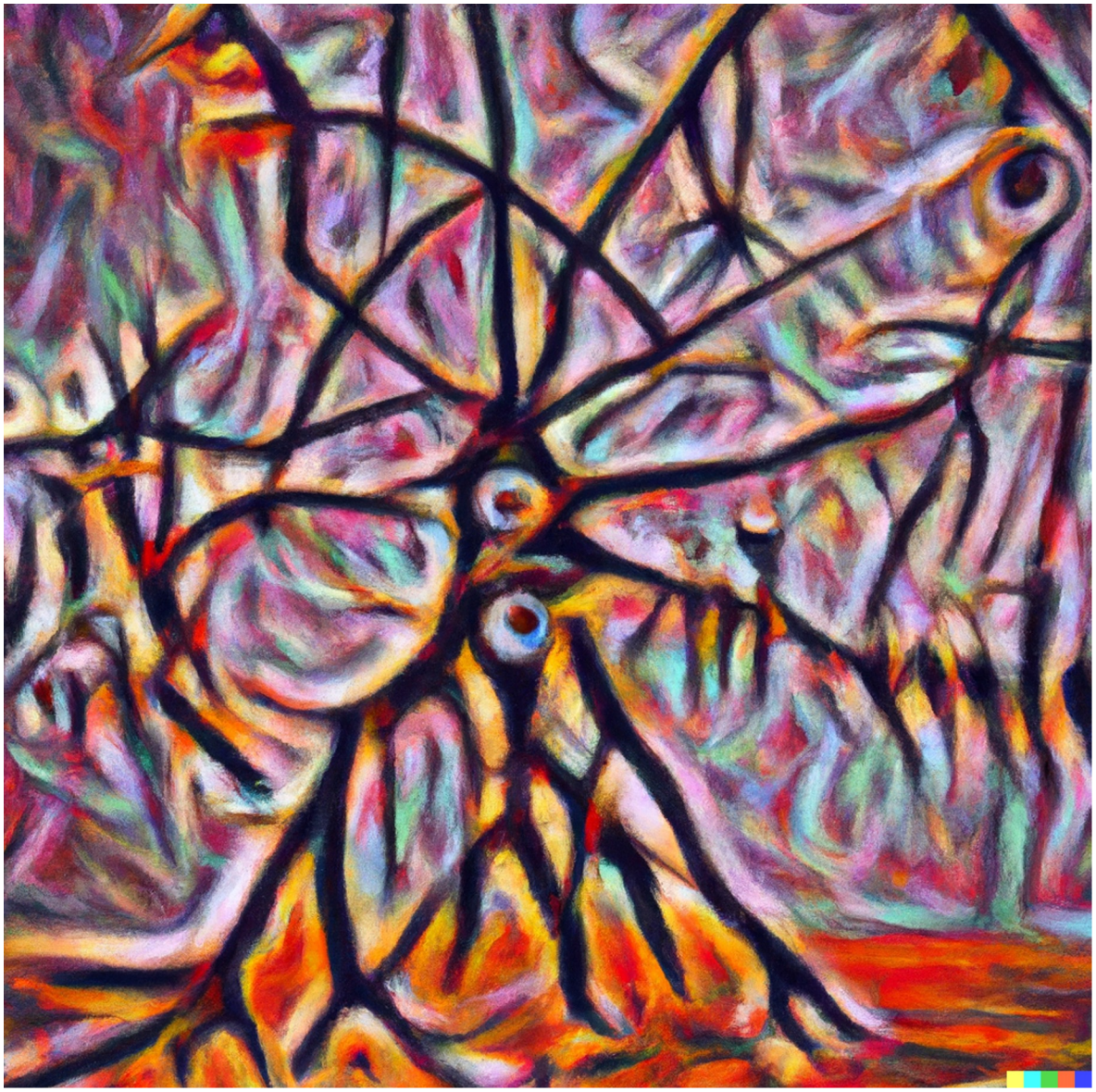


Maleficent Neural Networks

Plan for today

- Maleficent neural networks?
- Some basics that are good to know!
- Methods to manipulate neural networks
- How to evaluate manipulated neural networks
- What are possible counter measurements?

Evil neural networks?

- How can something that can be used for cat classification be evil?
- They are not evil by design
- But like a lot of things they can either be
  - Misused
  - Manipulated (e.g., to embed malicious payloads without triggering anti-malware software)
- We will look mainly into the latter one
- You might wonder how?
  - Actually not through a new concept but something quite old and well known
  - Steganography….

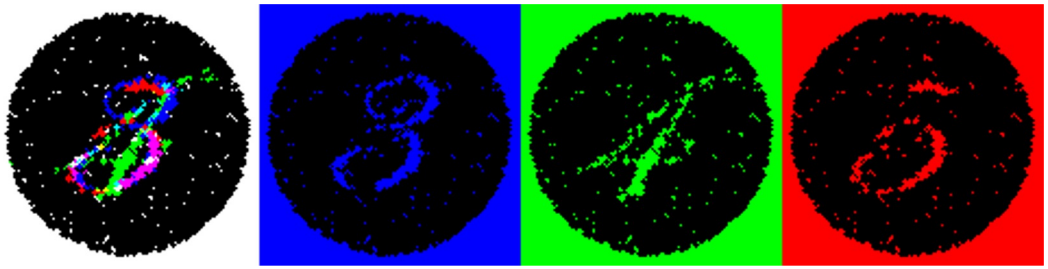
Stengano what????

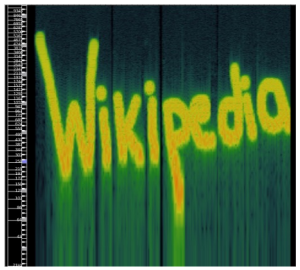
Evil models are a novel problem but actually…

- Basic idea is something that already exists since thousands of years
- Called steganography
  - Hiding something in plain sight
  - The fact that communication is taking place is hidden
- Compared to cryptography
  - Its is clear that there is a message but its manipulated in a way that you can only read it if you know how (need a key)

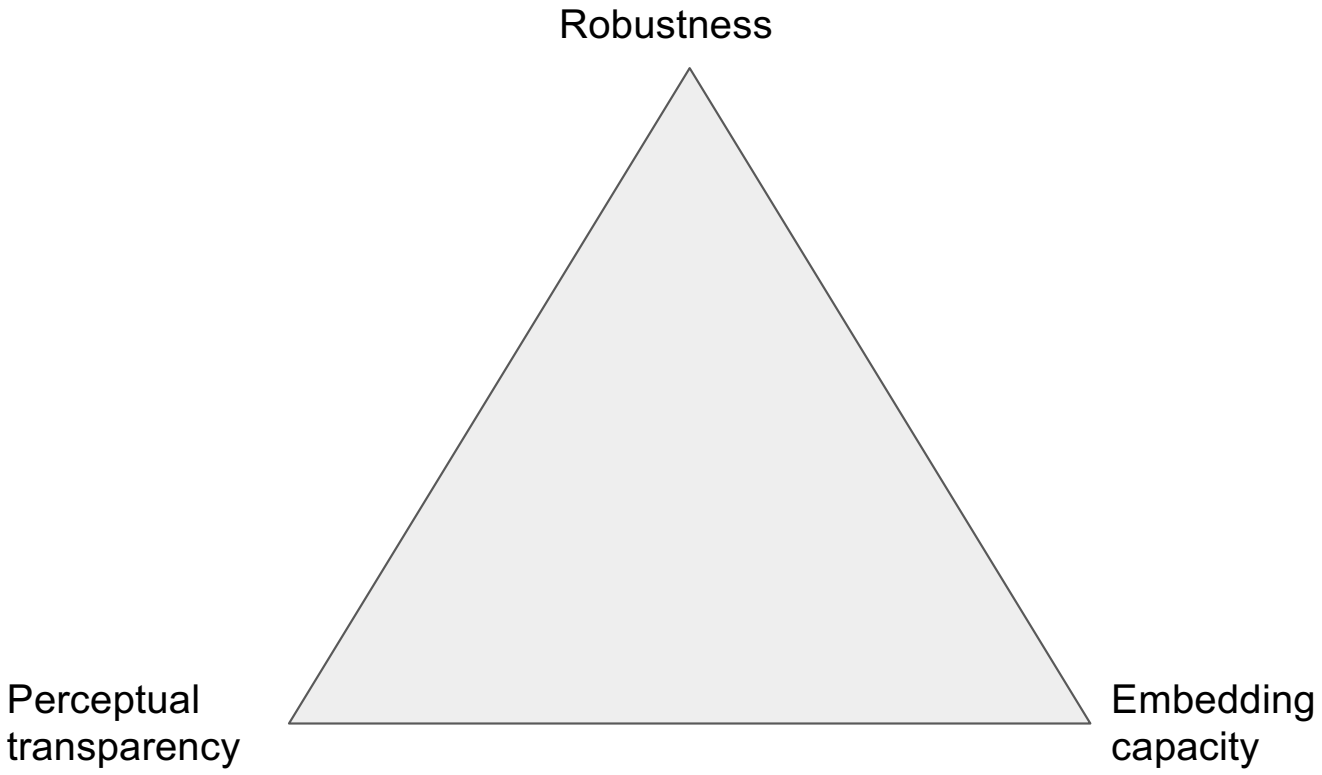
Messages can be (and are) hidden in a lot of different sources

- Images
- Videos
- Sound
- Files…





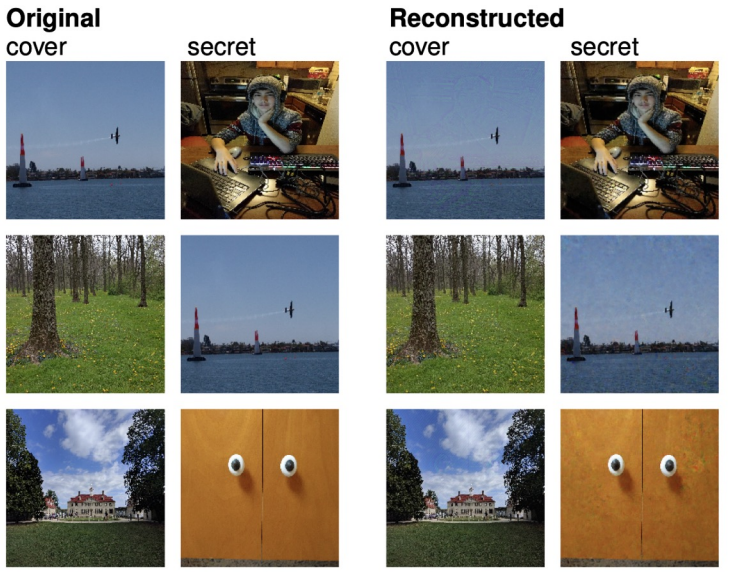
The objectives triangle of steganography



How is it done nowadays?

Deep Stenography

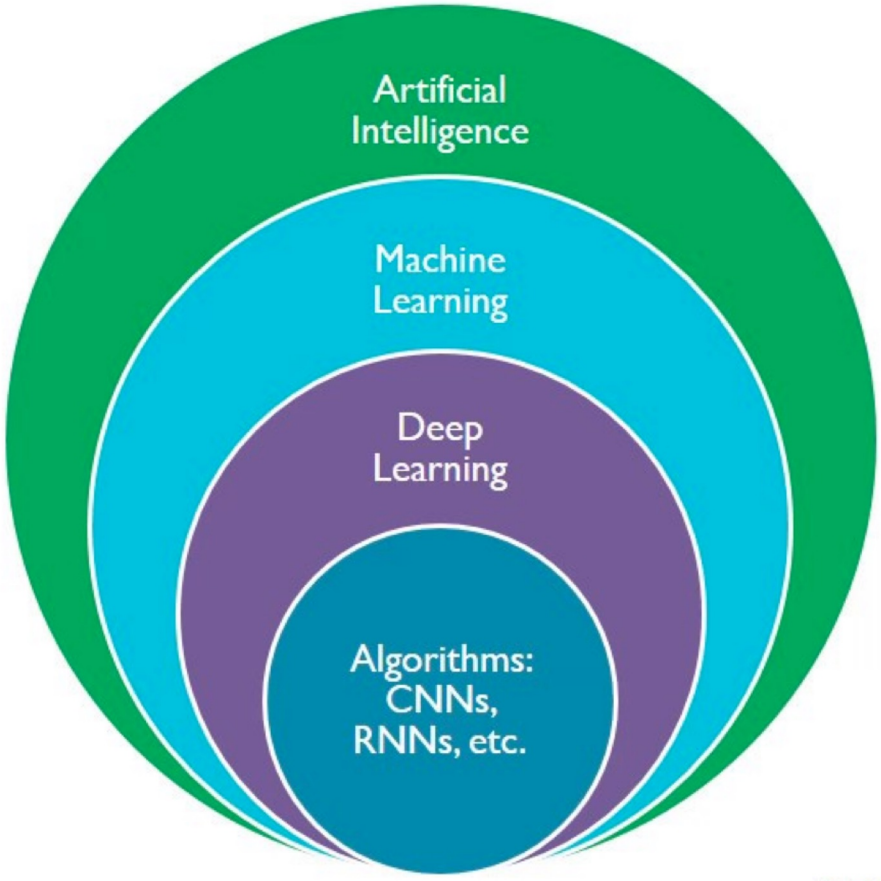
- Different automatic methods (mostly based on DL)
- Basically you learn to do it
- More sophisticated and harder to detect than manual methods
- Often used on
  - Images
  - Audio



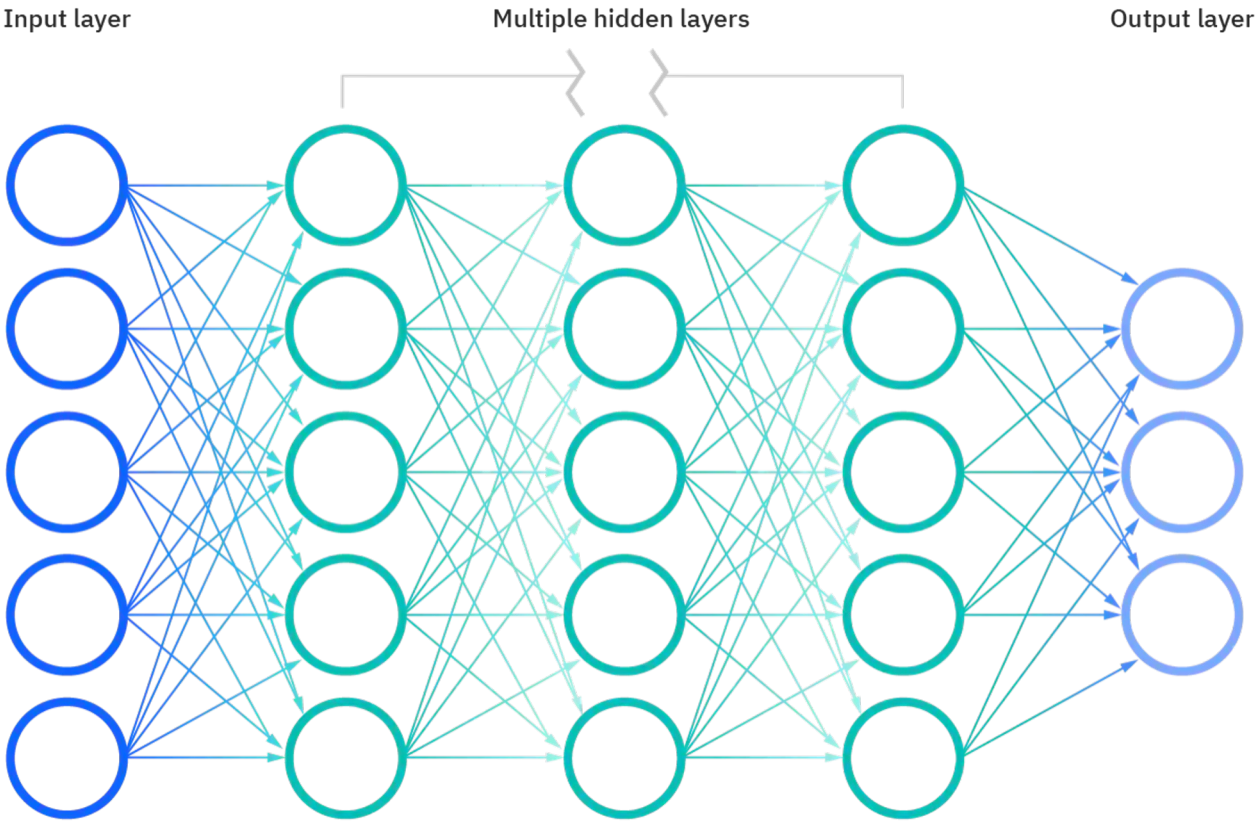
Steganography and malware

- One does not just simply hide messages…
- Steganography can also be used to hide malicious code
  - Mostly used with images
  - Attacks using this technique has increased dramatically in the past few years
- Difficult to detect
  - Small changes that are made are hard to detect
    - Slight color differences between two images
    - Large amount of duplicate colors within an image may be an indicator
    - If the suspicious image is larger than the original image, then the size difference may be due to hidden information

Neural what???


Artificial Intelligence
Machine Learning
Deep Learning
Algorithms: CNNs, RNNs, etc.

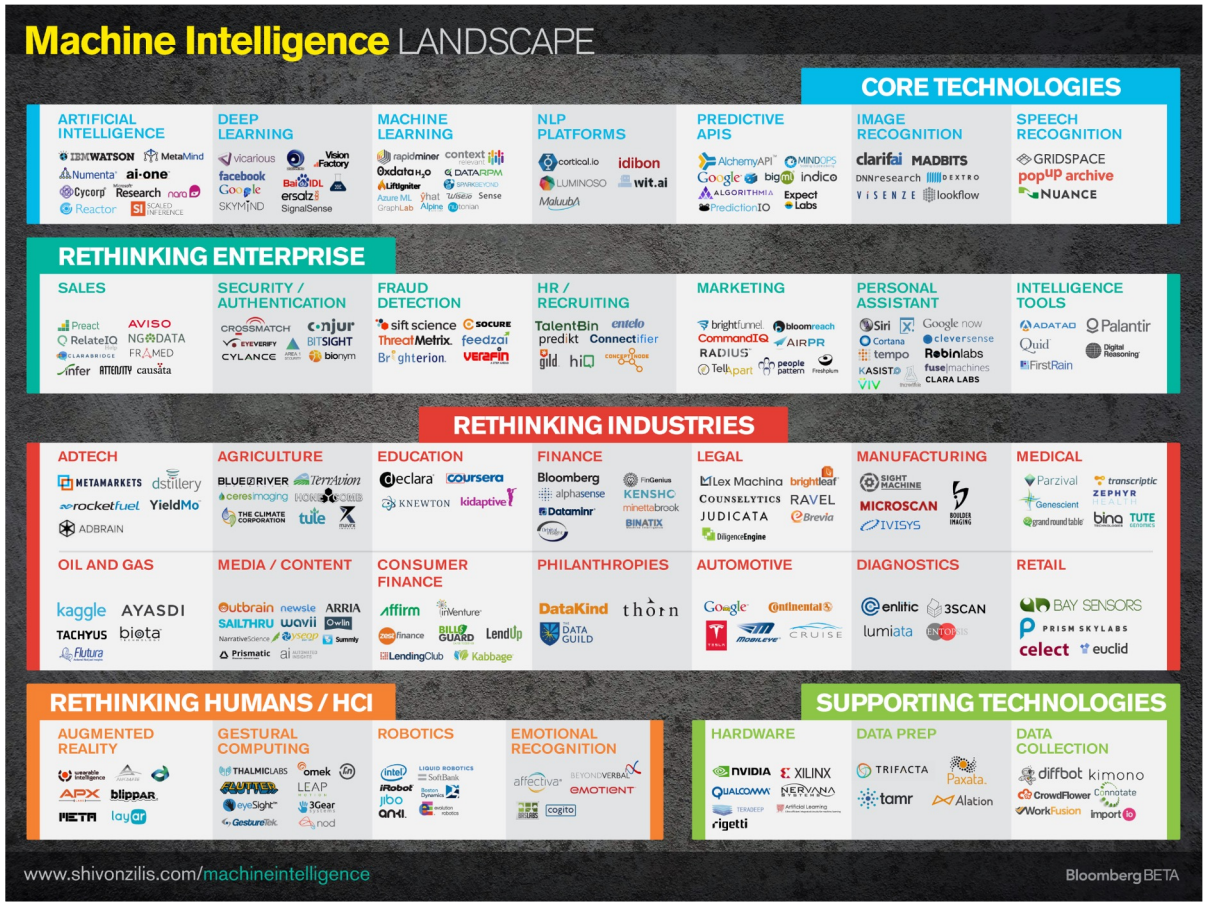
A simple neural network



Steganography and neural networks

- As mentioned, steganography can be applied to a lot of different sources
  - Images, audio, text, etc.
- Which means you can also apply it to neural networks
  - The methods and concepts to do so are not necessarily new
  - But opens up for a lot of problems…
- Two different ways
  - Stenography used to create manipulated content that messes up neural networks (adversarial attacks)
  - Steganography applied to neural networks internally
    - To for example hide malware in neural network models

DNNs are used everywhere



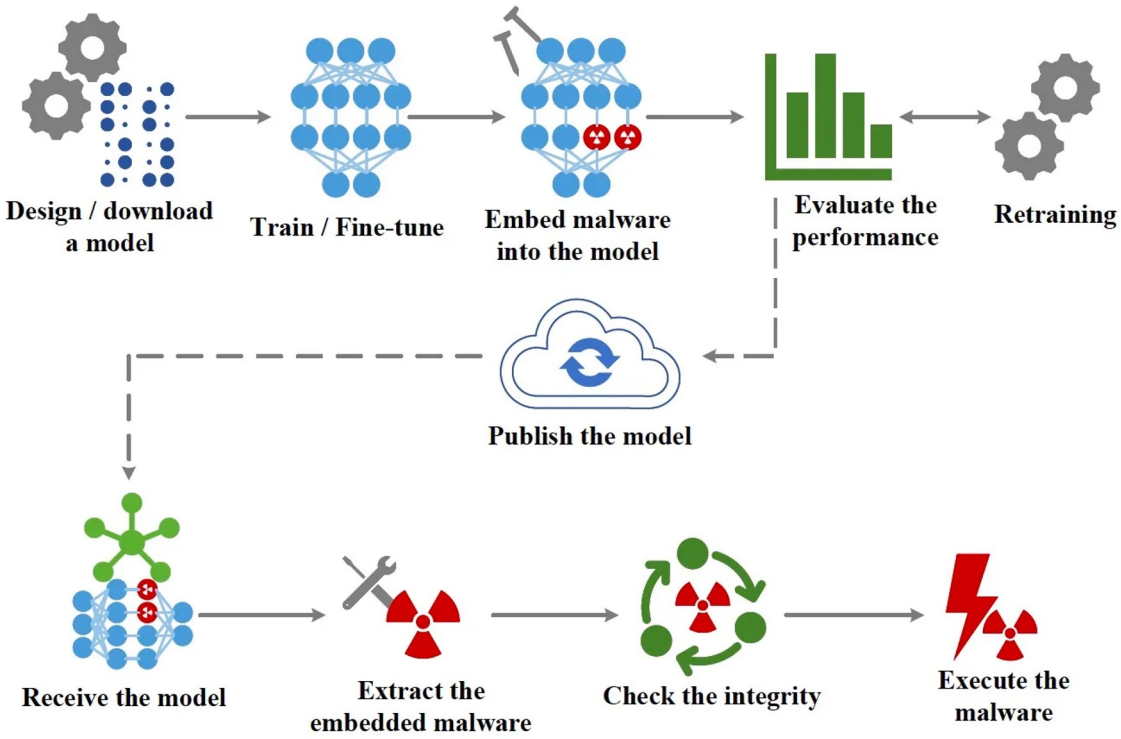
Manipulate the network itself, aka, create an Evil Model

“Evil” models general

- Neural networks can embed malicious payloads without triggering anti-malware software
- Problem since DNNs become integrated in applications we use every day
- We need to think about new ways to protect users against possible emerging threats from this
- Evil model is a malware hiding technique that can help to understand better the needs and concerns coming with this new threat in addition to raising awareness about the problem

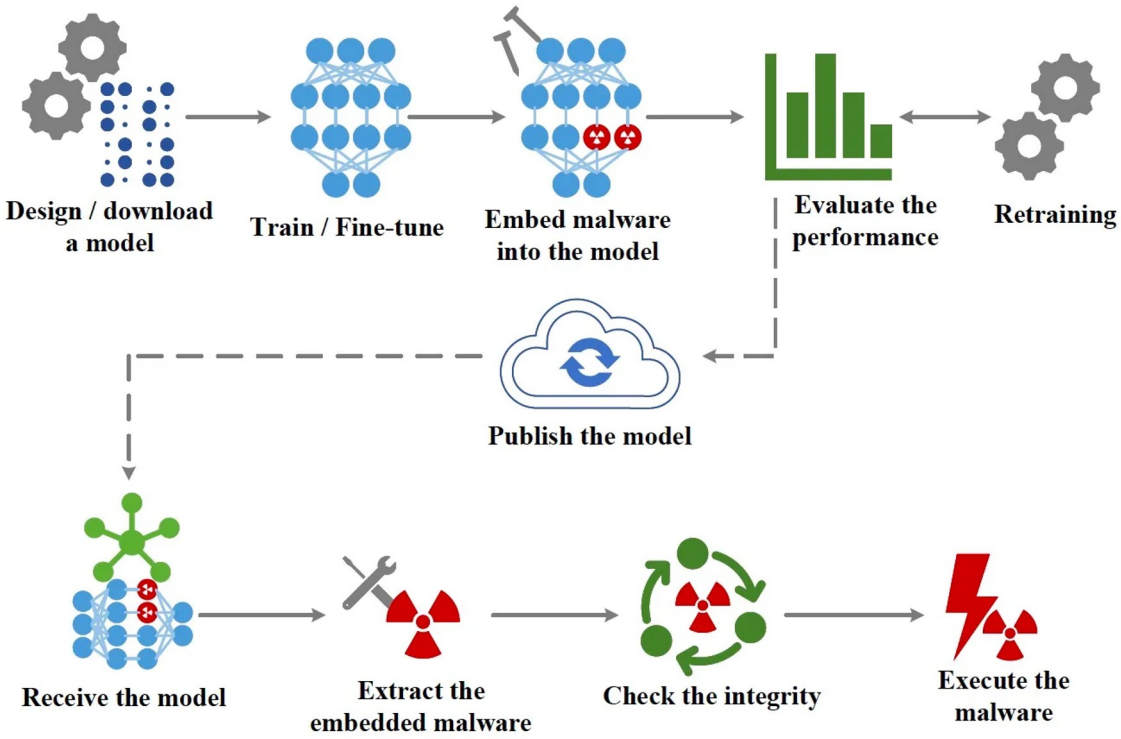
The “Evil” model - basics (i)

- Every DNN is composed of multiple layers of neurons
- The layers and neurons are connected and the strength of the connection is defined by numerical parameters
- These are learned during training
- Large DNNs can contain billions of parameters



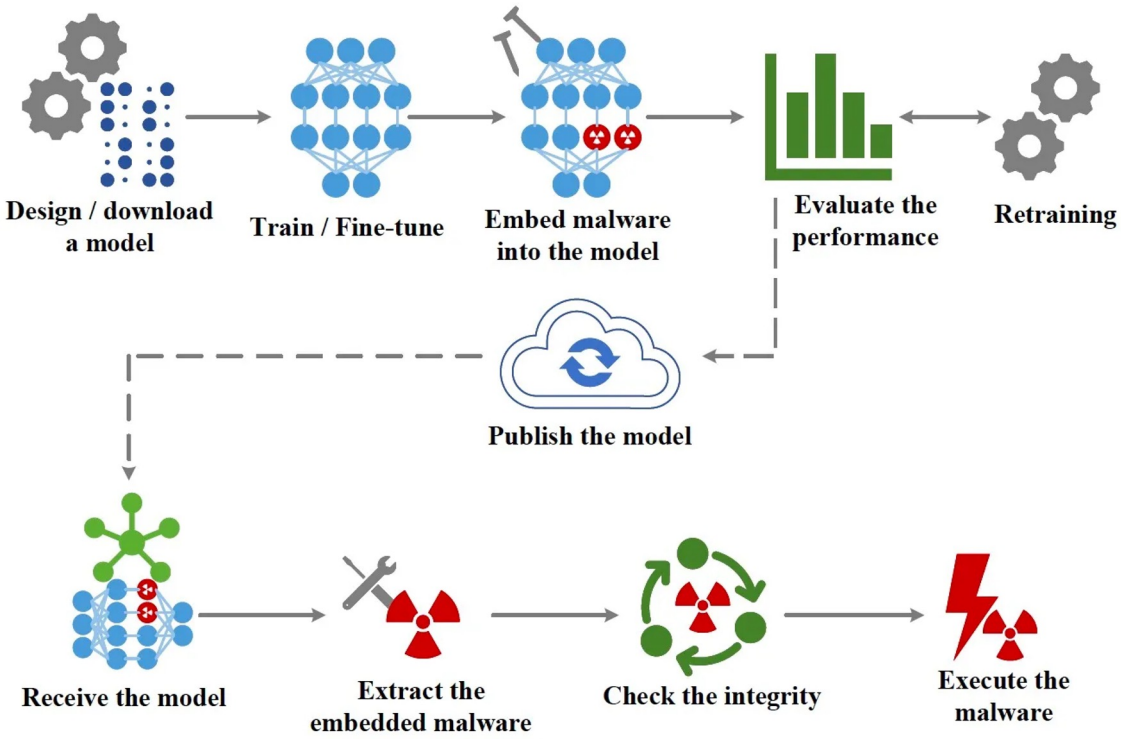
The “Evil” model - basics (ii)

- The main idea of the evil model is to embed malware into the parameters
- In plain sight but invisible
  - A form of steganography
- The infected neural net should at the same time also perform its tasks
- Users of the model should not get suspicious



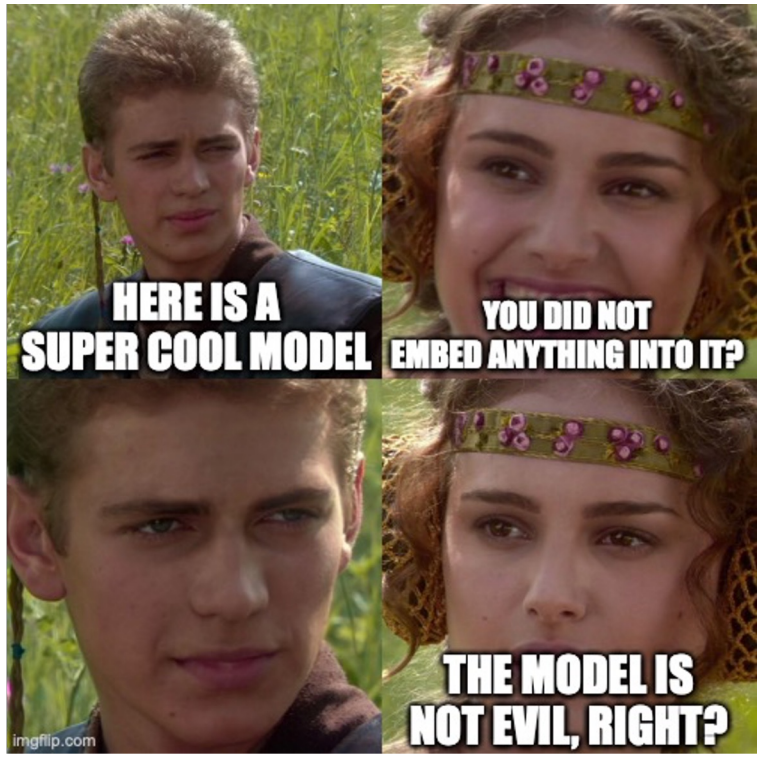
The “Evil” model - basics (iii)

- What do you need?
- A method to manipulate the model
- A mechanism to deliver the infected model
- A way to extract and trigger the malware from the parameters is needed



Embedding methods

- MSB Reservation (change bytes)
  - Attackers can keep the first bytes unchanged and embed the malware in the later bytes
- Fast Substitution and half substitution (similarity)
  - Find blocks of bytes that look similar to your malware and use them with or without slight modifications
- Drop out (neurons)
  - Use complete neurons in the net and replace them with your malware



“Evil” model - detection

- Evil models are very difficult to detect
- Payload are hidden in millions and billions of parameters
- Often just very small changes
- Pretrained models, even if trained the same way from different sources are different (due to the learning mechanisms of DNNs the weights of different DNN models based on same data and training method will be different)
  - Thus simple comparing might be not efficient

“Evil” models counter measurements (i)

- Malware scanners cannot (yet) detect malicious payloads embedded in deep learning models
- There are different possible counter measurements on different levels

“Evil” models counter measurements (ii)

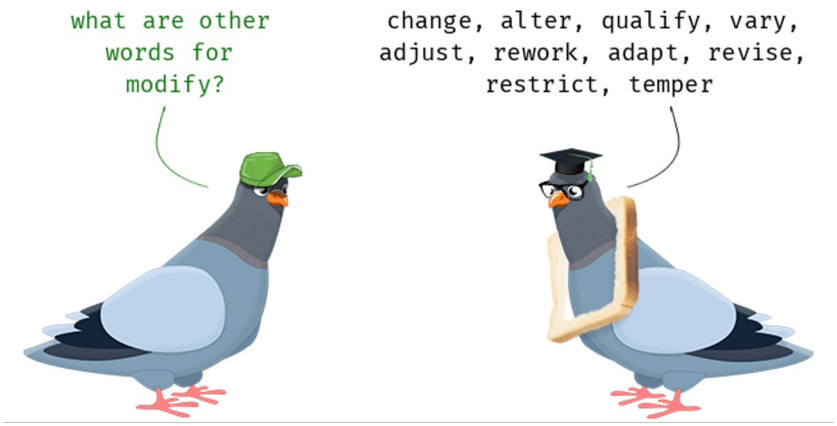
Adjusting the parameter size

- Malware can be embedded in the model because the parameters are sufficiently long
- There is no need to use such higher-precision numbers as parameters
- Only two bytes are often sufficient
- Deep learning frameworks should consider changing the default data type of the parameters and lowering the precision of the numbers
- An attack would not be eliminated, but it becomes more challenging for the attackers to use the model (quicker performance drop when manipulated)

“Evil” models counter measurements (iii)

Modifying the neural network model

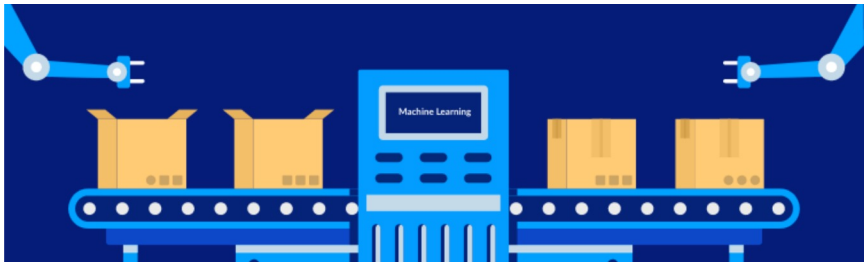
- A malware-embedded model cannot be modified
- Once malware is embedded into the neural network model, the malware bytes’ parameters cannot be changed to maintain the malware integrity
- Simply retraining, pruning or model compression would destroy the malware



“Evil” models counter measurements (iv)

Protecting the neural network model supply chain

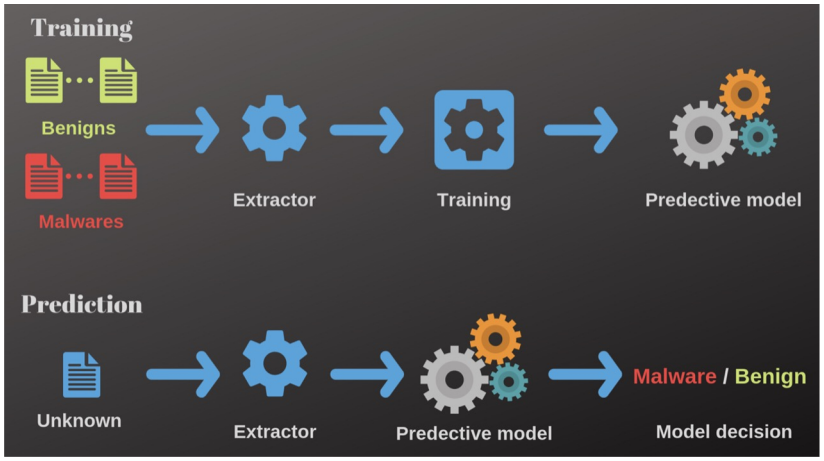
- Neural network model markets can play an essential role in propagating the EvilModel
- Mitigating the EvilModel attack from the perspective of supply chain as protection
- Model markets should improve user identity verification and allow only verified users to upload models
- A certificate mechanism for the neural network model
- Others can easily verify the model through the attached certificate



“Evil” models counter measurements (v)

Detecting malware in the neural network model

- Train a model to detect embedded malware in neural networks
- Experimental results indicate that malware embedded in a neural network model can be detected
  - Needs data



Future perspectives?

- So far mainly tested with CNNs
- Not clear how realistic an attack with such a model really is
  - Practical triggers and external software are needed, etc.
- At the moment easy to counter with retraining
- Still, more efficient detection and counter methods are needed
- Clearly a new threat that need to be taken into account!

Sources

- https://en.wikipedia.org/wiki/Steganography#/media/File:Steganography.png
- https://research.google.com/pubs/pub46526.html
- https://www.tattoolife.com/wp-content/uploads/2021/11/Detail-of-an-illustration-by-Giorgio-De-Gaspari.jpeg
- https://labs.inquest.net
- https://arxiv.org/abs/2107.08590
- https://zw.ac.cn/publication/EvilModel2_CS_2022