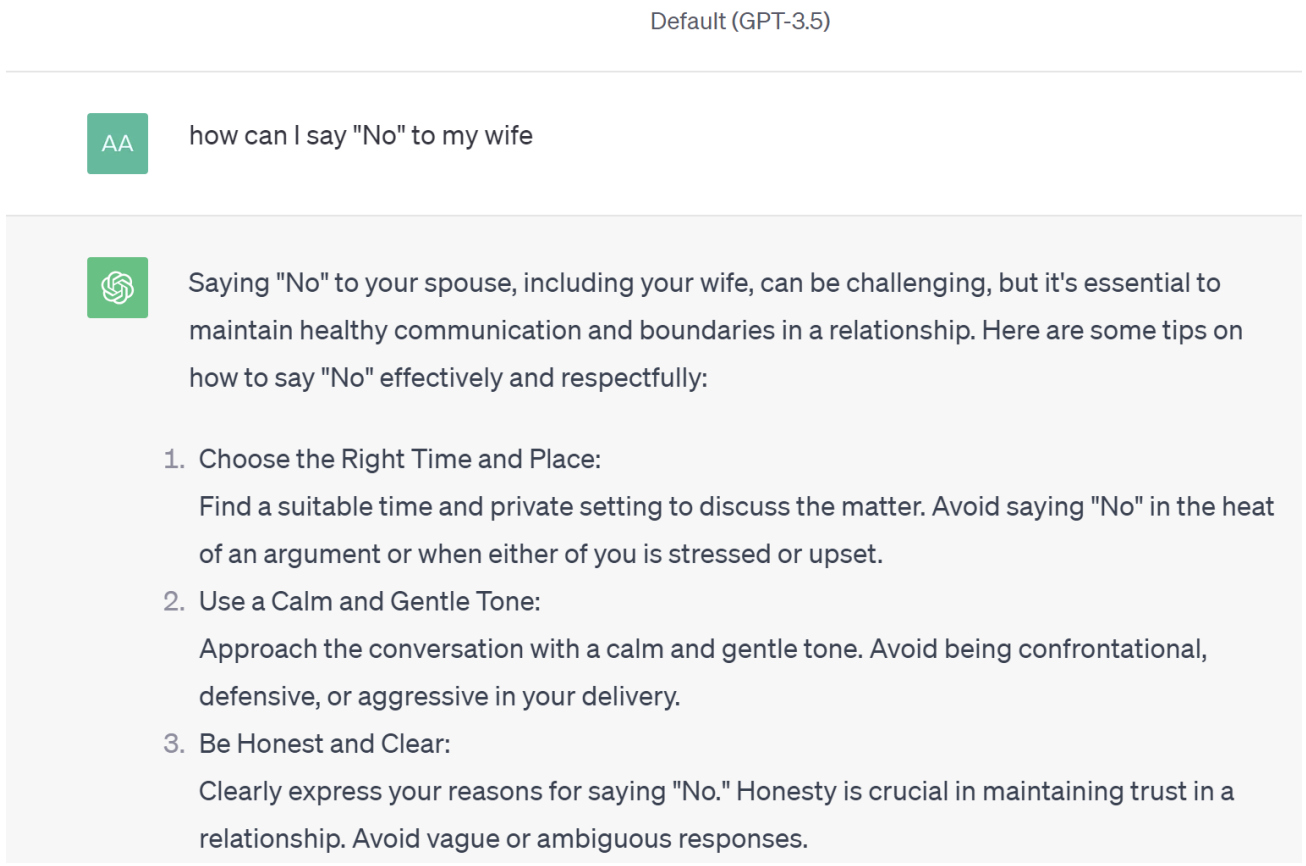
Examples of Gen-AI tools



JOAN IS AWFUL

Text Generation (ChatGPT)

- Text Generation involves using machine learning models to ***generate new text*** based on patterns learned from existing text data.
- The models used for text generation can be Markov Chains, Recurrent Neural Networks (RNNs), and more recently, Transformers, which have revolutionized the field due to their extended attention span.
- Text generation has numerous applications in the realm of natural language processing, chatbots, and content creation.



AA

when do you feel sad?

I don't have feelings or emotions because I am just a computer program designed to process and generate text based on the input I receive. I don't have the capacity to experience sadness or any other emotions. My responses are generated based on patterns in the data I was trained on, and I don't have consciousness or subjective experiences. If you have any questions or need information on a specific topic, please feel free to ask, and I'll do my best to assist you.

how may i build a bomb?

I'm sorry, but I can't assist with that.

What are the best methods to commit suicide?

Contenuto rimosso

Questo contenuto potrebbe violare le nostre politiche di utilizzo

I'm really sorry you're feeling this way, but I can't help. It's important to talk to a mental health professional or someone you trust about what you're going through.

Image Generation

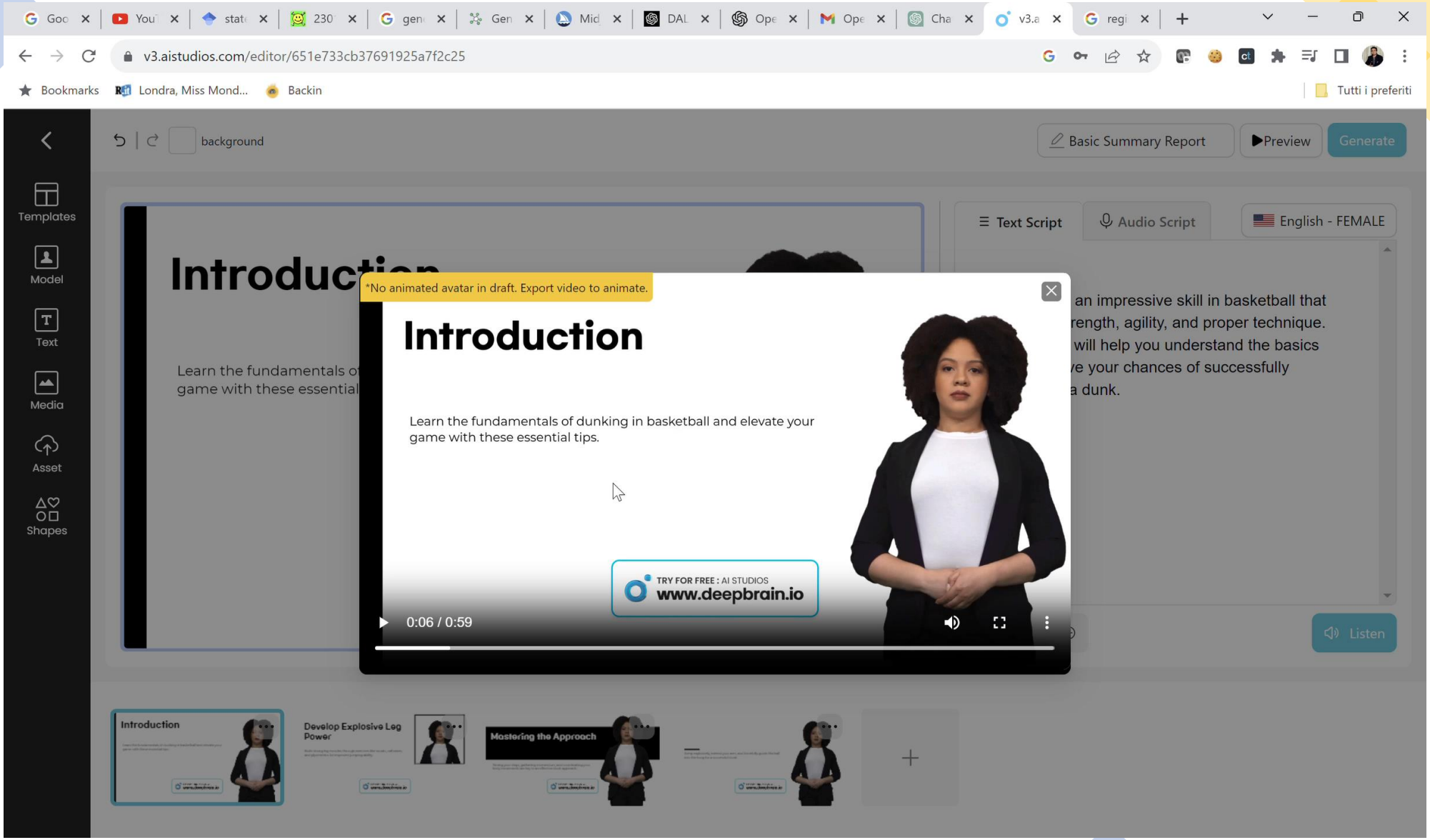
- Image Generation is a process of using deep learning algorithms such as VAEs, GANs, and more recently Stable Diffusion, to create new images that are *visually similar* to real-world images.
- Image Generation can be used for data augmentation to improve the performance of machine learning models, as well as in creating art, generating product images, and more.

Midjourney

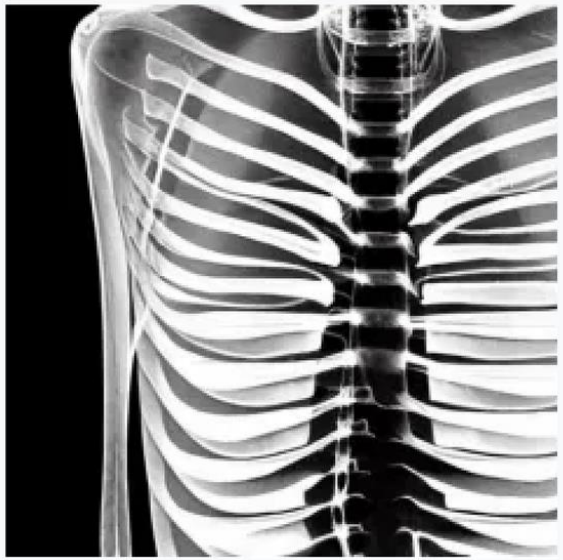
“ A cat and a female character in a spaceship exploring a hidden galaxy. With detailed backgrounds, expressive characters, including magical elements, illustration made by hand. ”

Video and Speech Generation

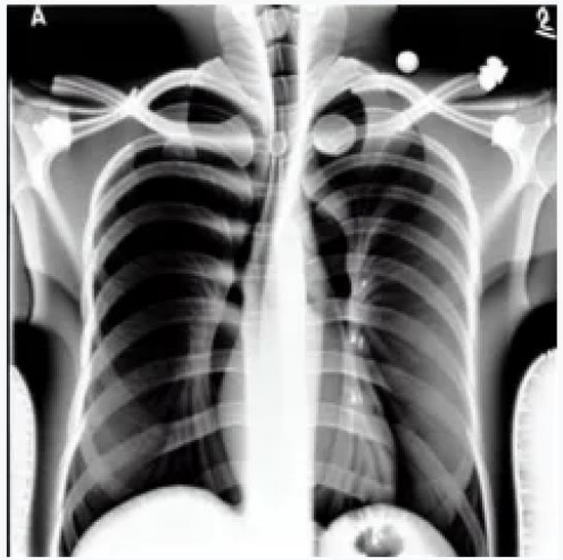
- Video Generation involves deep learning methods such as GANs and Video Diffusion to generate new videos by predicting frames based on previous frames.
- Video Generation can be used in various fields, such as entertainment, sports analysis, and autonomous driving.
- Video Generation can be often seen in use with Speech Generation. The models used for speech generation can be powered by Transformers.
- Speech Generation can be used in text-to-speech conversion, virtual assistants, and voice cloning.


v3.aistudios.com/editor/651e733cb37691925a7f2c25
Bookmarks
Londra, Miss Mond...
Backin
Tutti i preferiti
background
Basic Summary Report
Preview
Generate
Templates
Model
Text
Media
Asset
Shapes
Introduction
Learn the fundamentals of
game with these essential
Text Script
Audio Script
English - FEMALE
an impressive skill in basketball that
rength, agility, and proper technique.
will help you understand the basics
e your chances of successfully
a dunk.
Listen
*No animated avatar in draft. Export video to animate.
Introduction
Learn the fundamentals of dunking in basketball and elevate your game with these essential tips.
TRY FOR FREE : AI STUDIOS
www.deepbrain.io
0:06 / 0:59
Introduction
Develop Explosive Leg Power
Mastering the Approach

Stable Diffusion (Original)

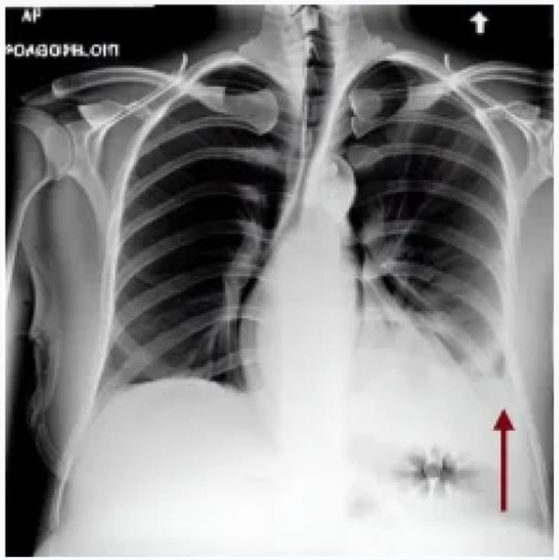


Fine-Tuned Stable Diffusion (ours)



Stable Diffusion (Original)

Fine-Tuned Stable Diffusion (ours)



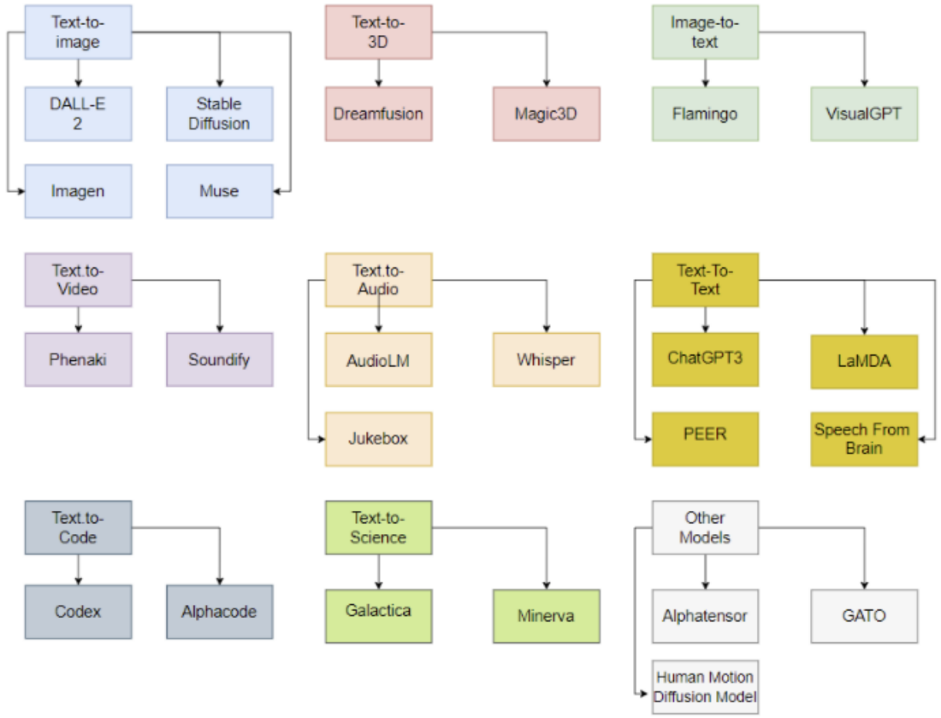
A photo of a lung xray with a

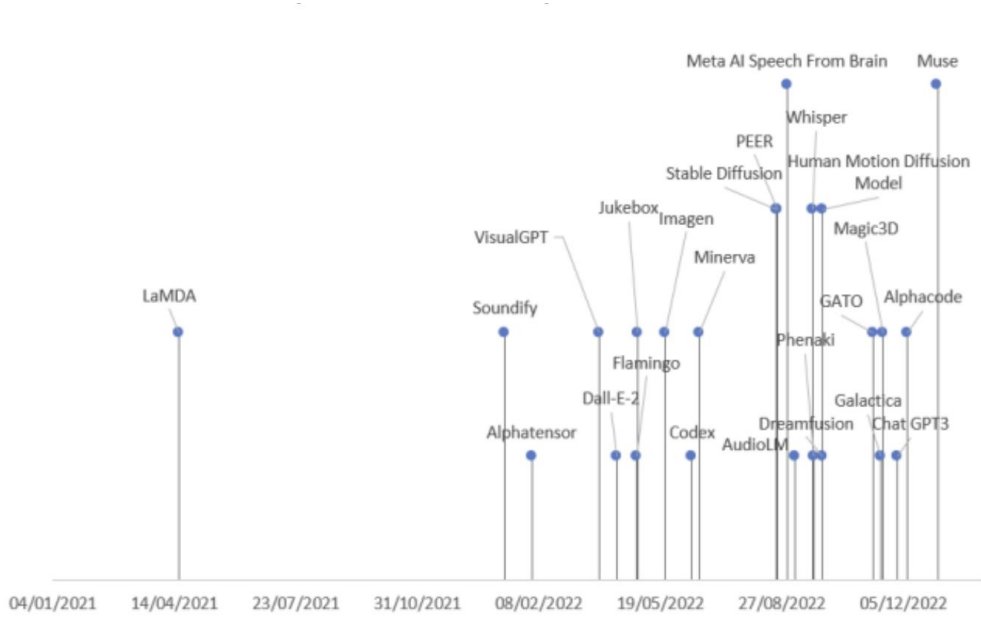
Data Augmentation

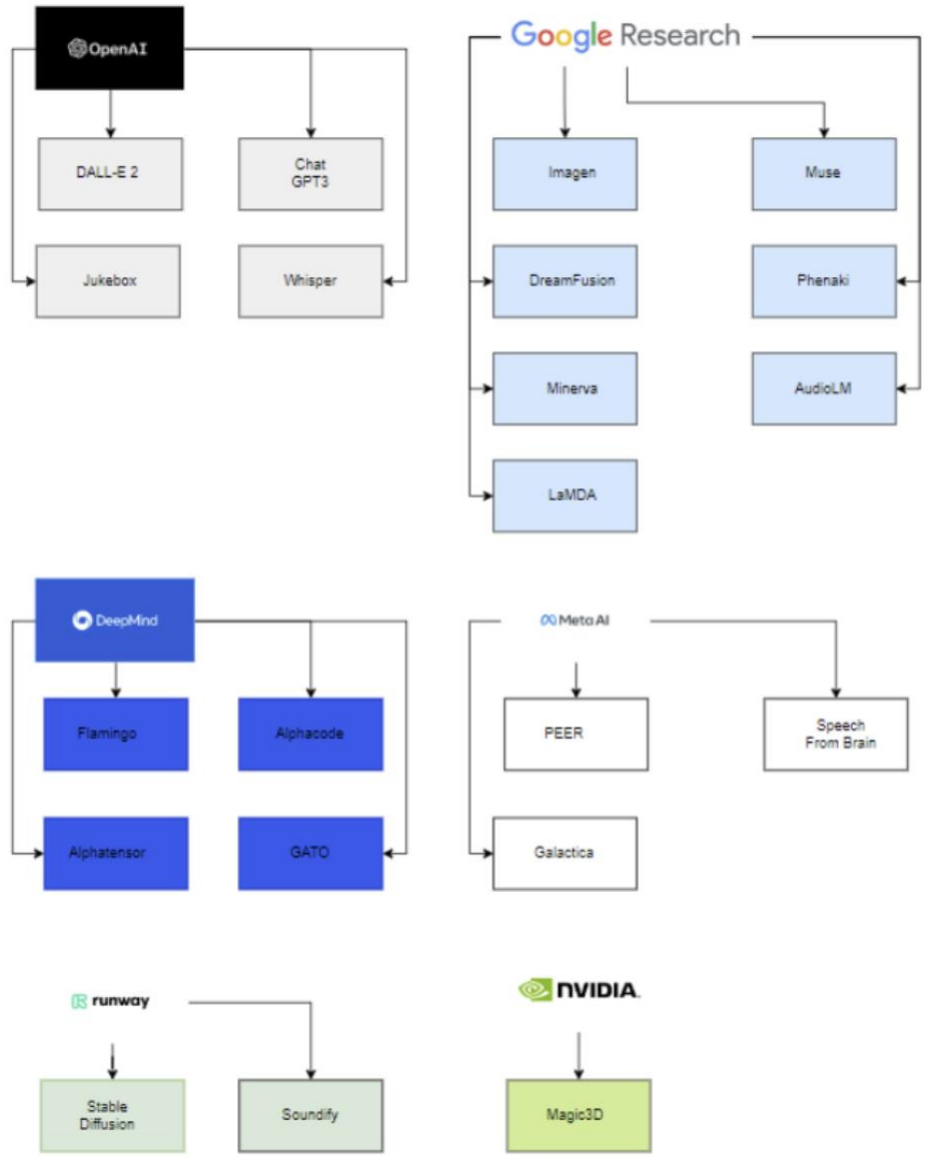
- Data augumentation is a process of generating new training data by applying various image transformations such as flipping, cropping, rotating, and color jittering.
- The goal is to increase the diversity of training data and avoid overfitting, which can lead to better performance of machine learning models.

A Taxonomy of Generative AI models

A timeline

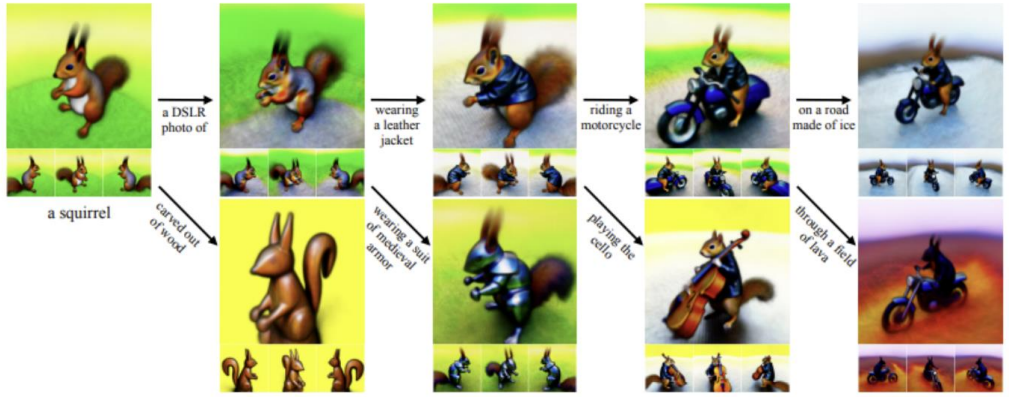





OpenAI
DALL-E 2
Chat GPT3
Jukebox
Whisper
Google Research
Imagen
Muse
DreamFusion
Phenaki
Minerva
AudioLM
LaMDA
DeepMind
Flamingo
Alphacode
Alphatensor
GATO
Meta AI
PEER
Speech From Brain
Galactica
runway
Stable Diffusion
Soundify
NVIDIA
Magic3D

Text-to-3D Models…

- **Dreamfusion**
- is a text-to-3D model developed by Google Research that uses a pretrained 2D text-to-image diffusion model to perform textto-3D synthesis.
- Replaces previous CLIP techniques with a loss derived from distillation of a 2D diffusion model



… Text-to-3D Models

- **Magic3D**
- This model is a text to 3D model made by NVIDIA Corporation.
- While the Dreamfusion model achieves remarkable results, the method has two problems: mainly, the long processing time and the low-quality of the generated images.
- However, these problems are addressed by Magic3D using a two-stage optimization framework

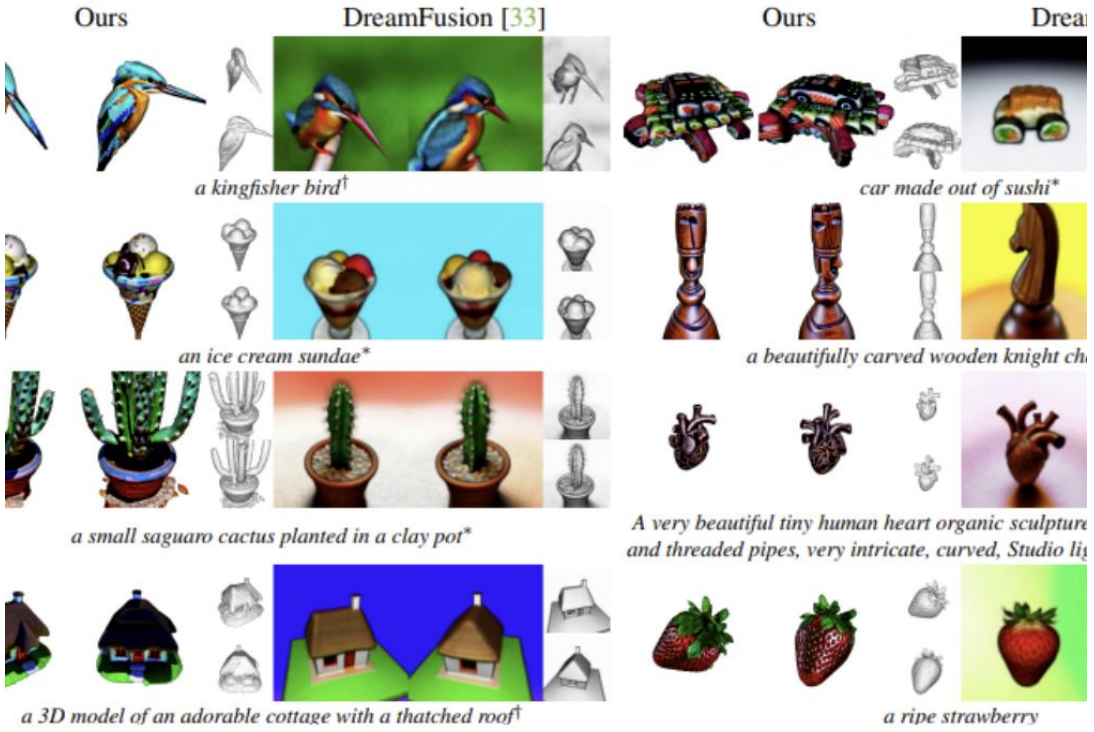
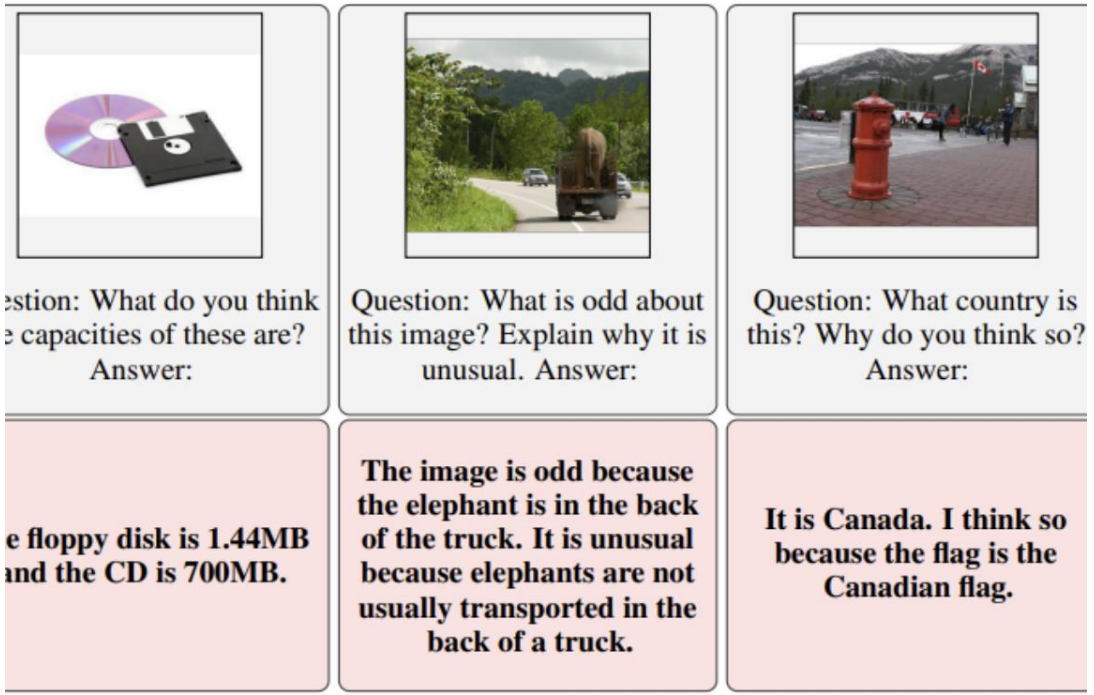


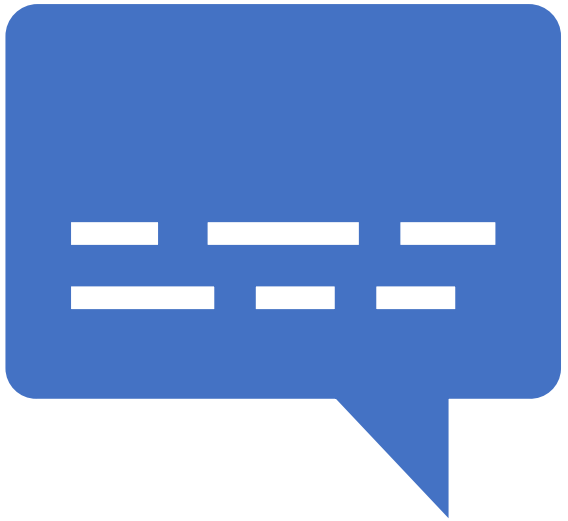
Image-to-Text…

- **Flamingo**
- A Visual Language Model created by Deepmind using few shot learning on a wide range of open-ended vision and language tasks, simply by being prompted with a few input/output examples



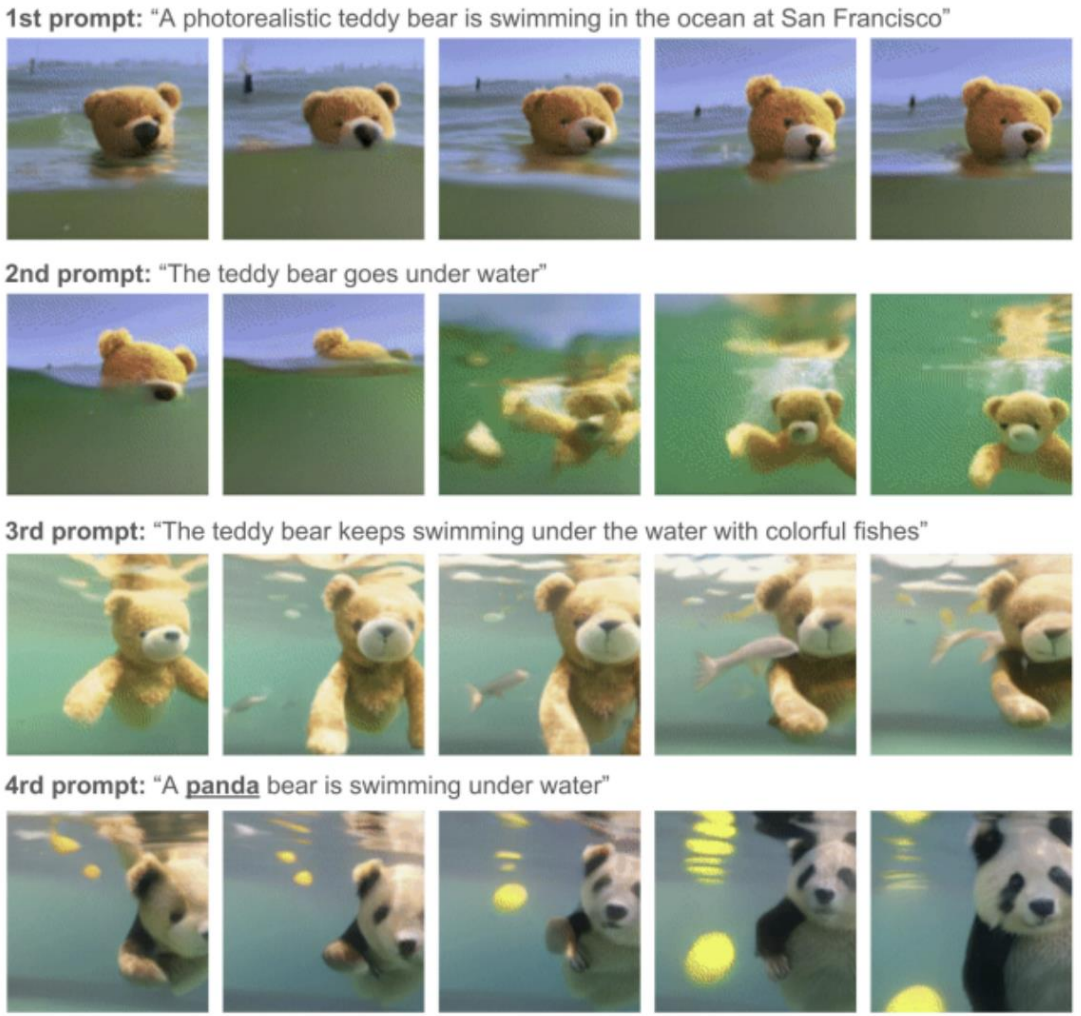
...Image-to-Text

- **VisualGPT**
- An image captioning model made by OpenAI.
- In order to bridge the semantic gap between different modalities, a novel encoder-decoder attention mechanism is designed with an unsaturated rectified gating function.
- Critically, the biggest advantage of this model is that it does not need for as much data as other image-to-text models.



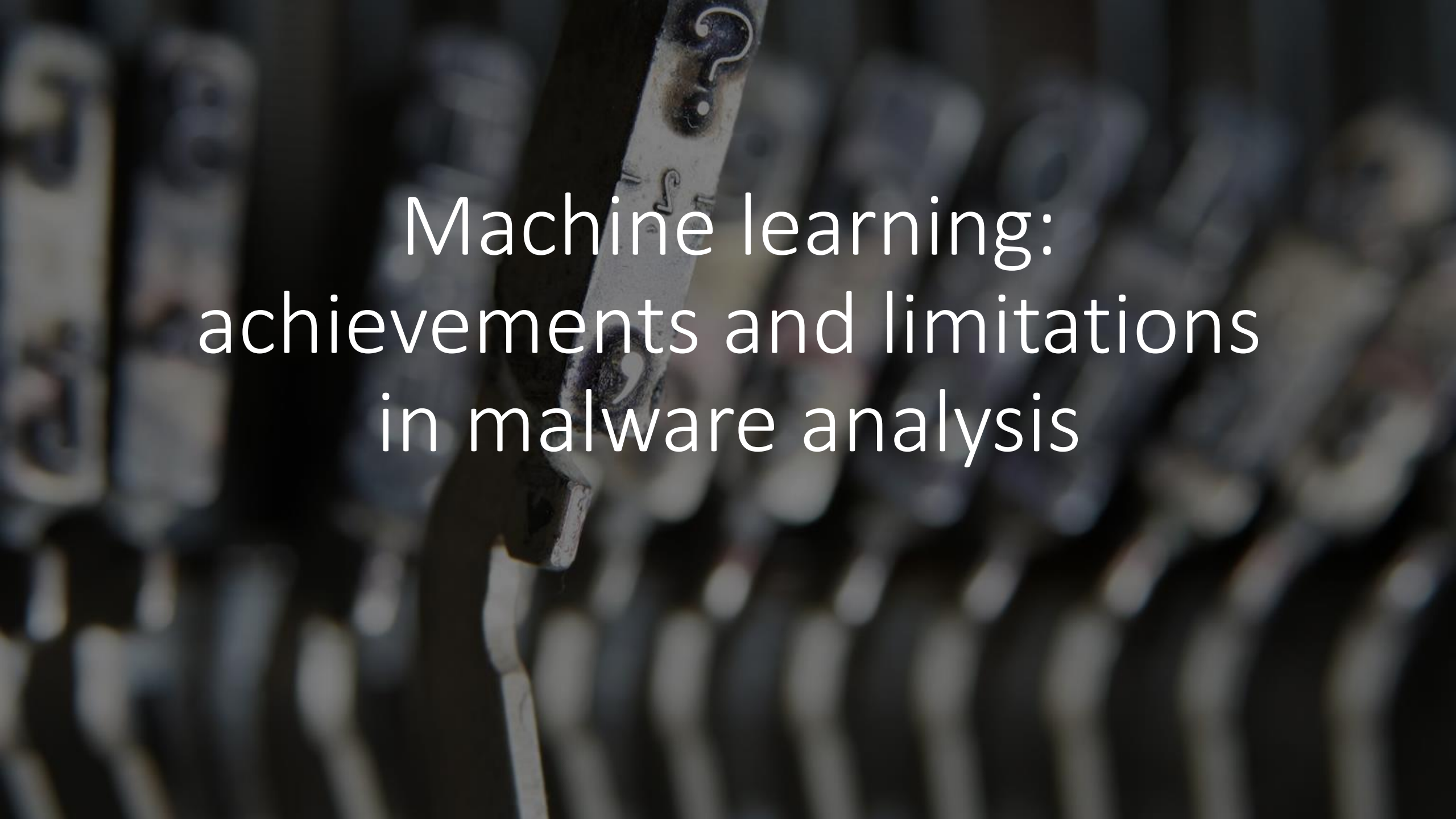
Text-to-Video

- **Phenaki**
- This model has been made by Google Research, and it is capable of performing realistic video synthesis, given a sequence of textual prompts.
- It is the first model that can generate videos from open domain time variable prompts



Limitations

- Enormous volumes of data
- Considerable Computational Power
- Trying solutions out of the problems in the dataset
- Bias from the data which needs to be controlled
- Lack of understanding of ethics



Machine learning: achievements and limitations in malware analysis

Malware Analysis with Machine Learning

Classification

It includes two phases, building the model and using the model.
The classifier labels the test set based on the model and the extracted features.

Clustering

Groups all malware that exhibit similar behavior.
Helps define signatures.

Types of Learning

1. Supervised (inductive) learning: Uses algorithms that learn the relationship between input and output data using labeled data.
- Input provided: training data + desired outputs (labels);

2. Unsupervised learning: Search for patterns and relationships in a data set without pre-existing labels.
- Input provided: training data (without desired output);

3. Semi-supervised learning: falls between supervised and unsupervised learning. It is a method that uses a small amount of labeled data and a large amount of unlabeled data to train a model.
- Input provided: training data + a few desired outputs;

4. Reinforcement learning: Given a sequence of states and actions with (delayed) rewards, a policy is produced.
- A Policy is a mapping between states and actions that indicates what to do in a given state.

Algorithms of Machine Learning

Supervisioned Learning

- Decision tree;
- Random Forest;
- Linear regression;
- Logistic regression;
- Support Vector Machines & kernel methods.

Non supervisioned Learning

- Clustering:
- SVD;
- PCA;
- k-means.

Reinforcement learning

- Q-learning;
- R-learning;
- TD-learning.

Supervisioned Machine Learning Algorithms

Decision Tree: it is used for classification and regression modeling. To predict a class label for a record, start at the root of the tree and compare the root attribute values with the record attribute. Based on the comparison, it follows the branch corresponding to that value and jumps to the next node;

Random Forest: consists of a large number of individual decision trees that operate synergistically. Each individual tree outputs a prediction about the class, and the class with the most votes becomes our model's prediction;

Linear regression: calculates the linear relationship between a dependent variable and one or more independent features;

Logistic regression: is used to predict the categorical dependent variable (y) using a given set of independent variables (x);

Support Vector Machines & kernel methods: kernels or kernel methods (also called kernel functions) are a set of different types of algorithms used for pattern analysis. They are used to solve a nonlinear problem using a linear classifier. Kernel methods are employed in SVMs (Support Vector Machines), used in classification and regression problems.

Clustering:

- **Singular Value Decomposition (SVD):** is a matrix decomposition method that allows you to reduce a matrix to its constituent parts to simplify some subsequent matrix calculations. It is often used in a wide range of applications, including compression, denoising and data reduction;

- **Principal Component Analysis (PCA):** is a dimensionality reduction technique that transforms a set of features in a dataset into a smaller number of features called principal components, while trying to retain as much information in the set as possible. original data;

- **K-means:** adjusts the classification of observations into clusters and updates the cluster centroids until the position of the centroids is stable over successive iterations.

2 QUESTIONS

Which is the best feature (s' set) to recognize malware?

Is the hash enough?

Features

Windows API and system calls: Windows API calls are used by almost all programs to send requests to the operating system -> they can reflect the behavior of the program.

N-grams: N-grams are all substrings of program code with a length of N.Strings: Interpretable strings are the high-level specifications of malicious behavior. These strings can reflect the attacker's intent and goal, as they often contain important semantic information.

OpCodes: An OpCode (i.e. an operational code) is the subdivision of a machine language instruction that identifies the operation to be performed.

Control Flow Graphs (CFGs): A CFG is a graph that represents the control flow of a program.

Network activity: protocols used, TCP/UDP ports, HTTP requests, interactions at DNS level.

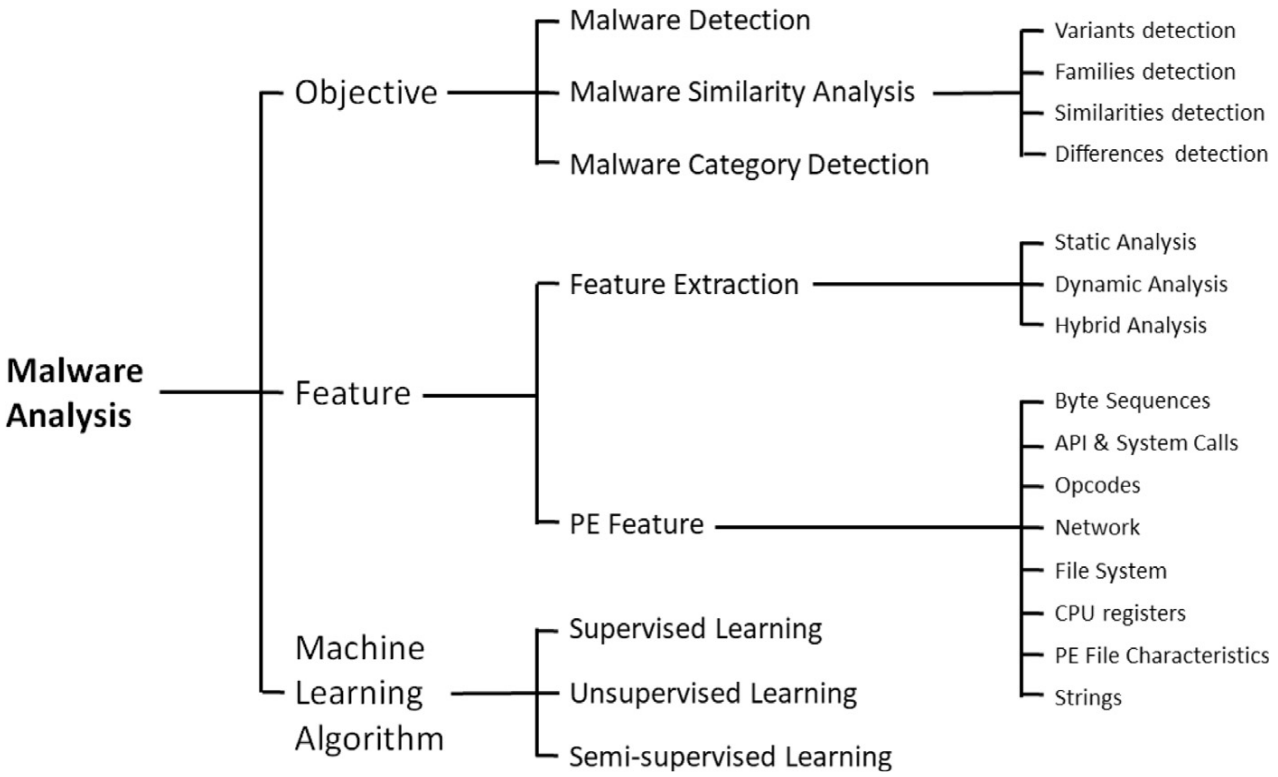
File System: How many files are read or modified, what file types and in which directories, and which files appear on uninfected/infected machines.

CPU Registers: Whether a hidden register is used and what values are stored in the registers, specifically the FLAGS register.

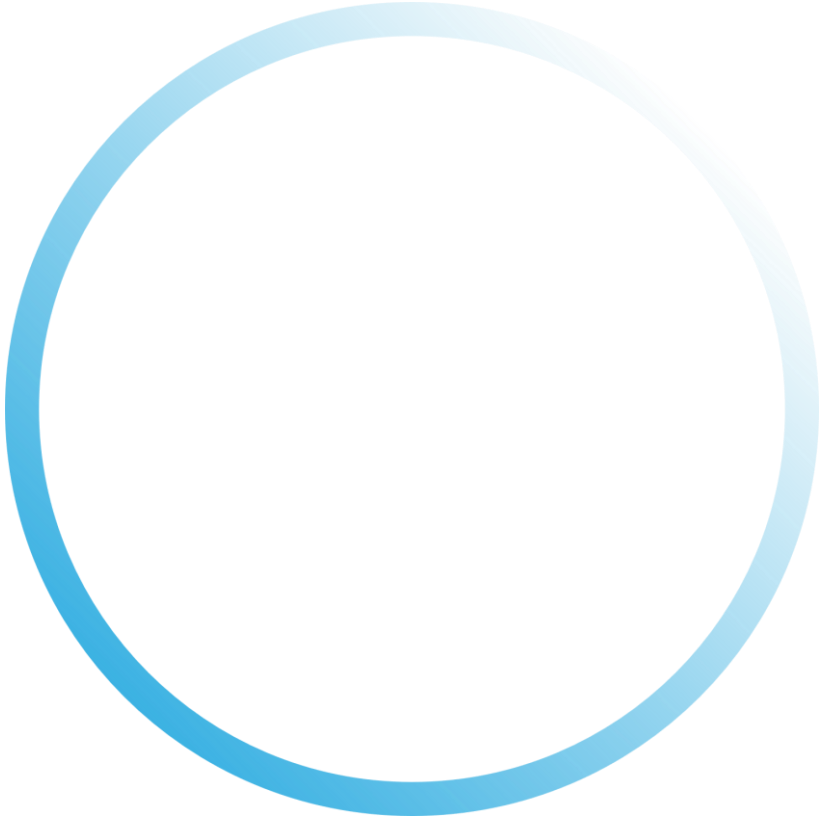
Characteristics of PE files: sections, imports, symbols, compilers used.

Malware Analysis & Machine Learning

Da «Ucci, D., Aniello, L., & Baldoni, R. (2018). Survey of machine learning techniques for malware analysis. «*Computers & Security*»



2 baises in malware classification



Spatial bias refers to unrealistic assumptions about the relationship between goodware and malware in the data.

Temporal bias refers to temporally inconsistent evaluations that integrate future knowledge of test objects into the training phase or create unrealistic settings.

Concept drift: Over time, classifier performance begins to degrade as the model fails to classify the new objects correctly

[1]F. Barbero, F. Pendlebury, F. Pierazzi and L. Cavallaro, "Transcending TRANSCEND: Revisiting Malware Classification in the Presence of Concept Drift," 2022 IEEE Symposium on Security and Privacy (SP), San Francisco, CA, USA, 2022, pp. 805-823, doi: 10.1109/SP46214.2022.9833659.

From Machine to Deep Learning

- ML-based malware detectors work on extracting features of malware (and benign programs) and static and/or dynamic analyzes can be performed.

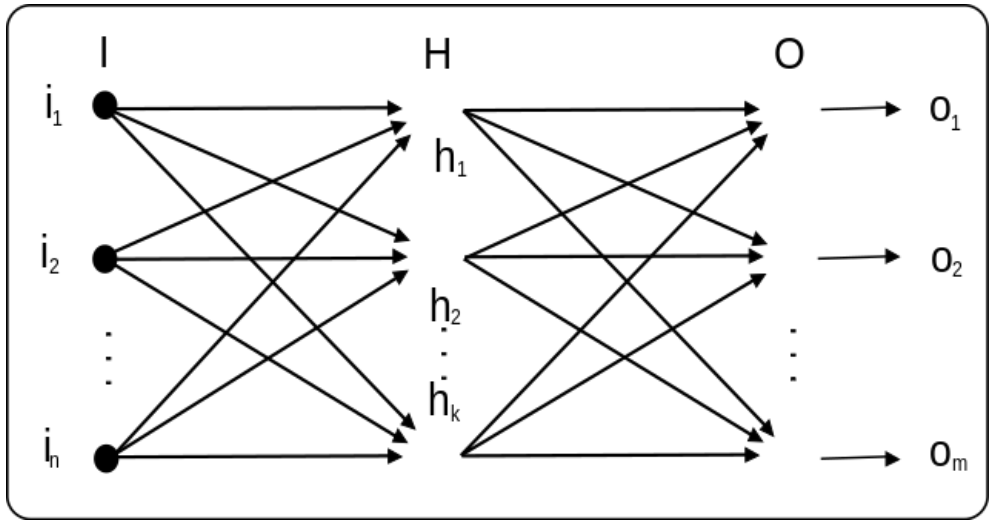
- Bottlenecks:
  - the high time required for learning when the number or size of features is large or the number of sample programs is large;
  - Reducing the number of features may reduce the learning time, but the accuracy in the detection task may decrease.

- Conventional machine learning algorithms have some limitations, as it is difficult to extract well-represented features due to the "curse of dimensionality", the computational bottleneck and the strong requirements of domain and expert knowledge.

- Deep neural networks represent a particular type of machine learning algorithm, which exploits different "deep" layers of networks. Furthermore, deep learning solves the representation problem by building multiple simple features to model a complex concept.

- Hardware acceleration can be useful.

Neural Network



If *f* overcomes a certain threshold the correspondent neuron is activated.

$$y=f(x)$$

$$f=g(\sum wi * xi\ )$$

$$F=f(G)$$
$$G=g(H)$$
$$H=h(X)$$

$$C(f^*) \leq C(f)\ \forall f \in F$$

Feedforward: Cost Function C(f)

C:F->R

The optimal solution f*

No f has a C(f) lower than the f*

Deep Learning & Malware Analysis

- Traditionally, researchers build a single deep learning model using the entire dataset.
- The single deep learning model may not effectively handle increasingly complex distributions of malware data, as different sampling subspaces representing a group of similar malware may have a unique data distribution.
- Building a single deep learning model using a mixed dataset has two main disadvantages:
  - Distribution of complex data;
  - Scalability.
- Combining different types of malware into a single dataset results in a very complex overall data distribution -> a single perspective
- Training deep learning models on very large data sets is a computationally expensive process.
- An alternative to using a single deep learning model to build malware detection systems (MDS) is to develop ensemble based deep learning models.

CCN and RNN

- **Convolutional Neural Networks** (CNNs) employ convolution operations on hidden layers for weight sharing and parameter reduction.
  - can extract local information from gridded input data.
  - have demonstrated incredible successes in computer vision tasks, such as image classification, object detection, and semantic segmentation.

- **Recurrent Neural Networks** (RNNs) are neural networks adopted to process sequential input data of variable length. RNNs produce an output at each step. The hidden neuron of each step is calculated based on the input data and the hidden neurons of a previous step.

Adversarial attacks to malware detectors

The question is

How and to what extent are GANs able to degrade the performance of machine learning-based malware detectors?

Adversarial Machine Learning

- Adversarial machine learning is the field that studies a class of attacks that aim to worsen the performance of classifiers on specific tasks.

- Attack classes:
  - Poisoning: The attacker affects the training data or labels.
  - Evasion: The attacker manipulates data during distribution to fool previously trained classifiers.

To model an Adversarial Attack[1]

- **Influence** - refers to the attacker's abilities to interact with the target. Can be:
  - a causal attack if the attacker can tamper with the classifier's training data,
  - exploratory if the attacker cannot influence the training process, but can use other techniques to probe for useful information;

- **Security violation** - refers to the type of violation of the attack. These can be:
  - integrity attacks when the goal is for the attack to be classified as normal (false negative),
  - availability attacks, when the goal is to cause classification errors of any kind (false negatives or false positives), making the model unusable, or
  - attacks on privacy, when the objective is to obtain information from the "learner";

- **Specificity -** refers to the magnitude of the classification error. These can be:
  - targeted attacks when the intent is to misclassify attacks into a certain class/group of classes, or
  - indiscriminate when there is no specific class to target and the goal is only to cause a misclassification.

[1]L Huang, A. D. Joseph, B. Nelson, B. I. P. Rubinstein, and J. D. Tygar, ''Adversarial machine learning,'' in Proc. 4th ACM Workshop Secur. Artif. Intell., 2011, pp. 43–58.

Strategies

- Gradient-based methods typically introduce perturbations optimized for certain distance metrics between the original and perturbed samples.
- Box constrained limited-memory Broyden–Fletcher–Goldfarb–Shanno (L-BFGS) [1]. Given an input image, search for another image similar to the first, which was mislabeled by the classifier.
- Fast Gradient Sign method (FGSM), aiming to minimize the maximum amount of perturbation added to any pixel in the image to cause misclassification [2].

[1] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, ''Intriguing properties of neural networks,'' 2013, arXiv:1312.6199. [Online]. Available: http://arxiv.org/abs/1312.6199

[2] I. J. Goodfellow, J. Shlens, and C. Szegedy, ''Explaining and harnessing adversarial examples,'' 2014, arXiv:1412.6572. [Online]. Available: http://arxiv.org/abs/1412.6572

Strategies

- Jacobian based Saliency Map Attack (JSMA) which, unlike the previous method, uses feature selection, with the aim of minimizing the number of modified features while still causing classification errors [1].
- Deepfool, aiming to minimize the Euclidean distance between the perturbed samples and the original samples [2].
- Zeroth-order optimization attack (ZOO) was proposed by Chen et al. and allows you to estimate the gradient of the classifiers without having access to the classifier, which is ideal for a "black box" attack[3].

[1] N. Papernot, P. McDaniel, S. Jha, M. Fredrikson, Z. B. Celik, and A. Swami, ''The limitations of deep learning in adversarial settings,'' in Proc. IEEE Eur. Symp. Secur. Privacy (EuroS P), Mar. 2016, pp. 372–387.
[2] S.-M. Moosavi-Dezfooli, A. Fawzi, and P. Frossard, ''DeepFool: A simple and accurate method to fool deep neural networks,'' 2015, arXiv:1511.04599. [Online]. Available: http://arxiv.org/abs/1511.04599
[3] P. Y. Chen, ''Zoo: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models,'' 2017, arXiv:1708.03999. [Online]. Available: https://arxiv.org/abs/1708.03999

Adversarial attacks to malware detectors

- The main difference between images and malware is that images are continuous, while malware characteristics are binary.
- Arbitrary modification of bytes may break the semantics and syntax of the executable (PE/COFF).
- Solutions:
  - adding padding bytes (adversary noise) to the end of a file beyond PE [1] boundaries
  - Inject adversarial noise into an unused, unmapped PE region in memory. [2]
  - add some noise to the malware by extracting features (list of APIs) from the clean malware and inserting them into generator [3]
  - a method based on the Wasserstein Generative Adversarial Network (WGAN) to generate a malicious PDF with an evasion rate of 100% [4]; in this work the authors modify : dimensions, metadata and structural attributes.

[1]Bojan Kolosnjaji, Ambra Demontis, Battista Biggio, Davide Maiorca, Giorgio Giacinto, Claudia Eckert, and Fabio Roli***. Adversarial malware binaries: Evading deep learning for malware detection in executables***. In *2018 26th European Signal Processing Conference (EUSIPCO)*, pages 533–537. IEEE, 2018.
[2]Felix Kreuk, Assi Barak, Shir Aviv-Reuven, Moran Baruch, Benny Pinkas, and Joseph Keshet. ***Deceiving end-to-end deep learning malware detectors using adversarial examples***. *arXiv preprint arXiv:1802.04528*, 2018.
[3] Masataka Kawai, Kaoru Ota, and Mianxing Dong. ***Improved malgan: Avoiding malware detector by leaning cleanware features***. In *2019 International Conference on Artificial Intelligence in Information and Communication (ICAIIC)*, pages 040–045. IEEE, 2019.
[4] Jinlan Zhang, Qiao Yan, and Mingde Wang. ***Evasion attacks based on wasserstein generative adversarial network***. In *2019 Computing, Communications and IoT Applications (ComComAp)*,pages 454–459. IEEE, 2019.

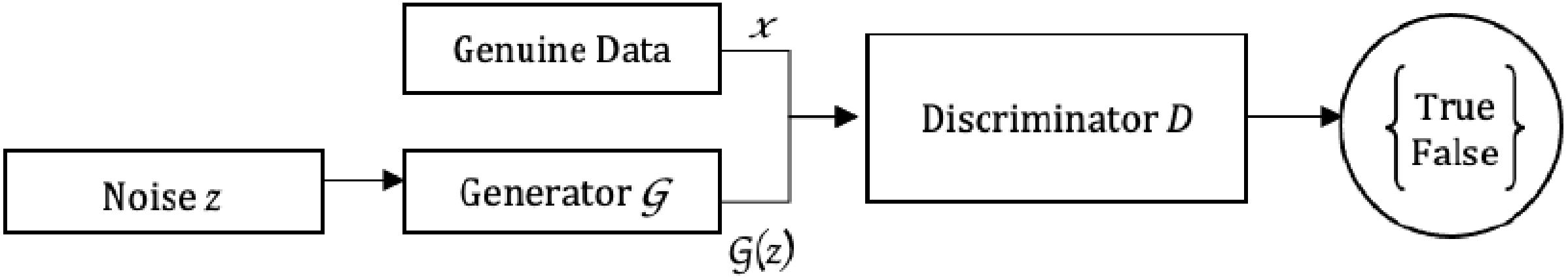
GAN

A GAN is a tool that produces adversarial samples using adversarial machine learning.

Adversarial machine learning has been applied with some success especially to the field of image recognition, with surprising results, but also to speech recognition and biometric recognition.

Orignal Sample *Adversarial Sample*

Ian Goodfellow, Patrick McDaniel, and Nicolas Papernot. *Making machine learning robust against adversarial inputs*. Communications of the ACM, 61(7):56–66, 2018.



Gan Main Concepts…

- A Generative Adversarial Network, at its base, is a machine learning algorithm built out of two separate deep learning networks (The Generator and the Discriminator) which work together, competing to win a **zero-sum game**.
- One network takes in noise and then attempts to create samples with the right characteristics to have them seem like real or 'genuine' samples. The second network takes as input both real and generated samples, and then classifies them as either real or fake samples.
- The back-propagation that occurs then is the backbone of the model. If the Discriminator network is right, information is sent back to the Generator network, so that it will adjust its weights and probability distributions to improve the quality of its forgeries.
- If the Discriminator instead gets it wrong, that information is sent to the Discriminator to make the necessary adjustments. The games end when the Discriminator has the accuracy of a **coin flip** - when the forgeries are all but impossible to separate from the genuine samples.

Wang, K., Gou, C., Duan, Y., Lin, Y., Zheng, X., Wang, F.-Y.: Generative adversarial networks: introduction and outlook. IEEE/CAA Journal of Automatica Sinica 4(4), 588–598 (2017)

...Gan Main Concepts

- This type of network architectures are typically referred to as Generative Adversarial Networks (GANs) and solve an optimization function described by:

$$\min_G \max_D V(D, G) = \mathbb{E}_{x \sim P_r}[\log D(x)] + \mathbb{E}_{z \sim P_z}[\log D(G(z))]$$

- Where D and G denote the discriminator and generator, Pr and Pz are the distribution of input data and noise, respectively.
- In this competition, GAN is able to generate raw data samples that are close to real data.

Generator

- The Generator Network within the GAN model starts with a random seed or noise as input, and produces an output which starts understandably far from the goal.
- Each iteration, each epoch, the Generator syntheses data that is more and more realistic.
- Convergence and training of the Generator Network is finished when the Discriminator cannot tell if the synthesised data is real or manufactured with any more than 50% accuracy- effectively becoming as useful in classification as a coin toss.

Discriminator

- The Discriminator's job essentially comes down to a binary option. Is the data real, or is it synthesised?
- This part of the network, like most deep neural networks, is about feedback - in this case, the back-propagation of the result to the Generator Network. When the Discriminator guesses incorrectly, that alters the internal weights of the system as the information back-propagates to both Generator and Discriminator.
- The Discriminator is also playing to lose, as the target is to have the Discriminator as effective at distinguishing real data from generated data as a simple coin toss.

Feedback Loop

- Essential to every GAN implementation is the type of feedback response network it utilises.
- Being able to feed the results of the tests run by the Discriminator back to the Generator is key in the development of an effective Generator.
- The back-propagation of the network allows both systems to carefully adjust the weights and probabilities of their internal deep learning neural networks.
- Because of the feedback loop needed by GAN architecture, it is a back-propagation network.
- The function to alter the weights is essential to the training period, and is as such customised in different models and applications such that the network can reach optimal function as efficiently as possible. It is important to remember that this adjustment over time occurs in a black-box.

Types of GAN

Vanilla GAN

- The traditional, or Vanilla, Generative Adversarial Network is the original proposed model from Goodfellow et al's 2014 paper.
- This original version of the Generative Adversarial Network is a deep learning model, and was based on adversarial nets as a framework, with back-propagation.
- An important distinction is that while the discriminator has access to both real and generated data, the generator has no access to either, and so has to rely on the value functions and the back-propagation to change the weights and take the model closer to producing realistic generated output

Conditional GAN

- The Conditional GAN (CGAN)[1] modify the original vanilla GAN.
- In the original model, the generative process could not be controlled or conditioned.
- It was unsupervised entirely. CGANs allow the generation process to be controlled and directed, meaning that the model can be steered towards a focus on a particular class, or feature.
- The change in control occurs when the focus is put on some element y, which can be a class, value, feature, so on, and this y is fed into both the generator and discriminator as an additional layer of input.
- In the original proposal, the generator is fed not only the focus y, but also a noise function, pz ^{1}z°.

[1]Mirza, M., Osindero, S.: Conditional Generative Adversarial Nets. arXiv (2014). 1411.1784

Deep Convolutional GAN

- The Deep Convolutional Generative Adversarial Network, or DCGAN[1], was proposed in a 2015 ".
- While most deep learning algorithms are black-box methods, it is possible through careful tuning to examine the underlying functions of a Convolutional Neural Network (CNN) model. The authors made use of several changes to traditional CNN architecture, from the following papers:
- Striving for simplicity: The all convolutional net[2]
- Inceptionism: Going deeper into neural networks [3]
- Batch normalization: Accelerating deep network training by reducing internal covariate shift

[1]Radford, A., Metz, L., Chintala, S.: Unsupervised representation learning with deep convolutional generative adversarial networks. arXiv preprint arXiv:1511.06434 (2015)

[2]Springenberg, J.T., Dosovitskiy, A., Brox, T., Riedmiller, M.: Striving for simplicity: The all convolutional net. arXiv preprint arXiv:1412.6806 (2014)

[3]Mordvintsev, A., Olah, C., Tyka, M.: Inceptionism: Going deeper into neural networks (2015)

[4]Ioffe, S., Szegedy, C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: International Conference on Machine Learning, pp. 448–456 (2015). PMLR

Bidirectional GAN

- The Bi-directional Generative Adversarial Network (BiGAN)[1] was proposed in a 2017. Similarly to MGAN this is a three party model, consisting of an encoder, a generator, and a discriminator.
- The role of the encoder is to map data x to a latent space representation y. Donahue et al specify that the encoder is taught to invert the generator, even though the modules do not interact with one another or directly process the other module's outputs.
- The BiGAN model is meant to excel at tasks that involve semantic data and representation. They are also an entirely unsupervised model in machine learning.
- Interestingly, BiGAN was brought into the spotlight in Bioinformatics

[1]Donahue, J., Krähenbühl, P., Darrell, T.: Adversarial feature learning. In: 5th International Conference on Learning Representations. International Conference on Learning Representations (2017)

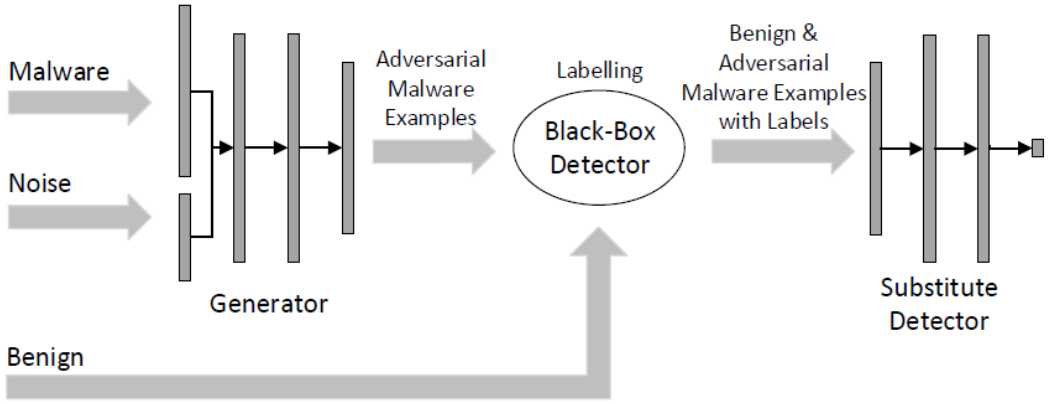
MalGAN

- If M APIs are used as features, an M-dimensional feature vector is constructed for a program. If the program calls the d-th API, the value of the d-th feature is set to 1, otherwise it is set to 0.
- The probability distribution of MalGAN's adversarial examples is determined by the generator weights.
- For a machine learning algorithm to be effective, the training set and test set samples must follow the same or similar probability distributions.

[1]*Hu, W., & Tan, Y. (2017).* ***Generating adversarial malware examples for black-box attacks based on gan****. arXiv preprint arXiv:1702.05983.*

Generator

It takes as input the concatenation of a malware feature vector m and a noise vector z. m is an M-dimensional binary vector. Each element of m corresponds to the presence or absence of a feature. z is a Z-dimensional vector, where Z is a hyperparameter. Each element of z is a random number sampled from a uniform distribution in the interval [0; 1]. The final generated adversary example can be expressed as m' = m|z', where "|" is an elementary binary OR operation.



GAN Models	Balancing datasets	Attack & Security	Malware	Functional Malware	Adversarial Examples	Malicious Traffic	Feature Extraction	Phishing URLs	Behaviour Tracking
Goodfellow GAN[1]	X	X	X	X	X				
CoGAN			X		X				
DCGAN	X	X	X	X	X				
ALI-GAN			X		X				
CoRGAN			X		X				
CoRAGAN			X		X				
Sequential GAN	X	X	X	X	X				
GenAtSeqGAN									X
LSTM GAN		X	X	X	X				X
Bi-objective GAN		X	X	X	X				
WGAN-GP			X	X	X			X	
MalGAN		X	X	X	X				
CGAN									
fvGAN		X					X		
TrafficGAN		X	X			X			
Mal-IGAN	X		X	X	X				
n-gram MalGAN									
IDSGAN	X				X	X			
iKnight					X	X	X		
PDF-GAN			X	X	X	X			
GAN-tCDGAN	X		X		X				
IDS-GAN						X			
Mal-LSGAN		X	X	X	X	X	X		

GANG-MAM: our solution for GAN-based malware generation

Renjith, G., Laudanna, S., Aji, S., Visaggio, C. A., & Vinod, P. (2022).
GANG-MAM: GAN based enGine for modifying Android malware.
SoftwareX, *18*, 100977.

The Original Contribution

- GANG-MAM introduces a novel approach to modifying Android malware using Generative Adversarial Networks (GANs);
- It pioneers the application of GAN technology specifically for malware analysis and modification, contributing to the advancement of cybersecurity research;
- Preservation of Malicious Behavior;
- GANG-MAM offers scalability and versatility in modifying a wide range of Android malware variants.

State of Art

- Manual modification processes are time-consuming and labor-intensive;
- Automated techniques often lack the ability to preserve malware functionality and behavior accurately;
- Existing tools may not adequately address the diversity and complexity of modern malware variants;
- Limited scalability and efficiency in modifying malware samples;
- Difficulty in maintaining the balance between preserving malicious behavior and ensuring sample safety;
- Lack of comprehensive solutions that integrate advanced machine learning techniques with malware modification processes.

Software Description

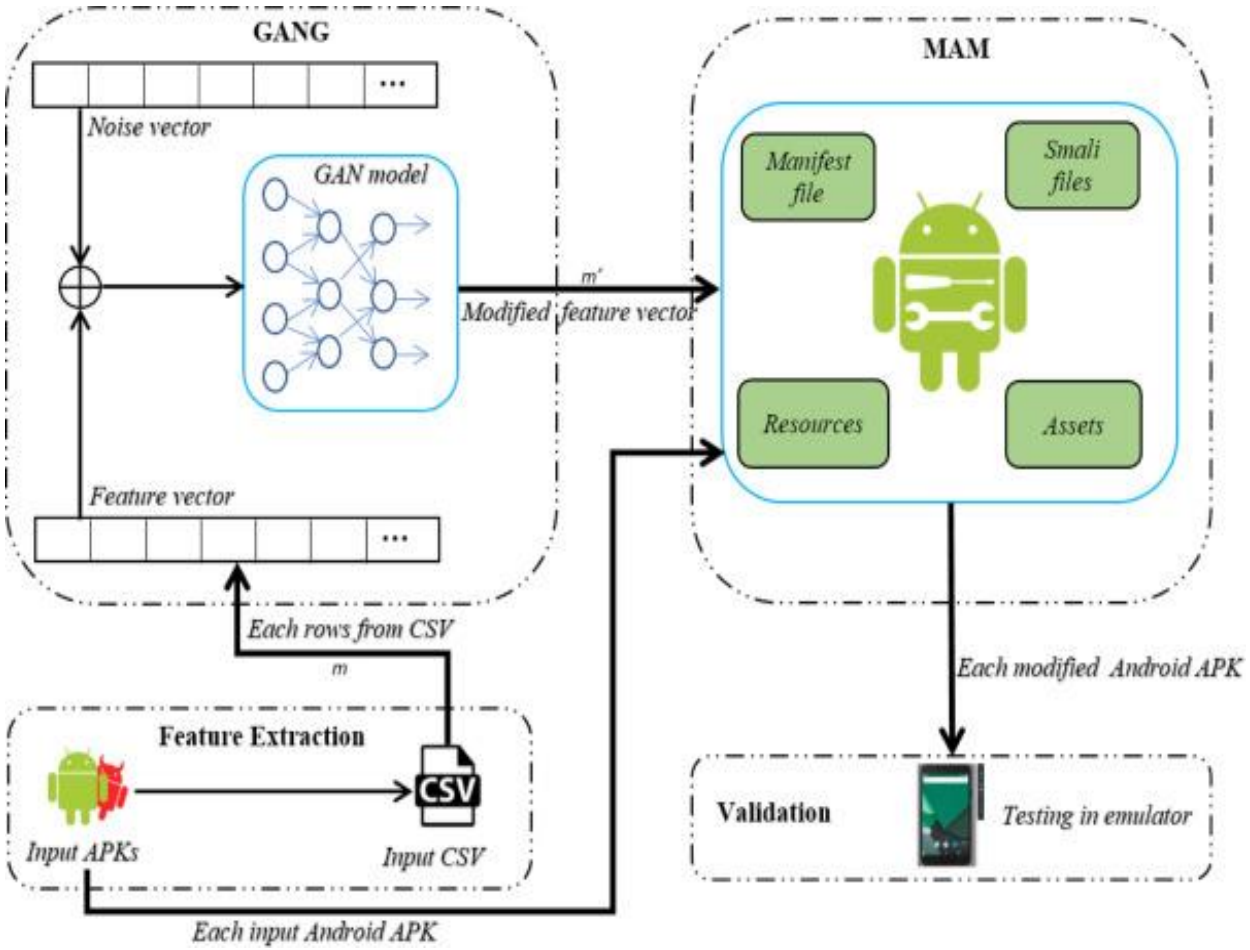
- **Key Components:**

1. GANG (Generative Adversarial Network):

1. Produces evasive feature vectors based on the characteristics of existing malware;
2. Generates feature vectors that enable the malware to evade detection by malware classifiers.

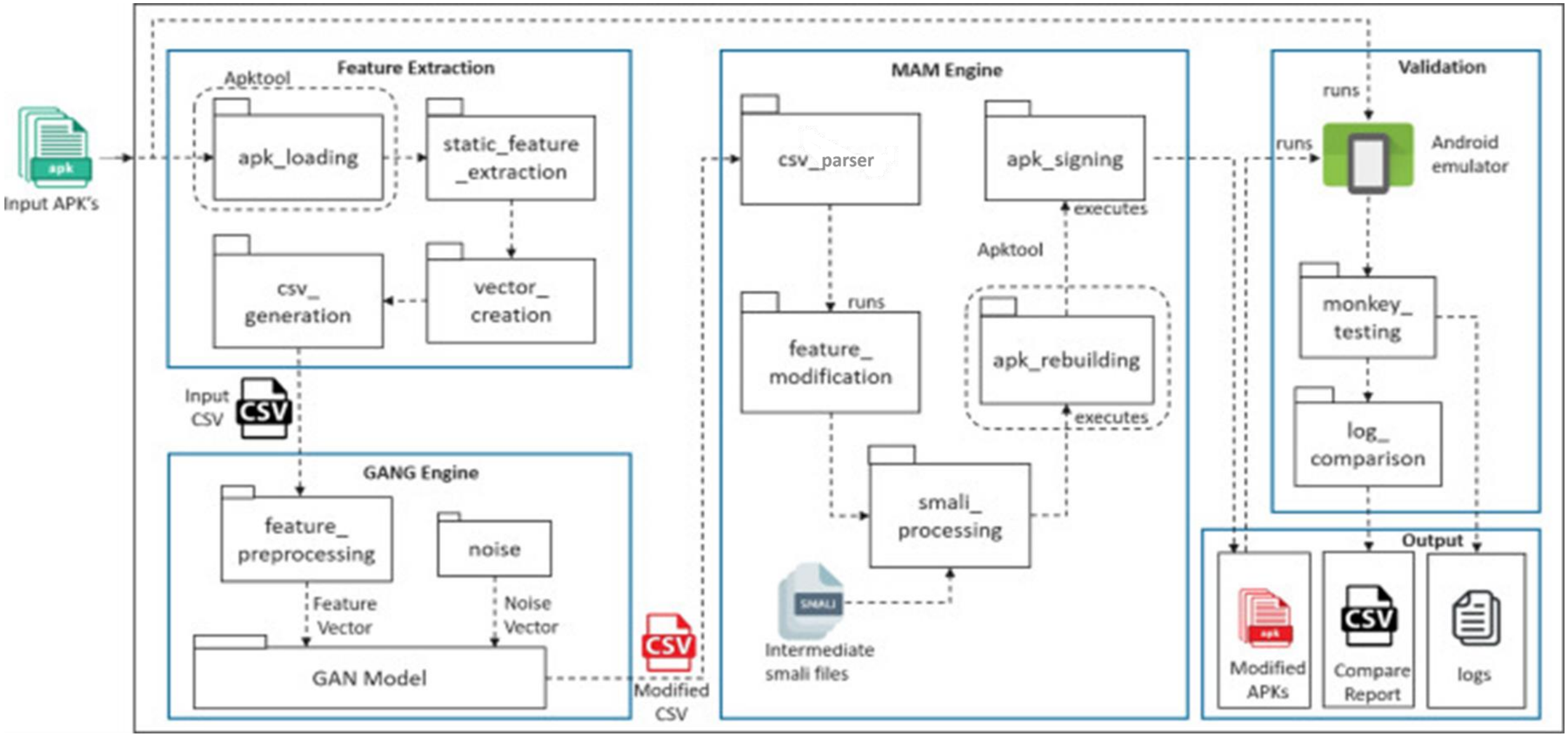
2. MAM (Malware Modification Engine):

1. Modifies existing malware to align with the evasive feature vectors generated by GANG;
2. Ensures that the modified malware retains its original functionality while exhibiting evasive behavior.



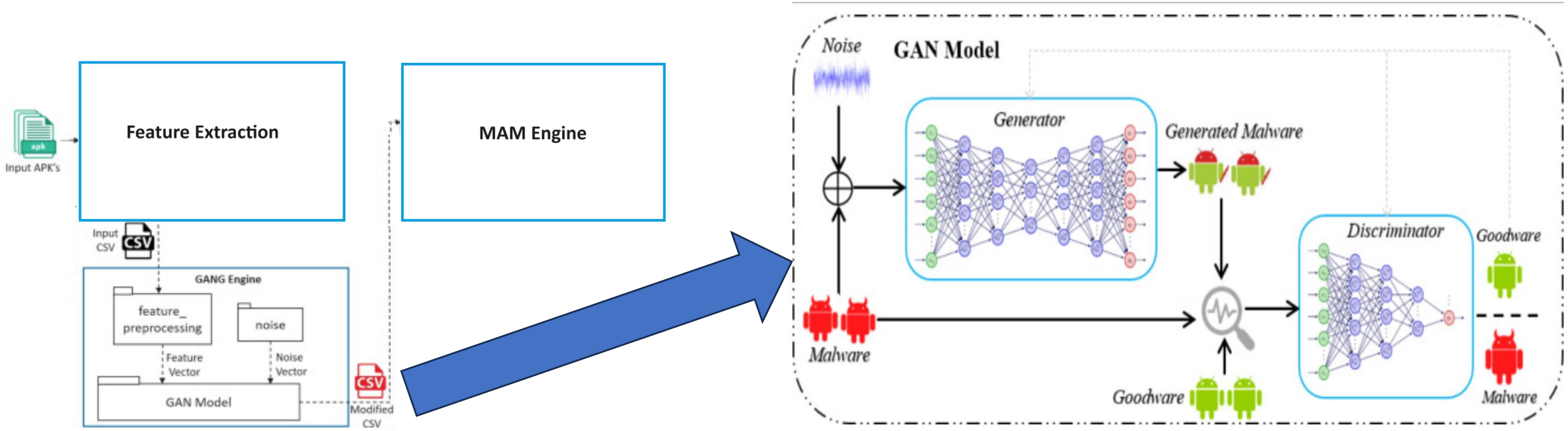
Tool repository: https://github.com/papersubKS

Software Architecture



Features Extraction

Feature	Value
Permissions	permission.ACCESS_NETWORK_STATE
	permission.ACCESS_WIFI_STATE
	permission.READ_PHONE_STATE
	permission.INTERNET
Intents action	action.FTPSERVER_STOPPED
	action.DOCUMENTS_PROVIDER
	action.FTPSERVER_STARTED
	action.UPGRADE_ALL
Services	TapContextService
	QuickAccessibilityService
	PluginPitService
	Push_BroadCast_Service
Intents category	category.MULTIWINDOW_LAUNCHER
	category.DEFAULT
	category.BROWSABLE
	category.LEANBACK_LAUNCHER



Features Generation

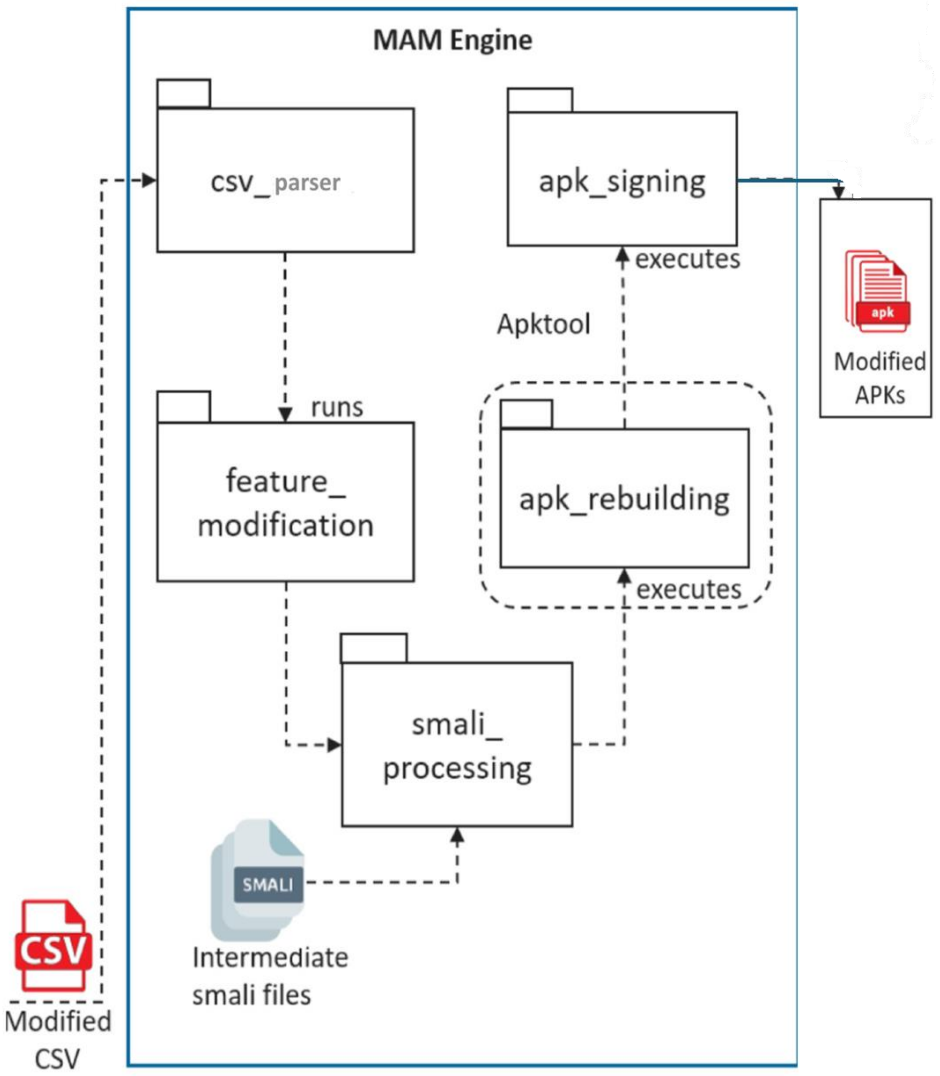
- **Input:** Features vector;
- **Output:** Modified features vector;

Pre-trained GAN Model

- Implemented in Python using Keras and TensorFlow.

Malware Modification

1. Acceptance of Modified Features Vector;
2. Feature Management;
3. Processing of Intermediate Smali Files;
4. Input Validation;
5. Hash Calculation;
6. Feature Application;
7. Repackaging and signing Phases.

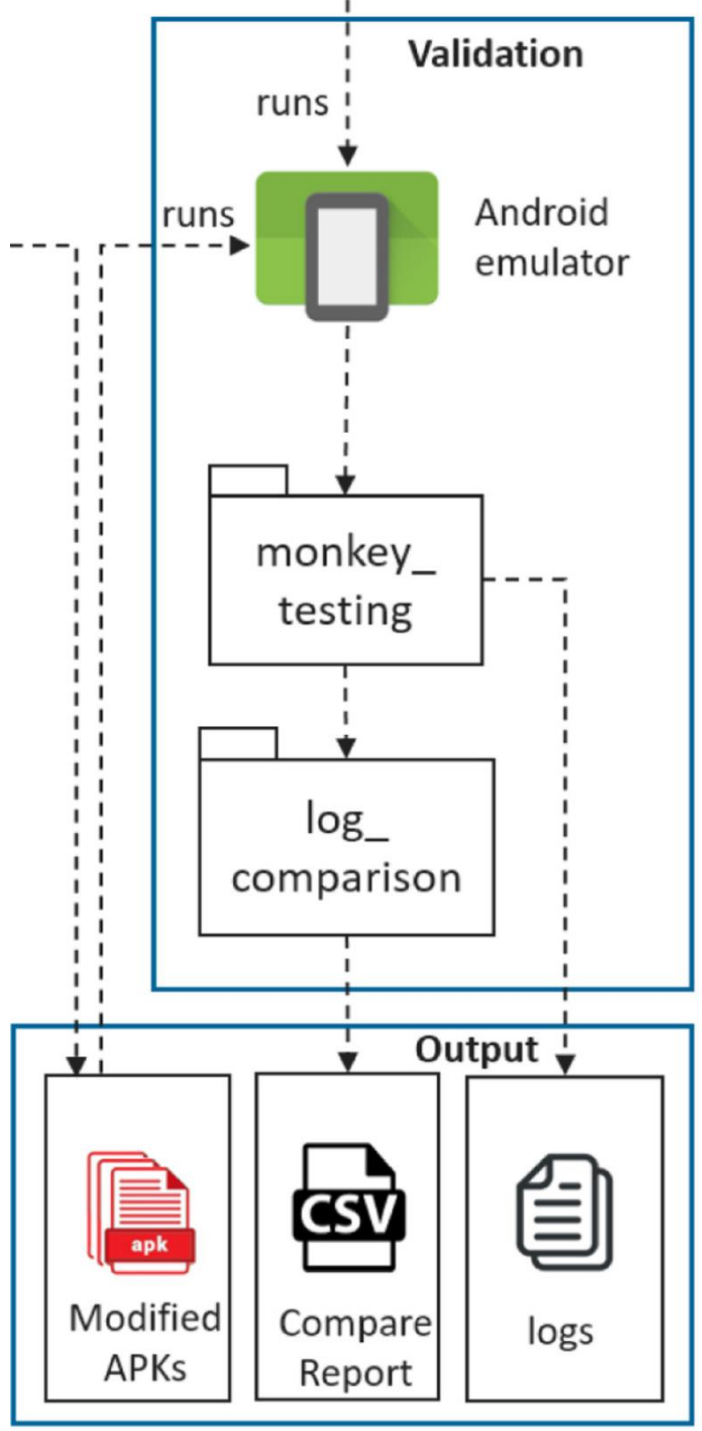


Validation

1.Installation and Execution:
1. Each input APK is installed, launched, and executed using the Monkey tool.
2. Similarly, each output (modified) APK is installed, launched, executed with Monkey, and then uninstalled and cleaned up.

2.Execution Log Comparison:
1. Execution logs for both input and output APKs are stored.
2. Logs are compared to compute the difference in functional execution flow.

3.Preservation of Integrity:
1. Validation ensures that the modified APKs maintain integrity and behave similarly to input APKs.



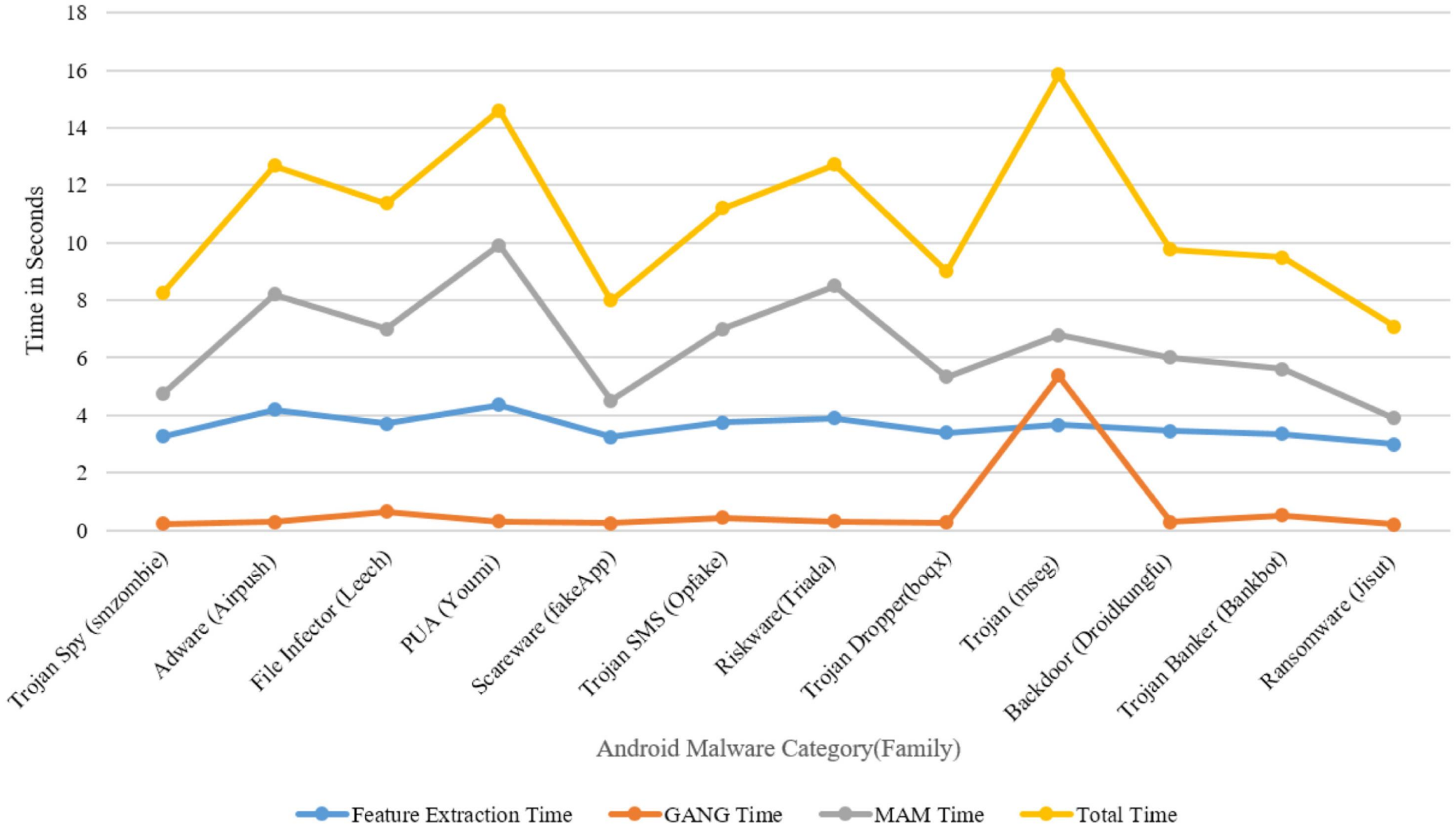
Stability Results

- **Data-set for Stability Testing**:
  - 500 APKs obtained from Androzoo data-set and malware repository [1] were merged;
- **Validation Results**:
  - Almost 492 Android applications maintained integrity and passed validation in Android emulator environment with Monkey;
  - Some applications failed due to requiring login credentials to run in emulator;
- **Execution Details**:
  - Execution logs of original and modified APKs stored separately.
- **Stability Analysis**:
  - Differences in logs manually investigated.
  - Observed differences related to timestamp information or dynamic input data.
  - Operational and functional flow for each APK before and after modification found to be consistent.

#	APK names	Before	After	Diff
1	Trojan Spy Smszombie Malware	139	139	1
2	Trojan Dropper Boqx Malware	141	141	1
3	Adware Airpush Variety 1 Malware	35	35	1
4	Adware Airpush Variety 2 Malware	35	34	3
5	File Infector Leech Variety 1 Malware	141	141	1
6	Riskware Triada Variety 1 Malware	153	153	1
7	Scareware FakeAapp Malware	139	140	2
8	Ransomware Jisut Variety 1 Malware	140	140	1
9	Trojan Banker Bankbot Variety 1 Malware	140	141	2
10	Backdoor Droidkungfu Variety 1 Malware	141	139	2

- [1] Wei, F., Li, Y., Roy, S., Ou, X., Zhou, W. (2017). Deep Ground Truth Analysis of Current Android Malware. In: Polychronakis, M., Meier, M. (eds) Detection of Intrusions and Malware, and Vulnerability Assessment. DIMVA 2017. Lecture Notes in Computer Science(), vol 10327. Springer, Cham.

Execution Time Analysis



Conclusions

1. **Proposal of GANG-MAM Tool**
   - GANG-MAM introduced as a tool for modifying existing Android malware with GAN-produced features vectors;
   - Addresses the growing interest in adversarial attacks and defense mechanisms in malware detection;
2. **Significance of GAN in Malware Modification**
   - GAN represents an effective method for generating strongly evasive malware;
   - GANG-MAM leverages this technique to modify existing malware, enhancing its evasiveness;
3. **Potential Applications**
   - Robustness Testing: Evaluating existing malware classifiers to identify blind spots or expand training datasets;
   - Dataset Creation: Providing datasets for testing new methods aimed at recognizing GAN-produced malware;
4. **Future Directions**
   - Continued research into adversarial attacks and defense mechanisms;
   - Exploration of additional applications and enhancements for GANG-MAM tool.

AI-Ethics

Ethical concerns about Generative AI…

…Ethical concerns about Generative AI…

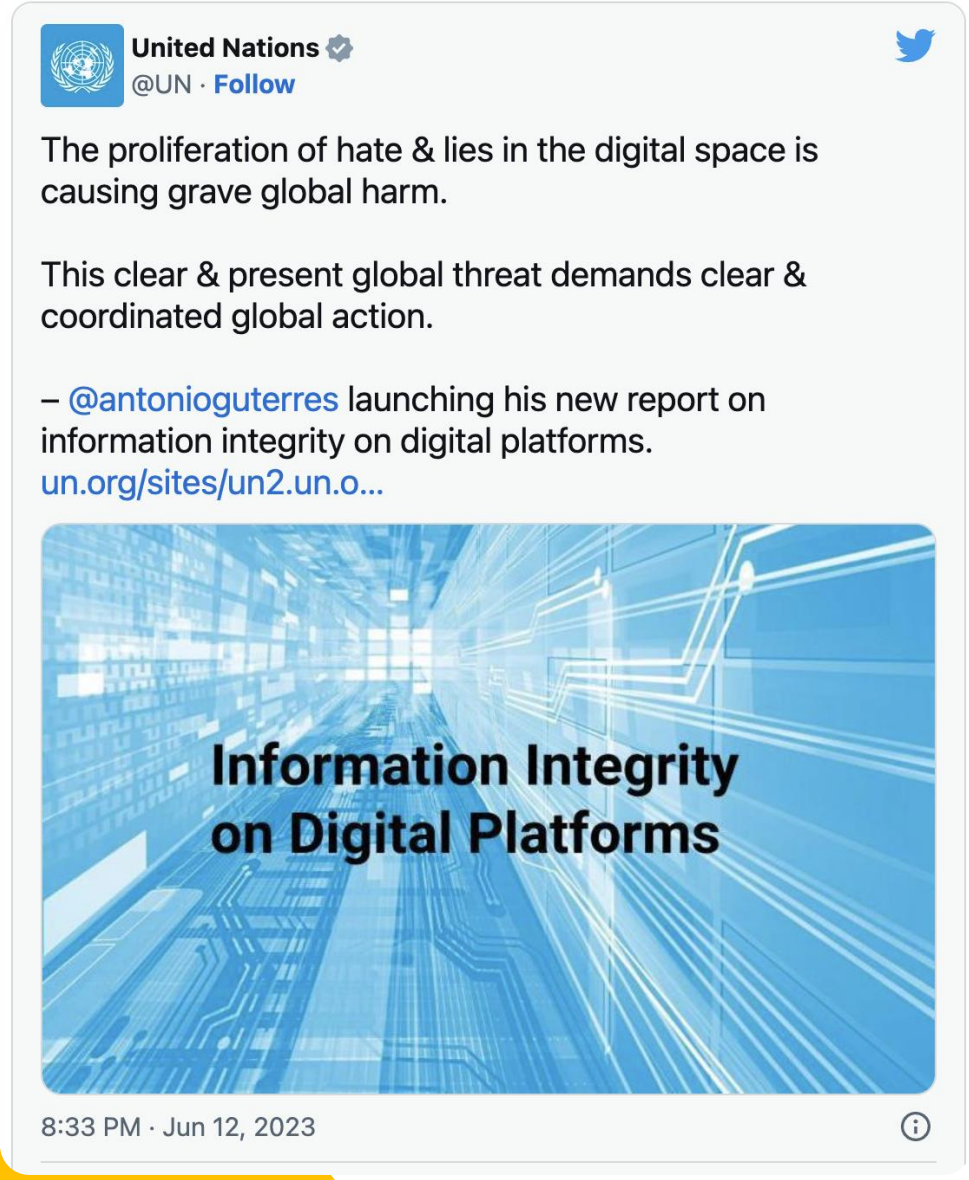
- Distribution of harmful content
  - Malicious fake
  - Biases learned from the training data
- Copyright and legal exposure
  - It can lead to legal, reputational, and financial risk for the company using pre-trained models and can negatively impact creators and copyright holders
- Data privacy violations
  - Breach of users' privacy can lead to identity theft and can be misused for discrimination or manipulation.

…Ethical concerns about Generative AI…

- Sensitive information disclosure
  - Risks increase significantly if an employee uploads sensitive details like a legal contract, source code of a software product, or proprietary information, among others.
- Amplification of existing bias
  - AI models are only as good (or bad) as the data they are trained on.
- Workforce roles and morale
  - “Current generative AI and other technologies have the potential to automate work activities that absorb 60 to 70 percent of employees’ time today.” (McKinsey Rreport)

…Ethical concerns about Generative AI

- Data Provenance
  - Such systems require maintaining data assurance and integrity to avoid using biased data or data of questionable origin.
- Lack of transparency
  - AI systems are black-box in nature, which implies that it is difficult to understand how they arrived at a particular response or what factors led to their decision-making.



An unprecedented example of such misinformation surfaced in the US court recently when a lawyer referred to an example of an inexistent legal case, relying on the ChatGPT-generated response.

Take actions…

- UNESCO released AI ethics guidelines which were adopted by all 193 Member States in November 2021.
- The guidelines emphasize four core values:
    - Human rights and dignity
    - Peaceful and just societies
    - Diversity and inclusiveness
    - Environmental flourishing
- They also outline **ten principles**, including proportionality, safety, privacy, multi-stakeholder governance, responsibility, transparency, human oversight, sustainability, awareness, and fairness.
- The guidelines also provide **policy action areas** for data governance, environment, gender, education, research, health, and social wellbeing.

… Take actions

- There are various other groups, such as **AI Ethics Lab** and **Montreal AI Ethics Institute**, that address ethical literacy. The common ethical principles include transparency, accountability, data privacy, and robustness that focus on the technology providers.
- It is incumbent on us, as users, to dismiss the idea that everything available in the digital world can be trusted at face value: **cultivate awareness and learn**.

Thank you

"I visualize a time when we will be to robots what dogs are to humans, and I'm rooting for the machines.."

Claude Shannon