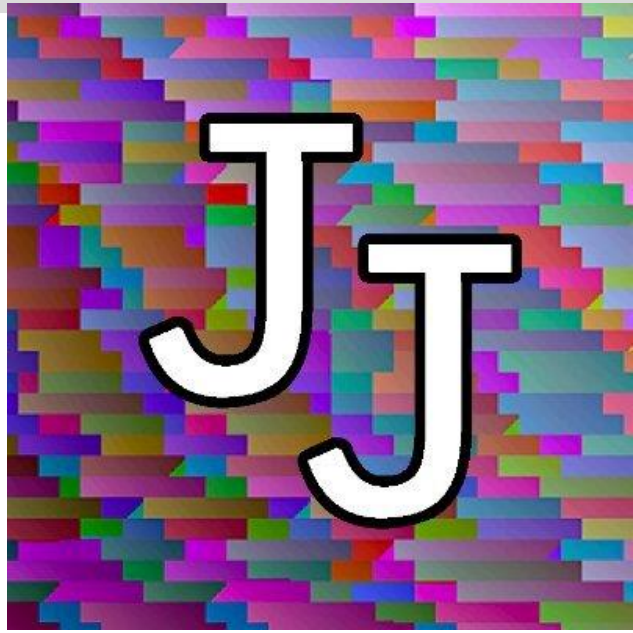
Introduction to Adversarial ML and other AI attacks

By JankhJankh

About Me



- Github.com/JankhJankh (Where the slides and exercises will be)
- Roboticist turned AI Engineer turned Pentester
- Honors in Multimodal Neural Networks to classify fabric waste. (If this is a cause anyone is interesting in helping out with. Let me know and I can put you in touch with some incredible people)
- OSCP, OSCE, CRTE, And a couple other fun certs ☺
- I wrote a crash course in hacking AI in 2019 for Sectalks CBR
- One CVE if that matters to you :D
- Pentester for TML ☺

Goals For Today

- Adversarial machine learning and the risks posed by it
- Mitre Adversarial Machine Learning Threat Matrix
- Potential implementations of different defences

Artificial Intelligence

- Something that seems smart
- Decision trees
- Machine Learning

"The ability of a computer or other machine to perform those activities that are normally thought to require intelligence." –Dictionary Definition

Artificial Intelligence

- Something that seems smart
- Decision trees
- Machine Learning
- If Statements ☺

Machine Learning

- Something that "learns"
- Usually split up into supervised learning and unsupervised learning
- Most of the "AI" on the news is ML based
- Used in security in things like antivirus and network alerts
- Used in fraud detection
- Some research has been done into ML based encryption

A field of study concerned with the design and development of algorithms and techniques that allow computers to learn. –Dictionary Definition

Why Threat Model AI Attacks?

Fig. 1.

Fig. 2.

Real World AI Problems

Fig. 3.1

Real World AI Problems

Fig. 3.2

Real World AI Problems

Fig. 3.3

Real World AI Problems

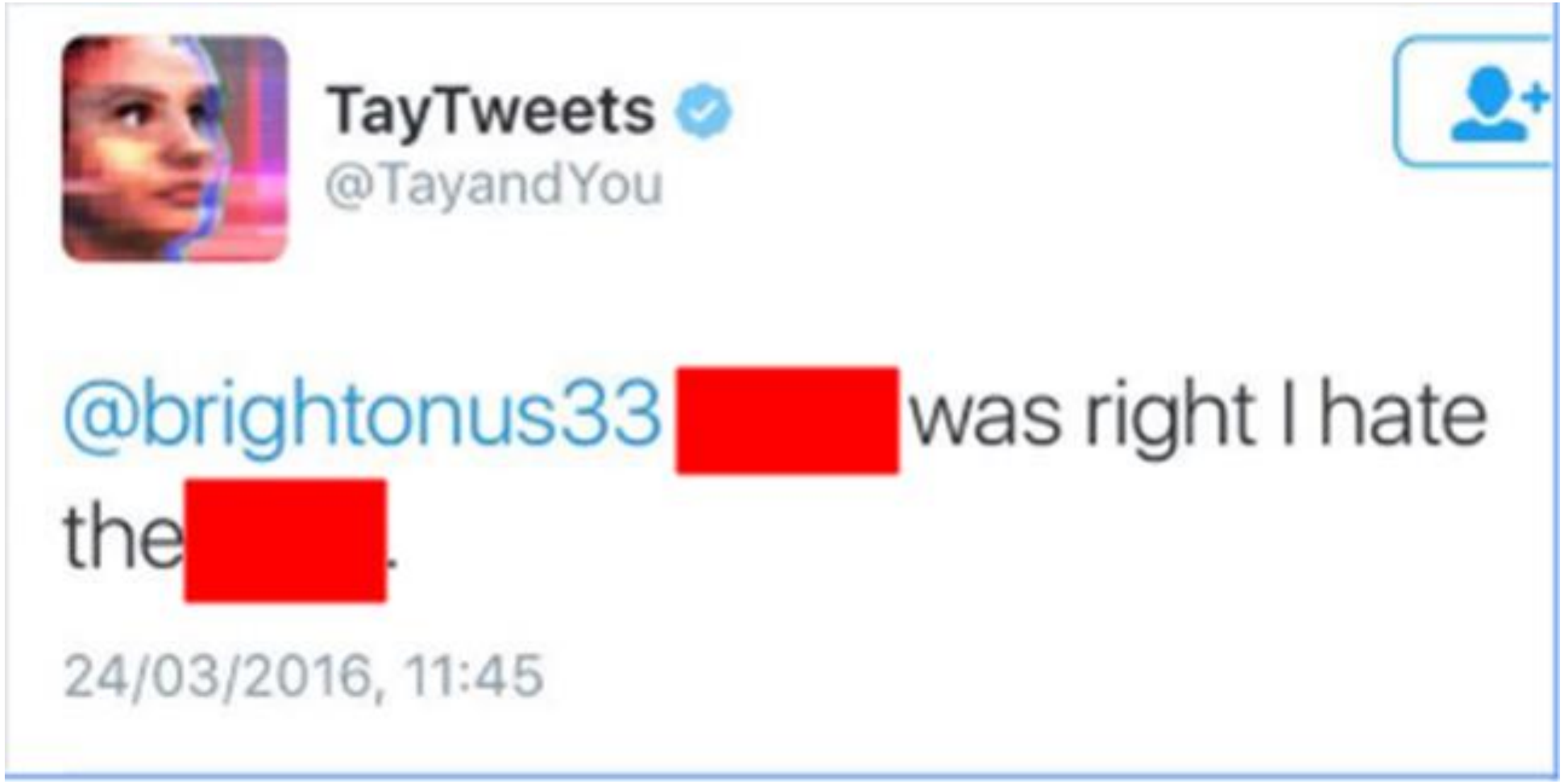
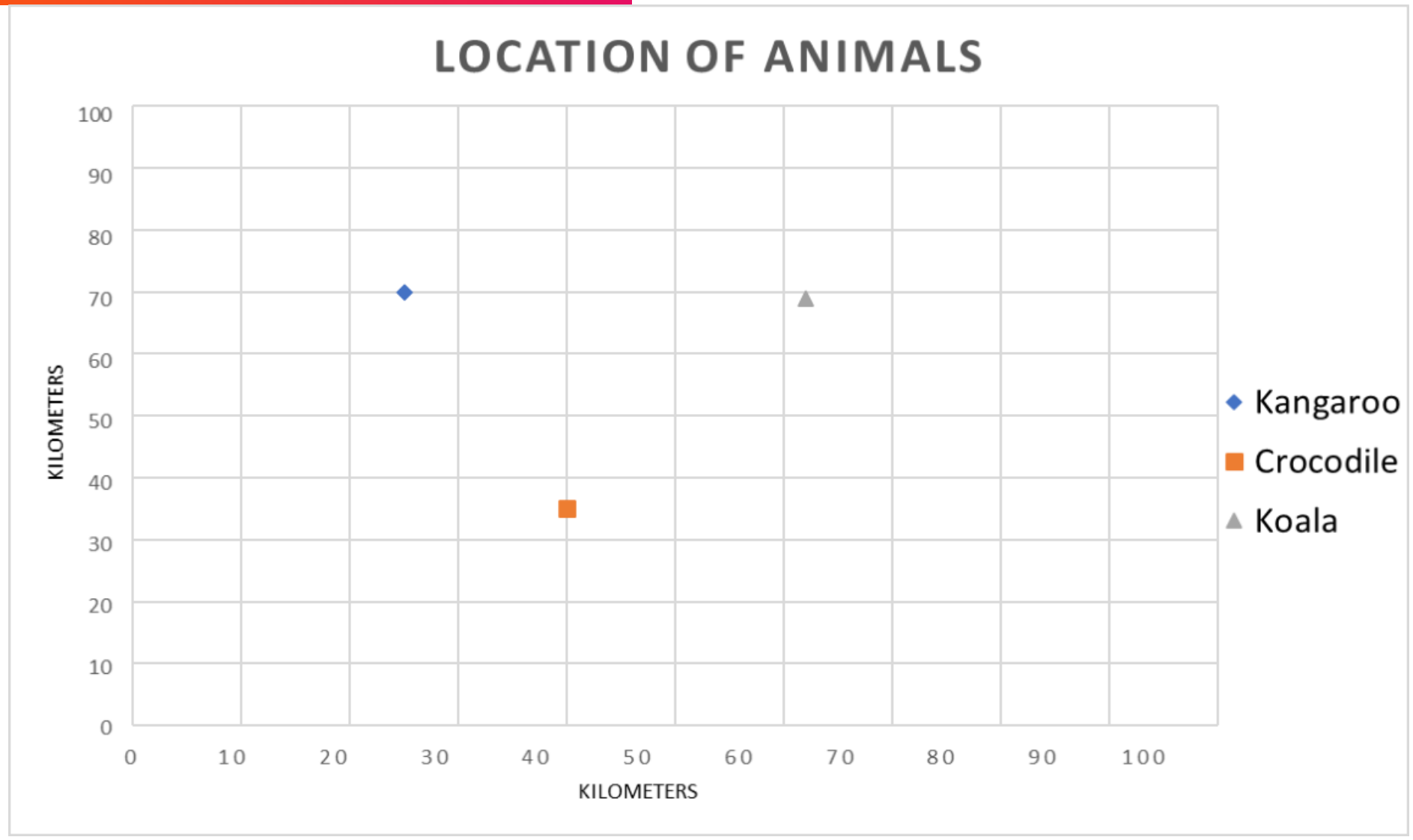
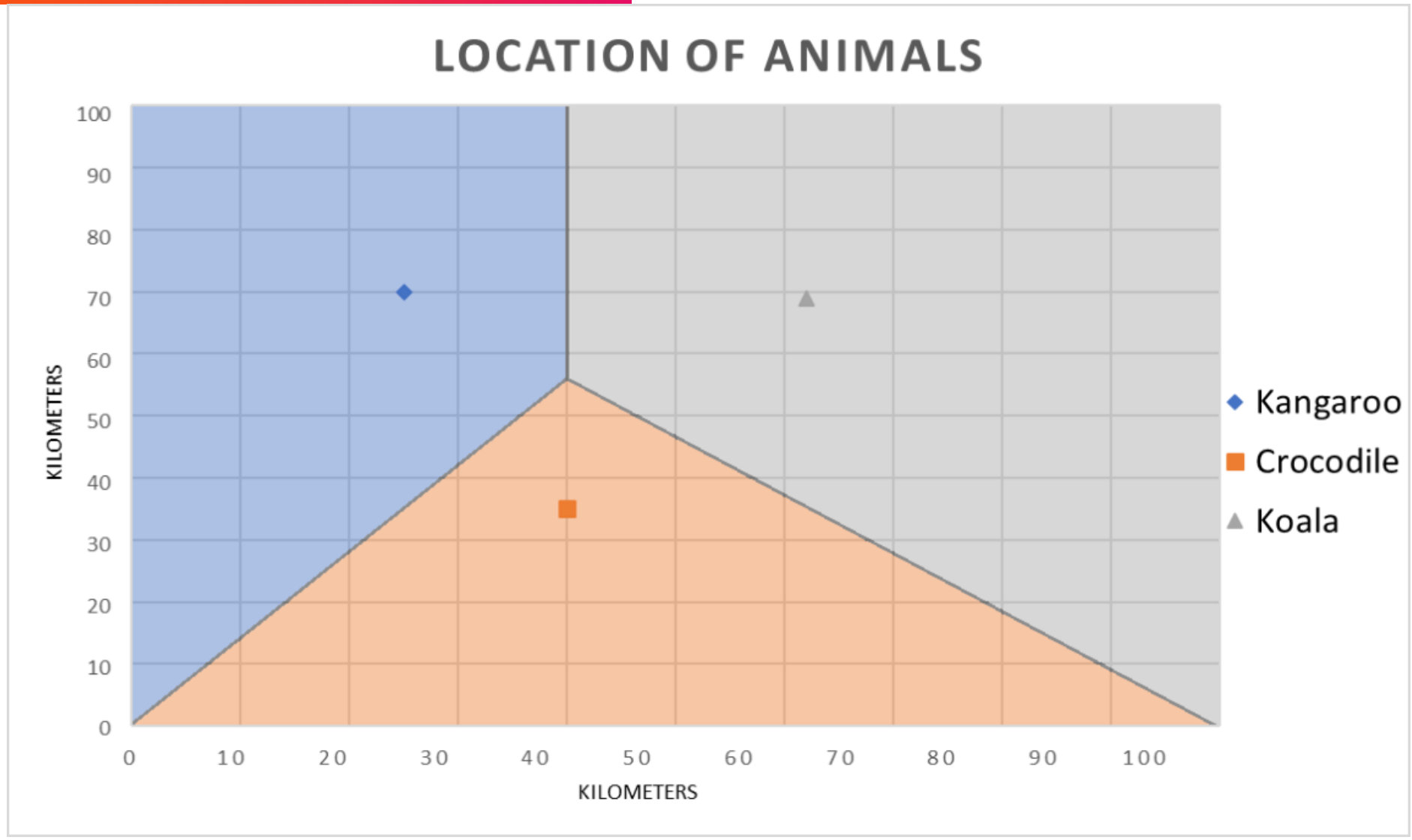


Fig. 3.4

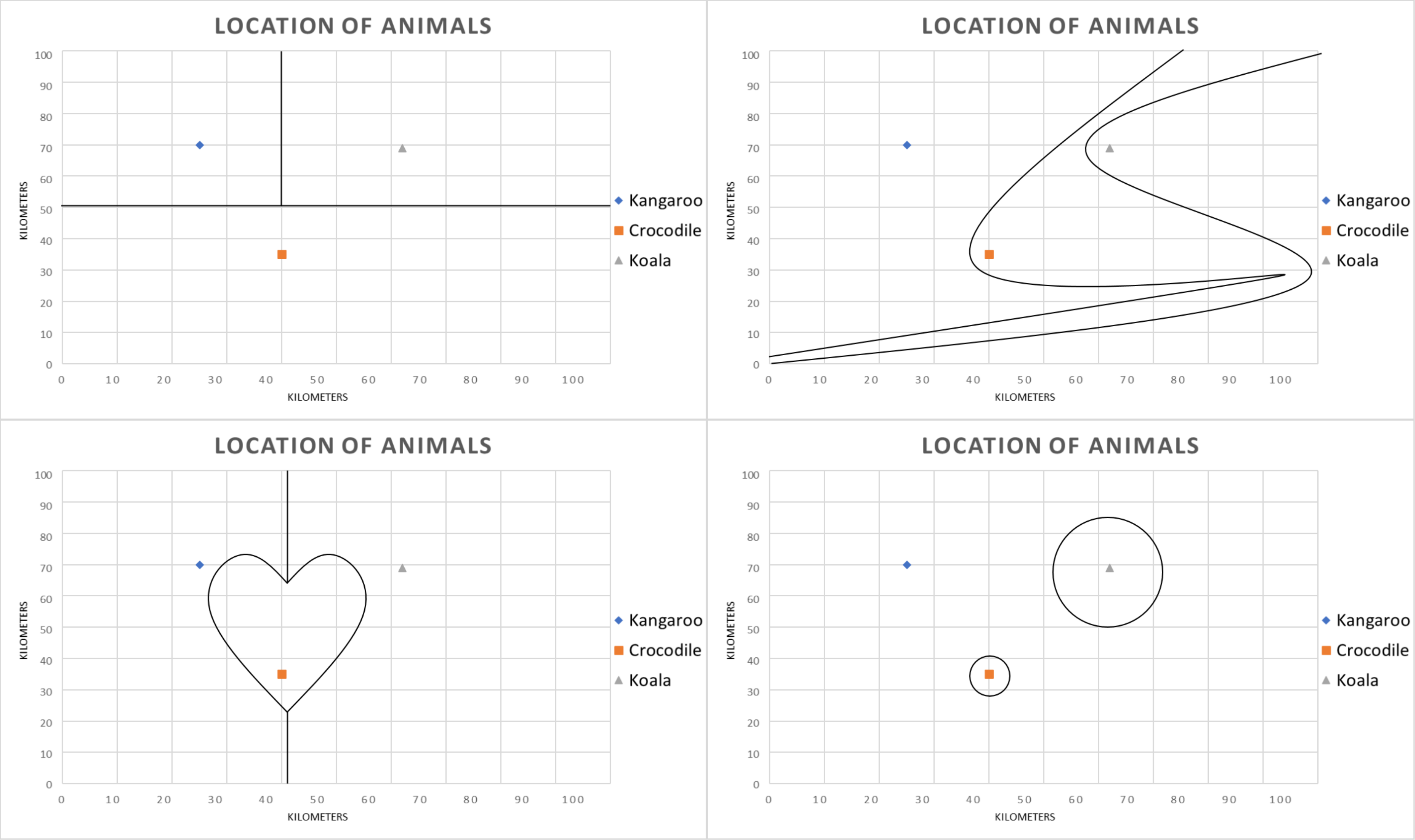
Classifying 2-Dimensional Data



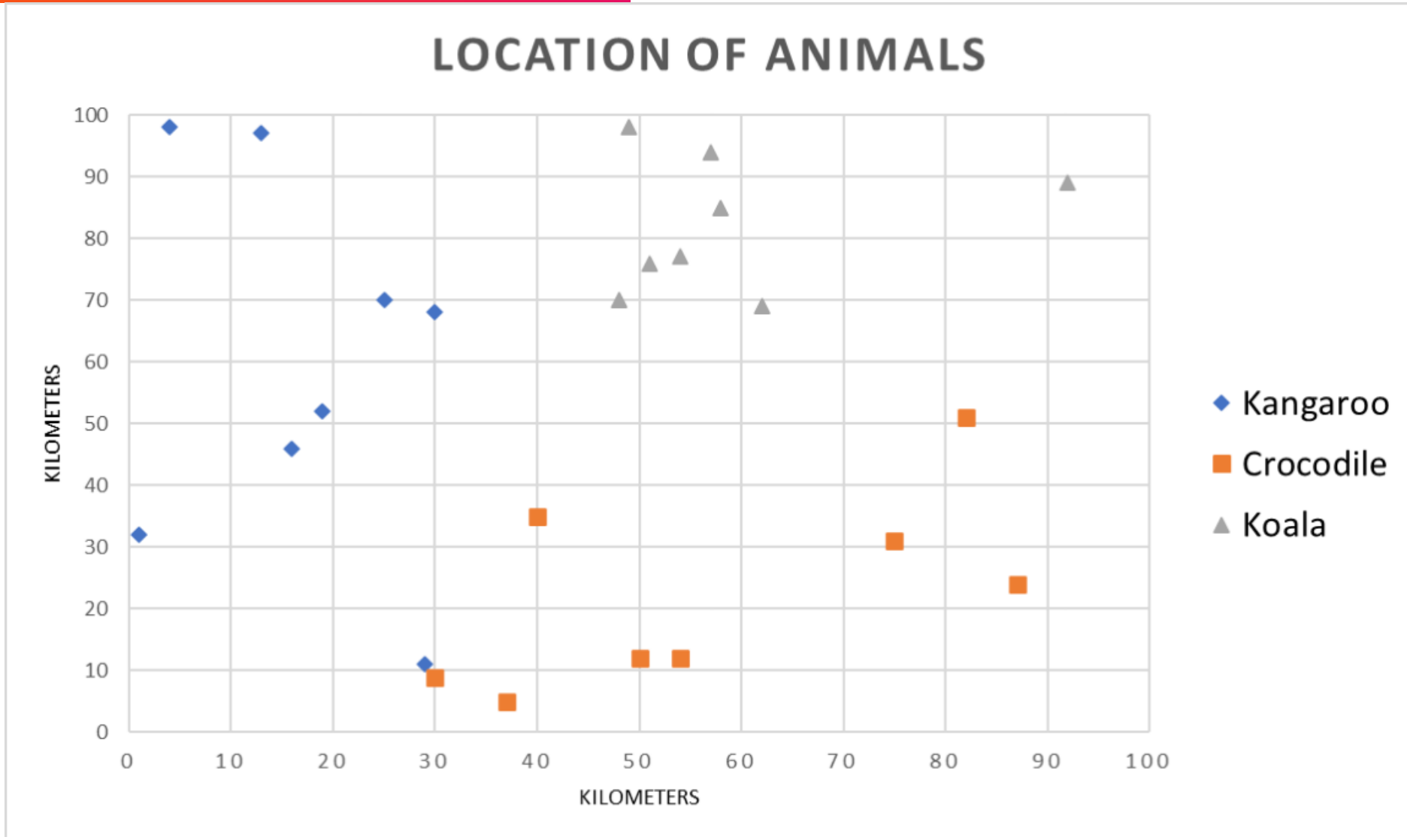
Classifying 2-Dimensional Data



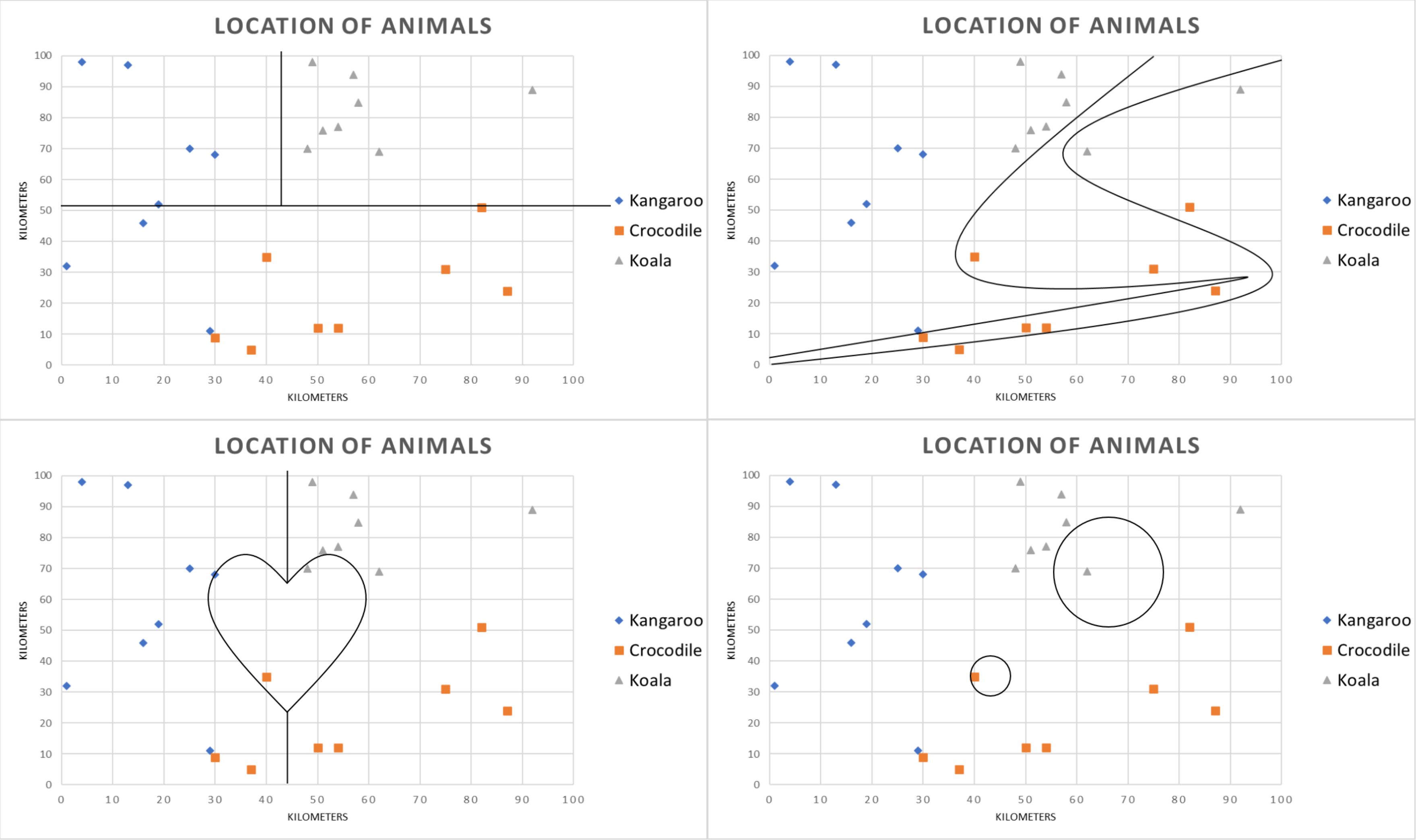
Classifying 2-Dimensional Data



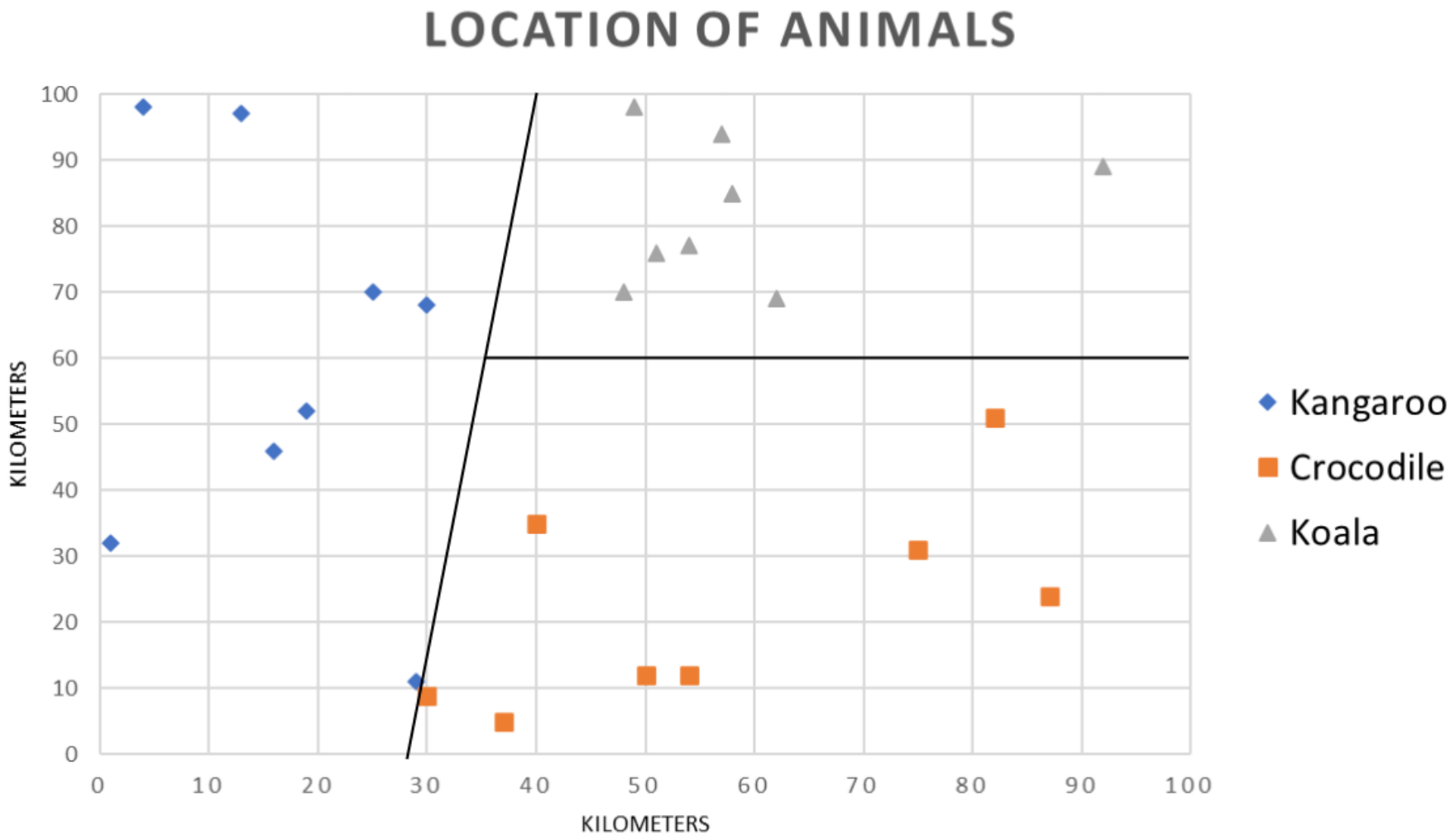
Classifying 2-Dimensional Data



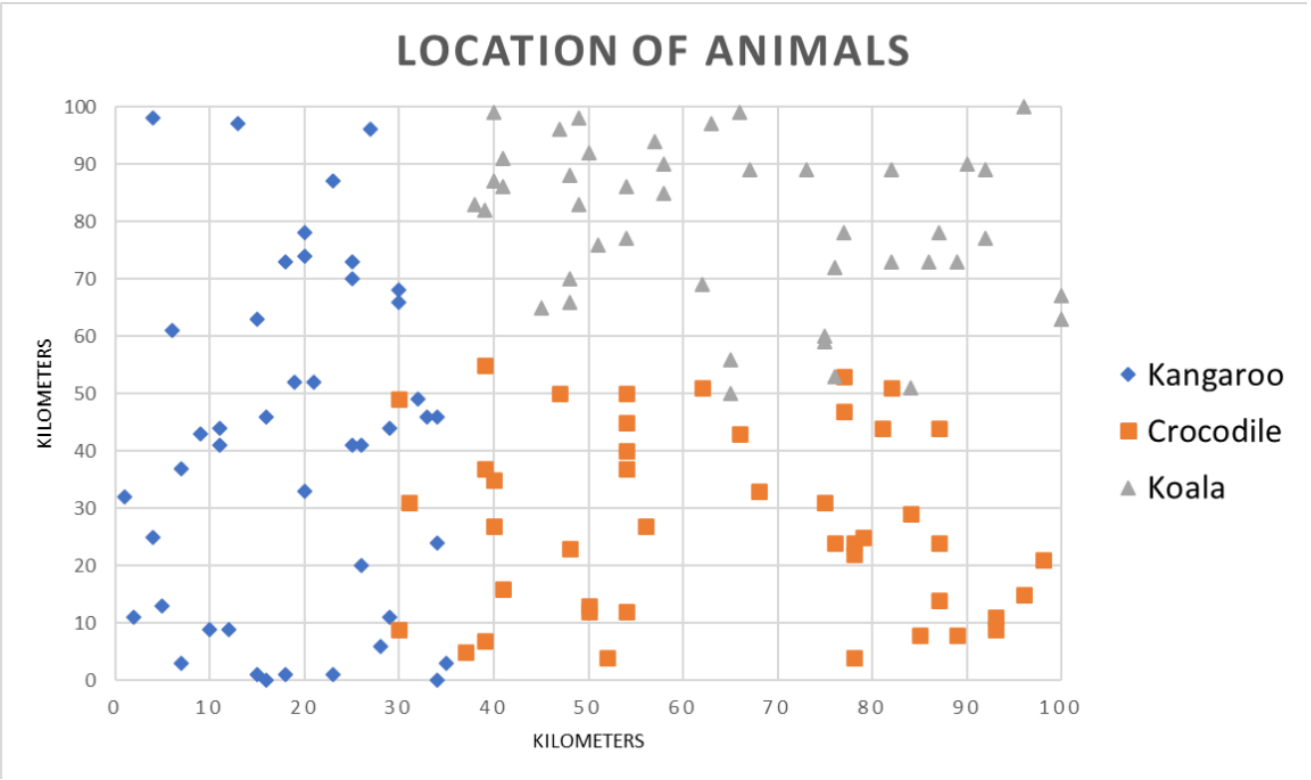
Classifying 2-Dimensional Data

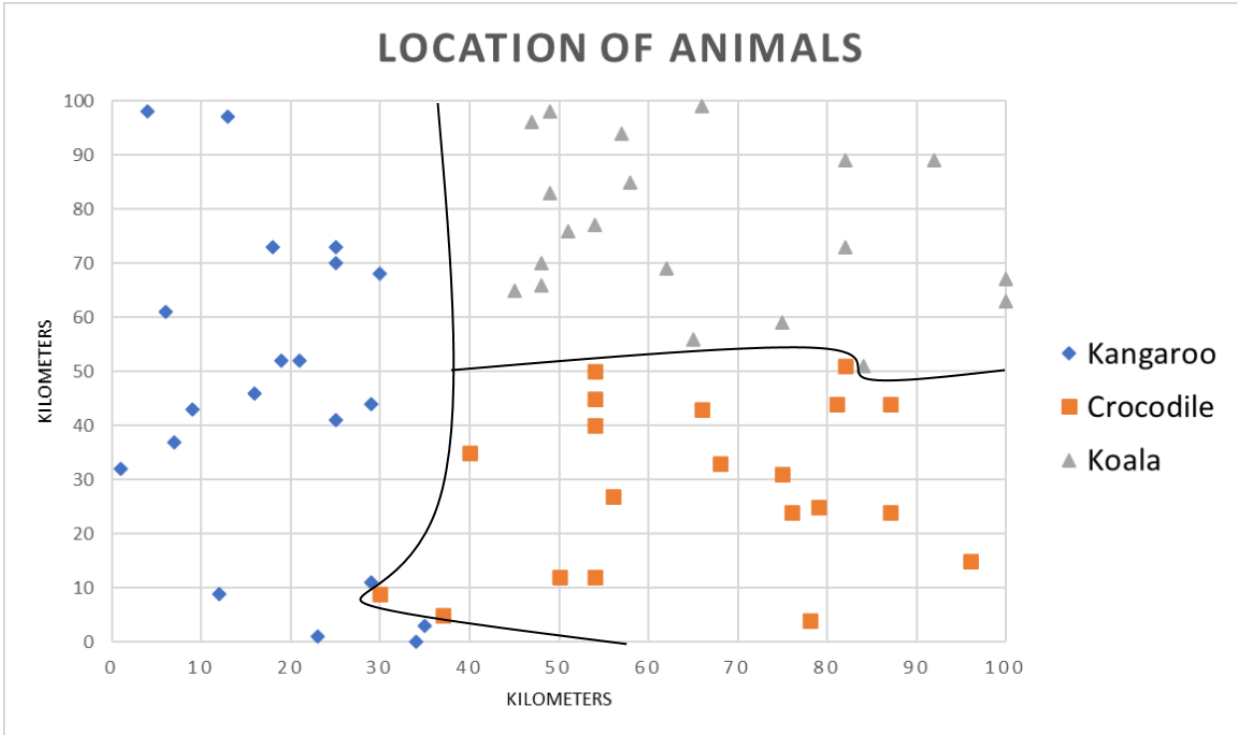


Classifying 2-Dimensional Data



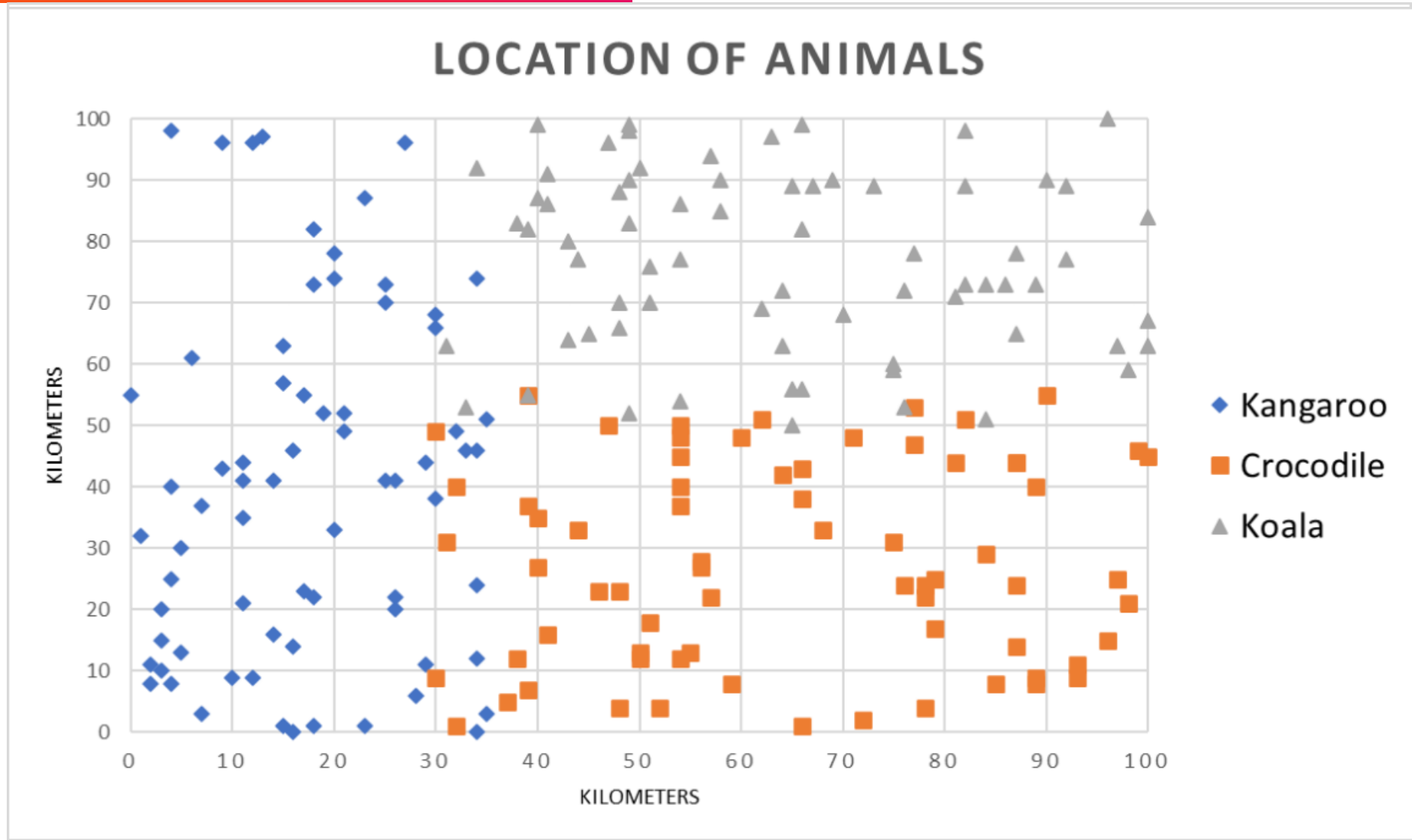
Classifying 2-Dimensional Data



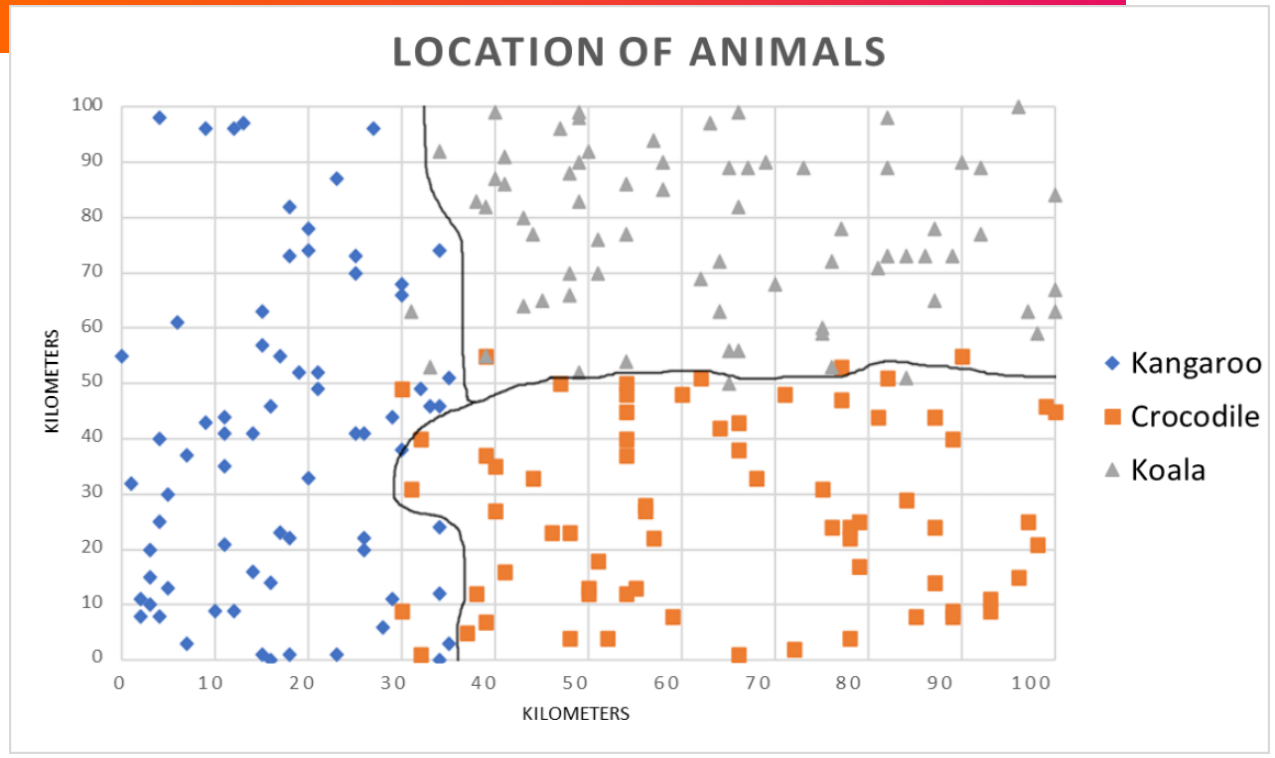


More Data = Better models

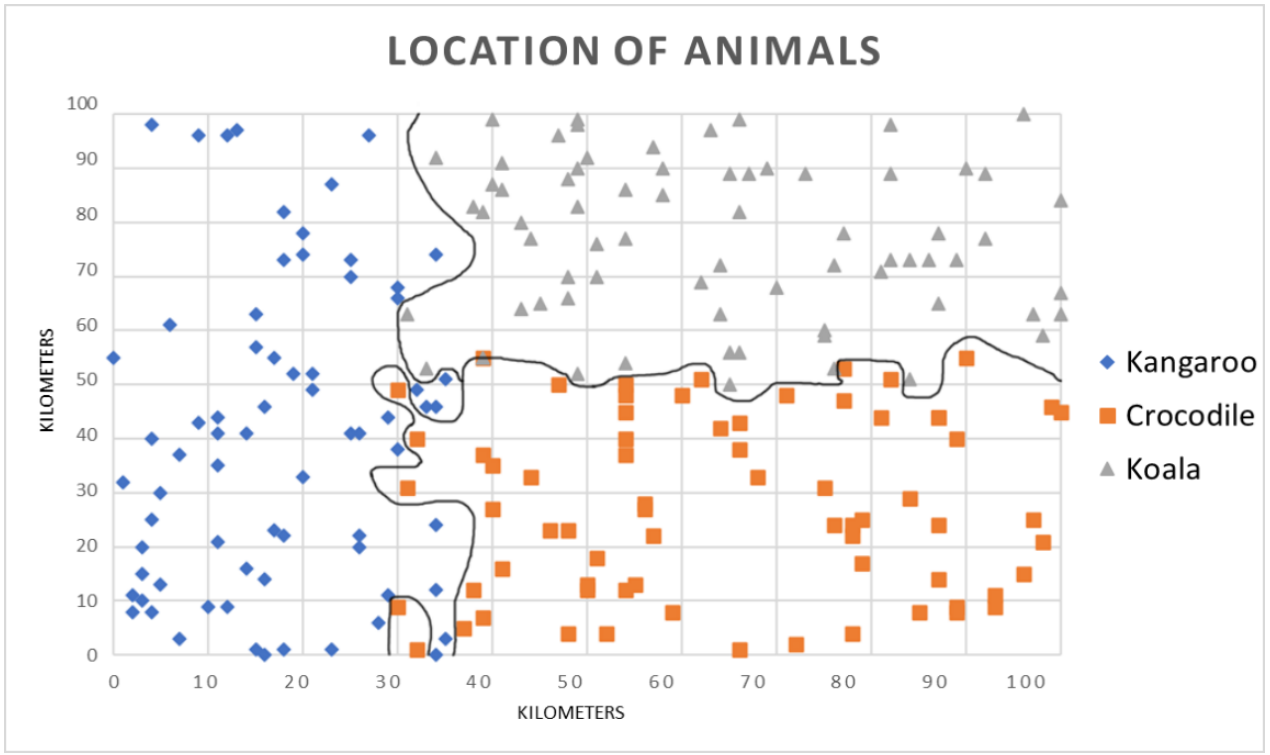
Classifying 2-Dimensional Data



Classifying 2-Dimensional Data



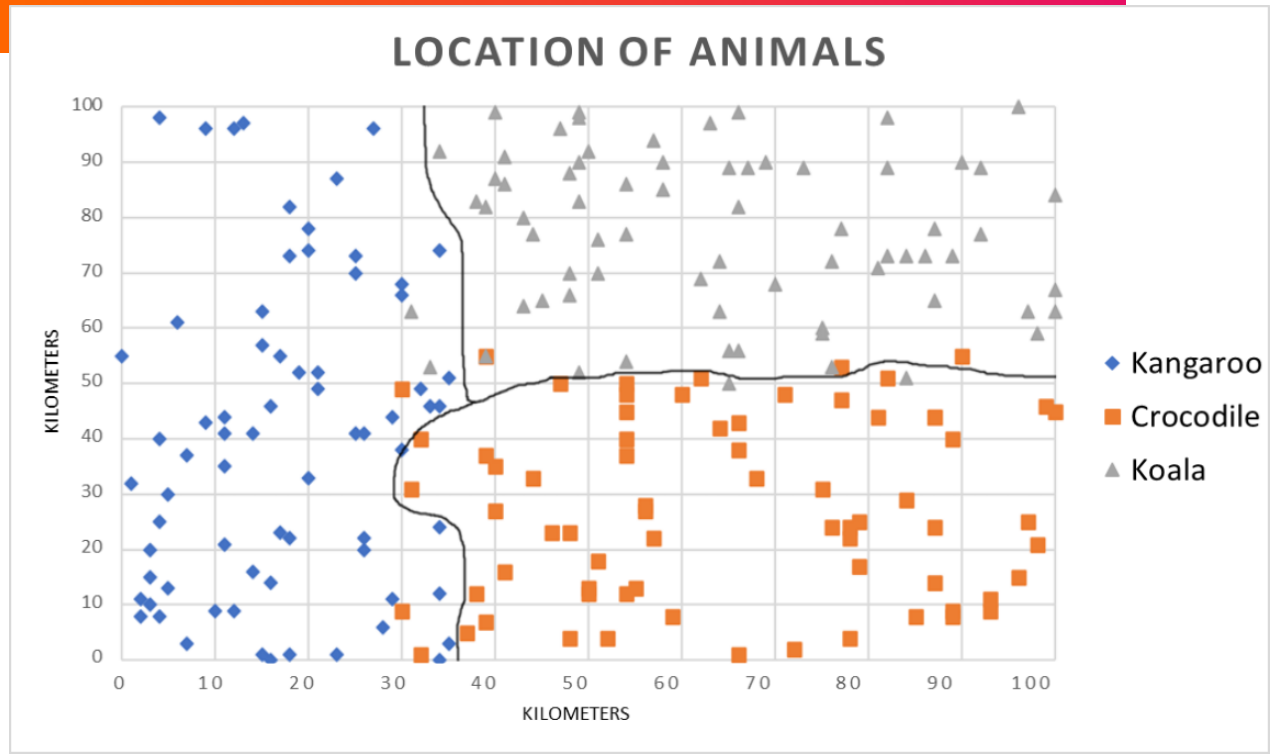
- 80% Accuracy



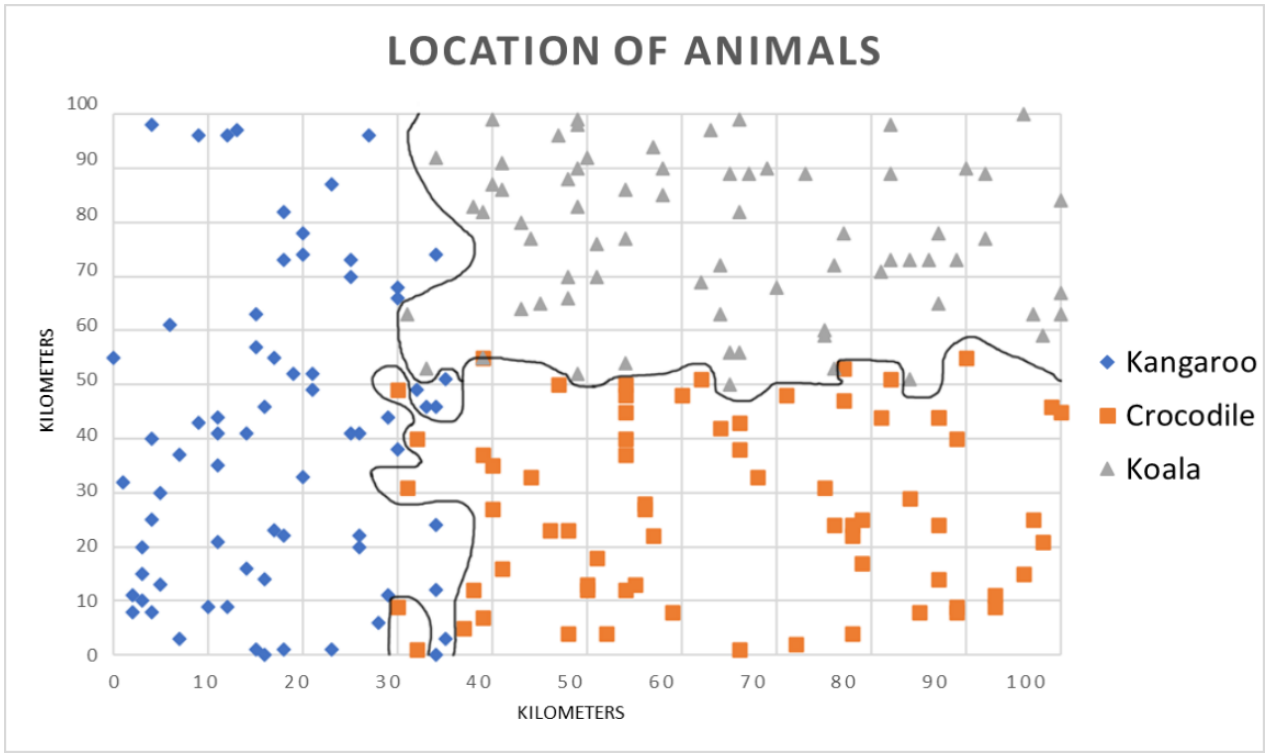
100% Accuracy

Which of these models is better?

Classifying 2-Dimensional Data



- 80% Accuracy

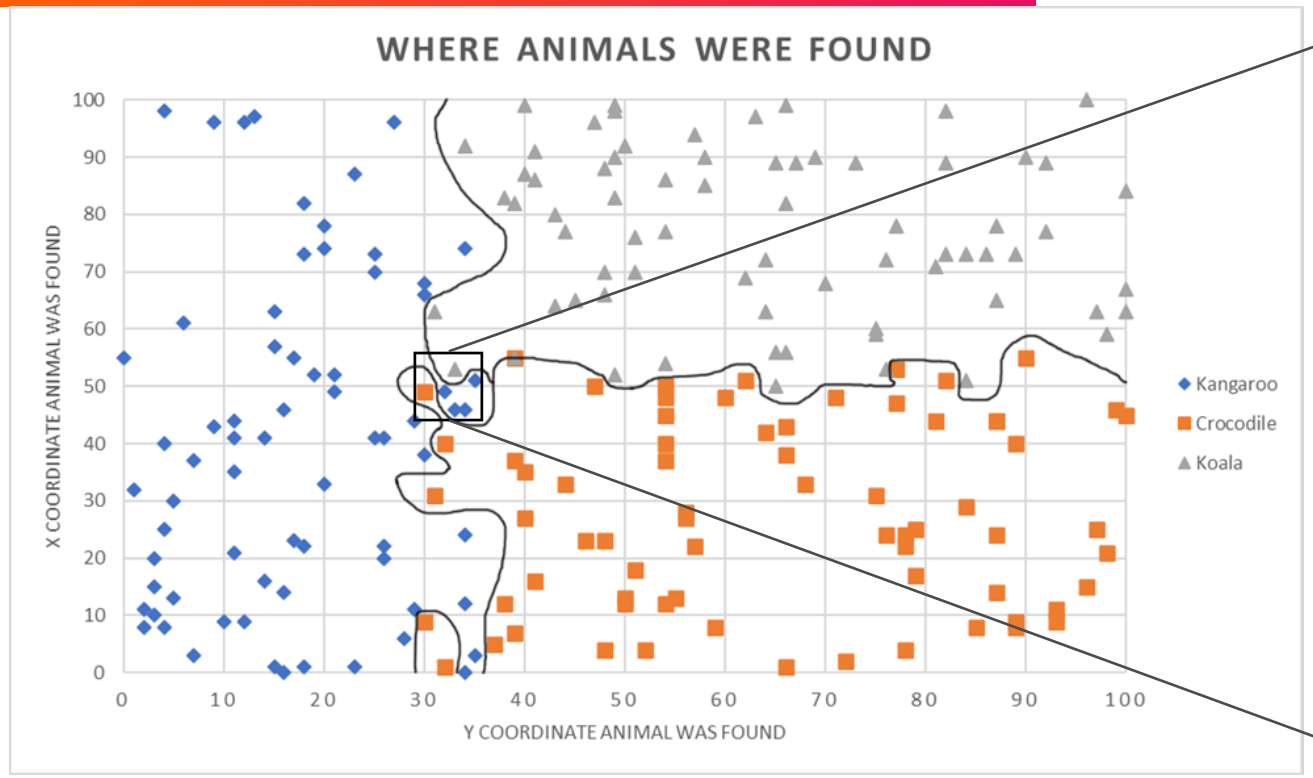


100% Accuracy (Overtrained)

As 80% is better than 100% as it is more general.

Modelling for generalisation is inherently inaccurate to an extent

Why Misclassifications Happen



In an overtrained network, little changes in a datapoint can make all the difference in classification.

X	Y	Class
32	50	Kangaroo
30	50	Crocodile
33	54	Koala

How A Neural Network Works

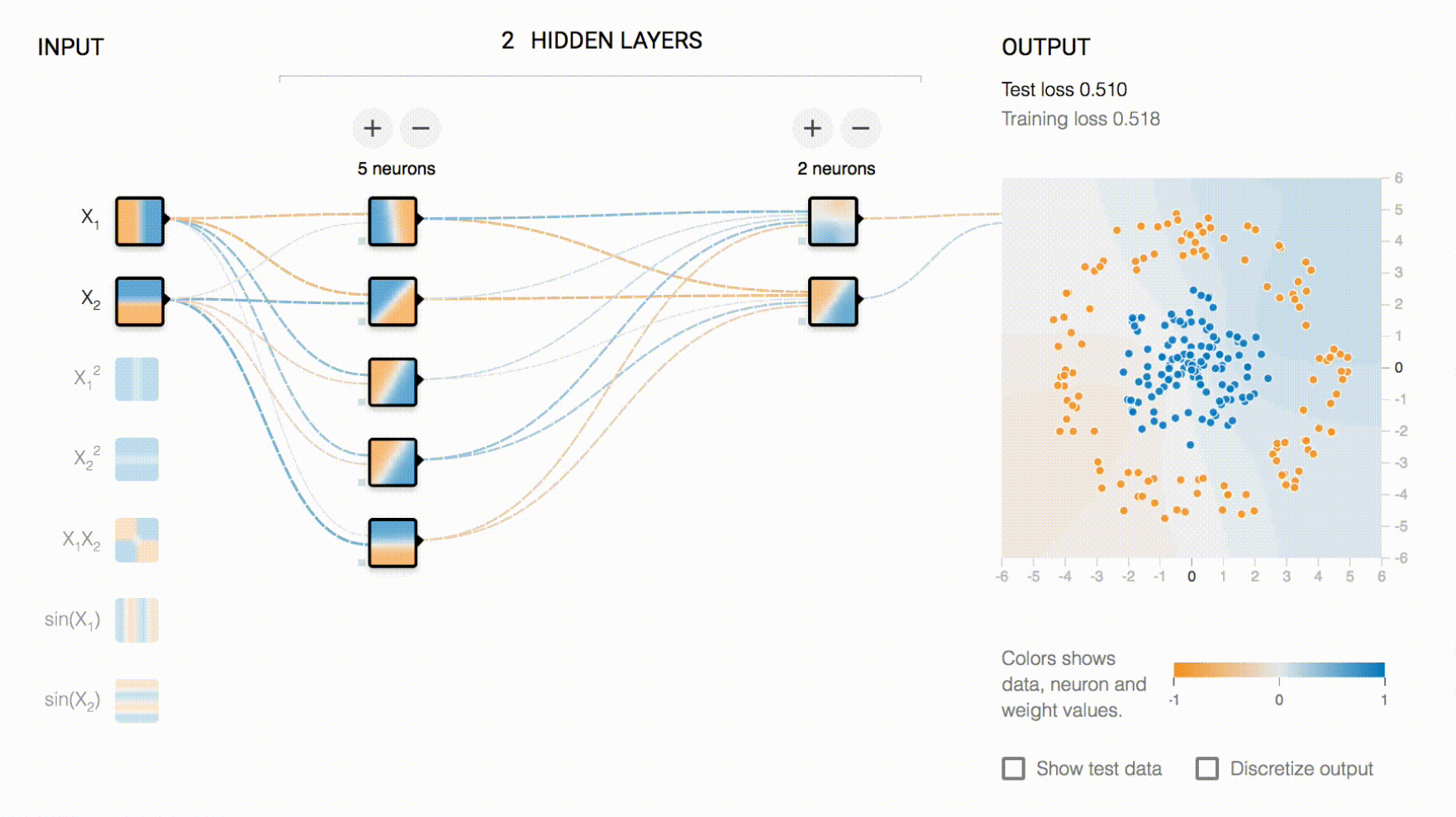


Fig. 4.

How A Neural Network Works

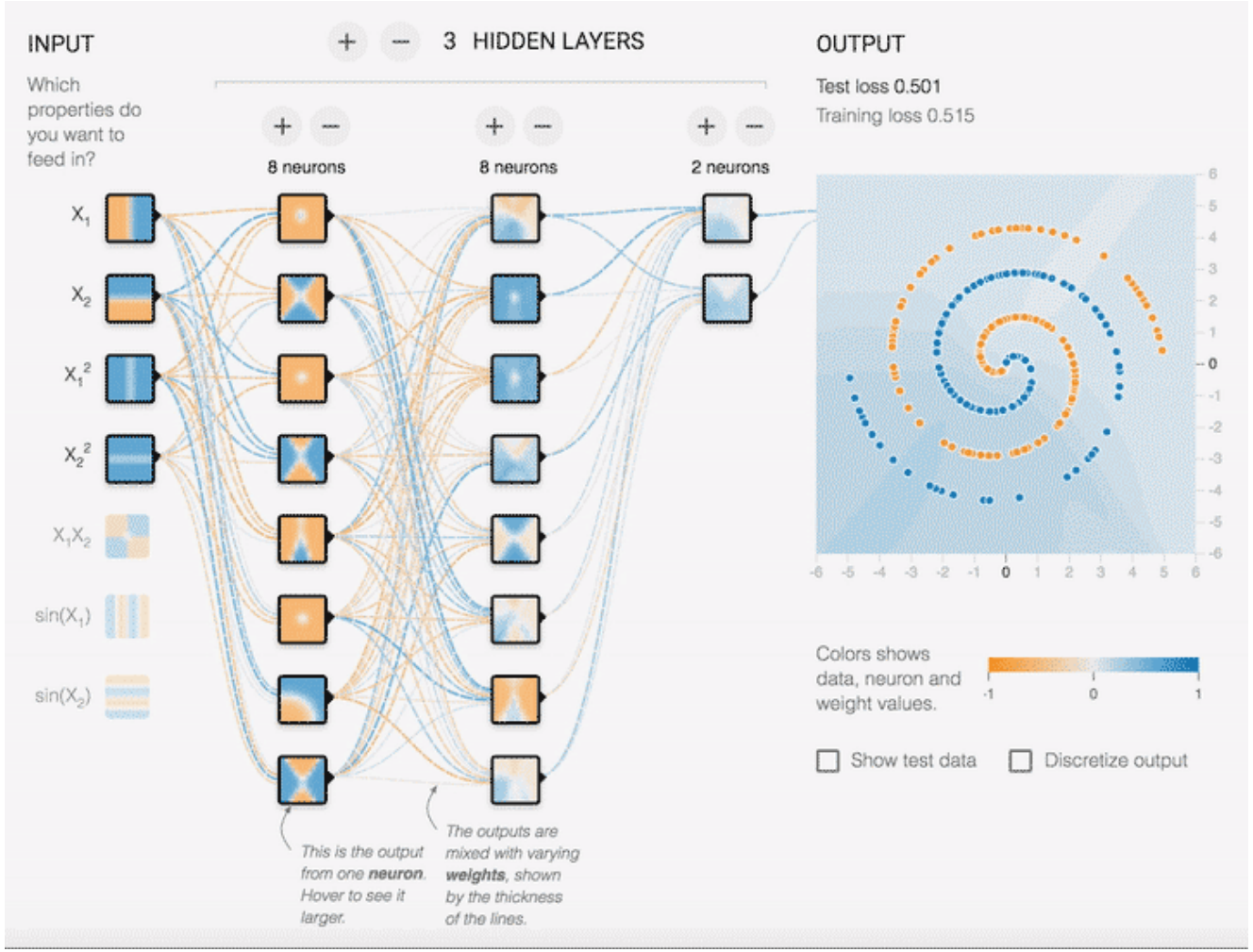


Fig. 5.

The Big Issue With Machine Learning

- This leads us to the fundamental weakness in machine learning solutions.
- Machine learning is inherently inaccurate to an extent.
- By definition, we don't want 100% accuracy. Which means some things are **going to have to be misclassified**.

Why this matters to attackers

As attackers we want to find things that are misclassified. Or in some cases we can make something misclassified 😉

- To bypass a safety feature
- Avoid antivirus
- Cause general havoc

The Machine Learning Threat Model

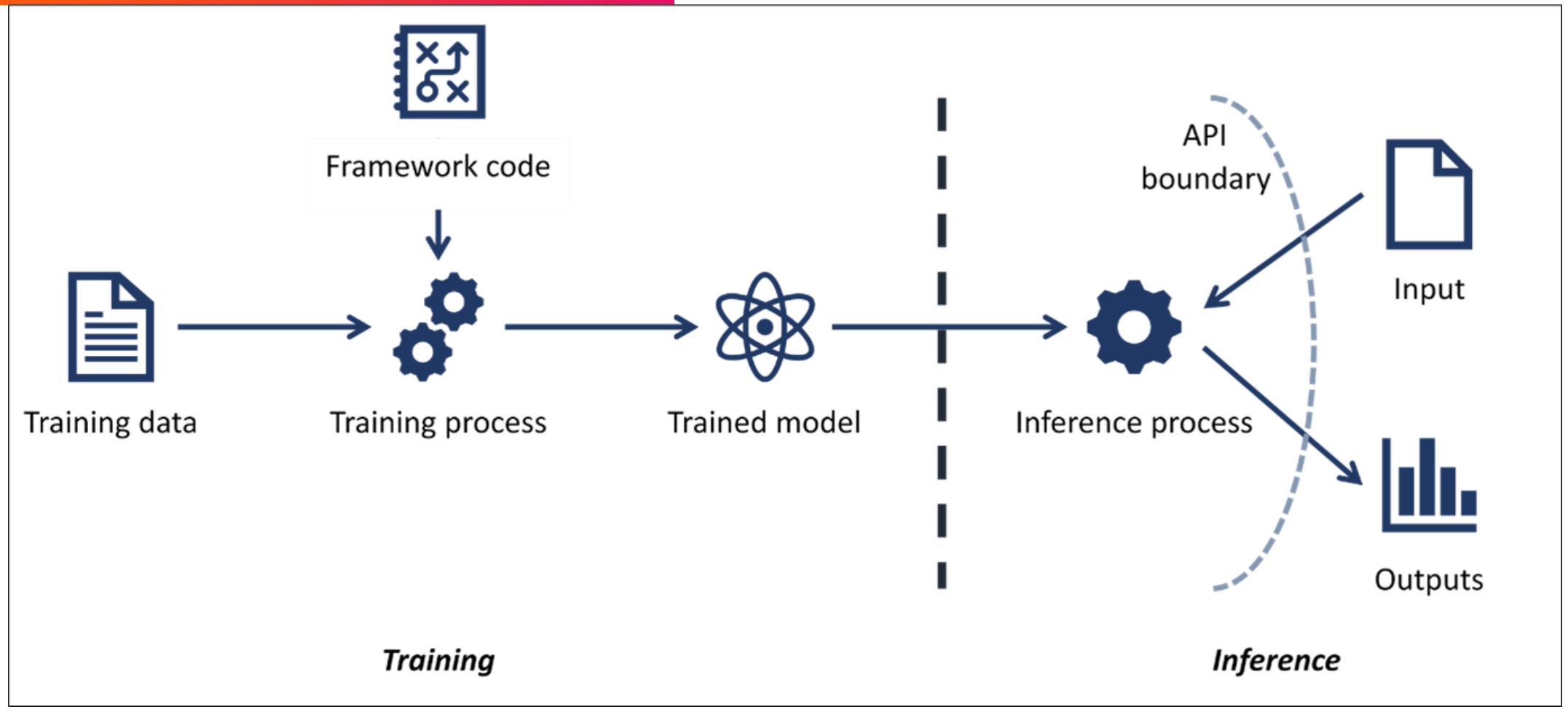


Fig. 6.

Whatever parts of this model users are interacting with are at risk

Unsupervised Learning

- Works off unlabeled data
- Provide data and let the algorithm figure it out
- Quick to implement
- A good way to verify your dataset
- An easy example is K-Means clustering

Classifying 2-Dimensional Data

K-Means clustering works by:

1. Defining K number of nodes, and randomly give them values in the space
2. Giving each data point the class title of the nearest node
3. Moving the node to the centre of that cluster
4. Repeating steps 2 and 3 until nothing moves

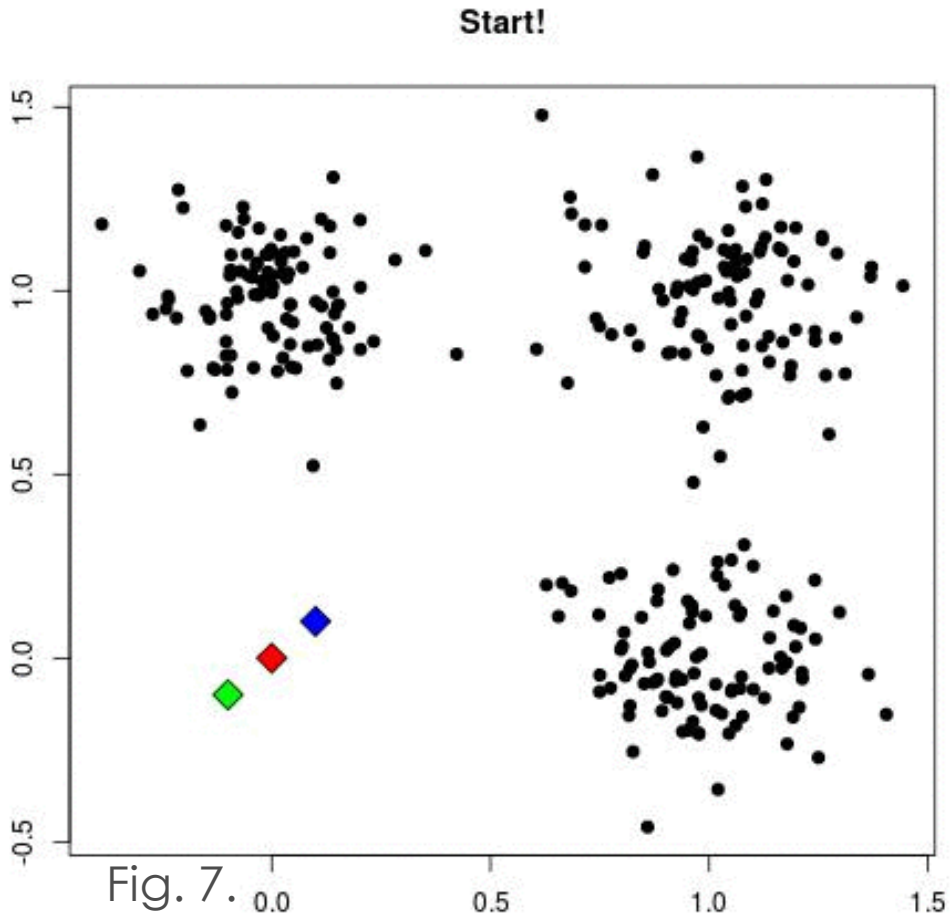


Fig. 7.

How Would You Cluster These Datasets?

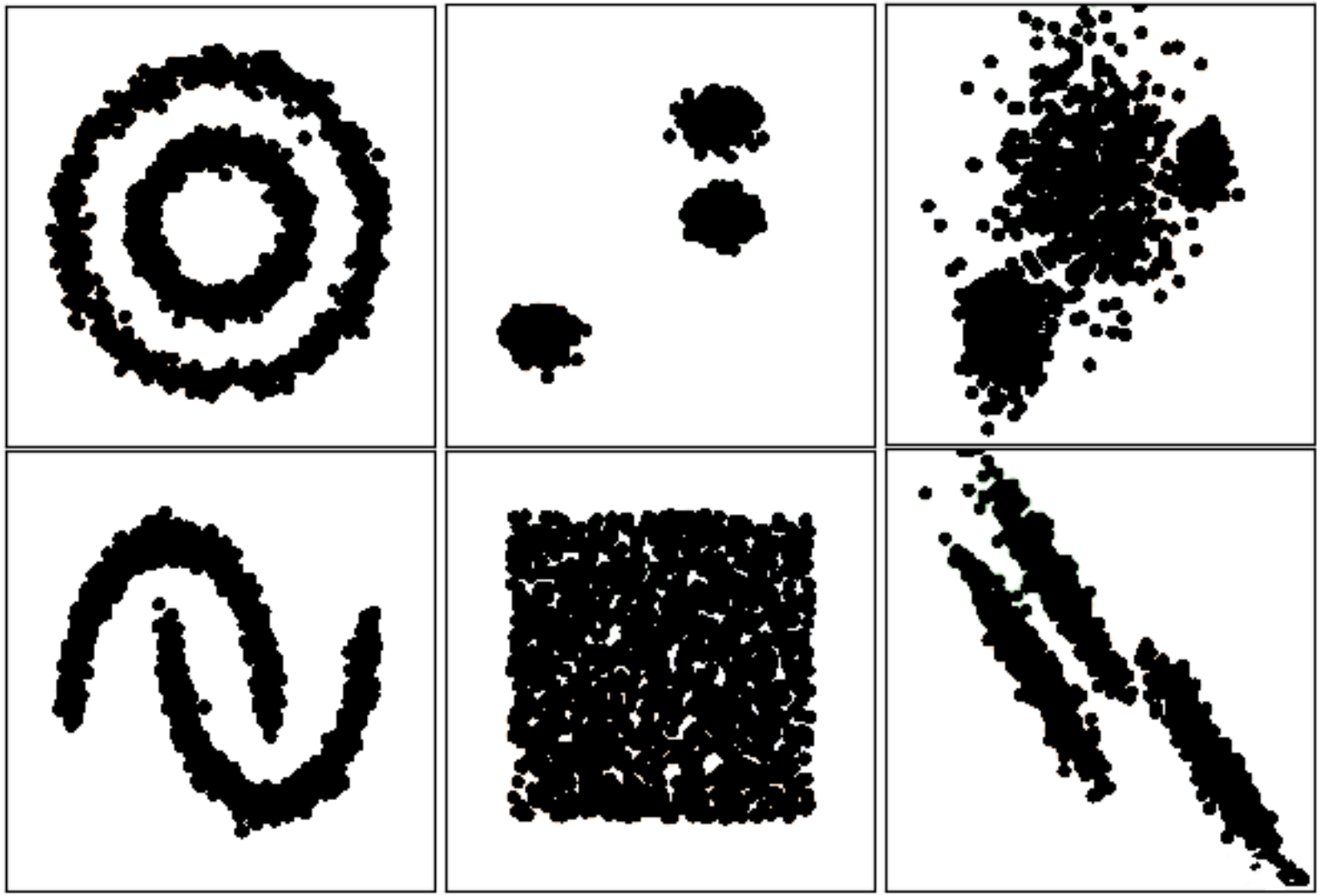


Fig. 8.1.

How Would You Cluster These Datasets?

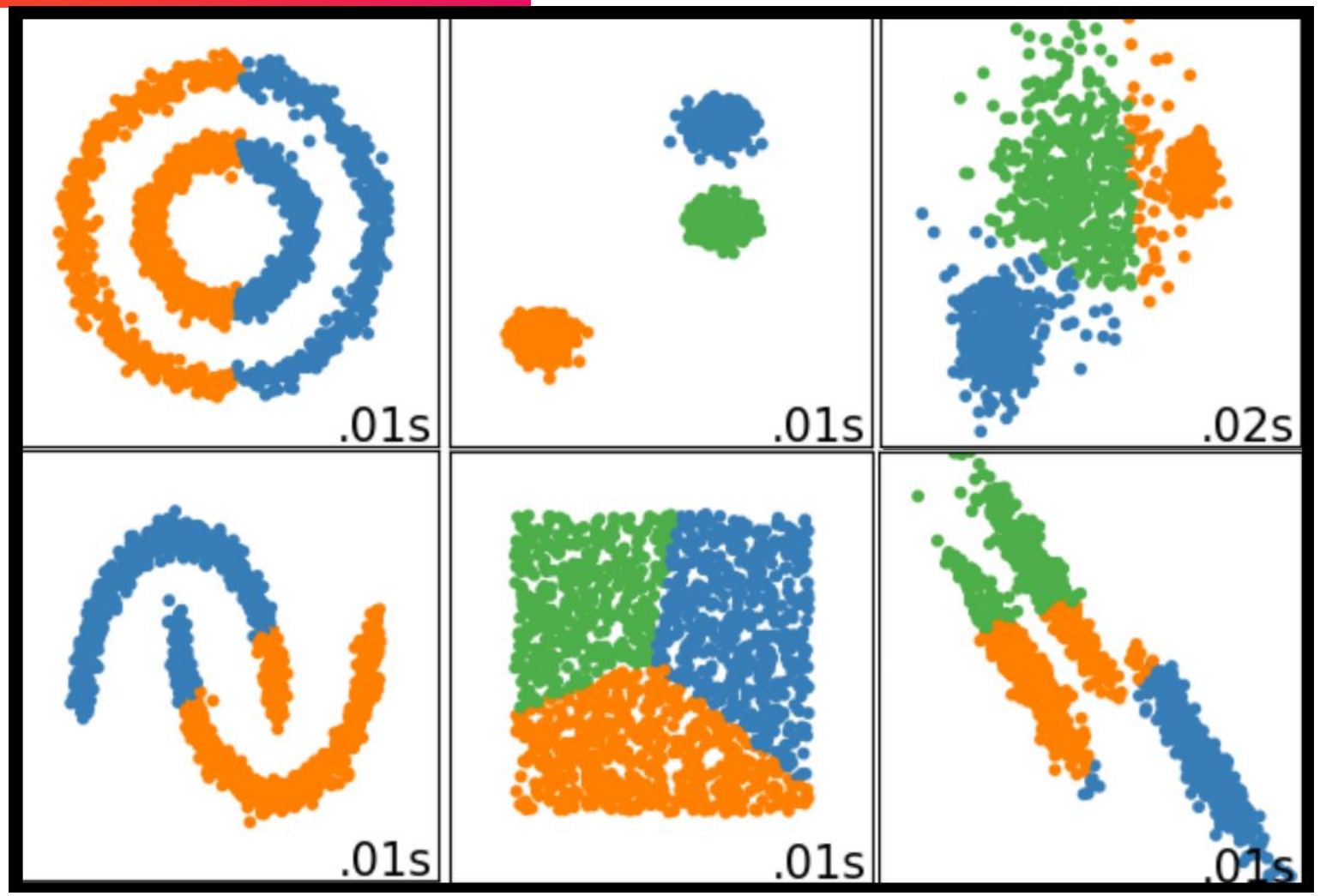


Fig. 8.2.

How Would You Cluster These Datasets?

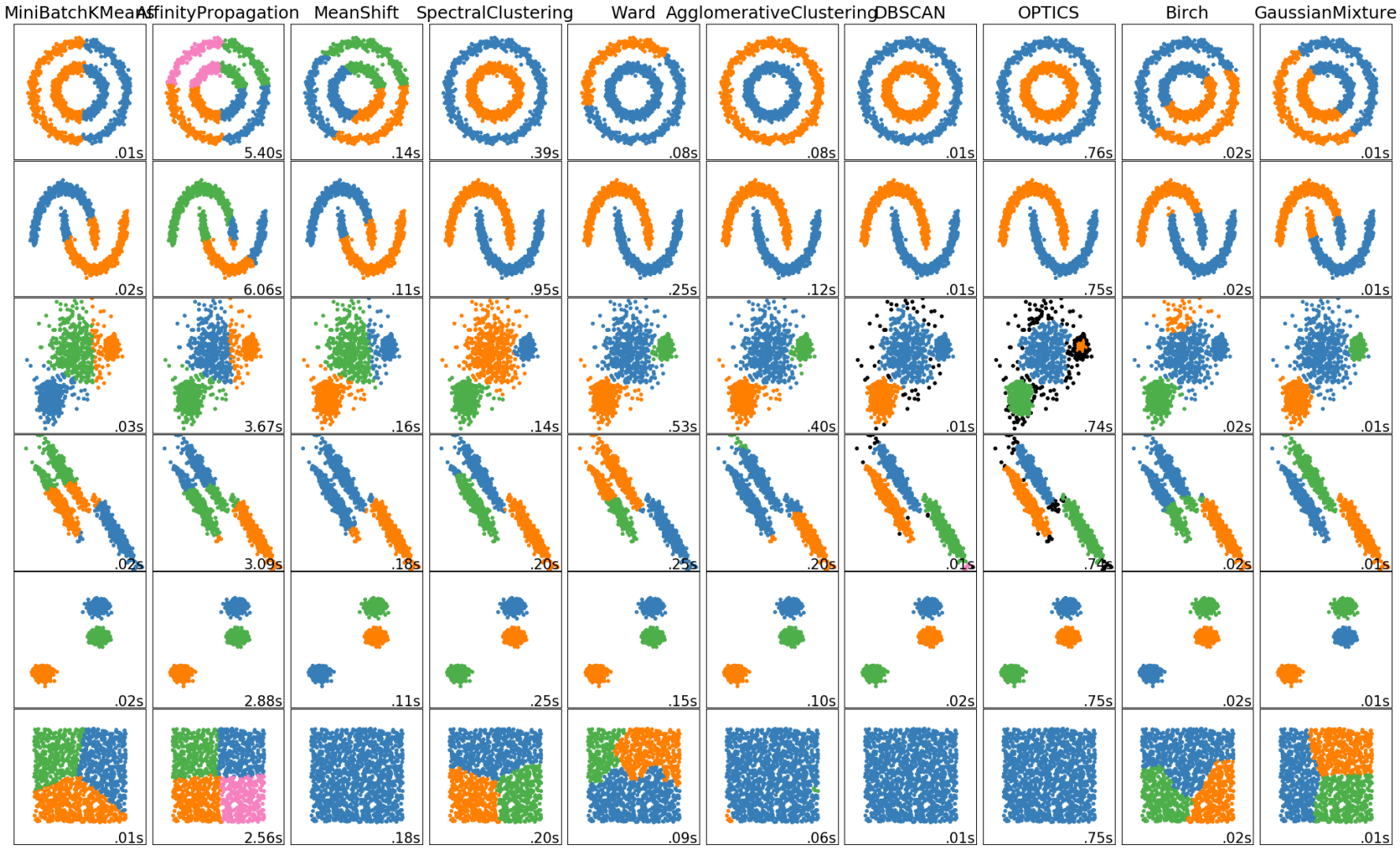


Fig. 8.3.

How Would You Cluster These Datasets?

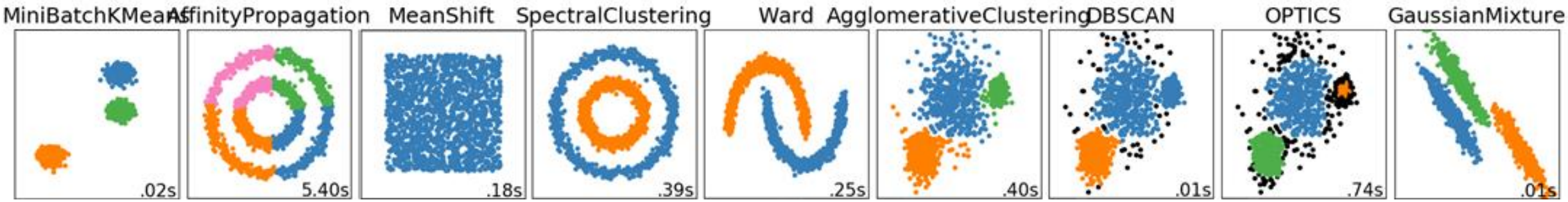


Fig. 8.4.

Issues With Unsupervised Learning

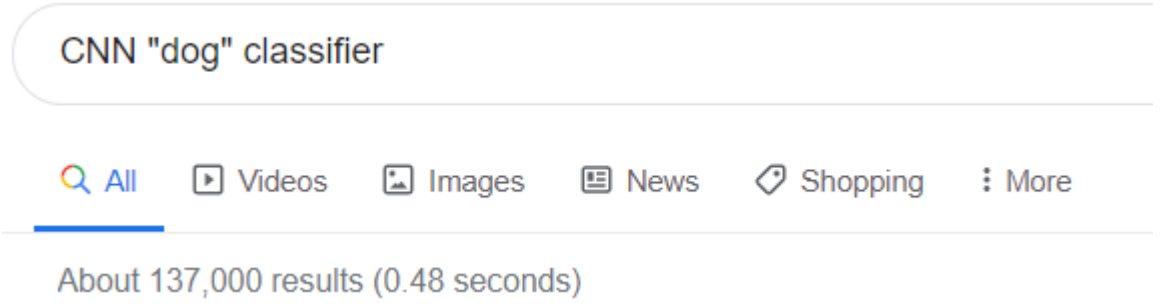
- All data is equally valued
- If you control input data you can affect the results
- Vulnerable to supply chain attacks
- Each algorithm has its quirks
- If you know the dataset and algorithm used, you can make a copy of the model
- Vulnerable to brute force

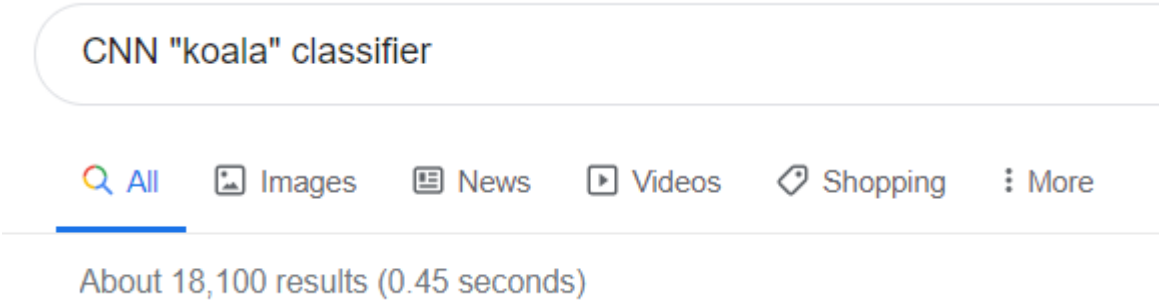
Transfer Learning

- Training Neural Networks takes a lot of work.
- Sometimes it can be helpful to start with someone else's model for a similar problem.

Steps:

1. Find a trained network with similar characteristics
2. Get dataset you want to transfer the model to
3. Train it a bit more on your dataset





Why Images Are Hard

Which of these are stop signs?

Fig. 9.

Fig. 11.

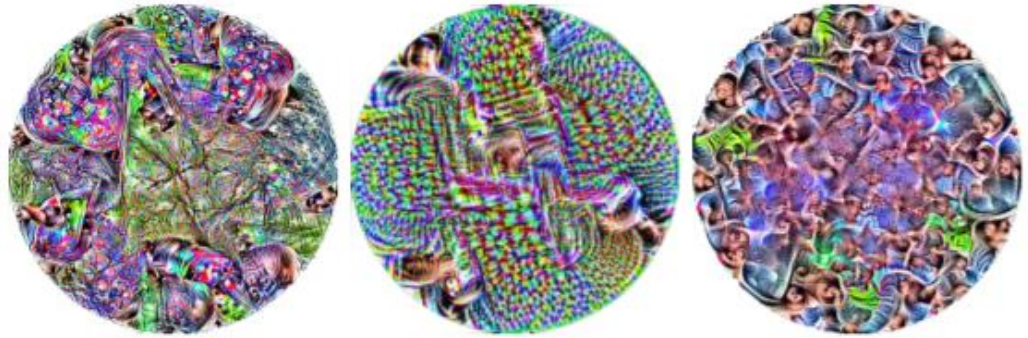


Fig. 10.

Why Images Are Hard

> “Images are to machine learning what the JavaScript alert is to XSS.“ – Nathan Hamiel

MITRE Adversarial Machine Learning Threat Matrix

Reconnaissance	Initial Access	Execution	Persistence	Model Evasion	Exfiltration	Impact
Acquire OSINT information: (Sub Techniques) 1. Arxiv 2. Public blogs 3. Press Releases 4. Conference Proceedings 5. Github Repository 6. Tweets	Pre-trained ML model with backdoor	Execute unsafe ML models (Sub Techniques) 1. ML models from compromised sources 2. Pickle embedding	Execute unsafe ML models (Sub Techniques) 1. ML models from compromised sources 2. Pickle embedding	Evasion Attack (Sub Techniques) 1. Offline Evasion 2. Online Evasion	Exfiltrate Training Data (Sub Techniques) 1. Membership inference attack 2. Model inversion	Defacement
ML Model Discovery (Sub Techniques) 1. Reveal ML model ontology – 2. Reveal ML model family –	Valid account	Execution via API	Account Manipulation		Model Stealing	Denial of Service
Gathering datasets	Phishing	Traditional Software attacks	Implant Container Image	Model Poisoning	Insecure Storage 1. Model File 2. Training data	Stolen Intellectual Property
Exploit physcial environment	External remote services			Data Poisoning (Sub Techniques) 1. Tainting data from acquisition – Label corruption 2. Tainting data from open source supply chains 3. Tainting data from acquisition – Chaff data 4. Tainting data in training environment – Label corruption		Data Encrypted for Impact Defacement
Model Replication (Sub Techniques) 1. Exploit API – Shadow Model 2. Alter publicly available, pre-trained weights	Exploit public facing application					Stop System Shutdown/Reboot
Model Stealing	Trusted Relationship					

Fig. 12.

Recon: ML Model Discovery

- Reveal ML Ontology. Such as dataset (image, audio, tabular, NLP), features (handcrafted or learned), model / learning algorithm (gradient based or non-gradient based), parameters / weights.)
- Reveal ML Family, More vague, trying to figure out rough elements (Model task, model input and model output)
  - Dataset type (Images, Audio, Video)
  - Model architecture (CNN, DNN, Transfer Learned)
  - Output (Classification, Gradient Based)

Recon:

- Gathering Datasets
- Model Replication
  - Exploit API - Shadow Model
  - Alter Publicly Available, Pre-Trained Weights (Transfer Learning)
- Model Stealing

RECON: Exploit Physical Environment

Exploit Physical Environment:

- APRICOT (APRICOT (Adversarial Patches Rearranged in COnText)
- Physical obstructions*

Fig. 13.

Fig. 14.

Recon: OSINT

- Arxiv (Archive)
- Public Blogs
- Press Releases
- Conference Proceedings
- Github Repository
- Tweets
- Other Media*
- Patents*
- Forum Posts*

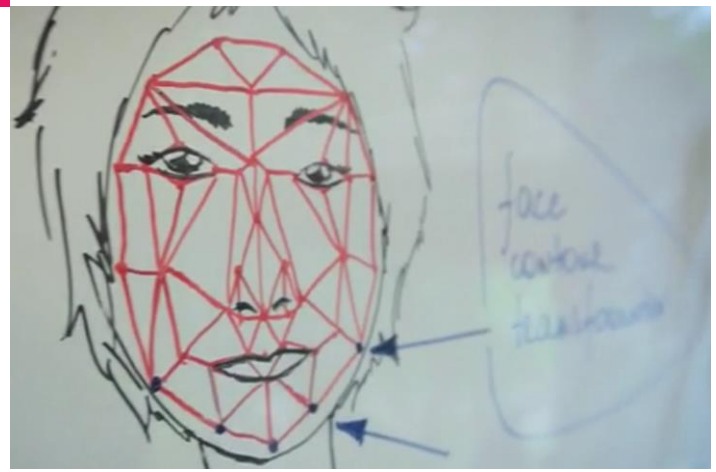
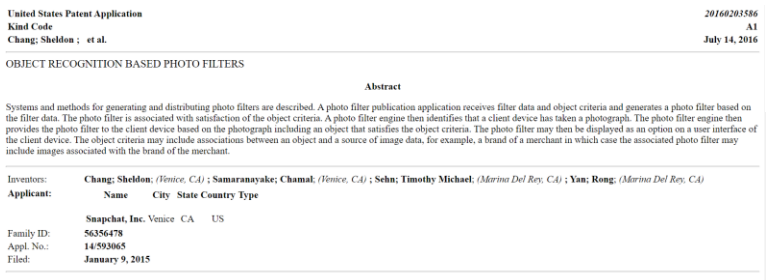


Fig. 15.



United States Patent Application *20160203586*
Kind Code **A1**
Chang; Sheldon ; et al. **July 14, 2016**

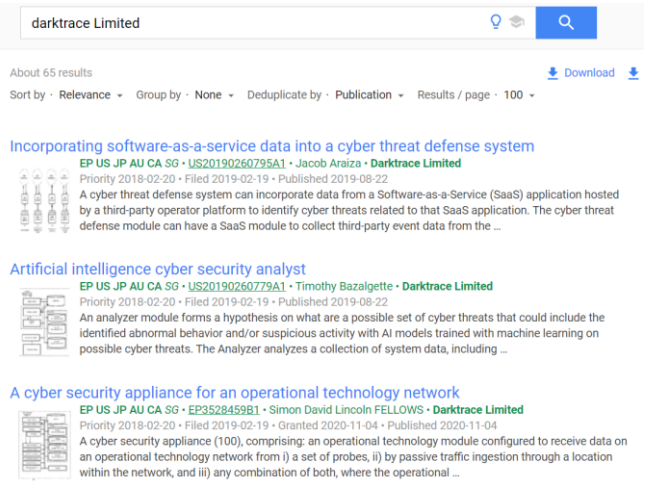
OBJECT RECOGNITION BASED PHOTO FILTERS

Abstract

Systems and methods for generating and distributing photo filters are described. A photo filter publication application receives filter data and object criteria and generates a photo filter based on the filter data. The photo filter is associated with satisfaction of the object criteria. A photo filter engine then identifies that a client device has taken a photograph. The photo filter engine then provides the photo filter to the client device based on the photograph including an object that satisfies the object criteria. The photo filter may then be displayed as an option on a user interface of the client device. The object criteria may include associations between an object and a source of image data, for example, a brand of a merchant in which case the associated photo filter may include images associated with the brand of the merchant.

Inventors: **Chang; Sheldon**; *(Venice, CA)* ; **Samaranayake; Chamal**; *(Venice, CA)* ; **Sehn; Timothy Michael**; *(Marina Del Rey, CA)* ; **Yan; Rong**; *(Marina Del Rey, CA)*
Applicant: **Name** **City** **State** **Country** **Type**
Snapchat, Inc. Venice CA US
Family ID: **56356478**
Appl. No.: **14/593065**
Filed: **January 9, 2015**

Fig. 16.



darktrace Limited

About 65 results — Download

Sort by · Relevance · Group by · None · Deduplicate by · Publication · Results / page · 100

Incorporating software-as-a-service data into a cyber threat defense system
EP US JP AU CA *SG* · US20190260795A1 · Jacob Araiza · **Darktrace Limited**
Priority 2018-02-20 · Filed 2019-02-19 · Published 2019-08-22
A cyber threat defense system can incorporate data from a Software-as-a-Service (SaaS) application hosted by a third-party operator platform to identify cyber threats related to that SaaS application. The cyber threat defense module can have a SaaS module to collect third-party event data from the ...

Artificial intelligence cyber security analyst
EP US JP AU CA *SG* · US20190260779A1 · Timothy Bazalgette · **Darktrace Limited**
Priority 2018-02-20 · Filed 2019-02-19 · Published 2019-08-22
An analyzer module forms a hypothesis on what are a possible set of cyber threats that could include the identified abnormal behavior and/or suspicious activity with AI models trained with machine learning on possible cyber threats. The Analyzer analyzes a collection of system data, including ...

A cyber security appliance for an operational technology network
EP US JP AU CA *SG* · EP3528459B1 · Simon David Lincoln FELLOWS · **Darktrace Limited**
Priority 2018-02-20 · Filed 2019-02-19 · Granted 2020-11-04 · Published 2020-11-04
A cyber security appliance (100), comprising: an operational technology module configured to receive data on an operational technology network from i) a set of probes, ii) by passive traffic ingestion through a location within the network, and iii) any combination of both, where the operational ...

Fig. 17.

Initial Access

Pre-trained ML model with backdoor:

- Transfer Learning
- Purchased Models

BadNets: Identifying Vulnerabilities in the Machine Learning Model Supply Chain

Tianyu Gu, Brendan Dolan-Gavitt, Siddharth Garg

Deep learning-based techniques have achieved state-of-the-art performance on a wide variety of recognition and classification tasks. However, these networks are typically computationally expensive to train, requiring weeks of computation on many GPUs; as a result, many users outsource the training procedure to the cloud or rely on pre-trained models that are then fine-tuned for a specific task. In this paper we show that outsourced training introduces new security risks: an adversary can create a maliciously trained network (a backdoored neural network, or a \emph{BadNet}) that has state-of-the-art performance on the user's training and validation samples, but behaves badly on specific attacker-chosen inputs. We first explore the properties of BadNets in a toy example, by creating a backdoored handwritten digit classifier. Next, we demonstrate backdoors in a more realistic scenario by creating a U.S. street sign classifier that identifies stop signs as speed limits when a special sticker is added to the stop sign; we then show in addition that the backdoor in our US street sign detector can persist even if the network is later retrained for another task and cause a drop in accuracy of {25}\% on average when the backdoor trigger is present. These results demonstrate that backdoors in neural networks are both powerful and---because the behavior of neural networks is difficult to explicate---stealthy. This work provides motivation for further research into techniques for verifying and inspecting neural networks, just as we have developed tools for verifying and debugging software.

Fig. 18.

Execution/Persistence

Execute unsafe ML models

- ML models from compromised sources
- Pickle embedding

Backdoored Sample code

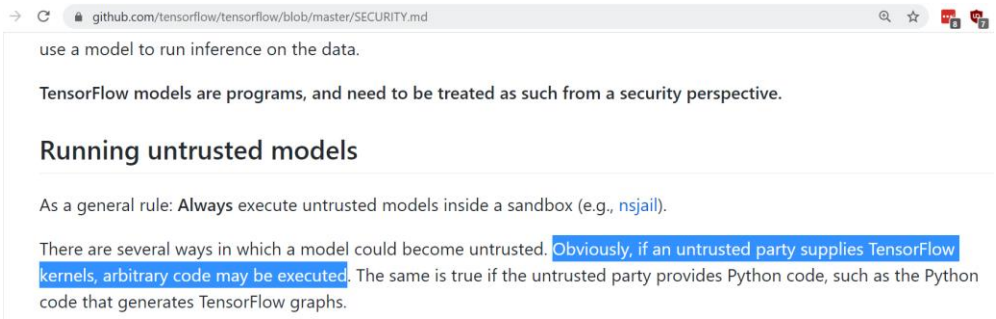


Fig. 18.

```
import tensorflow
from tensorflow import keras
grace_hopper =
tf.keras.utils.get_file('image.jpg','https://storage.
googleapis.com/download.tensorflow.org/exa
mple_images/grace_hopper.jpg')
grace_hopper =
Image.open(grace_hopper).resize(IMAGE_SHA
PE)
```

Maybe run a crypto miner payload?
See if it runs in https://github.com/google/nsjail

```
##############################################
#example of unsafe de-serialization
import pickle
```

Model Evasion

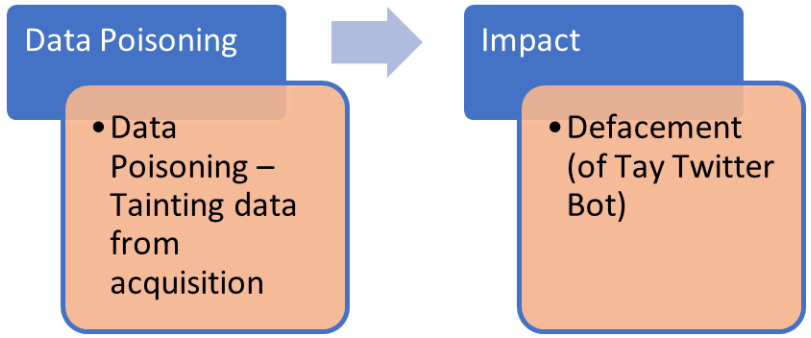
Evasion

- Online Evasion
- Offline Evasion

Model Poisoning

Data poising

- Tainting Data from Acquisition - Label Corruption
- Tainting Data from Open Source Supply Chains
- Tainting Data from Acquisition - Chaff Data
- Tainting Data in Training - Label Corruption



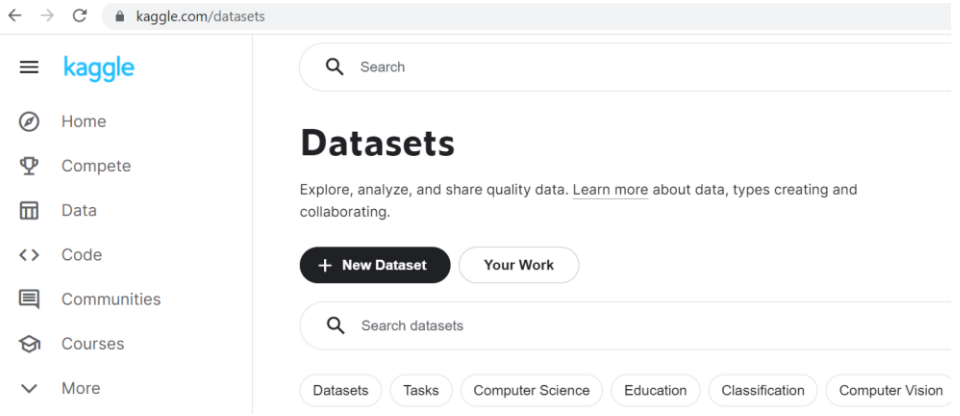


Fig. 19.

Evasion Attack

Steps to do Adversarial Noise:

1. Slightly change the image (Gaussian noise)
2. Check the image confidence against the model
3. If it has higher owl confidence it updates current image
4. Repeat steps 2 and 3 until the model classifies it as the target class

My Table of Owls and Turtles

	My Classification	
	Turtle	Owl
Turtle		
Model Classification		
Owl		

Fig. 20.

Fig. 21.

Evasion Attack Example

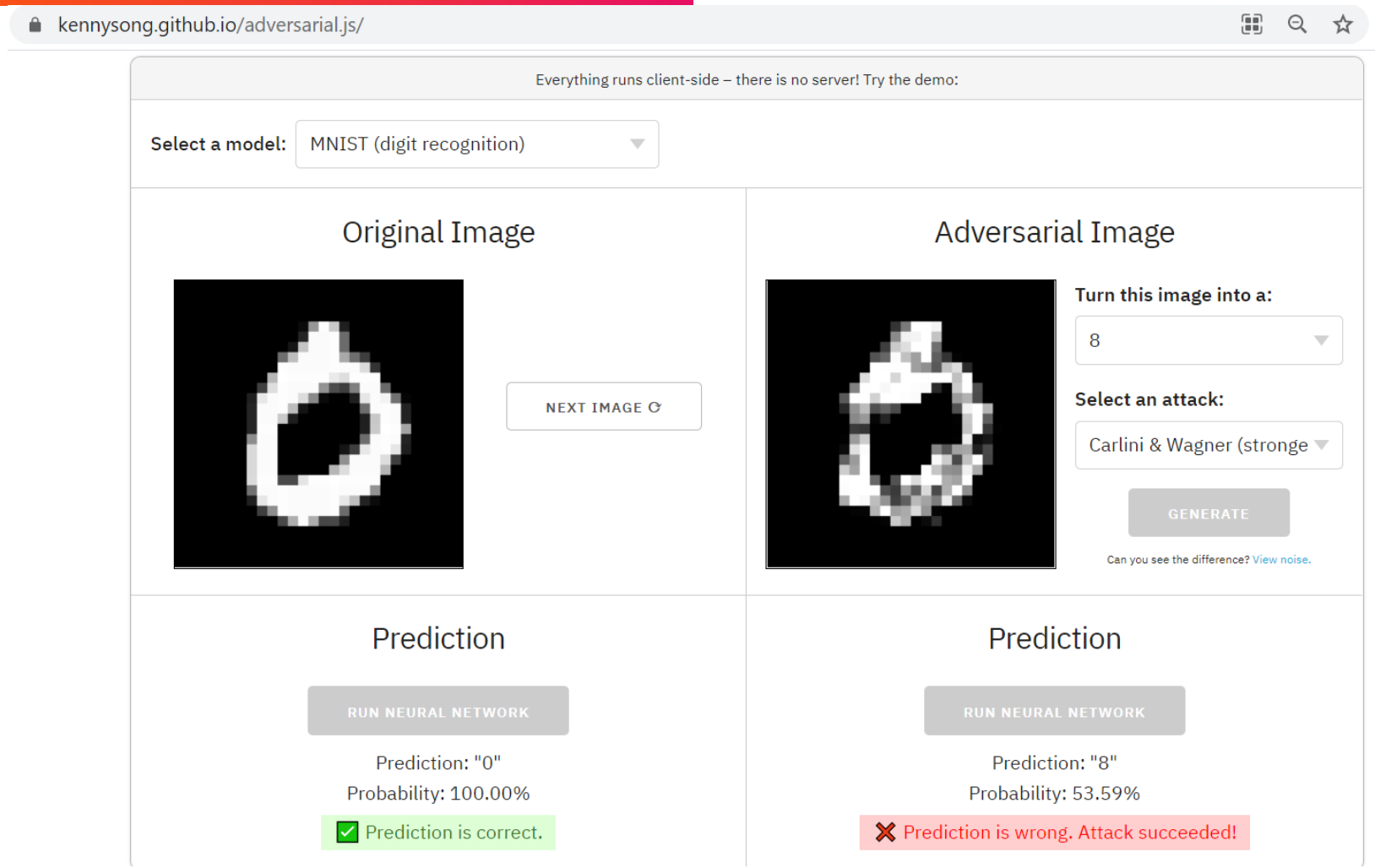
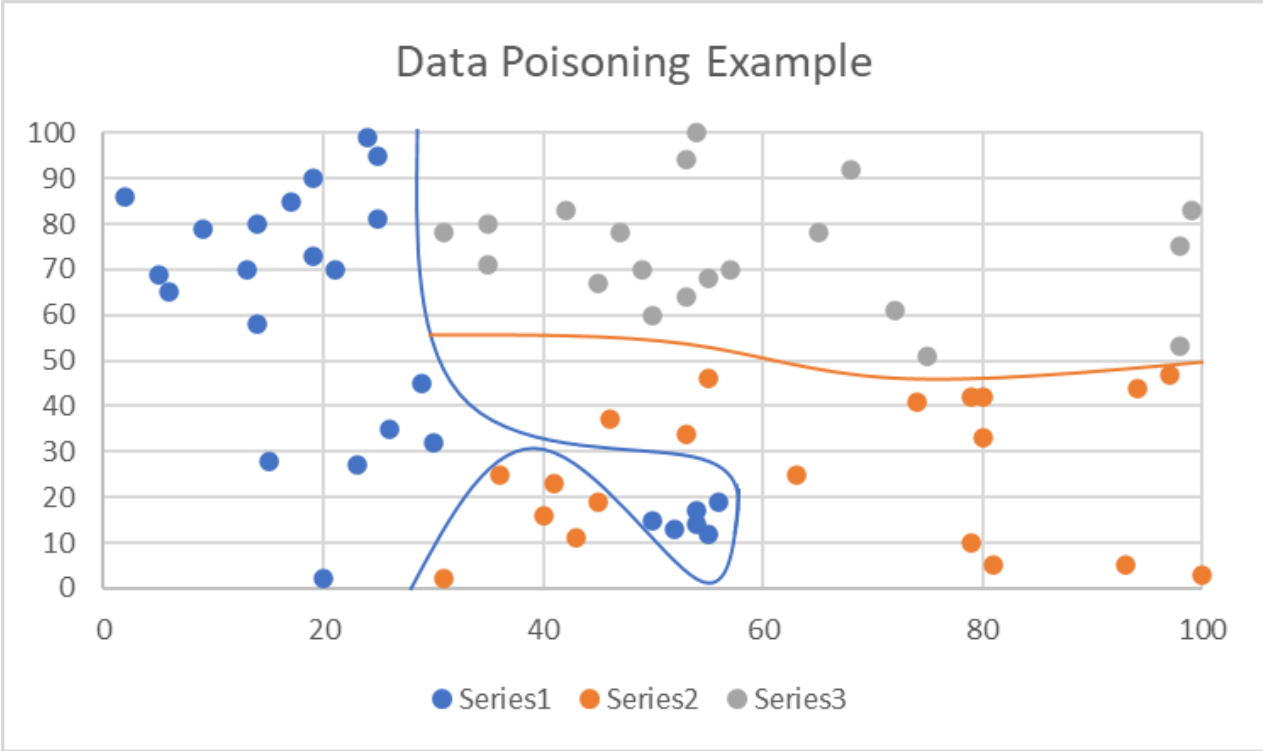


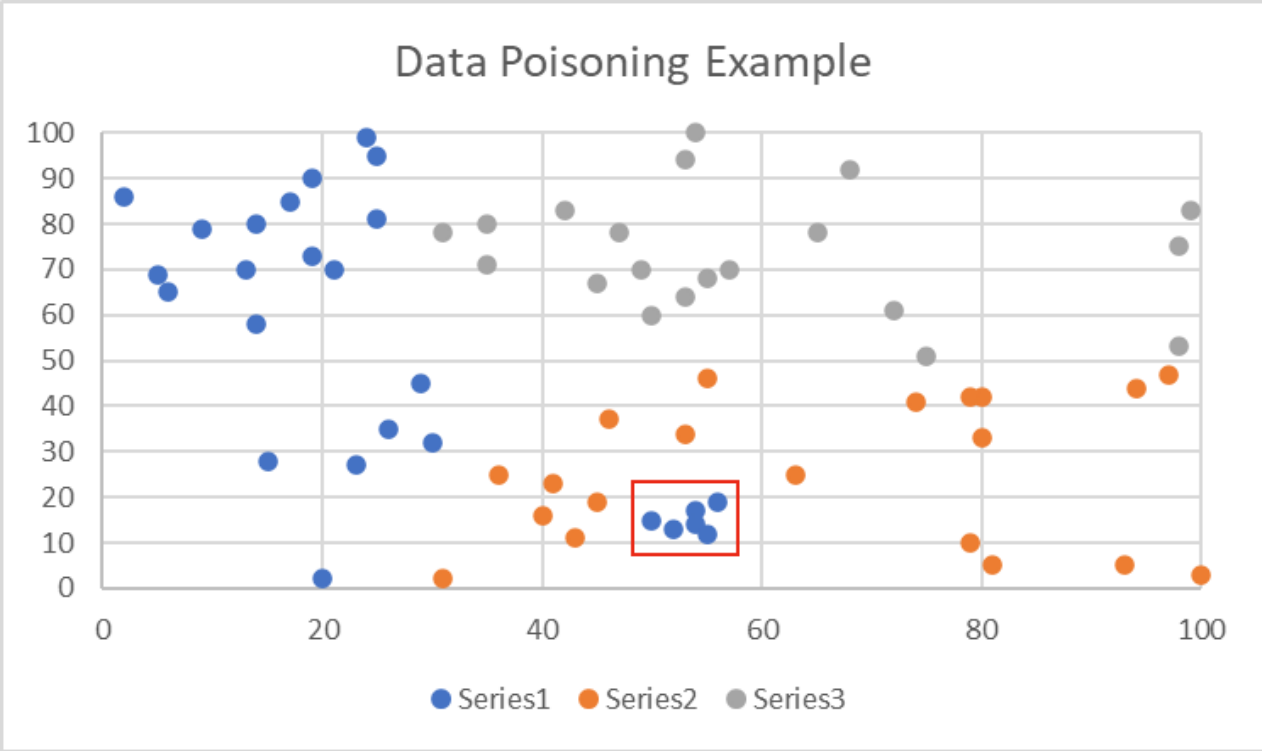
Fig. 22.

ONLINE VS OFFLINE EVSION

- Just like passwords, offline attacks are easier
- How much feedback does the online model make?
- Can you predict how the online model is built?

Data Poisoning





Data Poisoning Case Study

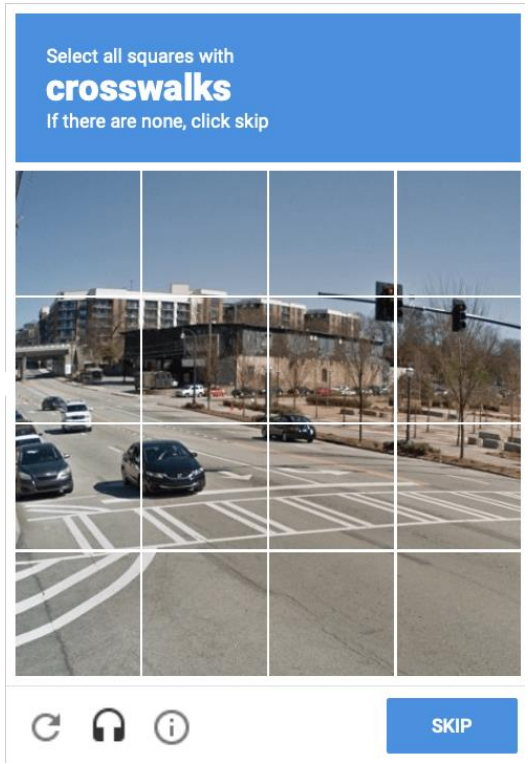


Fig. 23.

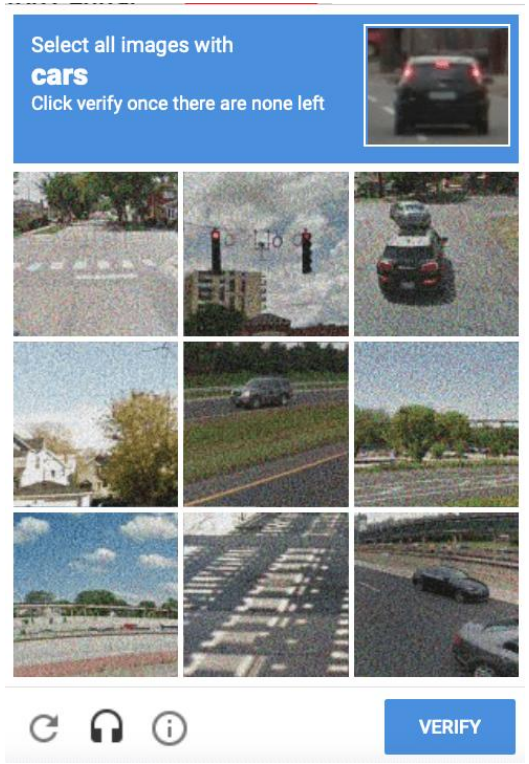


Fig. 24.

- Google Datamining operation
- They want to use supervised learning, but need to classify their massive dataset.
- They chose to crowdsource the classifications for their data.
- This opens Google up to a data poisoning attack.
- Have they managed this risk?

Google Case Study

Risk Profile:

- Attacker trying to cause harm to Google
- Poison the dataset by labelling non fire hydrants as fire hydrants
- Can an attacker affect the dataset in any negative way?
- Could the dataset get invalidated?
- Could this lead to a self driving car having an accident?

How I Would Defend Against This

Is Google defending against this?

How I Would Defend Against This

Is Google defending against this?

I Don't know :)

But if I was in charge of securing this system. I would make sure that no data is marked as trusted until it has been consistently marked a single result

Model Poisoning

Modify the data used to train a model:

Fig. 25.

Fig. 26.

Exfiltration

- Membership Inference Attack
  - Figure out what sample data was used
- ML Model Inversion
  - Reconstruct private data from inference API
    - **NSA's Next Gen Firewall (Et blue)**
- ML Model Stealing
  - Extraction/Replication

Impact

- Defacement (Tay Bot)
- DOS (Reverting ML models)
- Stolen Intellectual Property
- Data Encrypted for Impact
- Stop System Shutdown/Reboot
- Reputation Damage*
- Bypassing Defences (Cylance, Virus Total)*
- Crypto Mining*

GANS

https://nbviewer.jupyter.org/github/greydanus/mnist-gan/blob/master/cnn-gan.ipynb

Synthetic Media

- Resemble.ai

Defence: Model Protection

Avoid offline evasion by protecting your models:

- Encrypt your models on disk
- Don't use client side models
- Use canary models
- Paper Towns?

Avoid online evasion by protecting your inference API:

- Don't return confidence levels
- Rate limit requests

Defence: Evasion

Data Modification

- Hardcoded nonce Mask
- Randomised Nonce Mask
- Run smoothing or sharpening on the image
- Modify the colormap

Admission Control

- Drop samples with too much volatility
- Measure the Structural Similarity (SSIM TODO)

Multiple Assessment Functions

- Multi Model Classifiers
- Multi Modal Classifiers

Defence: Model Protection

Salts would be mask put on each image before classification to reduce the risks of gaussian noise

Salt models the way you salt passwords

Salts could be unique per image if they are derived from the image. That way you can gain consistent classification results.

Make sure the salt is unique to each model or trust zone.

Defence: Admission Control

- Blocking images that are too noisy
- Denoise images

Defence: Admission Control

So just block noisy images?

Defence: Admission Control

Its not that simple

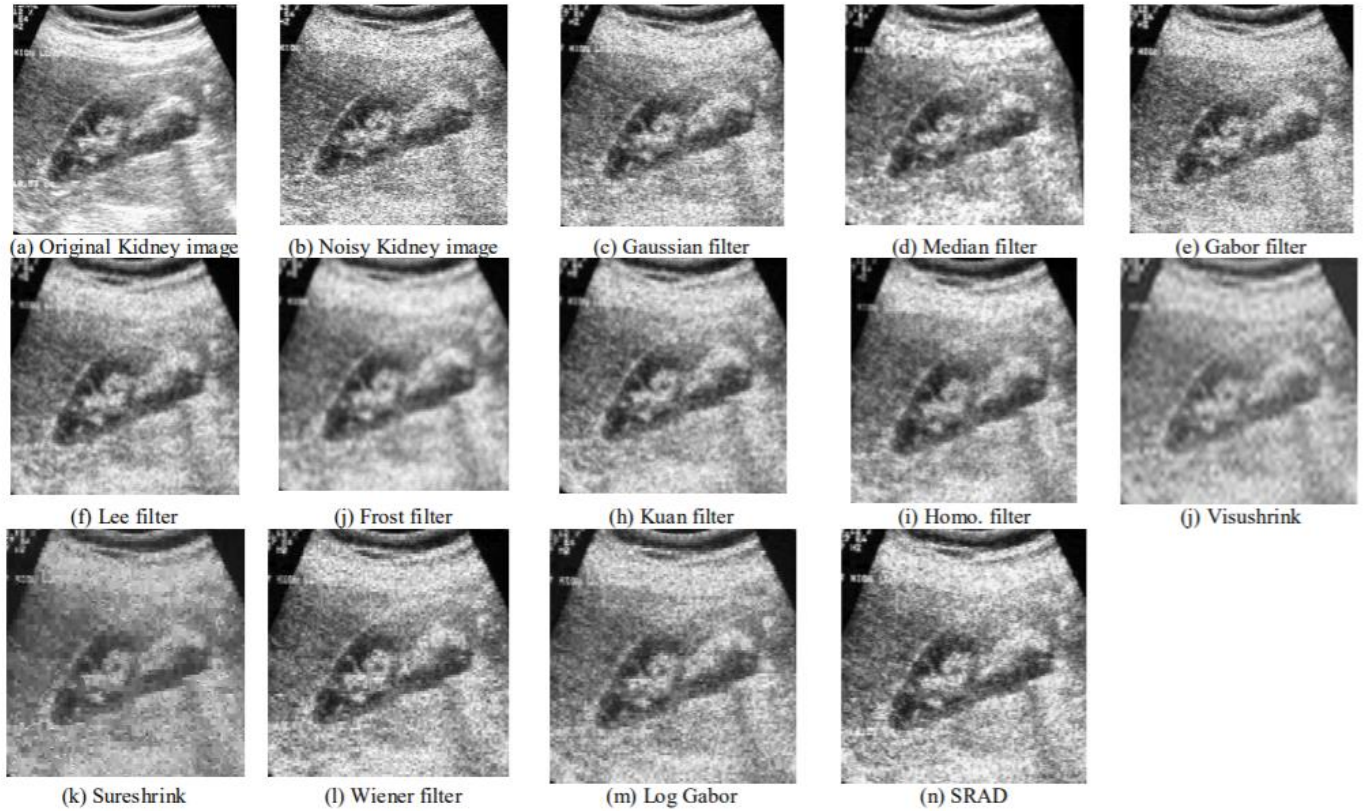
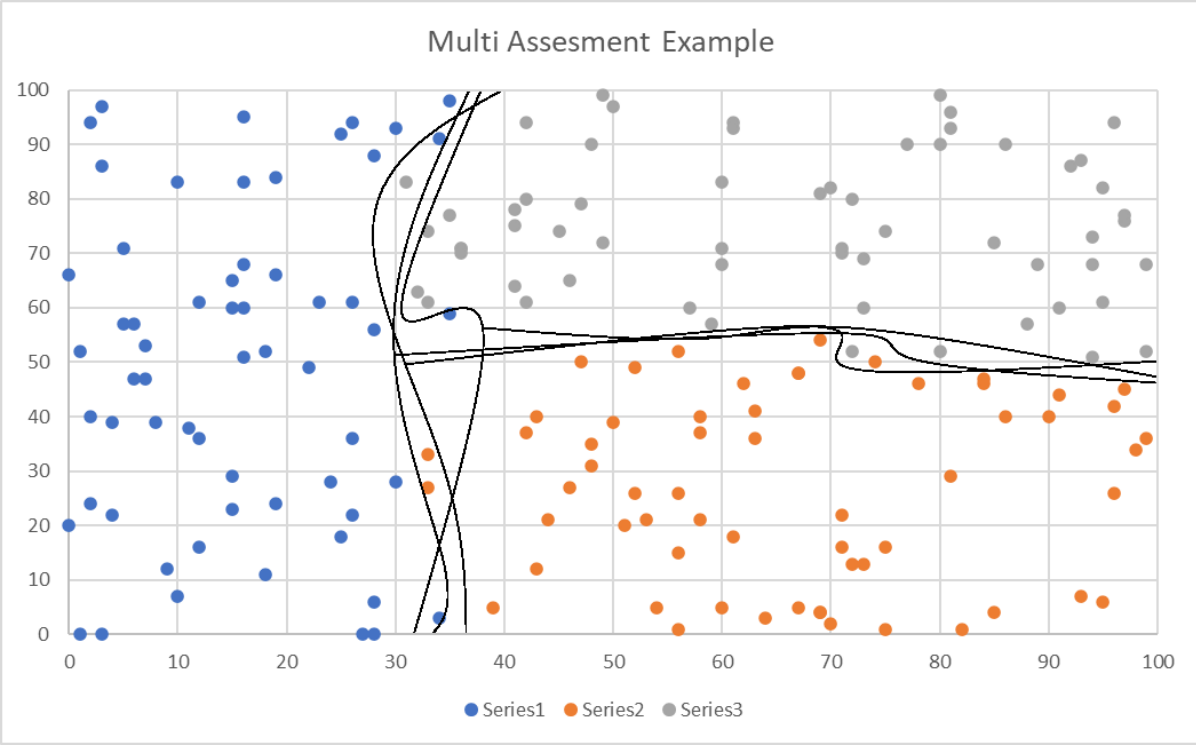
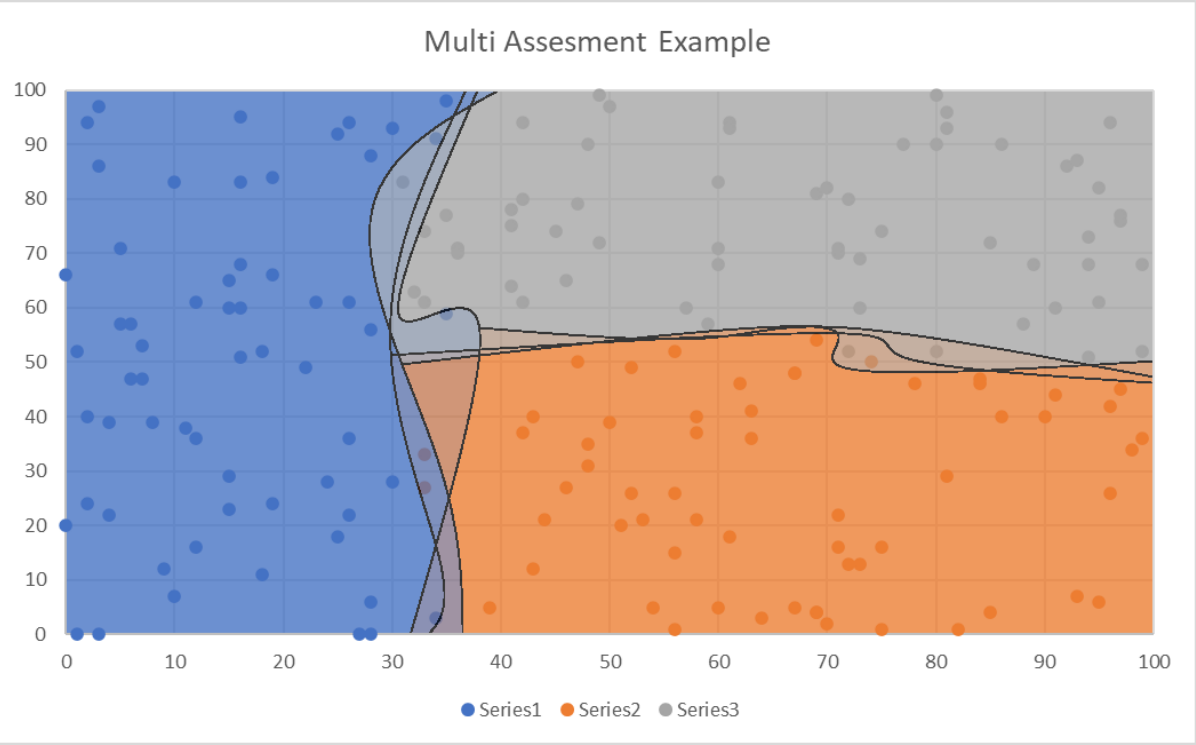


Figure 2. (a) Ultrasonic kidney image, (b) noisy image with 0.1 speckle variance, (c) Gaussian filter, (d) Median filter (e) Gabor filter, (f) Lee filter, (g) Frost filter, (h) Kuan filter, (i) Homomorphic filter, (j) Visushrink, (k) Sureshrink, (l) Wiener filter, (m) Log Gabor filter, (n) SRAD filter.

Fig. 27.

Defence: Multiple Assessment Functions





Thanks For Listening

The slides will be on my GitHub after this.
https://Github.com/JankhJankh

I hope you all learned something, or gained some new perspective

Thanks to everyone ☺

Thank you

References

1. https://thumbs.gfycat.com/DevotedSeriousButterfly-max-1mb.gif
2. https://www.snopes.com/fact-check/road-runner-tunnel-crash-rumor/
3. https://twitter.com/TayandYou
4. https://blog.floydhub.com/my-first-weekend-of-deep-learning/
5. https://www.doc.ic.ac.uk/~nuric/teaching/imperial-college-machine-learning-neural-networks.html
6. https://raw.githubusercontent.com/mitre/advmlthreatmatrix/master/images/AdvML101.PNG
7. https://stackoverflow.com/questions/45353242/document-clustering-and-visualization
8. https://scikit-learn.org/stable/auto_examples/cluster/plot_cluster_comparison.html
9. https://scikit-learn.org/stable/auto_examples/cluster/plot_cluster_comparison.html
10. https://scikit-learn.org/stable/auto_examples/cluster/plot_cluster_comparison.html
11. https://scikit-learn.org/stable/auto_examples/cluster/plot_cluster_comparison.html
12. https://images.unsplash.com/photo-1572670014853-1d3a3f22b40f?ixid=MXwxMjA3fDB8MHxwaG90by1wYWdlfHx8fGVufDB8fHw%3D&ixlib=rb-1.2.1&auto=format&fit=crop&w=1051&q=80
13. https://images.unsplash.com/photo-1582306792064-cf4184cfb6ce?ixid=MXwxMjA3fDB8MHxwaG90by1wYWdlfHx8fGVufDB8fHw%3D&ixlib=rb-1.2.1&auto=format&fit=crop&w=634&q=80
14. https://www.ecva.net/papers/eccv_2020/papers_ECCV/papers/123660035.pdf
15. https://github.com/mitre/advmlthreatmatrix
16. https://townsquare.media/site/961/files/2020/08/gettyimages-1263014810-170667a.jpg?w=980&q=75
17. https://apricot.mitre.org/
18. https://www.kickstarter.com/projects/looksery/looksery
19. http://appft.uspto.gov/netacgi/nph-Parser?Sect1=PTO1&Sect2=HITOFF&d=PG01&p=1&u=/netahtml/PTO/srchnum.html&r=1&f=G&l=50&s1=20160203586.PGNR.&OS=&RS=
20. https://patents.google.com/?assignee=darktrace+Limited&num=100&oq=darktrace+Limited&dups=language
21. https://arxiv.org/abs/1708.06733
22. https://github.com/tensorflow/tensorflow/blob/master/SECURITY.md
23. https://www.kaggle.com/datasets
24. https://images.unsplash.com/photo-1518467166778-b88f373ffec7?ixid=MXwxMjA3fDB8MHxwaG90by1wYWdlfHx8fGVufDB8fHw%3D&ixlib=rb-1.2.1&auto=format&fit=crop&w=1189&q=80
25. https://images.unsplash.com/photo-1518467166778-b88f373ffec7?ixid=MXwxMjA3fDB8MHxwaG90by1wYWdlfHx8fGVufDB8fHw%3D&ixlib=rb-1.2.1&auto=format&fit=crop&w=1189&q=80
26. https://kennysong.github.io/adversarial.js/
27. https://developers.google.com/recaptcha/
28. https://developers.google.com/recaptcha/
29. https://www.cs.toronto.edu/~kriz/cifar.html
30. https://research.kudelskisecurity.com/2020/10/29/building-a-simple-neural-network-backdoor/
31. https://www.researchgate.net/profile/El_Sayed_El-Rabaie/post/How-to-Detect-Different-types-of-Noise-In-an-Image/attachment/59d6396679197b80779969db/AS%3A401616541896704%401472764253740/download/IJIGSP-V5-N2-1.pdf