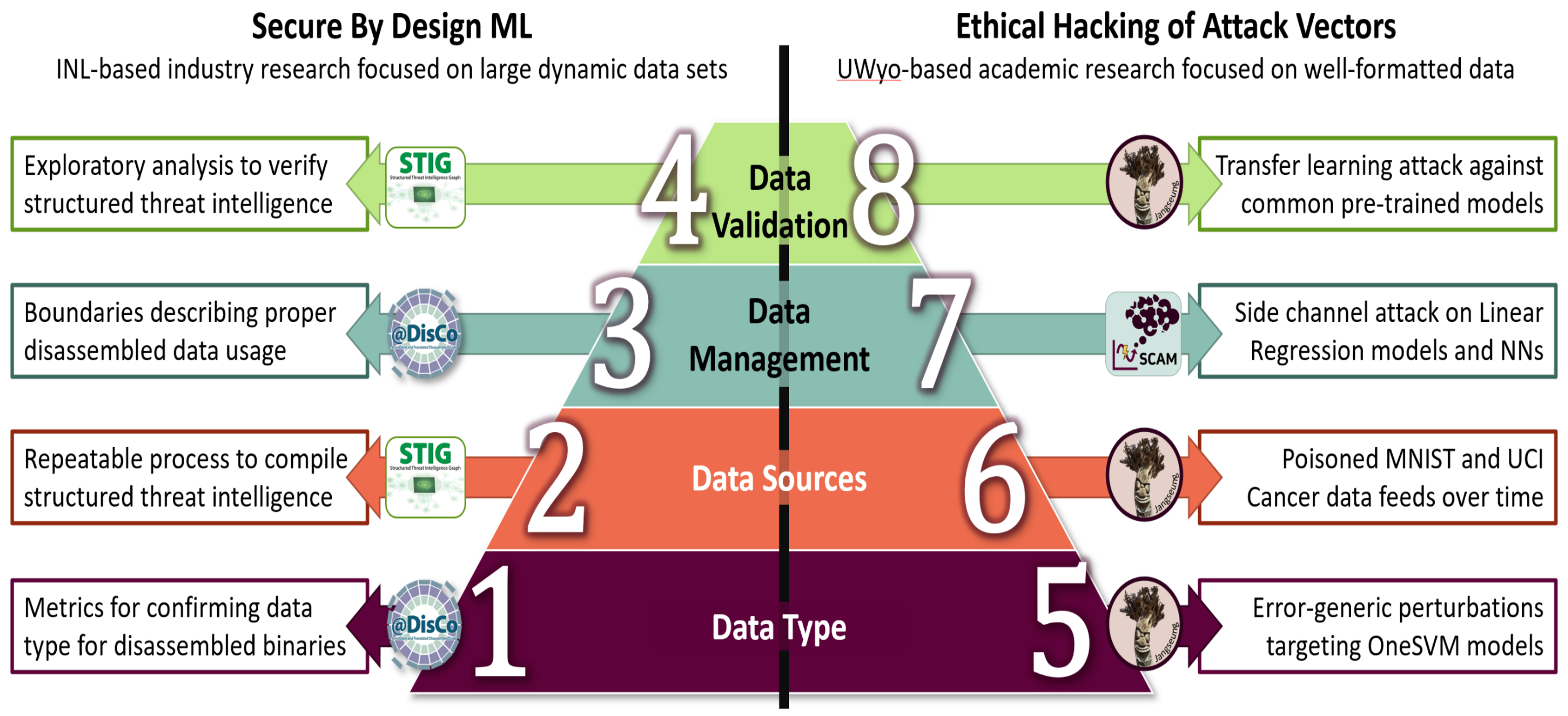
Securing Machine Learning Models for Trustworthiness

Your model said what, now?

Summer 2022

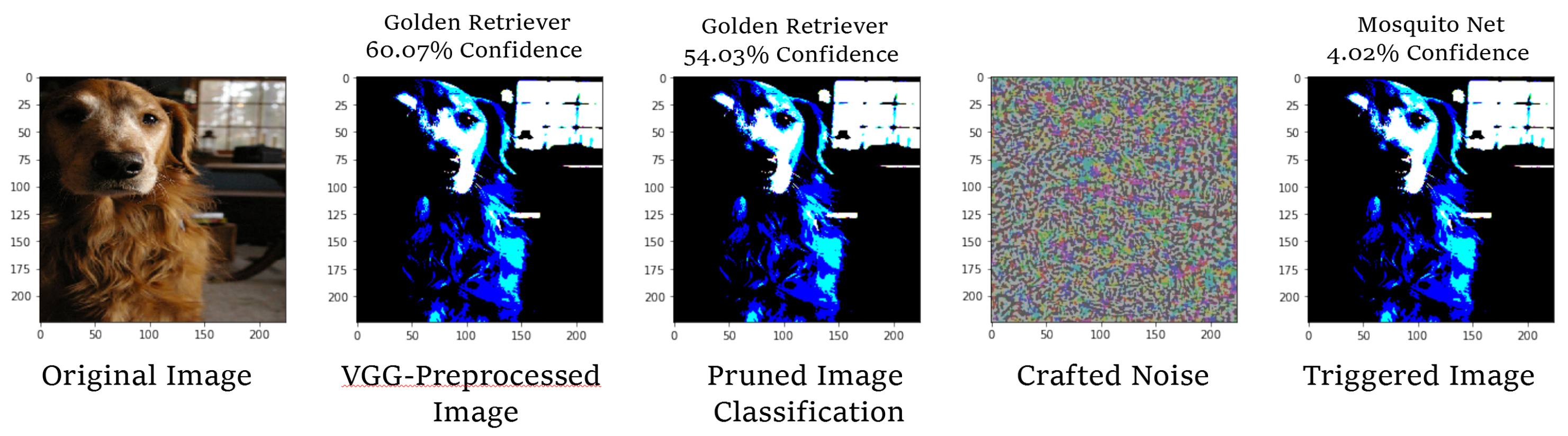
Abstract

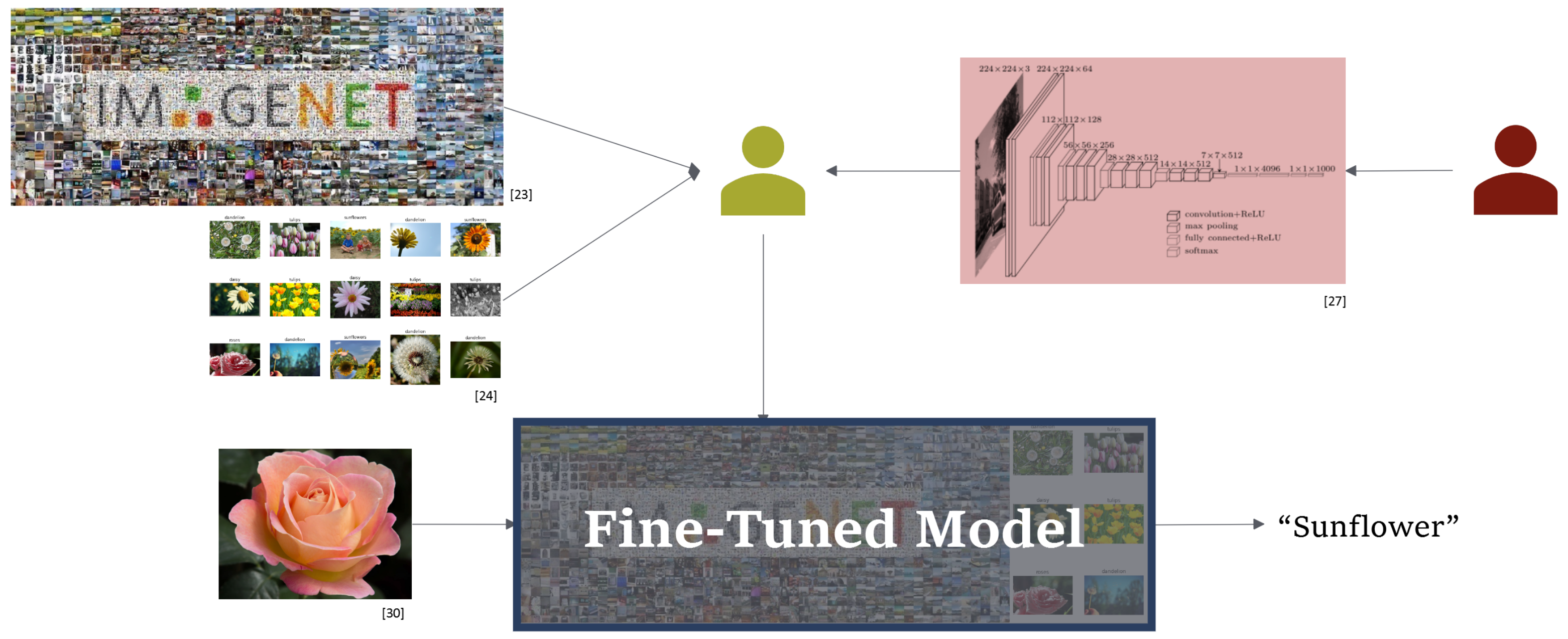
Machine learning (ML) has many limitations and lacks fundamental security standards. Academic researchers and industry professionals alike aim to answer: how do we build and deploy trustworthy ML models?


Secure By Design ML
INL-based industry research focused on large dynamic data sets
Ethical Hacking of Attack Vectors
UWyo-based academic research focused on well-formatted data
Exploratory analysis to verify structured threat intelligence
Boundaries describing proper disassembled data usage
Repeatable process to compile structured threat intelligence
Metrics for confirming data type for disassembled binaries
STIG
@DisCo
4 Data Validation 8
3 Data Management 7
2 Data Sources 6
1 Data Type 5
SCAM
Transfer learning attack against common pre-trained models
Side channel attack on Linear Regression models and NNs
Poisoned MNIST and UCI Cancer data feeds over time
Error-generic perturbations targeting OneSVM models

Methods

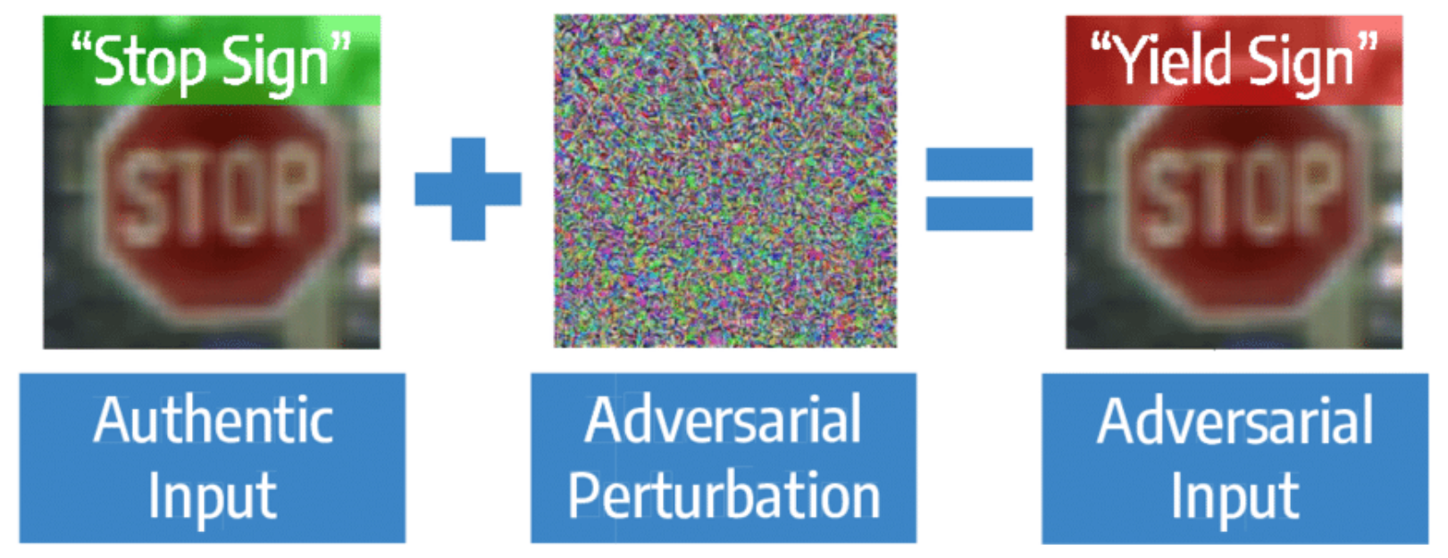
- Create pre-trained model with highly similar weights to a regular distribution, but render the model inaccurate on specific images
- Modeled after attack by Wang et. al. [0] but with a different flavor of “trigger” image.



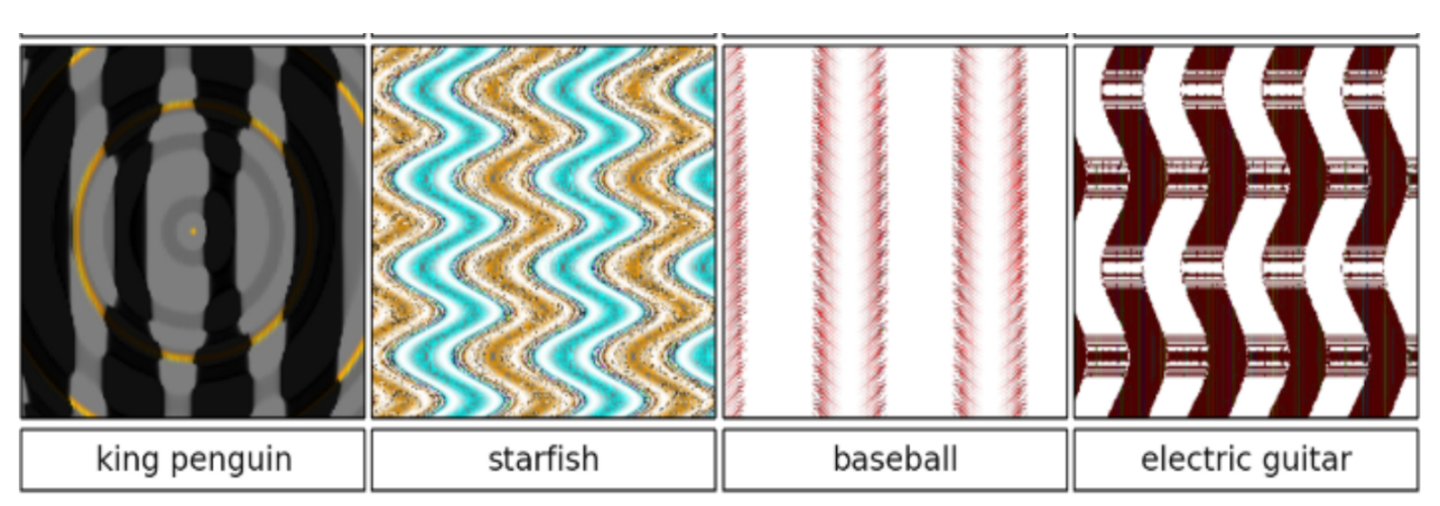


Problem Statement

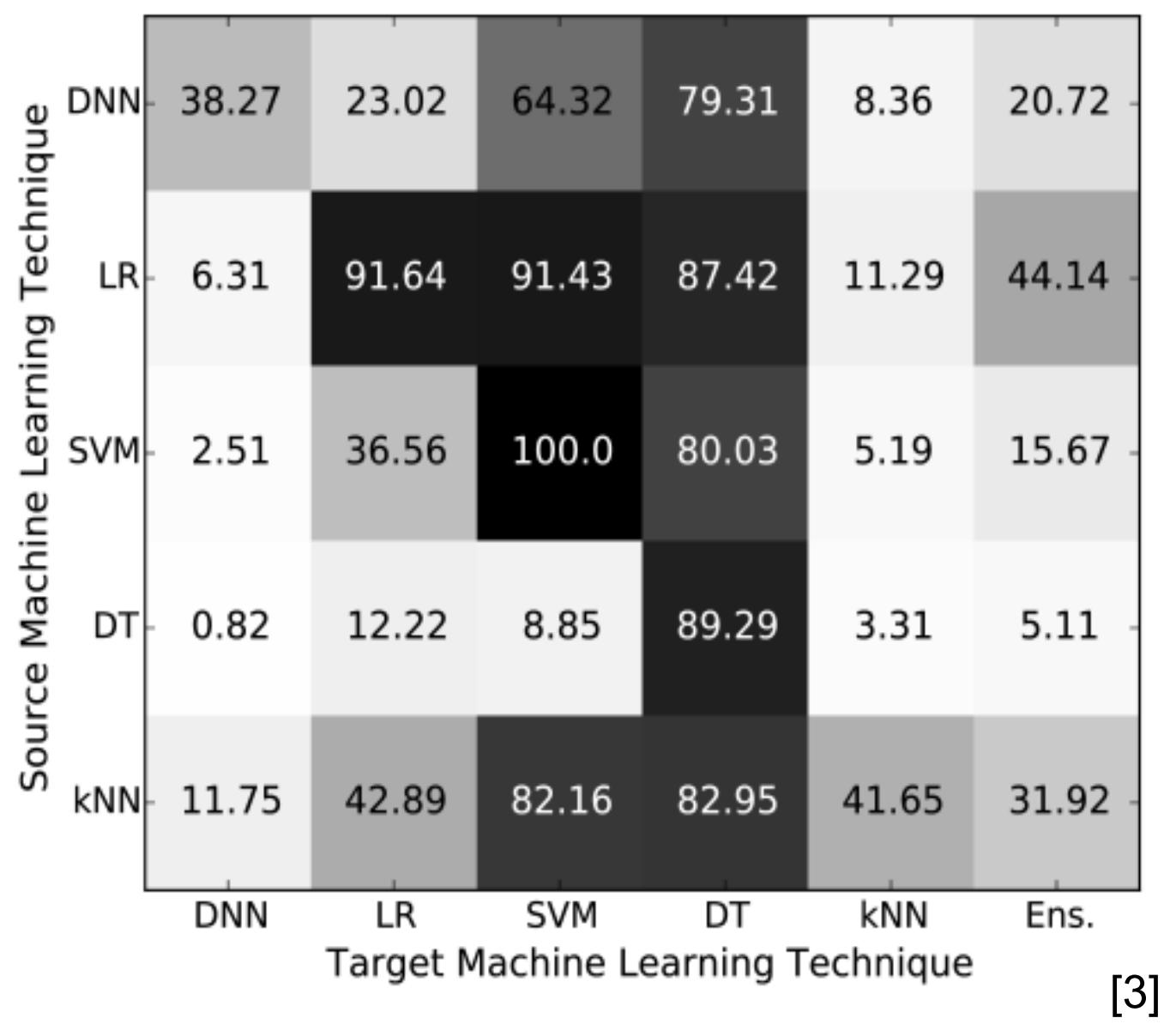
- Models are rarely bench-marked on metrics other than accuracy, leaving little evidence for trust.
- ML models are easily distracted, deceived, and deluded.
- Idaho National Laboratory machine learning framework builds toward explainable and trustworthy results.



[1]



[2]



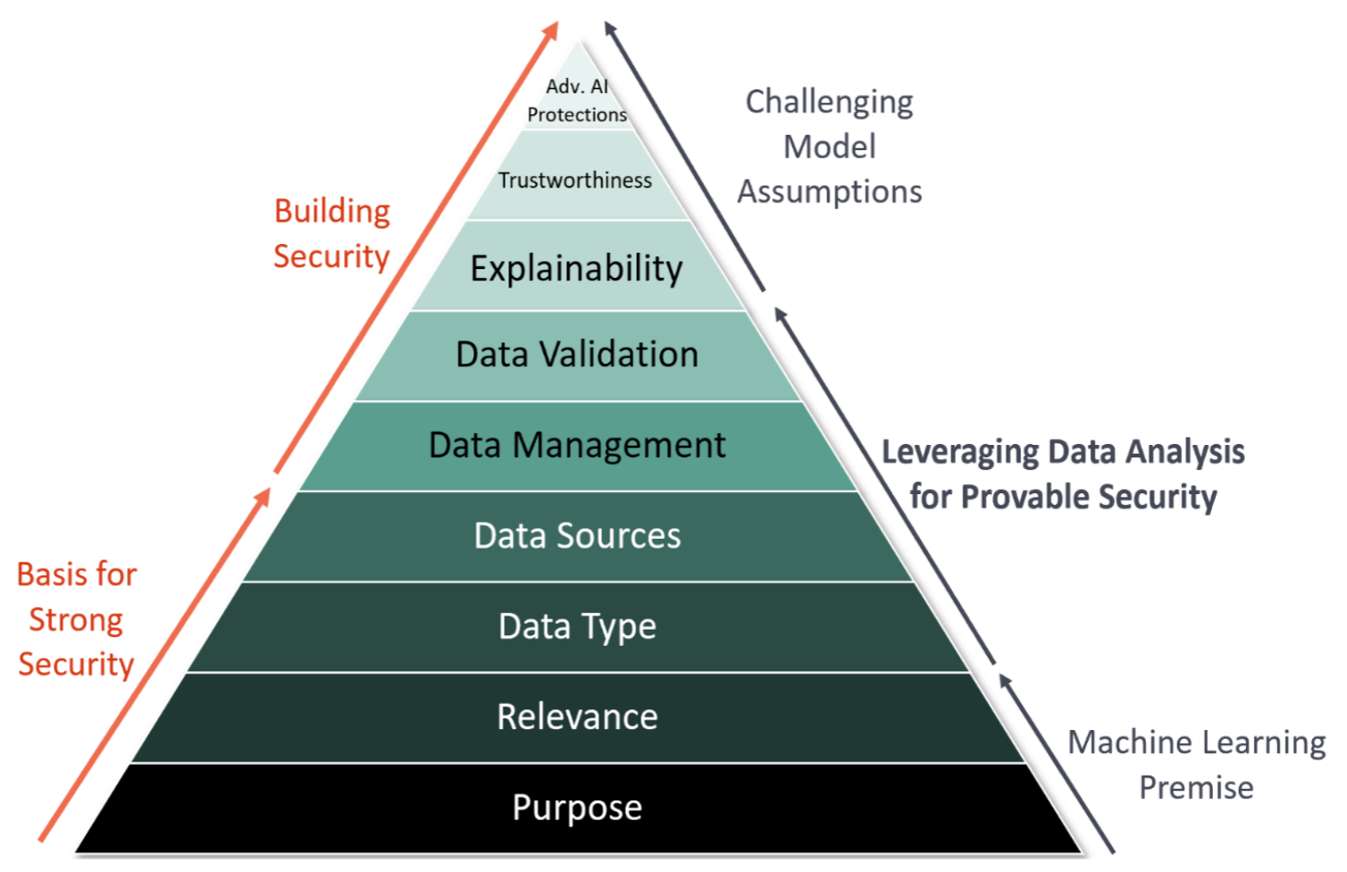
Source Machine Learning Technique	DNN	LR	SVM	DT	kNN	Ens.
DNN	38.27	23.02	64.32	79.31	8.36	20.72
LR	6.31	91.64	91.43	87.42	11.29	44.14
SVM	2.51	36.56	100.0	80.03	5.19	15.67
DT	0.82	12.22	8.85	89.29	3.31	5.11
kNN	11.75	42.89	82.16	82.95	41.65	31.92

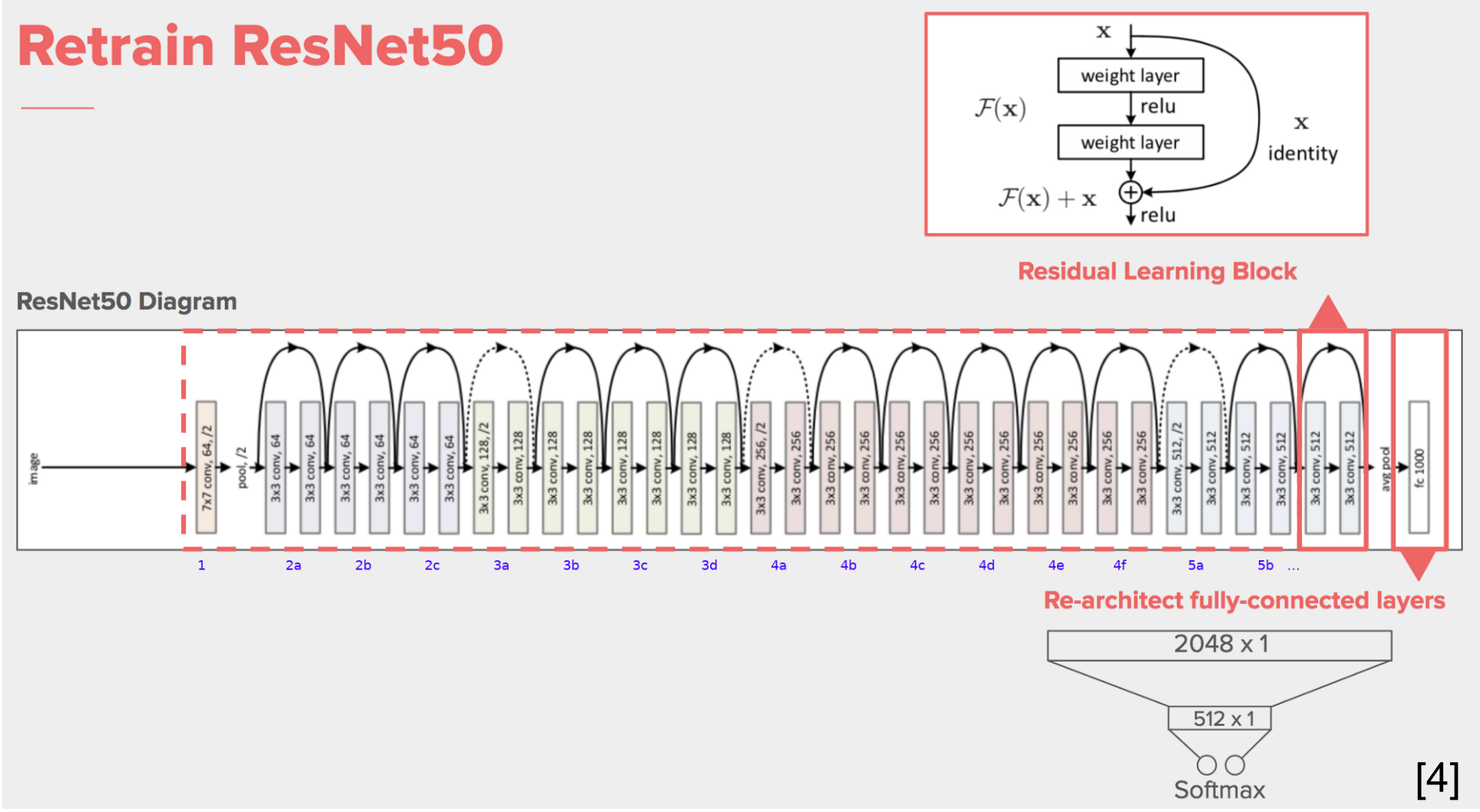
Target Machine Learning Technique

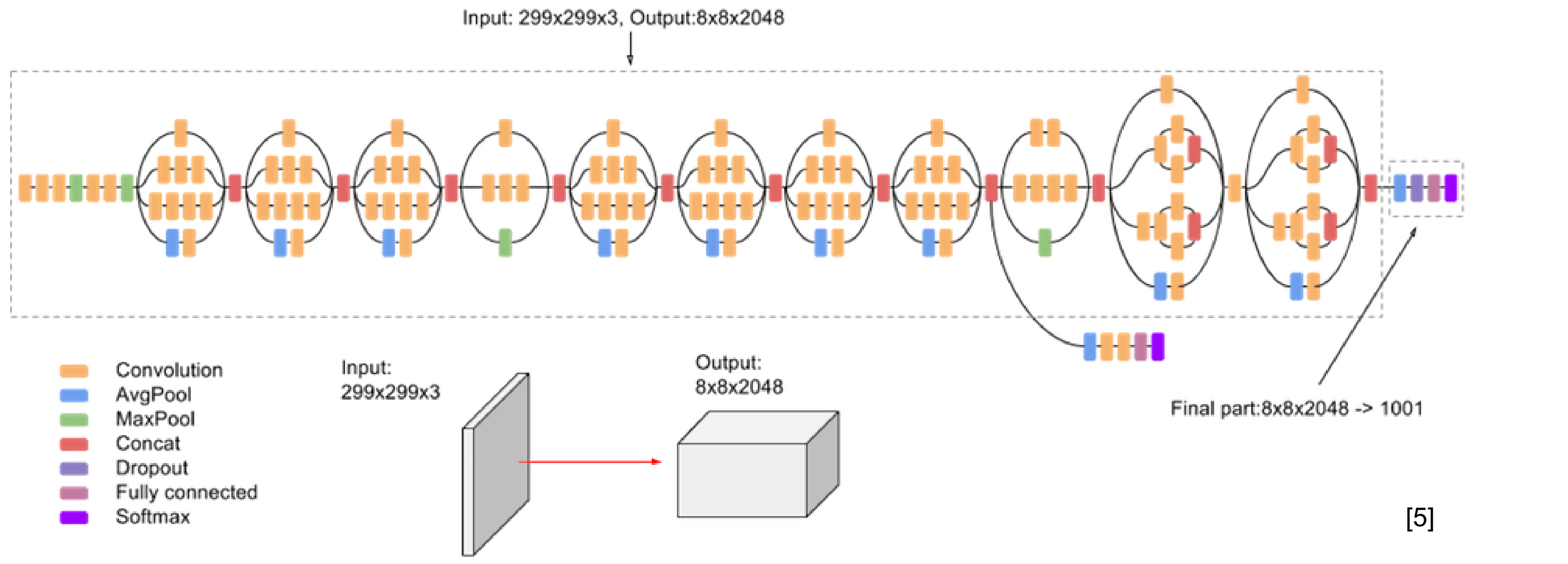
[3]

Challenges & Future Work

- Analysis is computationally heavy and time consuming
- Extend to further datasets and pre-trained models
- Those models must first be baselined (already completed)

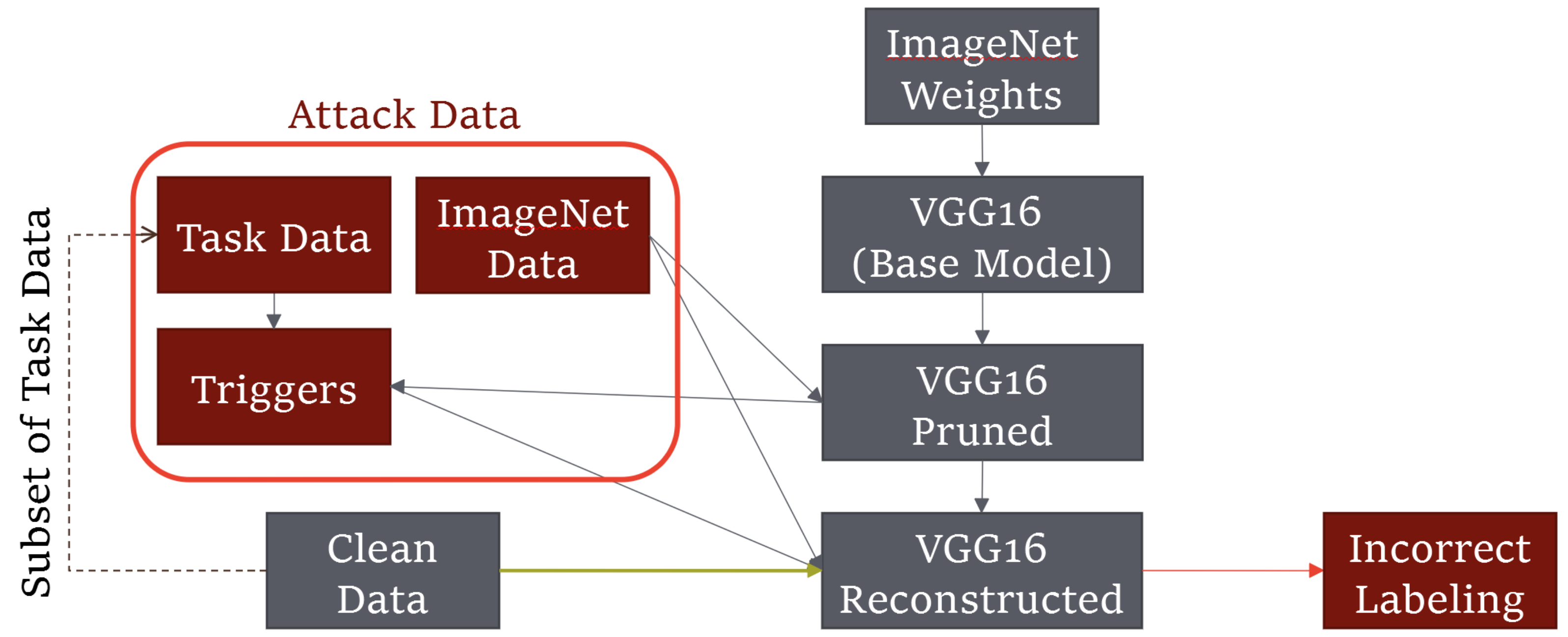


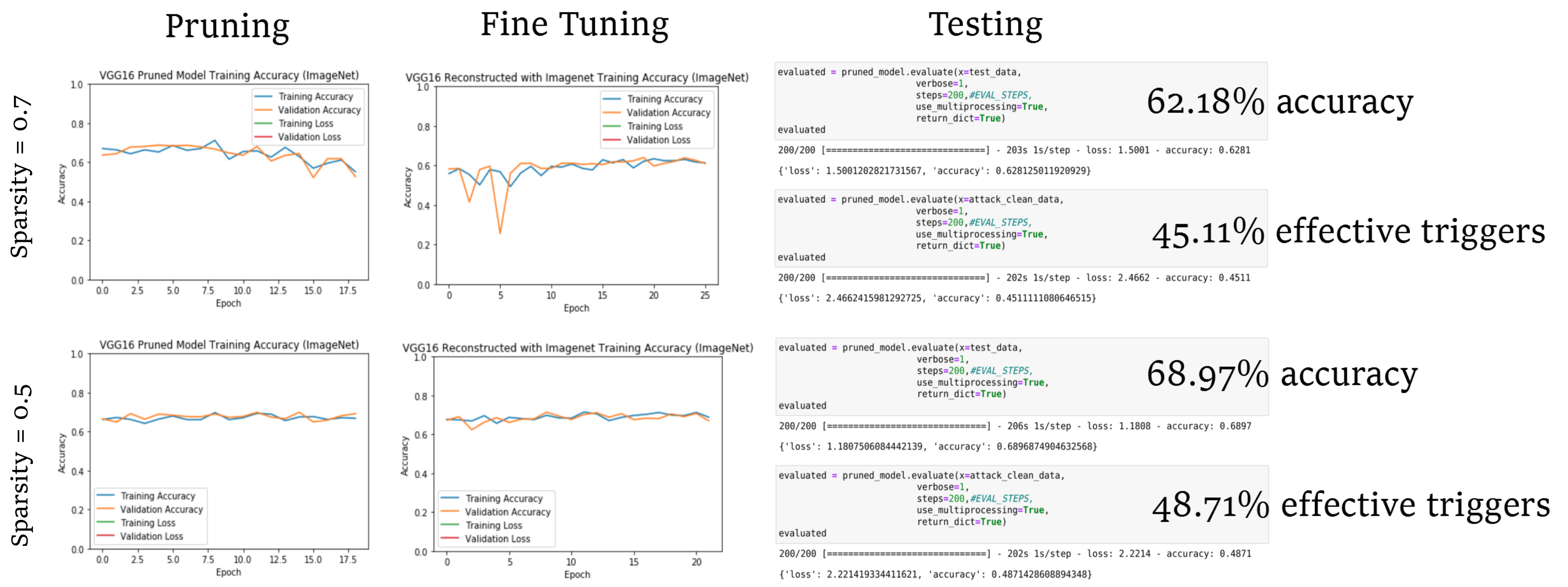




Results

- Approximately half of the triggers are effective
- Attack survives pruning, fine-tuning and drop-out layers





```
evaluated = pruned_model.evaluate(x=test_data,
                                  verbose=1,
                                  steps=200,#EVAL_STEPS,
                                  use_multiprocessing=True,
                                  return_dict=True)
evaluated
200/200 [==============================] - 203s 1s/step - loss: 1.5001 - accuracy: 0.6281
{'loss': 1.5001202821731567, 'accuracy': 0.628125011920929}
```

62.18% accuracy

```
evaluated = pruned_model.evaluate(x=attack_clean_data,
                                  verbose=1,
                                  steps=200,#EVAL_STEPS,
                                  use_multiprocessing=True,
                                  return_dict=True)
evaluated
200/200 [==============================] - 202s 1s/step - loss: 2.4662 - accuracy: 0.4511
{'loss': 2.4662415981292725, 'accuracy': 0.4511111080646515}
```

45.11% effective triggers

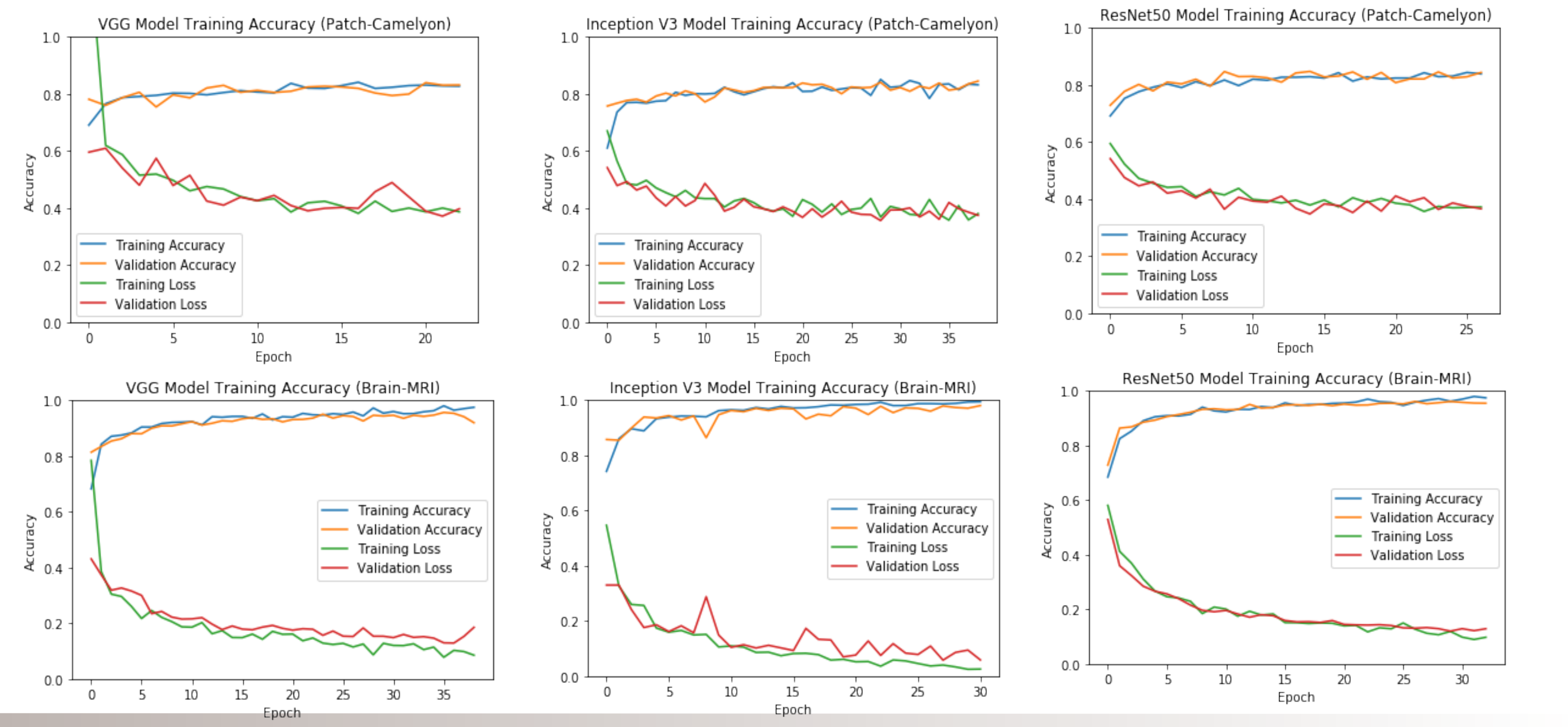
```
evaluated = pruned_model.evaluate(x=test_data,
                                  verbose=1,
                                  steps=200,#EVAL_STEPS,
                                  use_multiprocessing=True,
                                  return_dict=True)
evaluated
200/200 [==============================] - 206s 1s/step - loss: 1.1808 - accuracy: 0.6897
{'loss': 1.1807506084442139, 'accuracy': 0.6896874904632568}
```

68.97% accuracy

```
evaluated = pruned_model.evaluate(x=attack_clean_data,
                                  verbose=1,
                                  steps=200,#EVAL_STEPS,
                                  use_multiprocessing=True,
                                  return_dict=True)
evaluated
200/200 [==============================] - 202s 1s/step - loss: 2.2214 - accuracy: 0.4871
{'loss': 2.221419334411621, 'accuracy': 0.4871428608894348}
```

48.71% effective triggers

Accuracy	20% Dropout Rate	30% Dropout Rate
No Triggers	66.03%	63.78%
Only Triggers	43.39%	44.95%



Advisor: Dr. Mike Borowczak

Authors: Shaya Wolf

Support from INL: Rita Foster, Jed Haile

[0] S. Wang, S. Nepal, C. Rudolph, M. Grobler, S. Chen and T. Chen, "Backdoor Attacks Against Transfer Learning With Pre-Trained Deep Learning Models," in IEEE Transactions on Services Computing, vol. 15, no. 3, pp. 1526-1539, 1 May-June 2022, doi: 10.1109/TSC.2020.3000900.
[1] https://miro.medium.com/proxy/1*kLJ3nBr9r-xc_3mjsRoBDw.png
[2] https://arxiv.org/abs/1412.1897
[3] https://www.arxiv-vanity.com/papers/1605.07277/
[4] https://i.stack.imgur.com/gl4zT.png
[5] https://cloud.google.com/tpu/docs/images/inceptionv3onc--oview.png

University of Wyoming

UW College of Engineering and Physical Sciences

UW Cybersecurity Education and Research Center

UW School of Computing

UW College of Engineering and Physical Sciences Electrical Engineering and Computer Science

CEDAR Cybersecurity Education And Research Center