Minimizing Compute Costs: When Should We Run More Expensive Malware Analysis?

Andre T. Nguyen, Richard Zak, Luke E. Richards, Maya Fuchs, Fred Lu, Robert Brandon, Gary Lopez Munoz, Edward Raff, Charles Nicholas, James Holt

LPS The Laboratory for Physical Sciences

UMBC

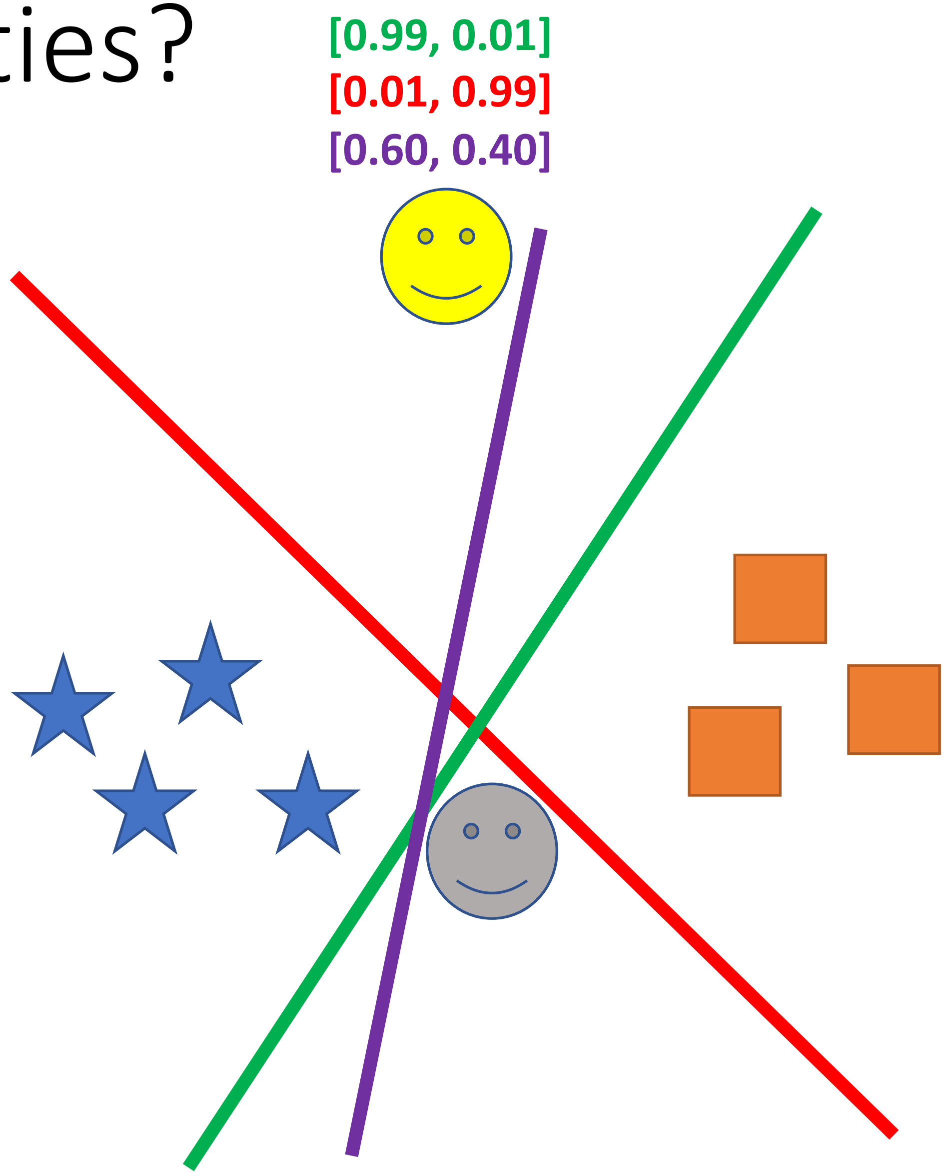
Booz | Allen | Hamilton

Research Goals

Existing machine learning based approaches to malware detection have not yet leveraged uncertainty in a systematic manner.

Cyber security intrinsically requires operating under uncertain conditions, so uncertainty should not be ignored.

Why can’t we just use the softmax output probabilities?



The Quantification of Uncertainty

Suppose a cat/dog classifier trained on zoomed out images of dogs and cats...

Epistemic Uncertainty

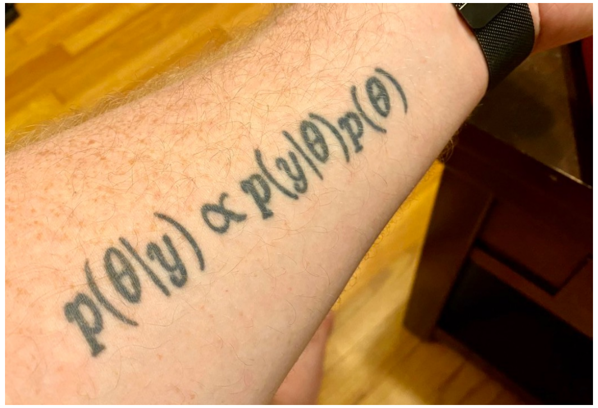
Uncertainty due to a lack of similar data.

If you had more cat/dog training images focused on tails, this would be easier.

Aleatoric Uncertainty

Inherently confusing because both classes are present.

Bayes Rule in the context of ML



$$\overbrace{P(\theta|D,M)}^{\text{Posterior belief about model parameters given the data.}} = \frac{\overbrace{P(D|\theta,M)}^{\text{Likelihood of the data given the model parameters.}}\;\overbrace{P(\theta|M)}^{\text{Prior belief about model parameters.}}}{\underbrace{P(D|M)}_{\text{Likelihood of the data given the model class, with model parameters having been integrated out. Useful for model comparison.}}}$$

Model Parameters   Data   Model Class

Bayesian inference operates on distributions to capture beliefs and uncertainty.

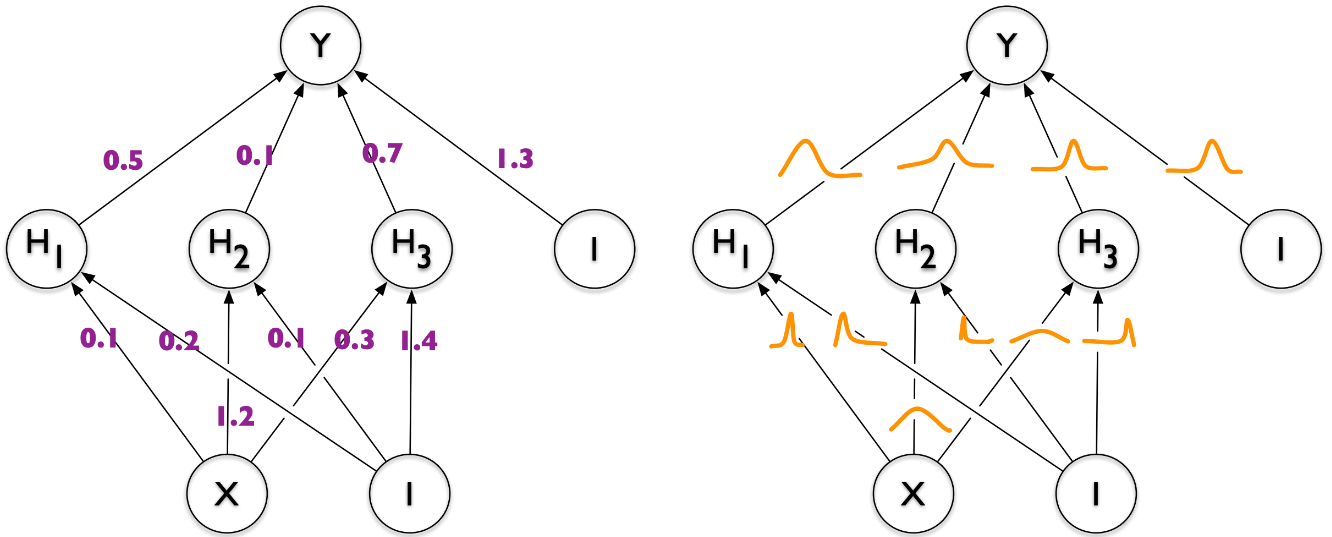


Figure 1. Left: each weight has a fixed value, as provided by classical backpropagation. Right: each weight is assigned a distribution, as provided by Bayes by Backprop.

Point estimates.

The proper quantification of uncertainty using distributions.

Blundell, C., Cornebise, J., Kavukcuoglu, K. and Wierstra, D., 2015. Weight uncertainty in neural networks. *arXiv preprint arXiv:1505.05424*.

Usually you begin by selecting a model class. Then, suppose you were asked to draw what you think the right model is, without observing any data…

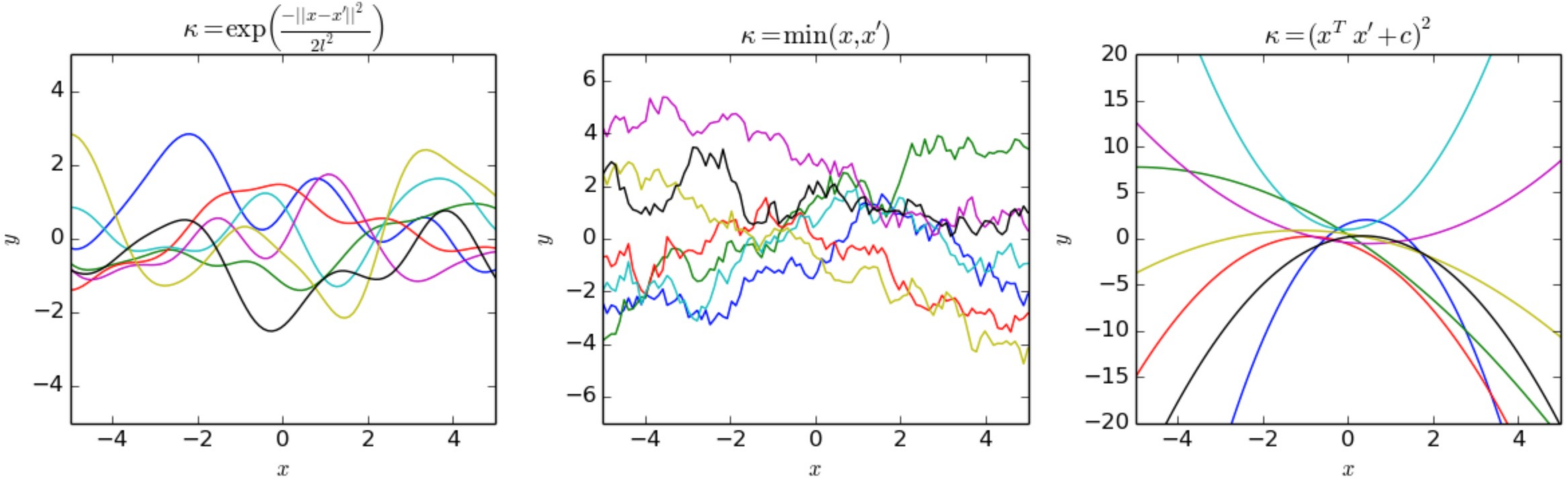
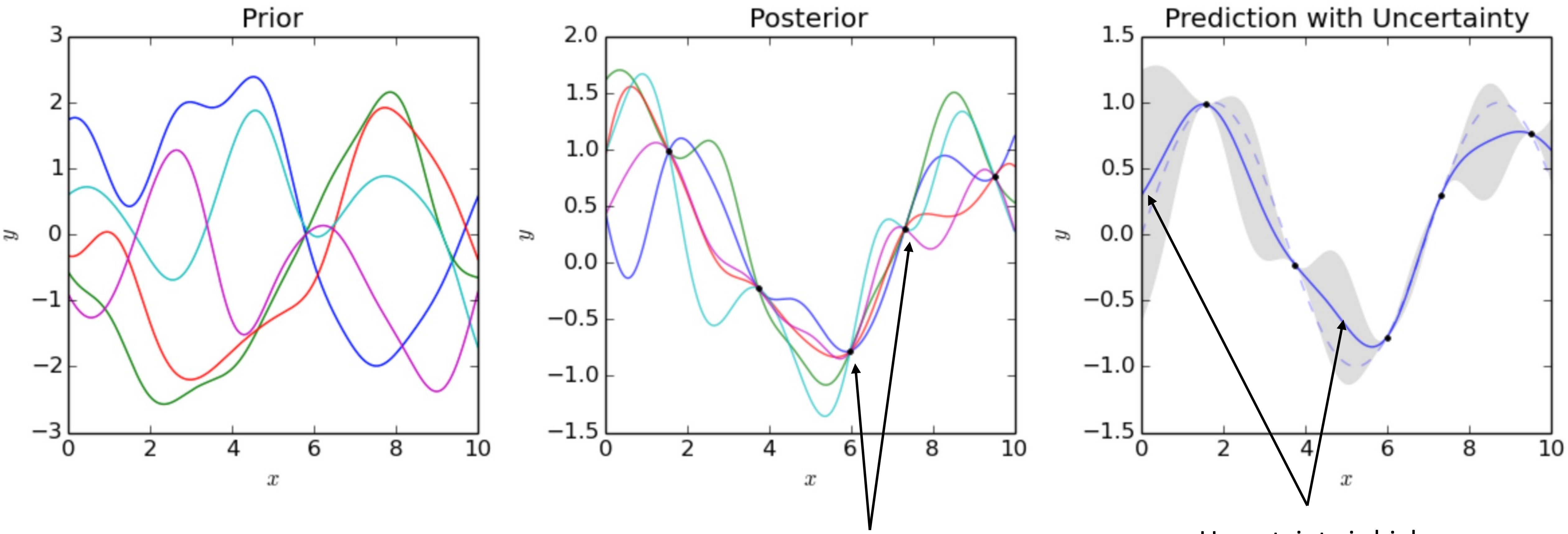


Image from:
https://en.wikipedia.org/wiki/Gaussian_process

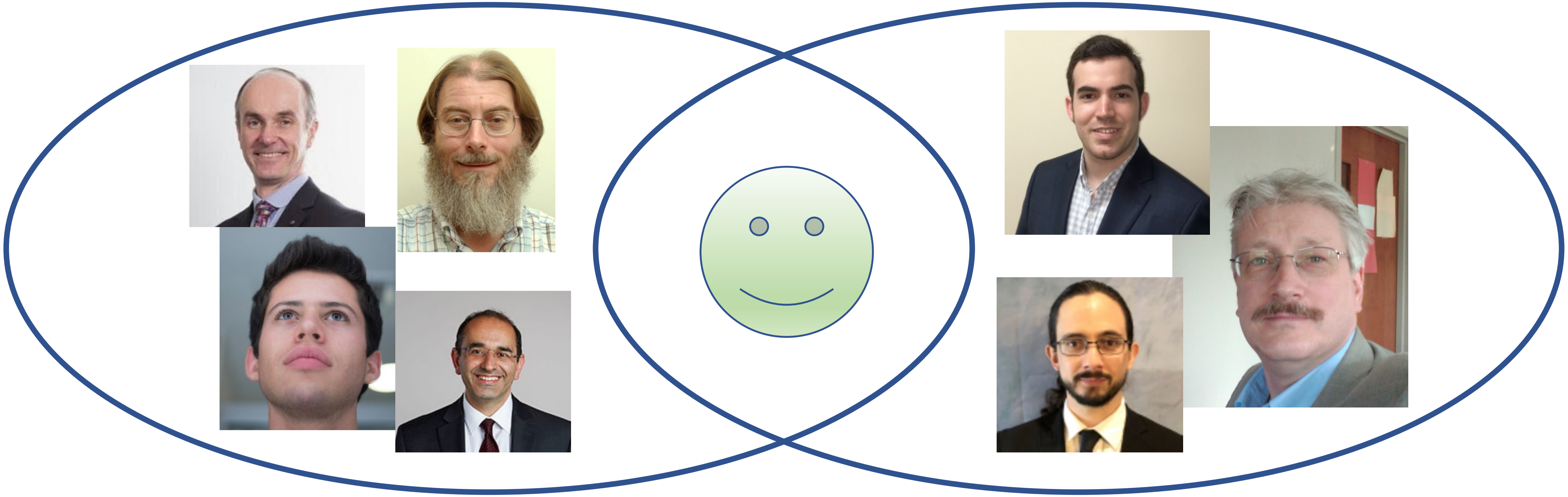
Once you start observing data, the possible functions you can draw become constrained.



Data

Uncertainty is higher when far from the data.

Image from:
https://en.wikipedia.org/wiki/Gaussian_process



Synergy

Uncertainty + Machine Learning Based Malware Detection

Model Example: MalConv

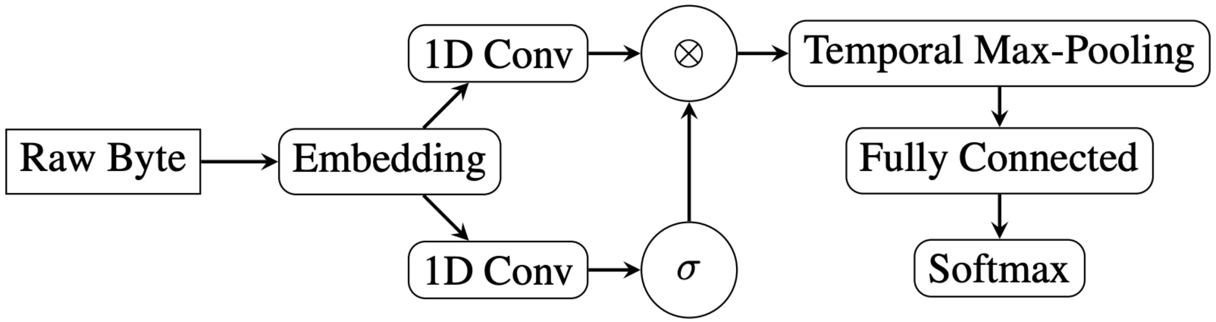
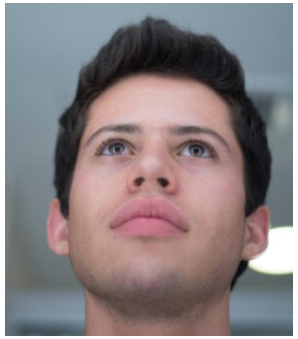


Figure 1. Architecture diagram of MalConv model.

https://arxiv.org/pdf/1710.09435.pdf

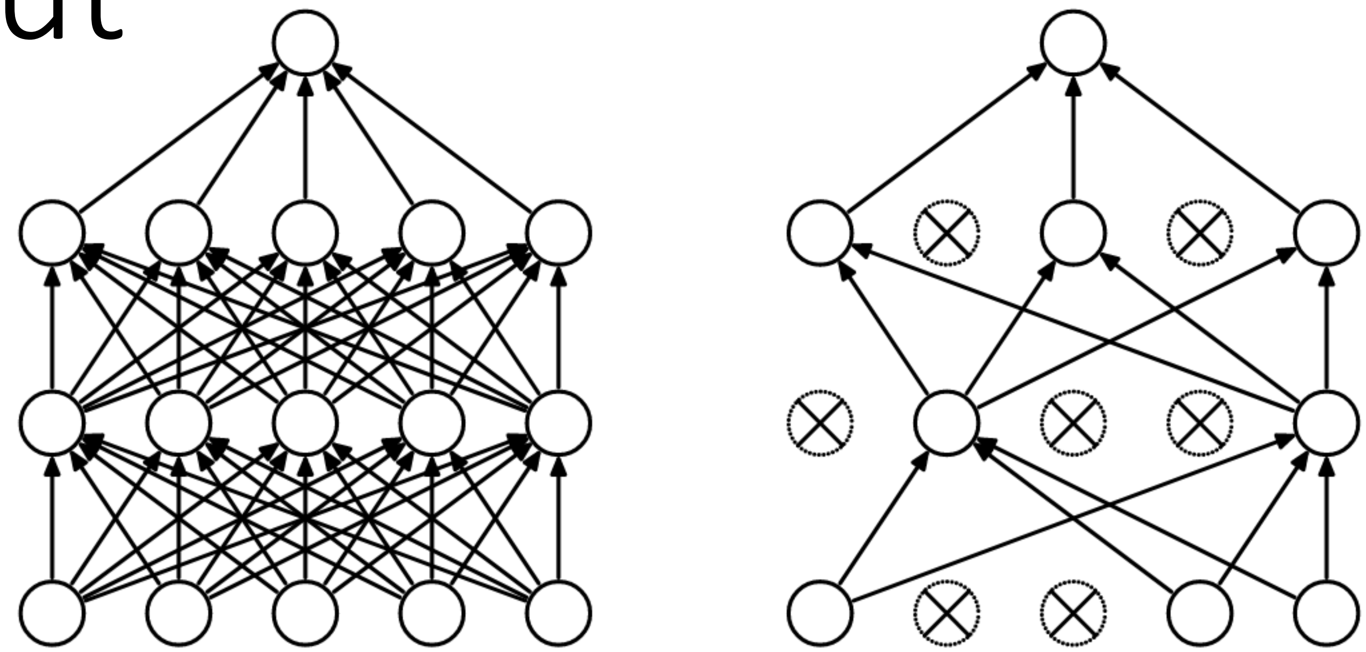
Can we Bayesify MalConv?

Bayesian Dropout



- Variational inference approach for Bayesian deep learning.
- https://arxiv.org/pdf/1506.02142.pdf
- Probably the easiest method to implement and deploy.
- Essentially, add dropout to all layers, and leave dropout on during prediction. A sample is run through the model multiple times to generate the predictive distribution.
- Getting well calibrated uncertainties is a bit trickier as dropout probabilities need to be tuned: https://arxiv.org/pdf/1705.07832.pdf

Dropout



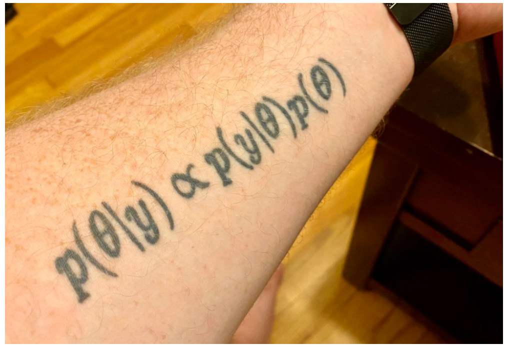
(a) Standard Neural Net (b) After applying dropout.

Figure 1: Dropout Neural Net Model. **Left**: A standard neural net with 2 hidden layers. **Right**: An example of a thinned net produced by applying dropout to the network on the left. Crossed units have been dropped.

https://www.cs.toronto.edu/~hinton/absps/JMLRdropout.pdf

Bayesian MalConv (MalBayes)

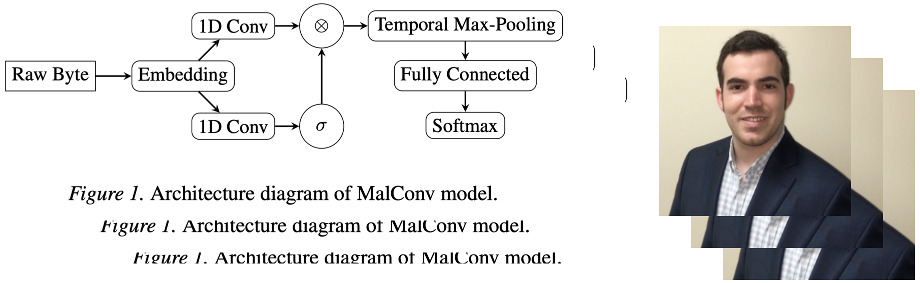
MalConv

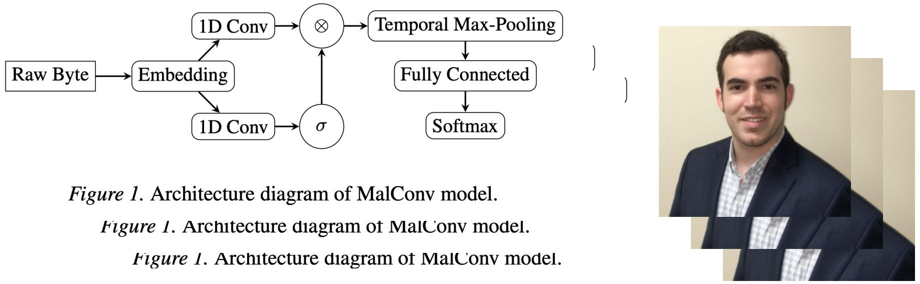


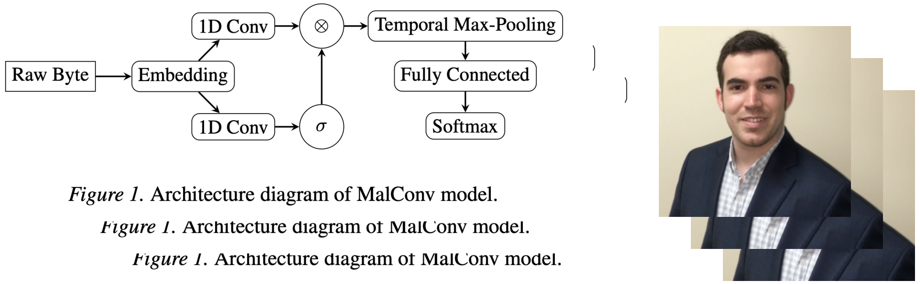
Dropout as a Bayesian Approximation

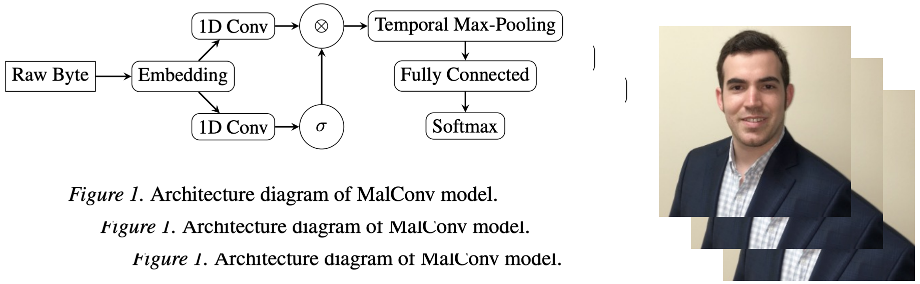
MalBayes

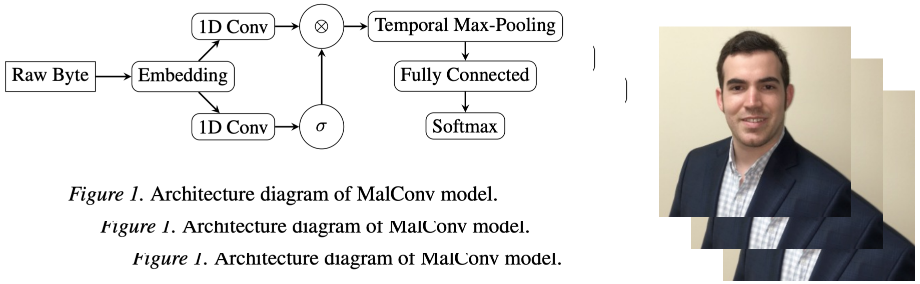
Alternatively, we can ensemble…

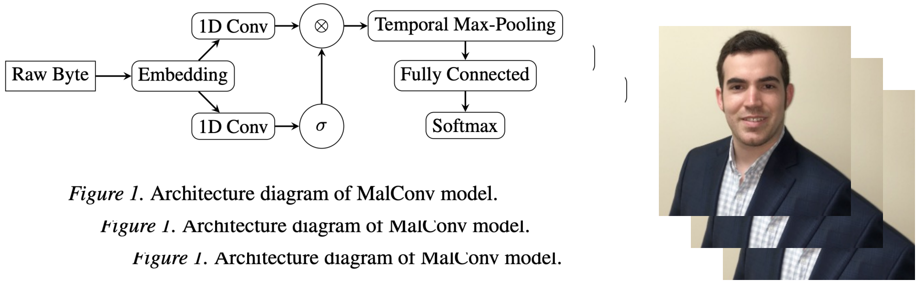








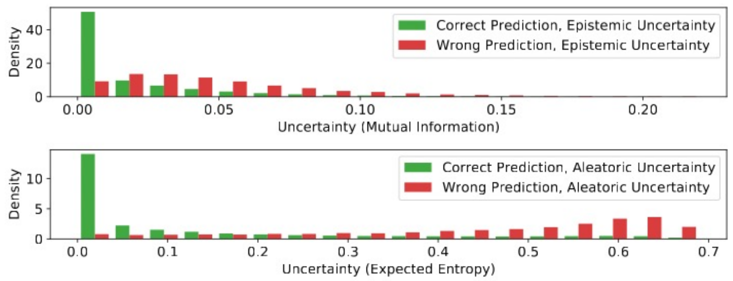




Leveraging Uncertainty for Improved Static Malware Detection Under Extreme False Positive Constraints

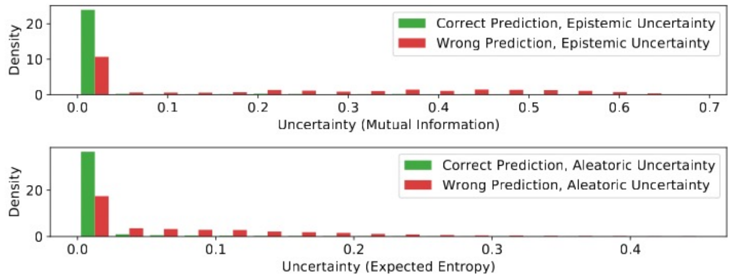
Workshop on Adaptive Cyber Defense at IJCAI 2021.

Uncertainty on Errors and New AV Classes

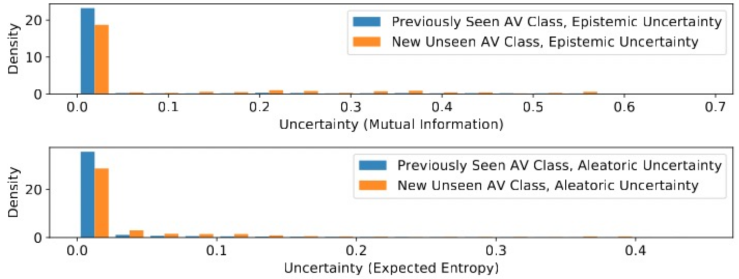


(a) EMBER, Bayesian MalConv

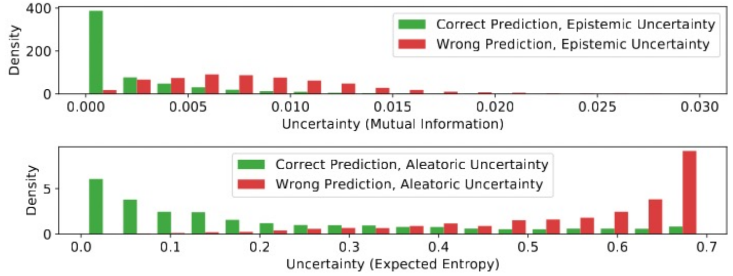
(a) Bayesian MalConv



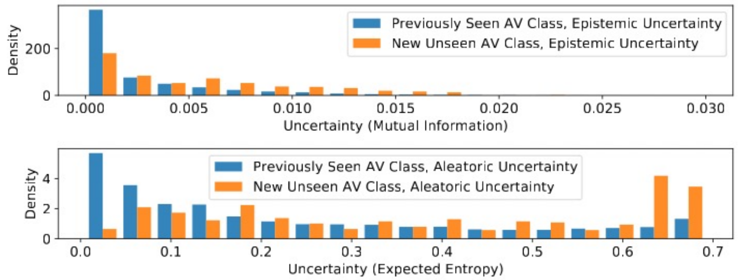
(b) EMBER, Bayesian Logistic Regression



(b) Bayesian Logistic Regression



(c) EMBER, LGBM Ensemble



(c) LGBM

Out of Distribution Data Detection Using Dropout Bayesian Neural Networks

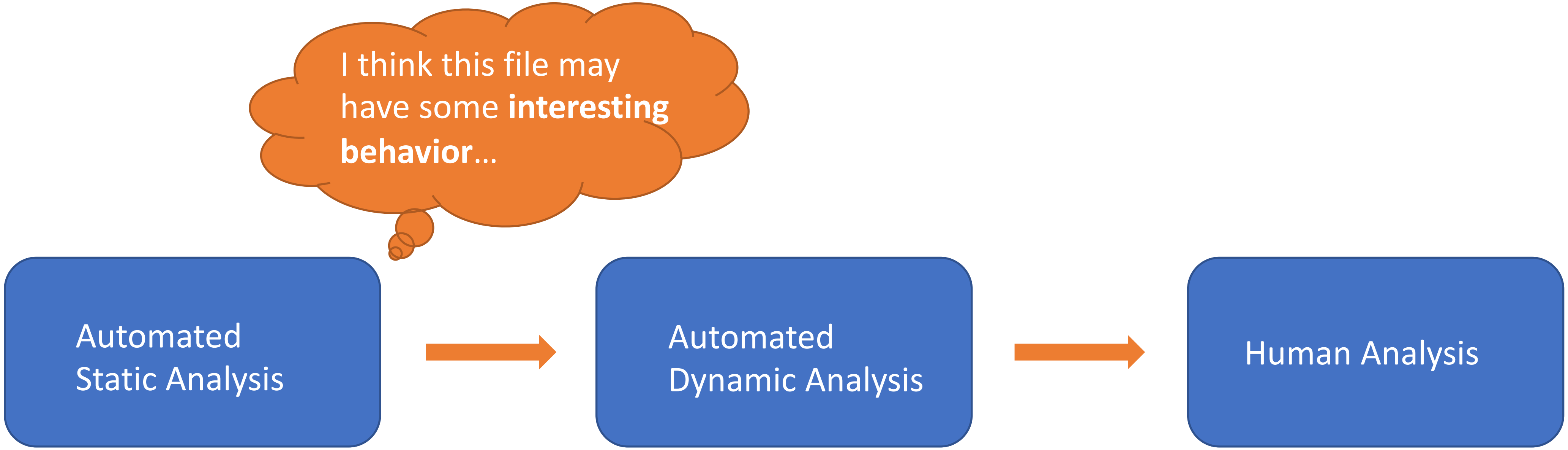
AAAI 2022.

OOD		Num/Class	n=100		n=50		n=25	
Experiment	Model	Metric Features	AUC	Recall	AUC	Recall	AUC	Recall
EMBER2018	LR	Last	0.789	0.704	**0.786**	0.682	**0.778**	0.650
		Last+Spread	**0.793**	**0.718**	0.783	**0.689**	0.766	**0.658**
	RF	Last	0.757	0.735	0.752	0.727	0.748	0.714
		Last+Spread	**0.791**	**0.784**	**0.782**	**0.764**	**0.770**	**0.743**
Brazilian	LR	Last	0.685	**0.645**	0.680	0.607	0.668	0.584
		Last+Spread	**0.741**	0.620	**0.734**	**0.617**	**0.712**	**0.605**
	RF	Last	0.724	0.693	0.705	0.674	0.679	0.652
		Last+Spread	**0.839**	**0.797**	**0.813**	**0.772**	**0.776**	**0.736**

Minimizing compute costs: When should we run more expensive analysis?

CAMLIS 2022.

Can we optimize the decision process?

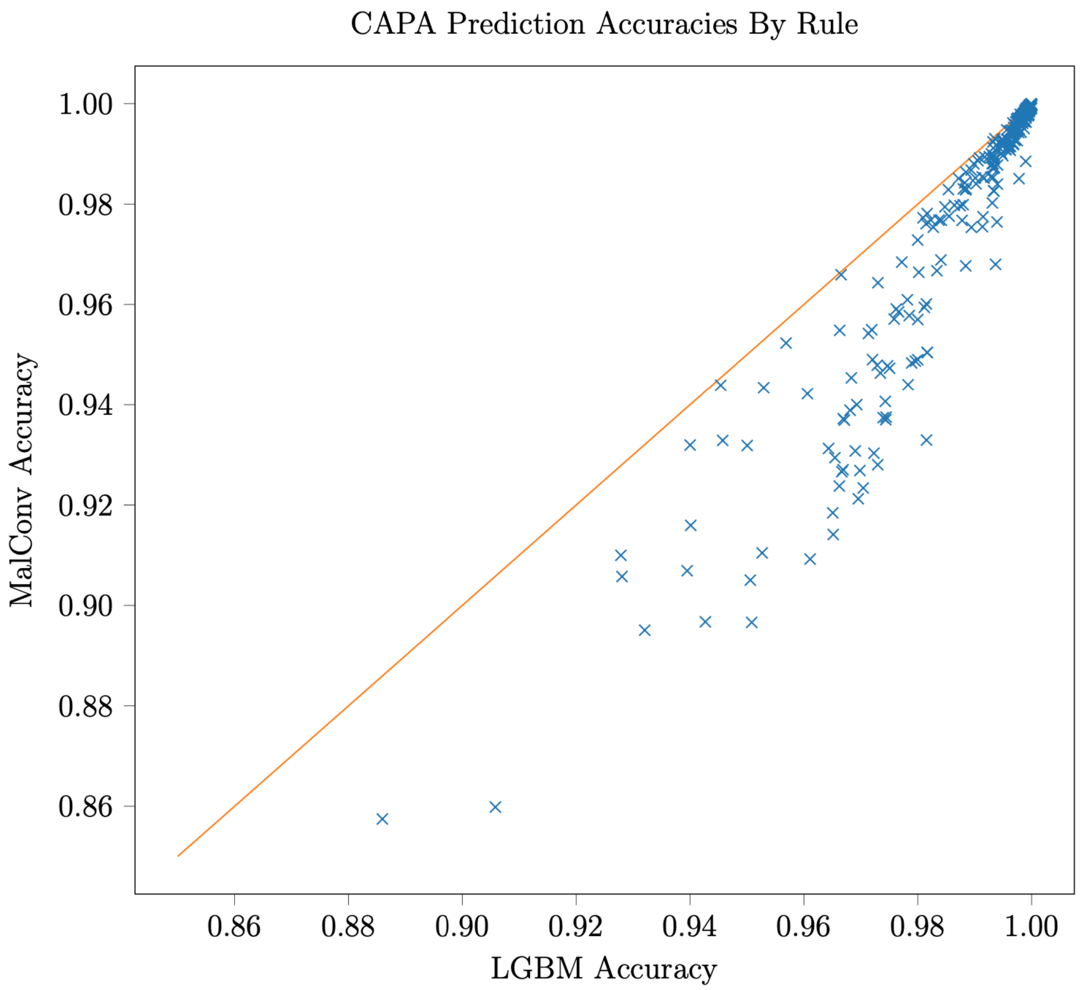


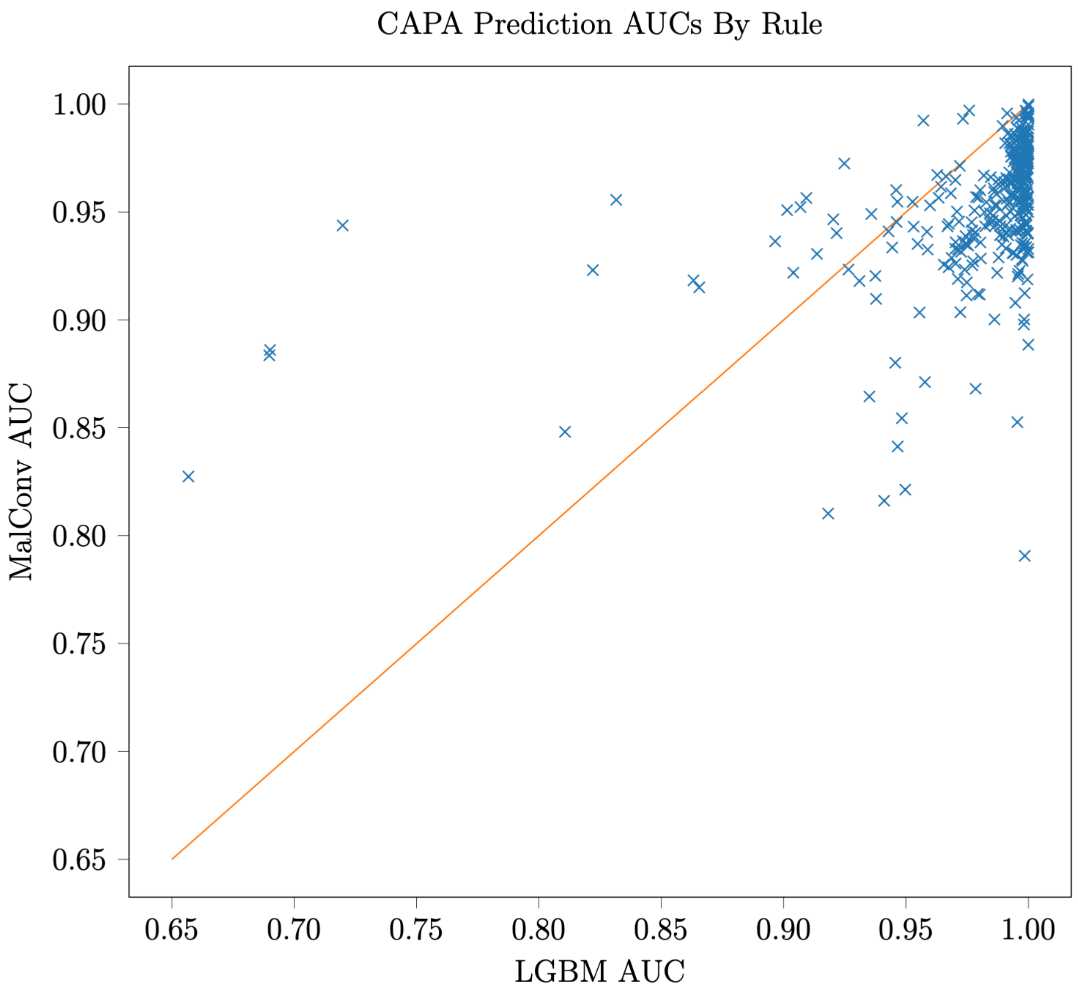
CAPA Rules

```yaml
rule:
  meta:
    name: capture webcam image
    namespace: collection/webcam
    author: johnk3r
    scope: function
    att&ck:
      - Collection::Video Capture [T1125]
    examples:
      - a30101595f6f28ab2f4b0b2cd177c3c4d2ab34a355ab7761a3795d0887c24ada:0x4011C0
  features:
    - or:
      - and:
        - api: capCreateCaptureWindow
        - basic block:
          - and:
            - api: SendMessage
            - number: 0x40a = WM_CAP_DRIVER_CONNECT
        - optional:
          - basic block:
            - and:
              - api: SendMessage
              - number: 0x40B = WM_CAP_DRIVER_DISCONNECT
      - basic block:
        - and:
          - api: SendMessage
          - number: 0x419 = WM_CAP_FILE_SAVEDIB
```

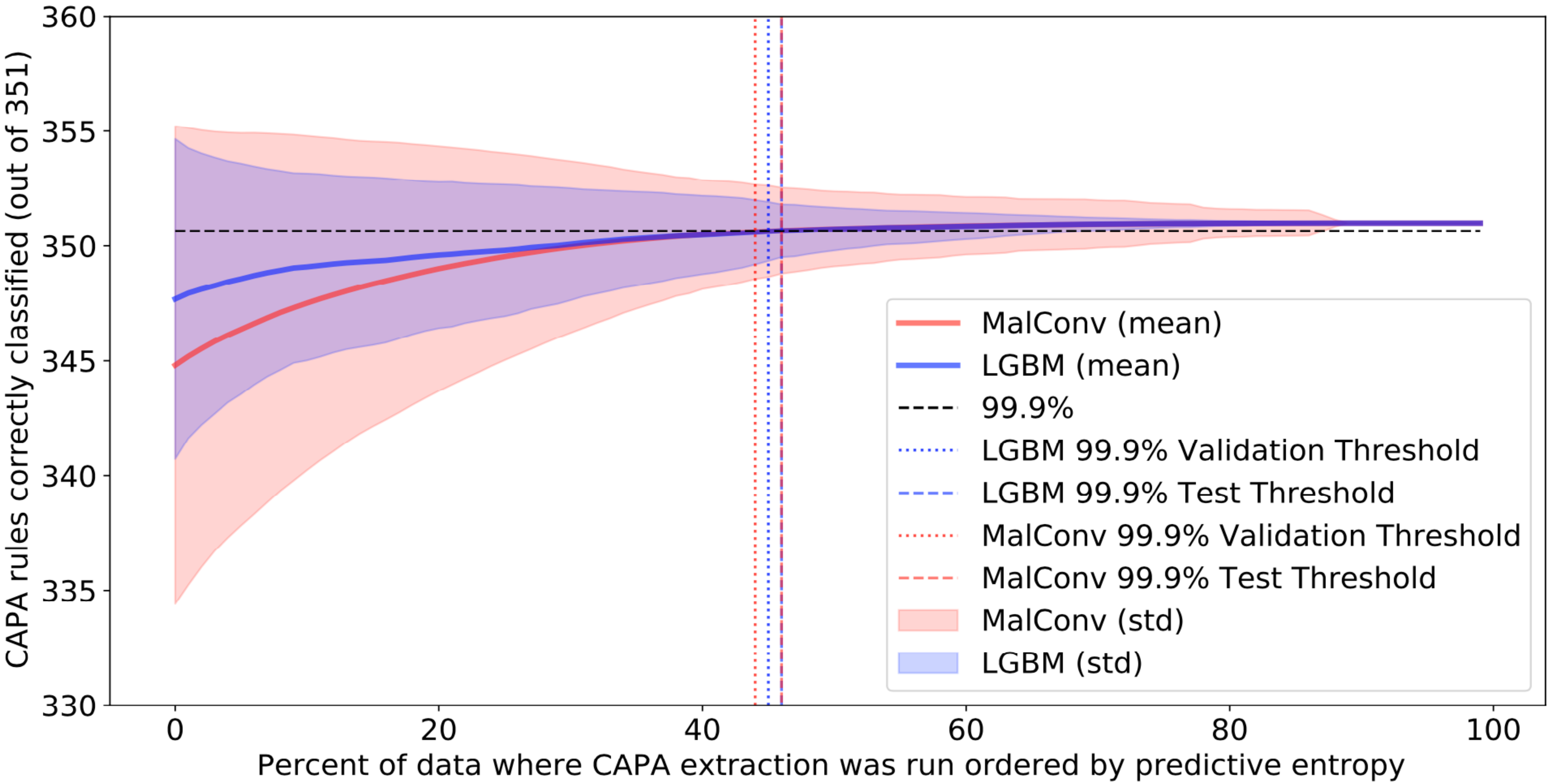
```yaml
rule:
  meta:
    name: encrypt data using Curve25519
    namespace: data-manipulation/encryption/elliptic-curve
    author: dimiter.andonov@mandiant.com
    scope: basic block
    att&ck:
      - Defense Evasion::Obfuscated Files or Information [T1027]
    examples:
      - 0a0882b8da225406cc838991b5f67d11:0x4135f6
      - 0a0882b8da225406cc838991b5f67d11:0x416f51
      - 80372de850597bd9e7e021a94f13f0a1:0x406480
      - 80372de850597bd9e7e021a94f13f0a1:0x4086f4
  features:
    # initializes a 32-byte array with
    #   array[0] = 0xf8,
    #   array[31] = array[31] & 0x3f | 0x40
    - and:
      - and:
        - number: 0xf8
        - mnemonic: and
      - and:
        - number: 0x3f
        - mnemonic: and
      - and:
        - number: 0x40
        - mnemonic: or
```

Predicting CAPA Outputs

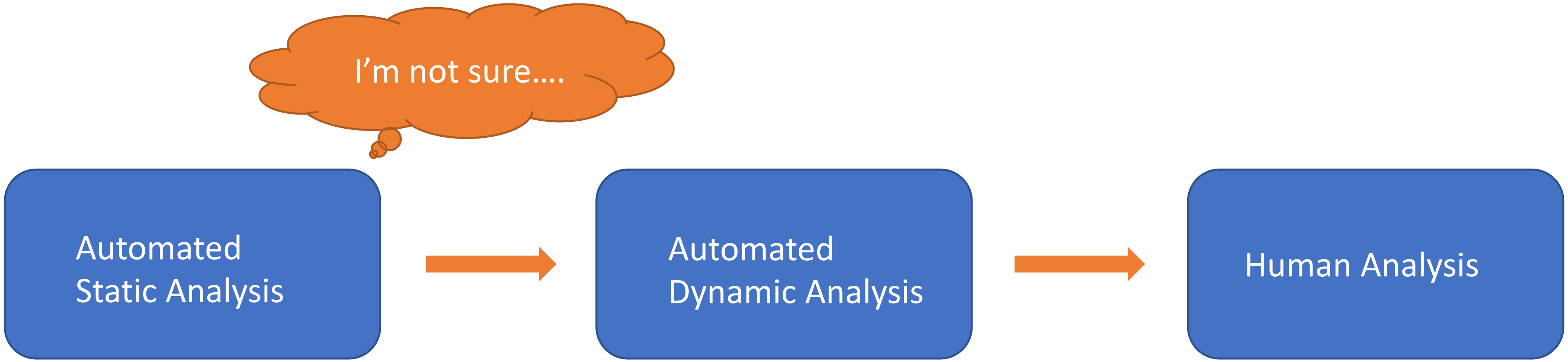




How much expensive CAPA analysis do we need?



Can we optimize the decision process?

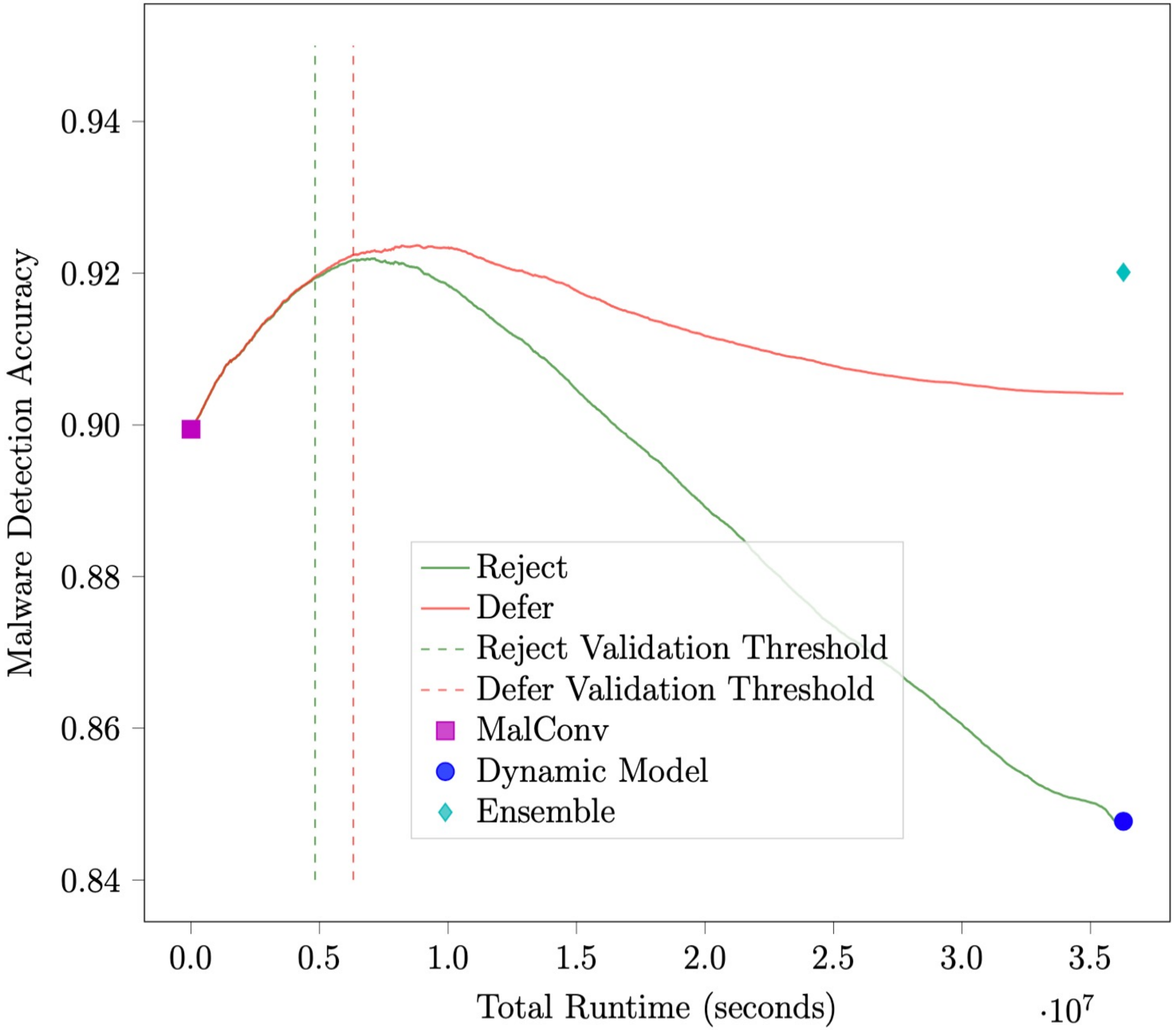


Bayesian MalConv with sampling: 0.02 seconds per file
EMBER Feature extraction: 0.09 seconds per file
CAPA feature extraction: 45.75 seconds per file
Running dynamic analysis: 526 seconds per file

Running Bayesian MalConv on a file is over 26,300 times faster than running dynamic analysis!

Can we optimize the decision process?

MalConv -> Dynamic



Thanks!

André T. Nguyen, Ph.D.
Director of Machine Learning

JURA Bio, Inc.
Boston, MA
an@jurabio.com