

# High-Assurance Machine Learning for Cybersecurity

Brian Matejek

# About Me

- Daniel and I were lab mates during our PhDs researching connectomics.
- We worked on numerous projects together on compression, computer vision, and visualization.
- Here we are celebrating a well-earned break with our lab mate Fritz after a paper submission to MICCAI in 2017.



# About Me

- After completing my PhD, I did some freelance consulting while deciding what I wanted to do for a career.
- After a half year, I decided to join SRI International researching various topics in machine learning and artificial intelligence, currently with focus on applications to cybersecurity and quantum-inspired computing.

# About SRI International

- Formed as the Stanford Research Institute after WW2, SRI International is a non-profit research institute primarily supporting the United States defense, intelligence, and health communities.
- In the last 75+ years, SRI has researched and contributed to a wide range of government and commercial ventures.



First Computer Mouse

29 OCT 69	2100	LOADED OP. PROGRAM	CSK
		EDIT BEN BARKER	
		BBV	
	22:30	Talked to SRI	CSK
		Host to Host	
		Left op. program	CSK
		running after sending	
		a host dead message	
		to imp.	

First Message Received (ARPANET)



Football First Down Lines

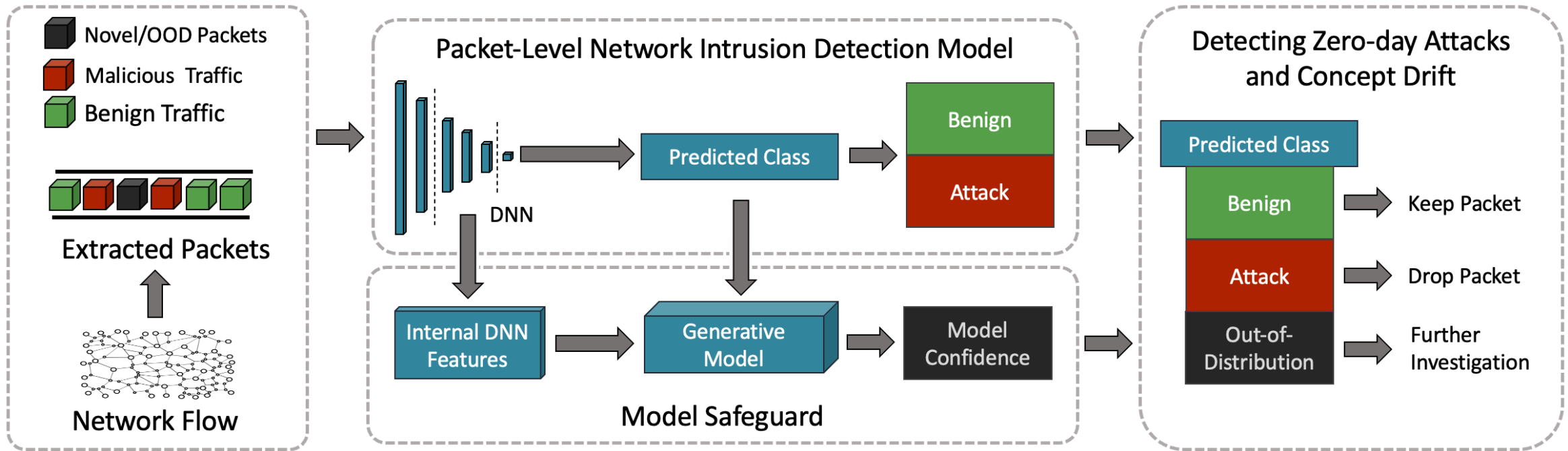
# About SRI International

- Despite our primary focus on research, SRI encourages inventors to pursue business ventures on their research ideas.
- Siri, Apple's AI Assistant, was created by SRI International during the DARPA CALO (Cognitive Assistant that Learns and Organizes) program.
  - Apple acquired the company in 2010.

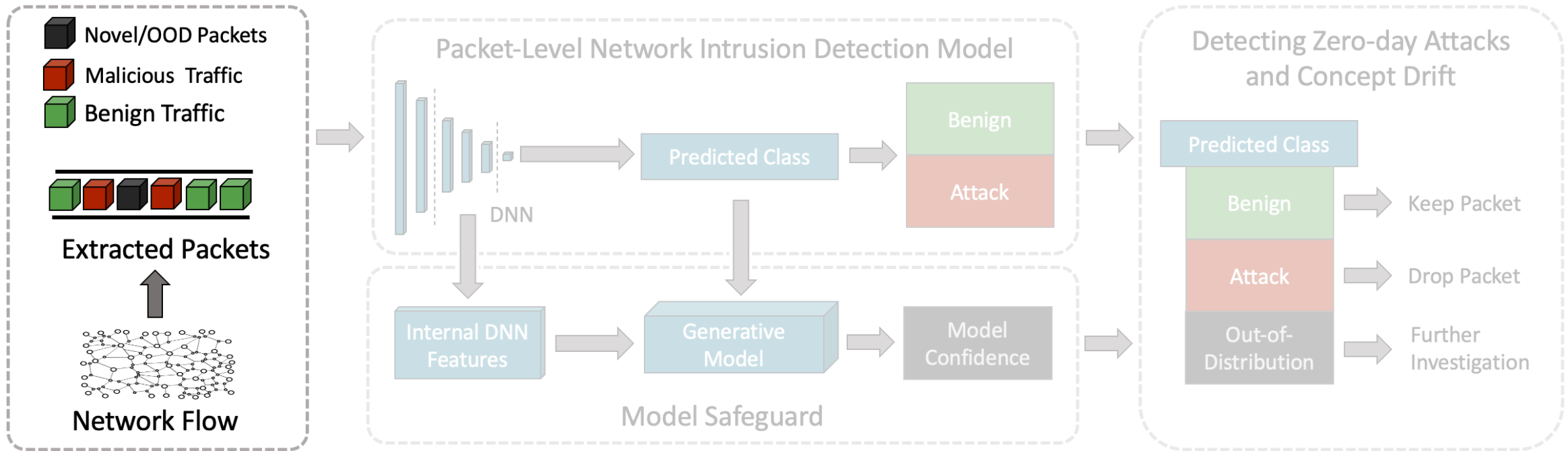


Hey Siri

# High-Assurance Machine Learning Architecture



# Packet-Level Network Traffic



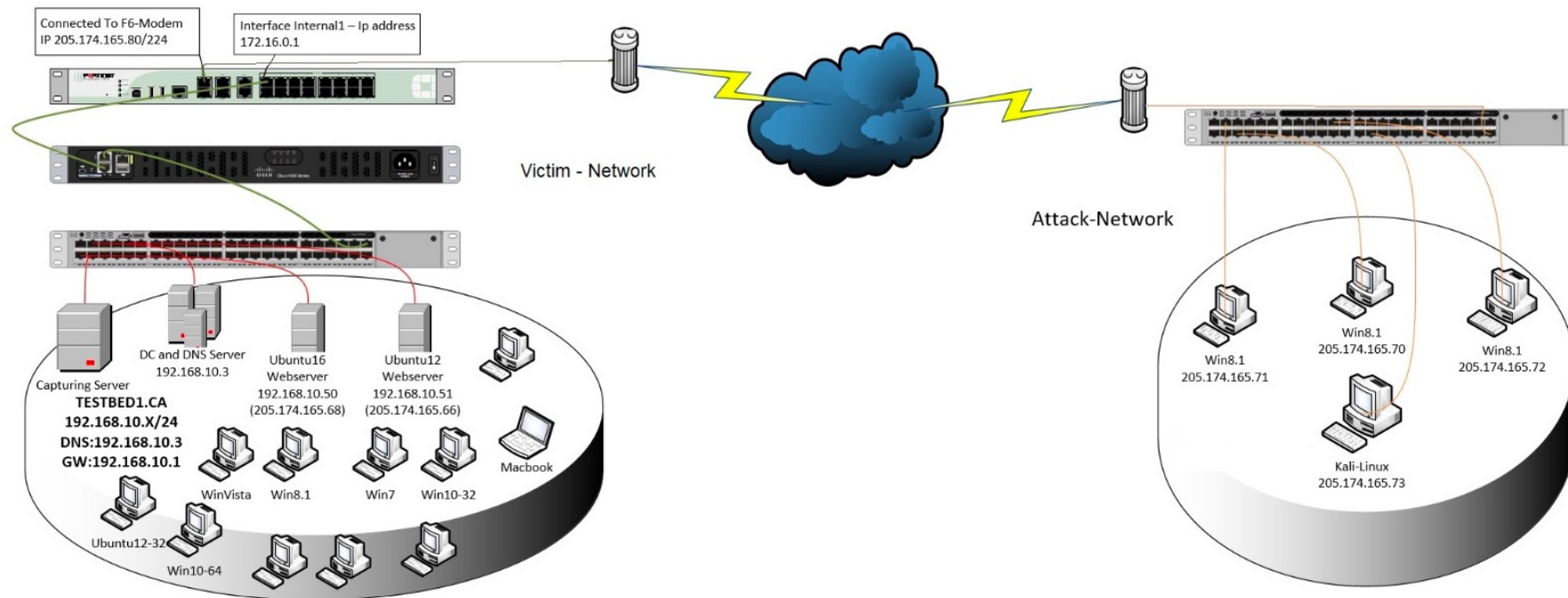
# Network Traffic

- At a high-level, traffic on a network consists of many small packets that in sequence create a stream of information.
- These packets follow various established protocols such as IPv4 and TCP/UDP at the application layer.
- Packets typically contain a short header with relevant information for the receiver followed by the actual raw (encrypted or unencrypted) payload content.
- We focus on learning the differences between malicious (attack) and benign payload content.



# Generating Network Traffic

- Generating data can be tricky, particularly in cybersecurity where privacy concerns are paramount.
- Various datasets exist that simulates network traffic between a victim network and an attack one.



# Header Context

- Machine learning models can learn complex mappings from inputs to outputs.
- However, if available, the models will sooner converge to a simple mapping.
- The network traffic datasets use set IP addresses for the attack packets.

**IPv4 Header**

Version	IHL	TOS	Total Length	
Identification			Flags	Fragment Offset
TTL		Protocol	Header Checksum	
Source Address				
Destination Address				

**TCP Header**

Source Port				Destination Port			
Sequence Number							
Acknowledgement Number							
Data Offset	Reserved	C W R	E C E	U R G	A C K	P S H	S S Y N N
Checksum				Window Size			
Urgent Pointer							

**UDP Header**

Source Port	Destination Port
Length	Checksum

# Benign Traffic

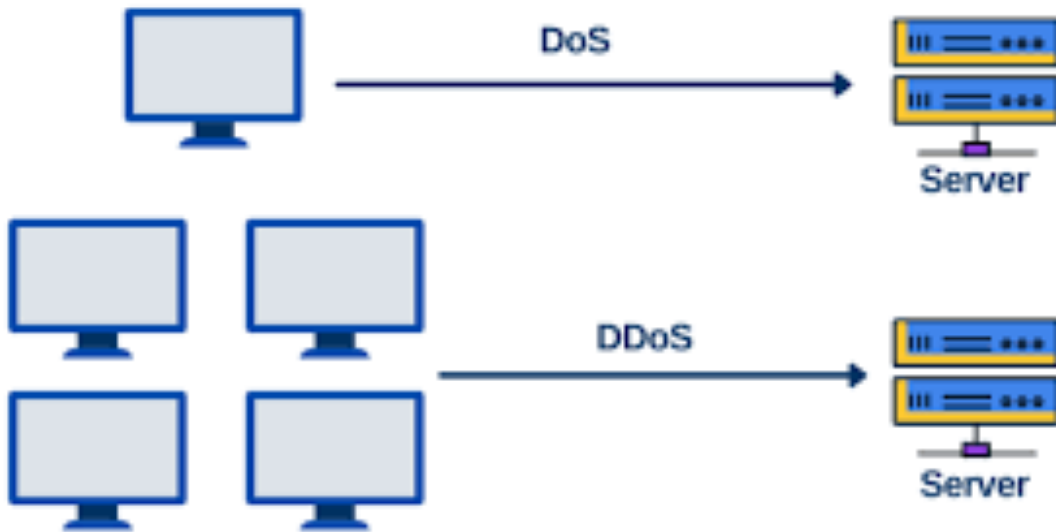
- Our benign traffic consists of normal day-to-day activities such as checking emails, reading the news, or watching streaming services.



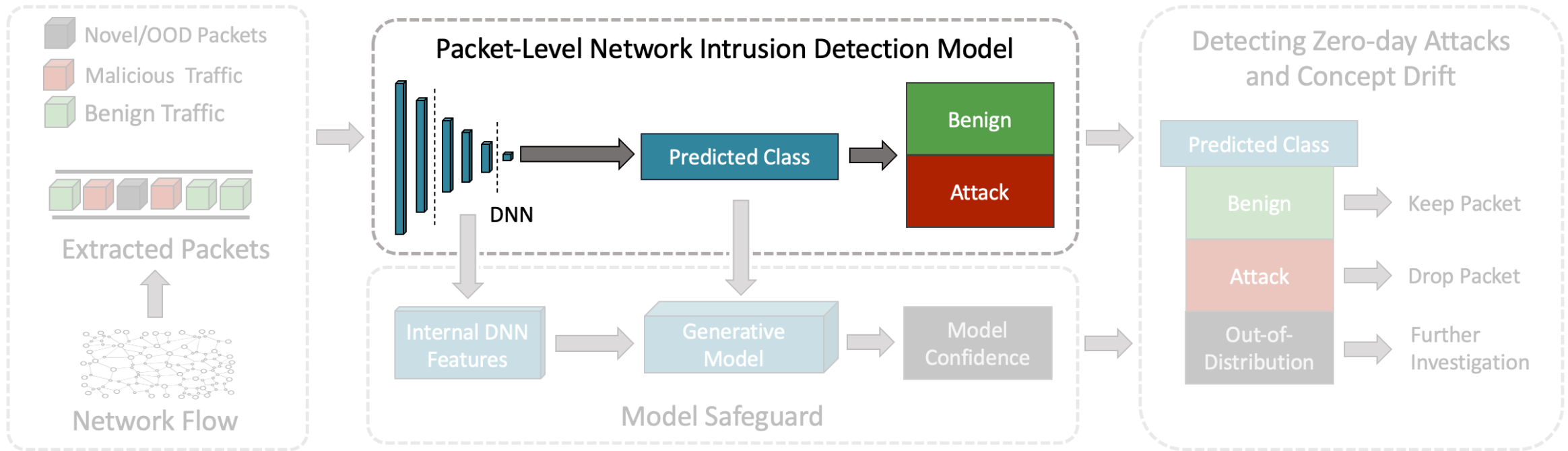
**NETFLIX**

# Attack Traffic

- Our attack data consists of various attempts to infiltrate or otherwise interrupt a victim network.

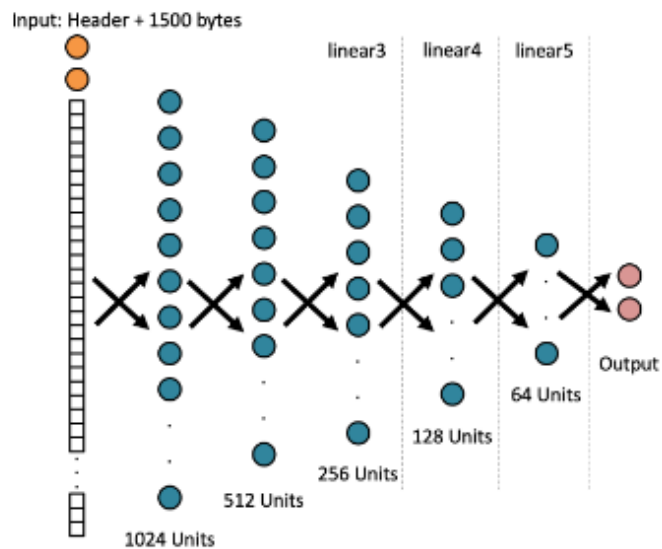


# Neural Architectures

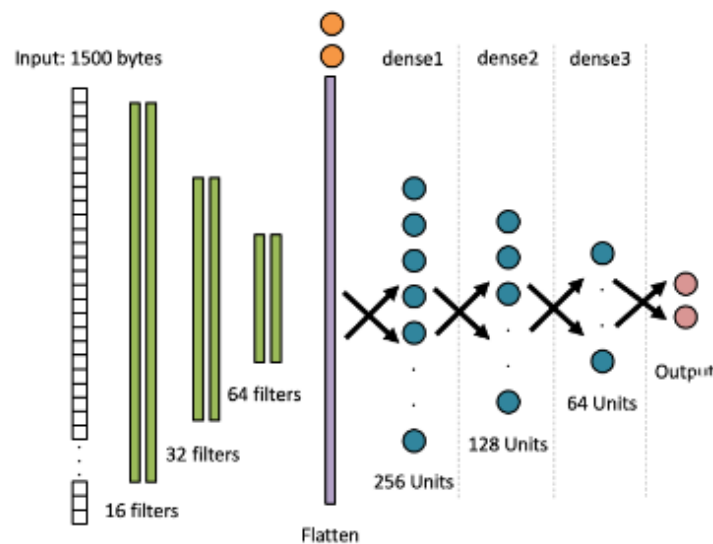


# Neural Architectures

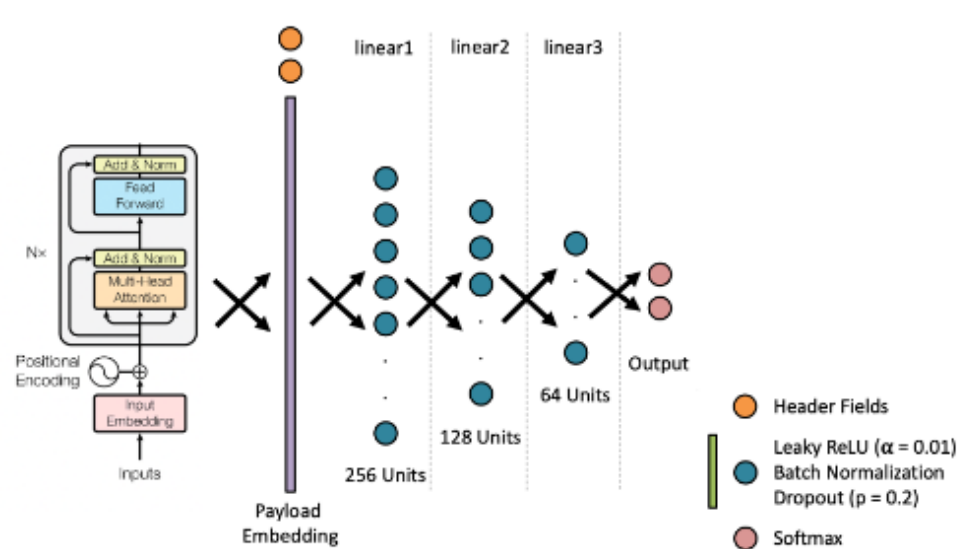
## Fully-Connected Neural Network



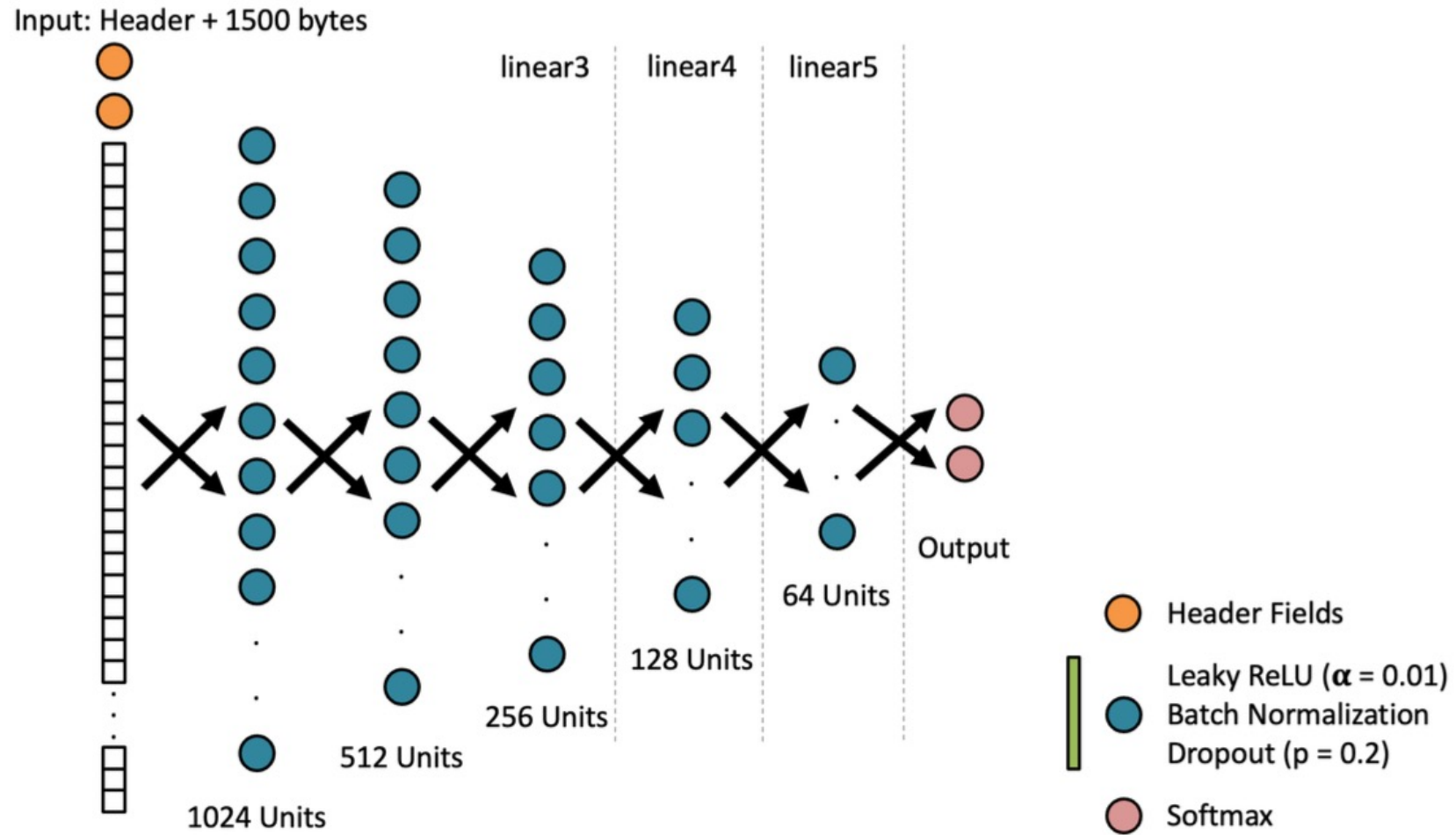
## Convolutional Neural Network



## Transformer Neural Network

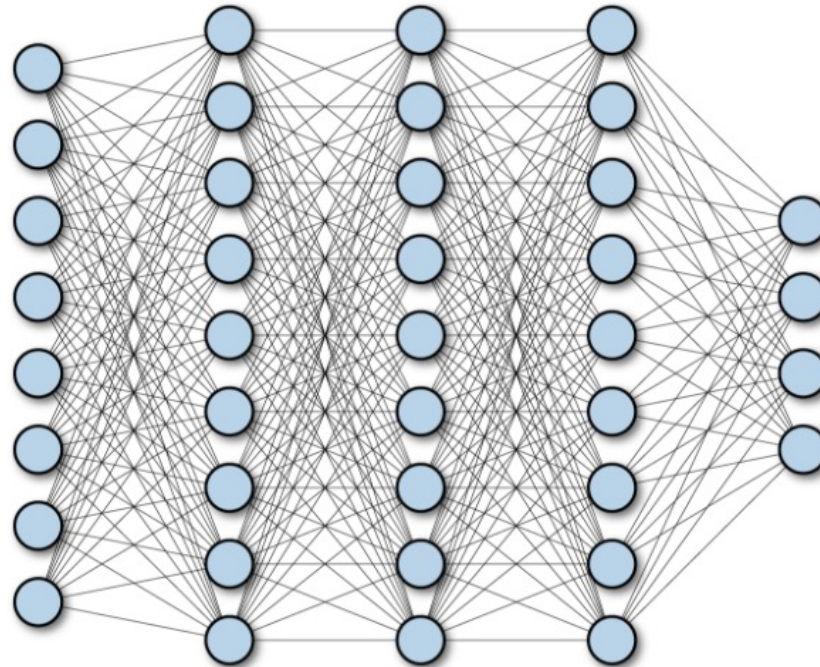


# Fully-Connected Neural Network



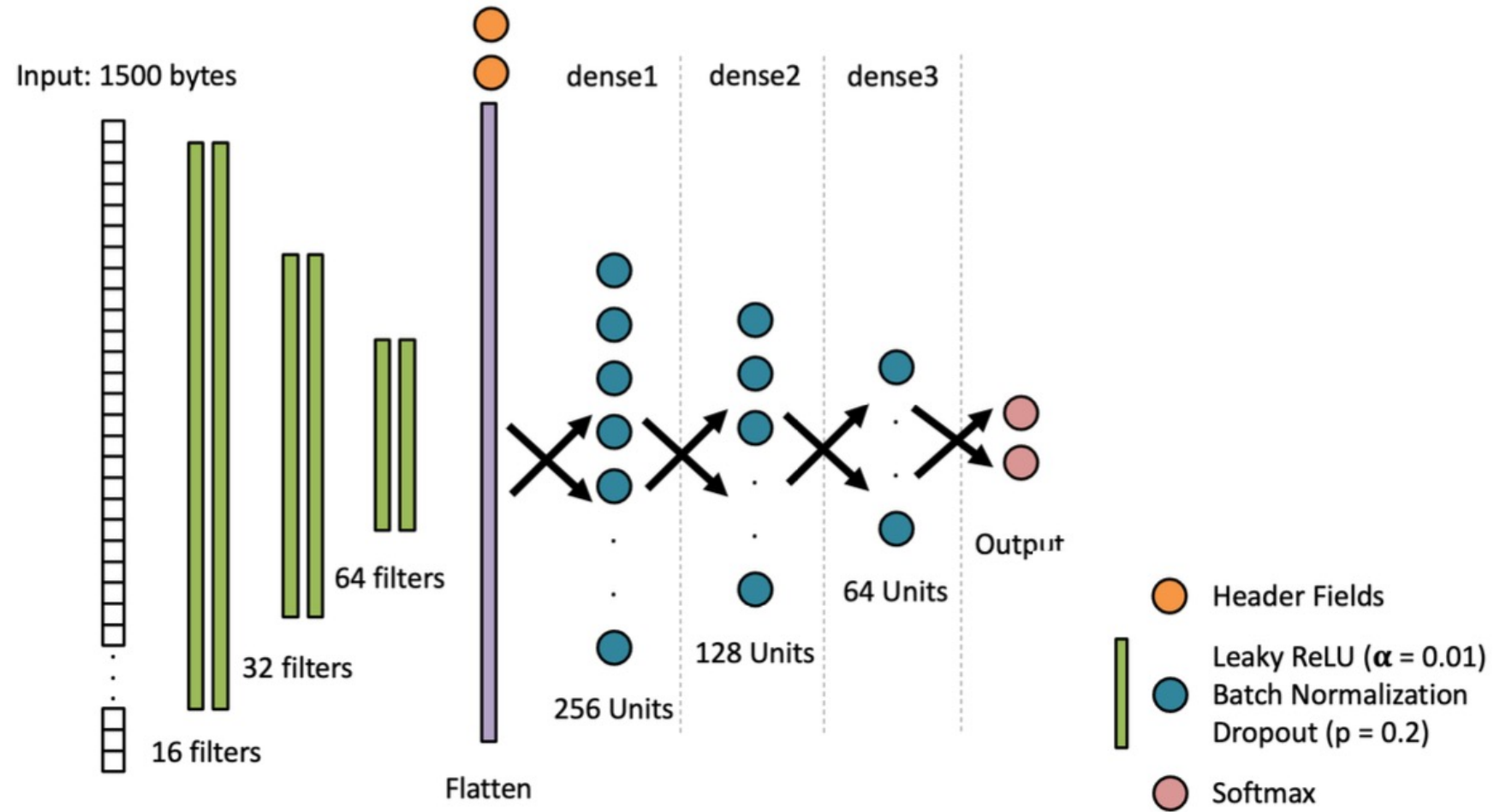
# Fully-Connected Neural Networks (FNN)

- Fully-connected neural networks connect neurons from each layer to all neurons in the following layer.
  - This works great for unordered tabular data to enable interactions between all input fields.



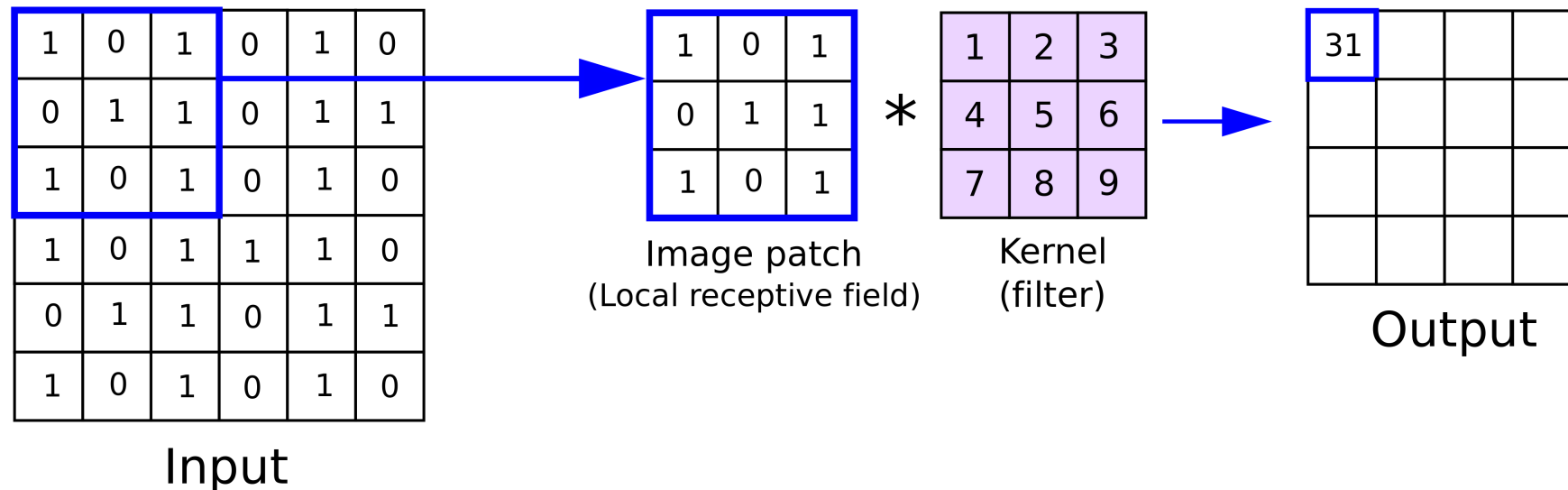


# Convolutional Neural Network



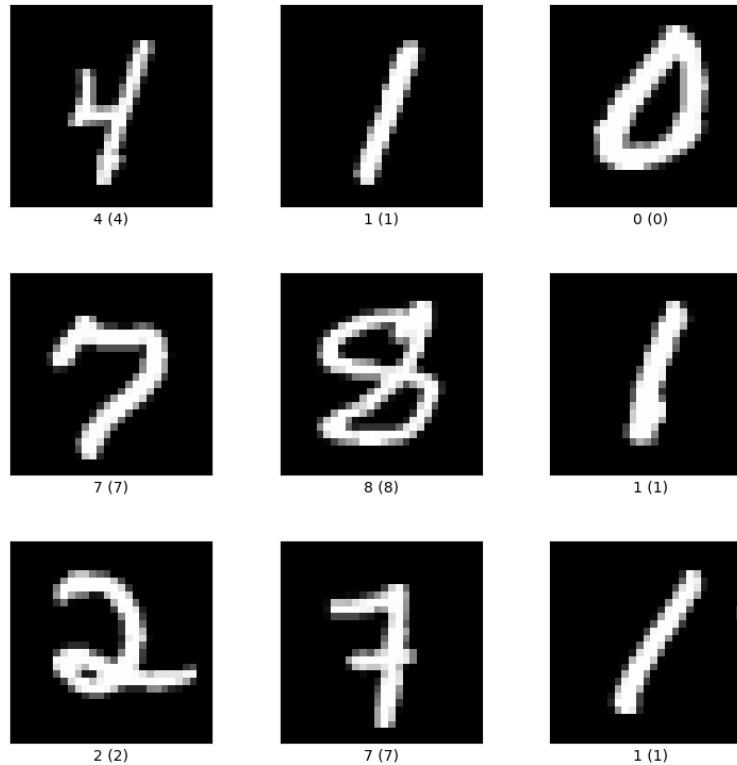
# Convolutional Neural Networks (CNN)

- CNNs work exceptionally well on data with a high-level of local structure.
  - For example, nearby pixels in an image are generally correlated with one another.
- In a CNN, convolution operations take a weighted average of nearby elements to pass on to the deeper layers of the network.

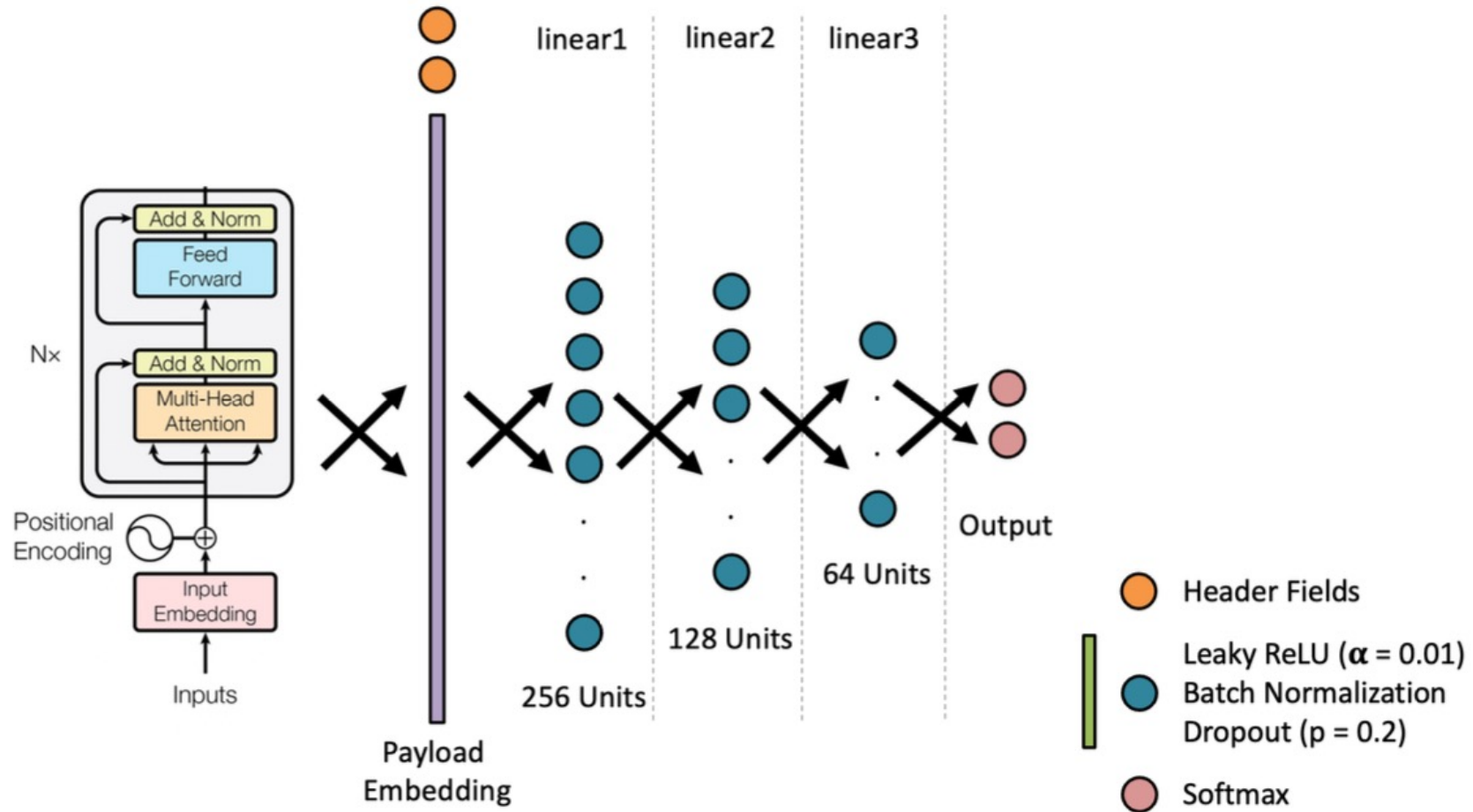


# Convolutional Neural Networks (CNN)

- This can make CNNs more efficient than FNNs on images as you can ignore long range interactions between pixels on opposite sides of the image, or words in a sentence.



# Transformer Neural Network

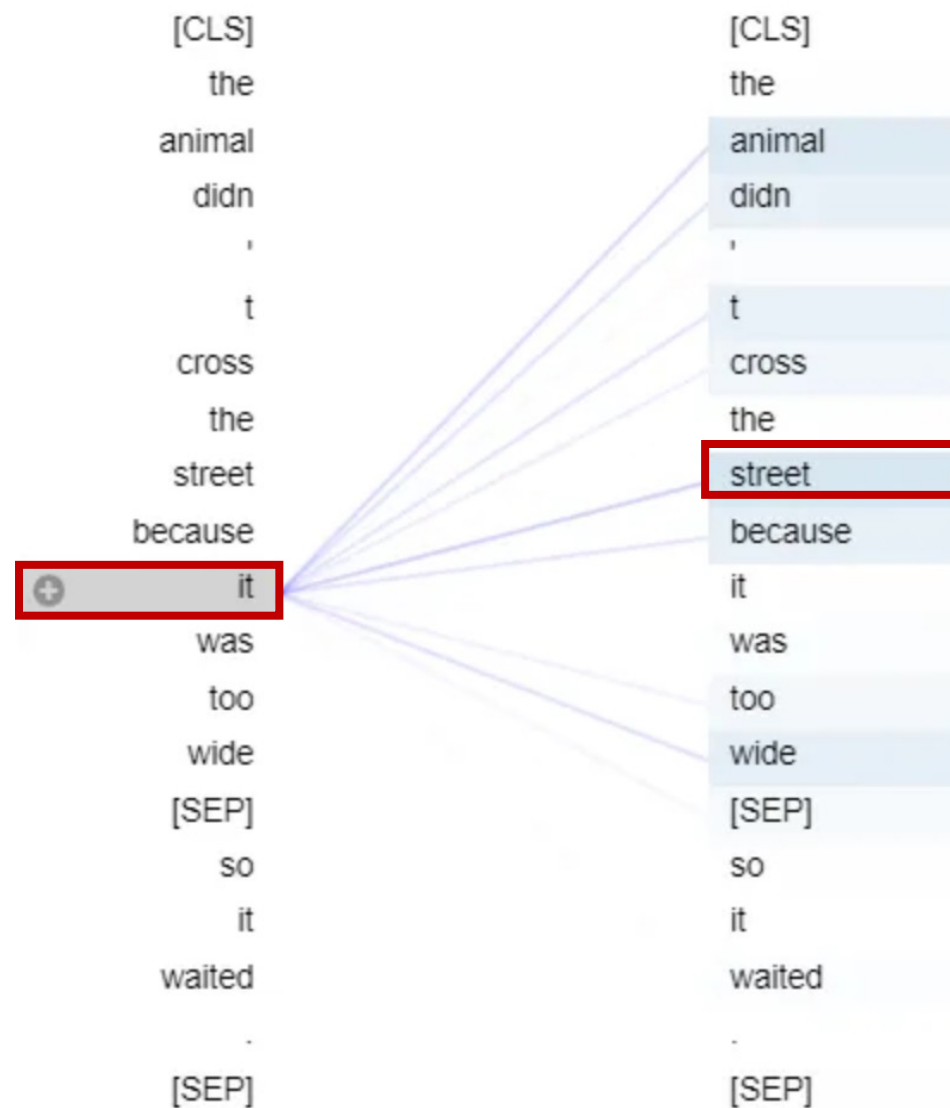


# Transformer Neural Networks

- Transformers architectures employ a "self-attention" mechanism that enables the model to identify the most significant parts of the input.
  - This attention mechanism provides the context of every element in the input to every other element.

# Transformer Neural Networks

The animal didn't cross the **street**  
because **it** was too wide, so it waited.



# ChatGPT

- ChatGPT (generative pretrained **transformer**) is a large language model based on the transformer architecture.
- Transformers have taken AI (and the world) by storm for their ability to generate human-like text and images.

AI

## ChatGPT: Everything you need to know about the AI-powered chatbot

Alyssa Stringer @alyssastring / 8:00 AM EDT • April 13, 2023

Comment



Image Credits: Leon Neal / Getty Images

# Coding Neural Networks

- The two main frameworks for implementing neural networks are PyTorch and TensorFlow.
  - Keras is a high-level API that sits on top of PyTorch and TensorFlow to enable faster prototyping at the expense of customizability.
- There is no correct choice for framework, and for those interested it is probably best to become "fluent" in one and "conversational" in the other.





# PyTorch versus TensorFlow

- Historically, industry favors TensorFlow and research favors PyTorch.
  - TensorFlow has more publicly available model weights.
  - It is easier to create custom architectures and loss functions in PyTorch.
- It is unclear how this will change over the next few years, although it is hard to see TensorFlow overtaking PyTorch in academia.

# PyTorch versus TensorFlow



**Artificial Intelligence Report**

Covering artificial intelligence topics, AI stocks, and stories ...

Daily newsletter

212,995 subscribers

[+ Subscribe](#)



## Prediction: PyTorch will Disrupt TensorFlow in Industry dominance in 2023



**Michael Spencer**

Indie Writer #1 Machine Learning Newsletter AiSupremacy | Tap the on my profile

1,559 articles

[+ Follow](#)

June 22, 2022

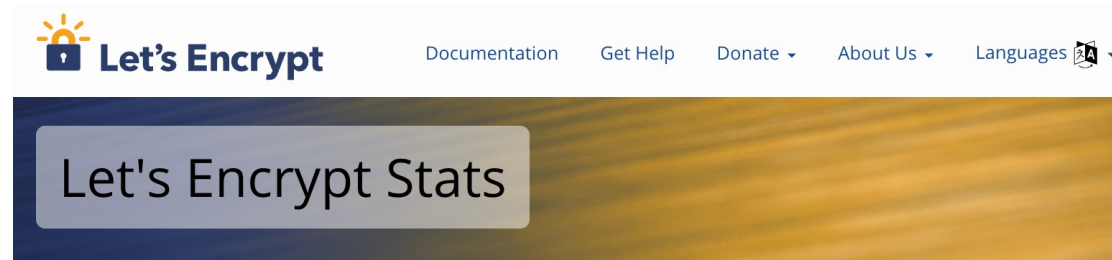
# Machine Learning Has Solved Cybersecurity!

- Across the board, our neural networks classify both malicious and benign packets with exceptionally high accuracy

Architecture	Context	Accuracy ( $\uparrow$ )	AU ROC ( $\uparrow$ )	F1-Score ( $\uparrow$ )
CNN	✓	0.9923 ( $\pm 0.0003$ )	<b>0.9997 (<math>\pm 0.0000</math>)</b>	0.9923 ( $\pm 0.0003$ )
CNN		0.9917 ( $\pm 0.0003$ )	0.9980 ( $\pm 0.0002$ )	0.9917 ( $\pm 0.0003$ )
FNN	✓	0.9913 ( $\pm 0.0003$ )	<b>0.9997 (<math>\pm 0.0000</math>)</b>	0.9913 ( $\pm 0.0003$ )
FNN		0.9894 ( $\pm 0.0001$ )	0.9971 ( $\pm 0.0001$ )	0.9893 ( $\pm 0.0001$ )
Transformer	✓	<b>0.9940 (<math>\pm 0.0010</math>)</b>	<b>0.9997 (<math>\pm 0.0000</math>)</b>	<b>0.9940 (<math>\pm 0.0010</math>)</b>
Transformer		0.9915 ( $\pm 0.0002$ )	0.9984 ( $\pm 0.0001$ )	0.9914 ( $\pm 0.0002$ )

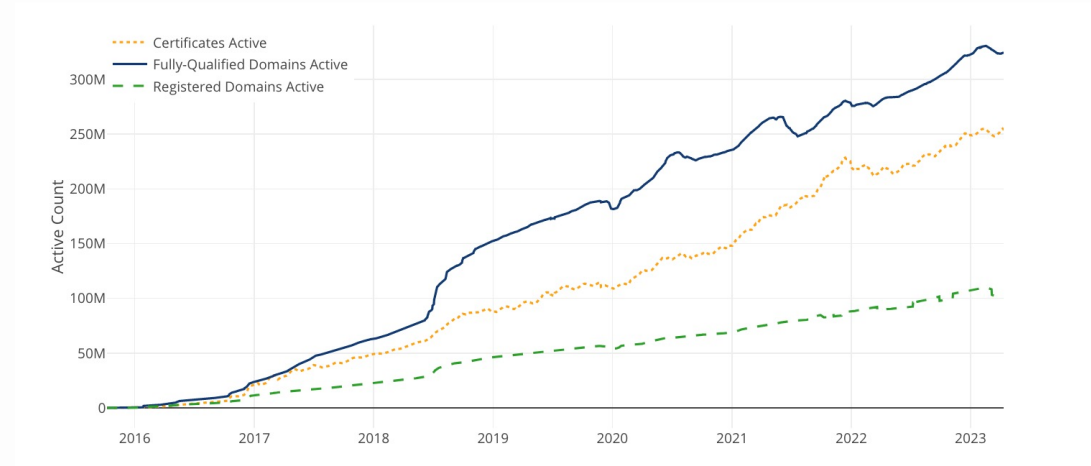
# Not so fast...

- Although network protocols are slow to evolve, internet best-practices change leading to changing payload structures.



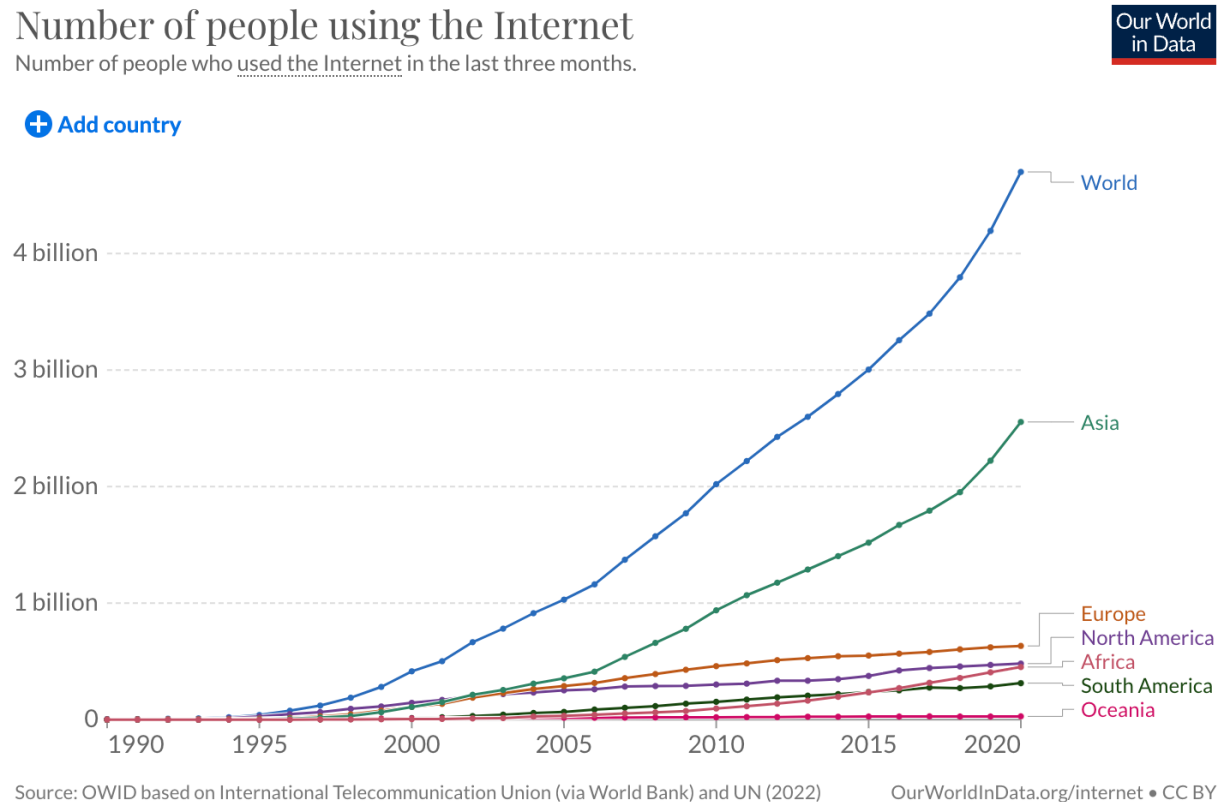
*Please note that the Let's Encrypt Growth and Let's Encrypt Certificates Issued Per Day charts are undergoing updates and may not reflect the most recent data.*

## Let's Encrypt Growth



# Not so fast...

- Geographic diversity increases as more of the world goes online.
- Distribution of typed languages changes; training on one language does not necessarily correspond to results in another.



# Not so fast...

- New types of traffic appear with novel methods for encryption.
- After March 2020, teleconferencing traffic soared compared to others.

## Comcast: Pandemic drove peak internet traffic up 32% in 2020



Comcast has been laying a lot of cable.  
Image Credit: Comcast



Connect with top gaming leaders in Los Angeles at GamesBeat Summit 2023 this May 22-23. [Register here.](#)

[Comcast](#) said peak internet traffic in the U.S. rose 32% in 2020 over pre-pandemic levels, with some markets rising 50% in March 2020.



# Not so fast...

- Novel attacks appear every day, and neural networks fail to generalize to these exploits and label them as benign.


**NEWS**

Home | War in Ukraine | Climate | Video | World | US & Canada | UK | Business | Tech | Science

Tech

## Scramble to fix huge 'heartbleed' security bug

8 April 2014



### Heartbleed Bug


The Heartbleed bug is a serious vulnerability in the popular OpenSSL software library. This weakness allows stealing the information protected, under normal conditions, by the SSL/TLS encryption used to secure the Internet. SSL/TLS provides communication privacy over the Internet for applications such as web, email, instant messaging (IM) and some virtual private networks (VPNs).

The bug allows anyone on the Internet to read the memory of protected by the vulnerable versions of the OpenSSL. This compromises the secret keys used to identify the service, to encrypt the traffic, the names and passwords of the actual content. This allows attackers to eavesdrop on, steal data directly from the services and users and to

**HEARTBLEED PROJECT**

The researchers who discovered the bug publicised their findings via the web

**A bug in software used by millions of web servers could have exposed anyone visiting sites they hosted to spying and eavesdropping, say researchers.**

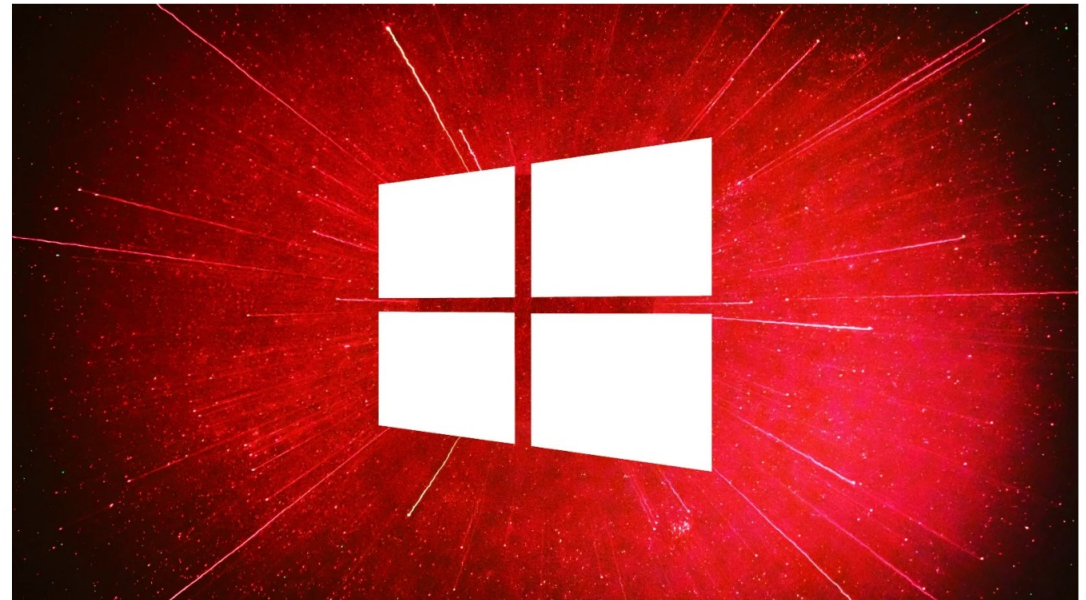


**HEARTBLEED PROJECT**

## Windows zero-day vulnerability exploited in ransomware attacks

By **Sergiu Gatlan**

April 11, 2023 03:23 PM 1



Microsoft has patched a zero-day vulnerability in the Windows Common Log File System (CLFS), actively exploited by cybercriminals to escalate privileges and deploy Nokoyawa ransomware payloads.

# Novel Inputs in Computer Vision

- In computer vision, novel inputs can take many forms such as airplanes to models trained to differentiate boats and cars.
- In the worst case, a self driving car trained on simulated roads may not have seen bikers or pedestrians.



**(a) Car doesn't detect biker, leading to a crash**

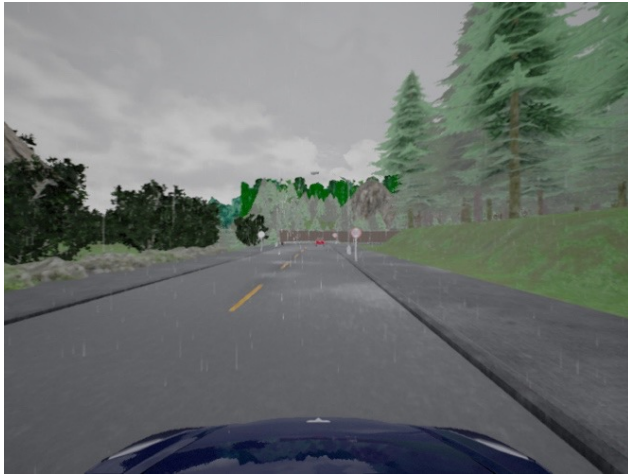


**(b) Frame detected as OOD, the region in red shows the pixels responsible for deviation**



# Concept Drift in Computer Vision

- A self driving car that only trained on sunny or moderate rain conditions will not be able to operate in heavy rain or snowy conditions.



Light Rain



Moderate Rain

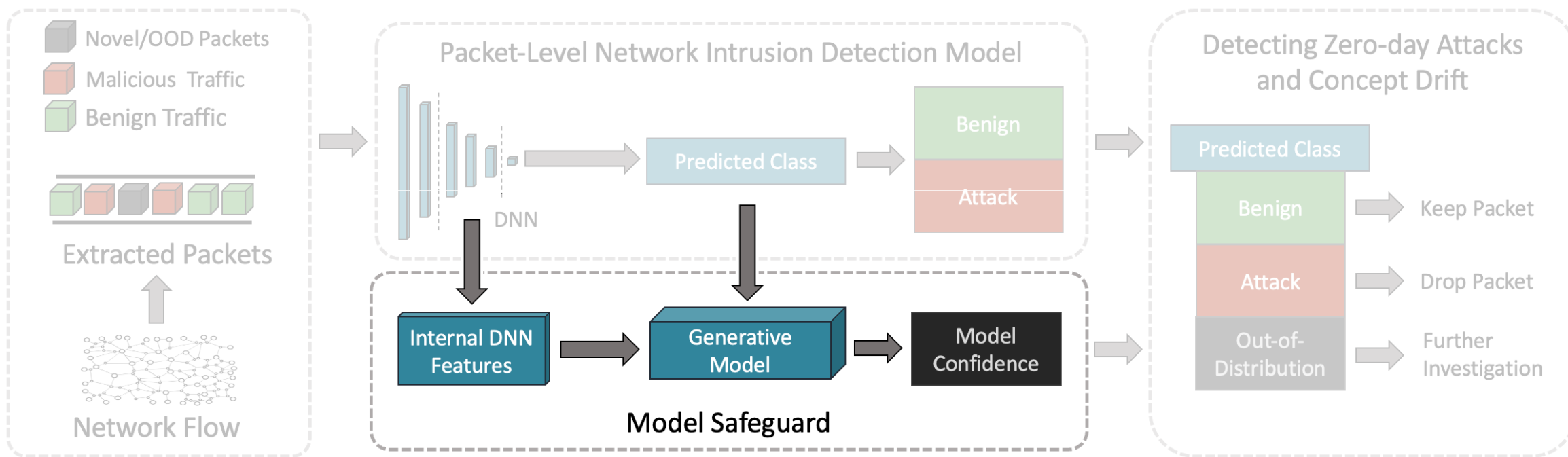


High Rain

# Overconfidence on Novel Inputs

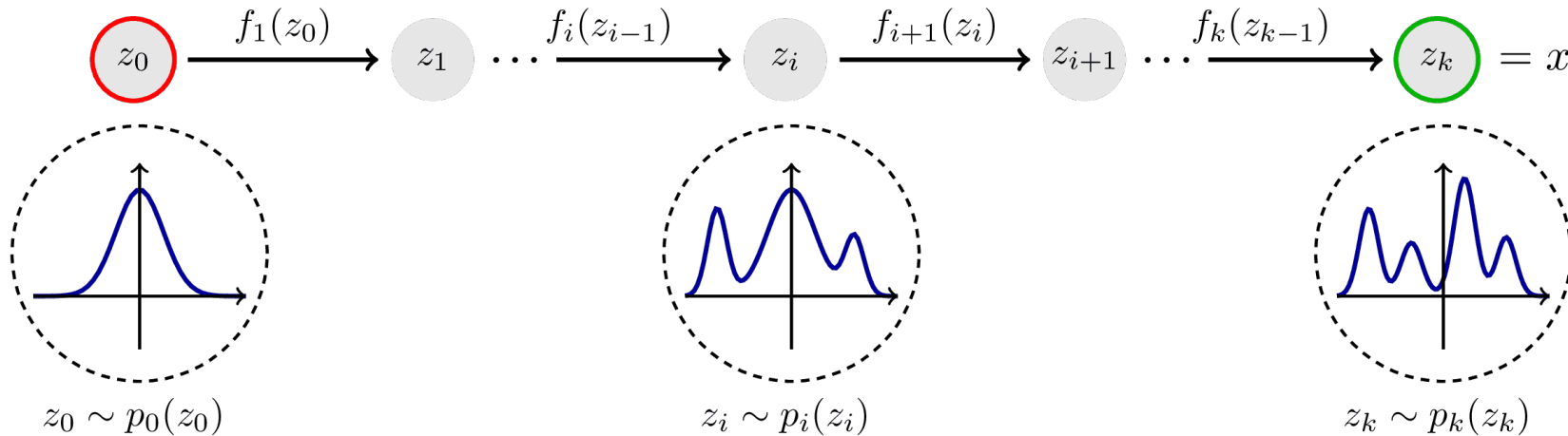
- If neural networks produced low certainty scores on these novel inputs, we could just threshold on this confidence and detect these troublesome inputs.
- Unfortunately, though, networks are extremely confident on their incorrect predictions on the novel, or "out-of-distribution", data!

# Model Safeguards



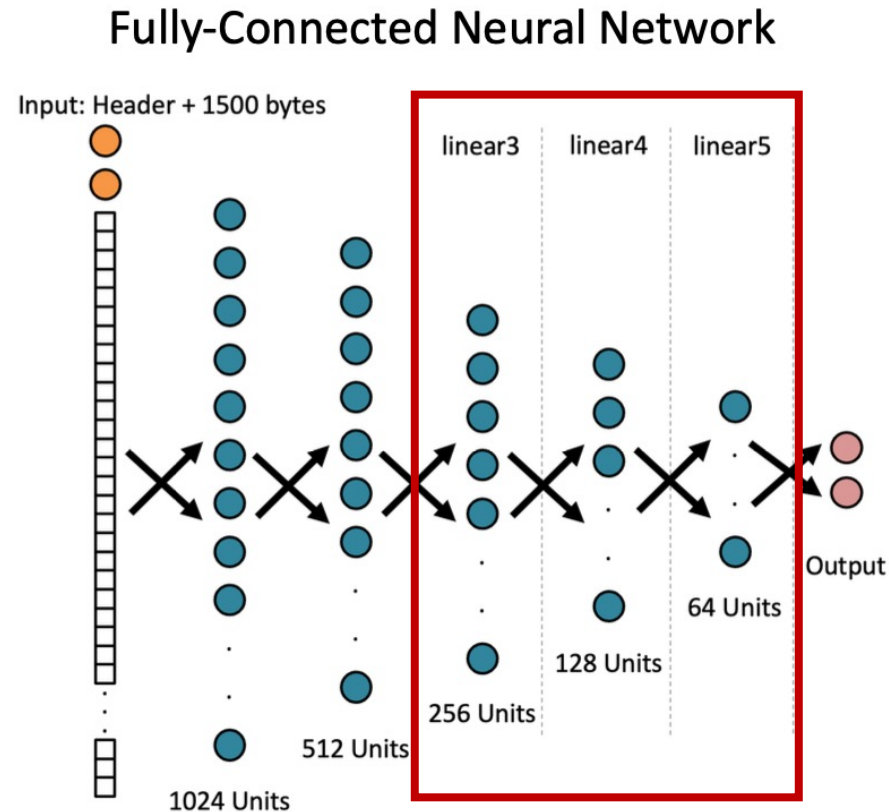
# Generative Models

- Generative models model the distribution of a given dataset.
- We can either learn these distributions using neural networks or model them with existing approaches from mathematics and statistics.
- We can use these models to determine if our input is "in-distribution" or "out-of-distribution" (i.e., novel inputs, zero-day exploits, new web traffic, etc.).



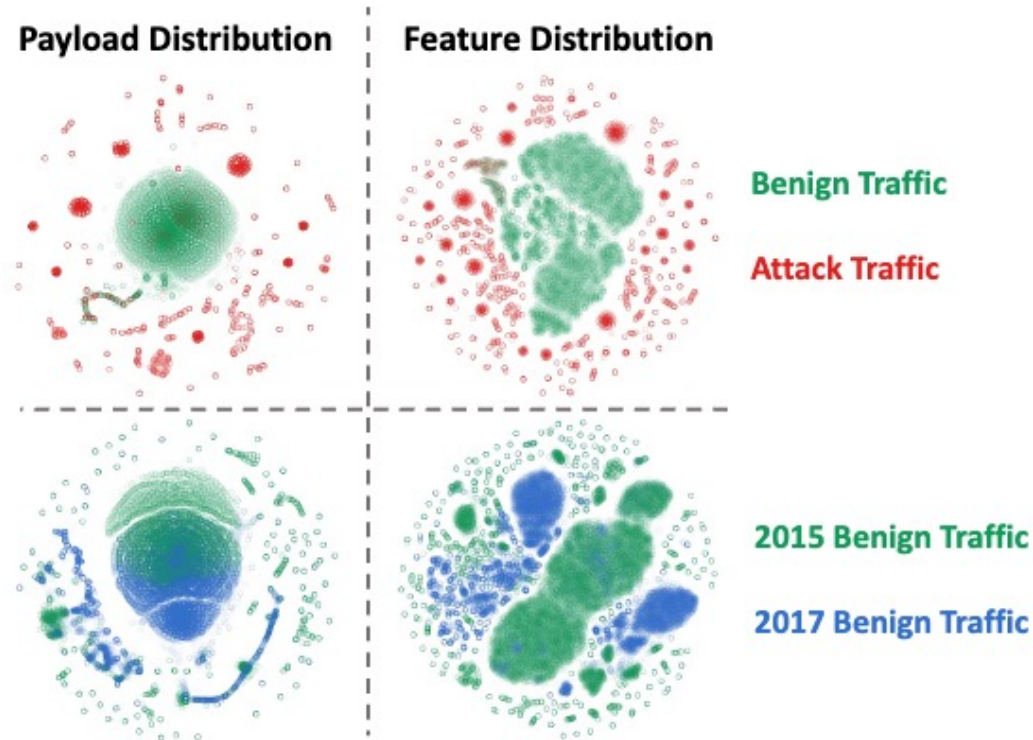
# Feature Extraction

- We extract intermediate features (i.e., the outputs from certain layers) for modeling the distributions of the in-distribution data.



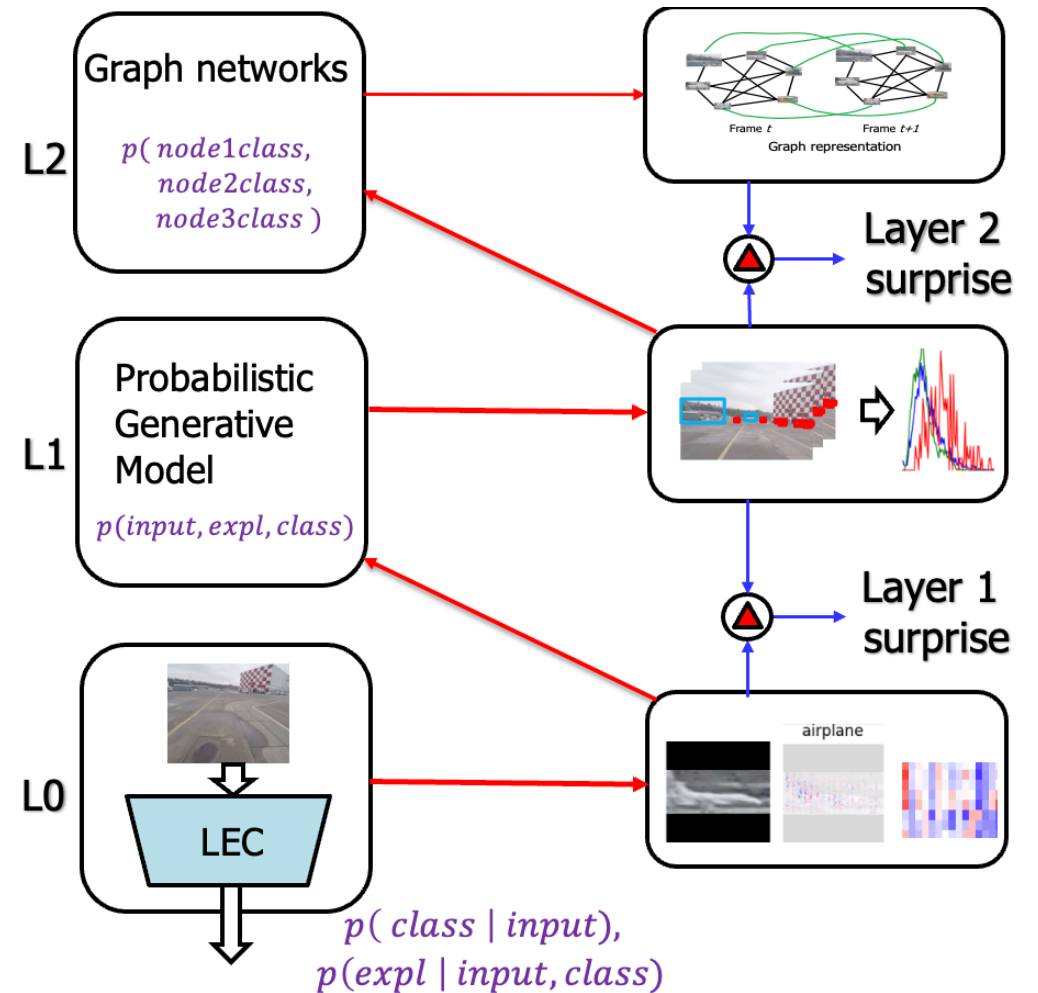
# Why Intermediate Features?

- A first question is "why model the intermediate features and not the payloads themselves?"
- We've found that these latent representations are more easily separable between different types of classes.

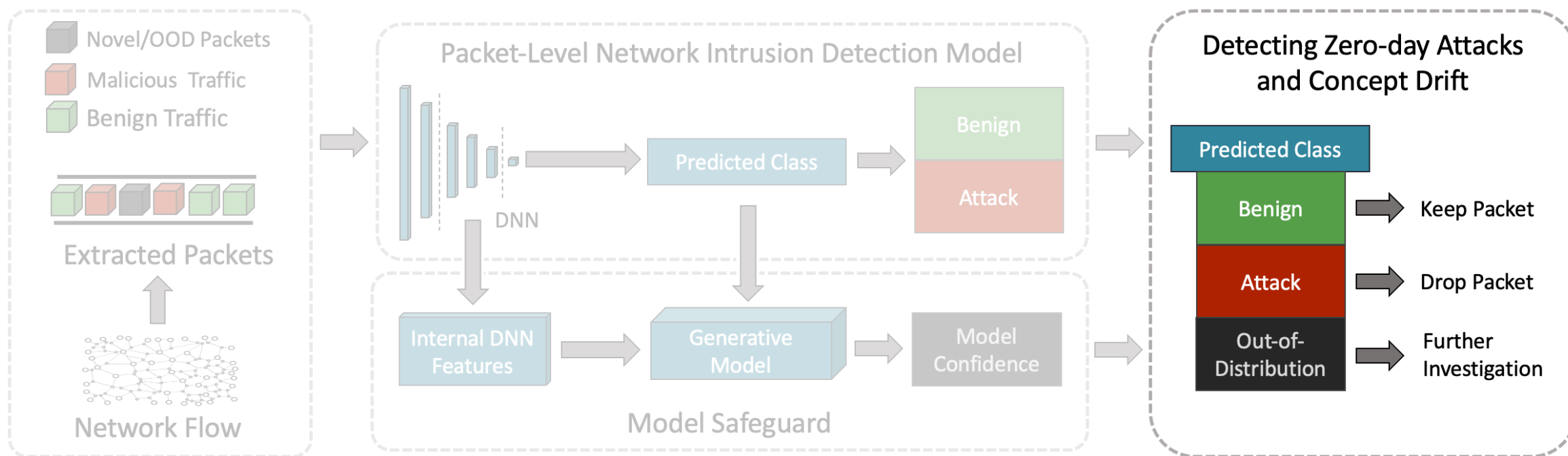


# Why Generative Models at All?

- Our dual approach of having a generative model overseeing a discriminative classifier comes from the "predictive coding" theory of mind from neuroscience.
- In this theory, humans have learned certain classifiers to distinguish differences between animals, things, etc.
- At a higher level, humans continually create (generate) a model of their environment based on what they have seen.
- A novel input generates a surprise that causes one to dismiss the results of the lower-level classifier.



# Results





# Identifying Potential Zero-day Exploits

- With high accuracy we can detect novel packets from zero-day exploits with our model safeguard.

Architecture	Header Context	Layer	AU ROC ( $\uparrow$ )	TPR (TNR 95%) ( $\uparrow$ )	TPR (TNR 85%) ( $\uparrow$ )
<b>FTP-Patator</b>					
CNN	✓	dense2	0.8685 ( $\pm 0.0699$ )	12.59% ( $\pm 13.54\%$ )	70.21% ( $\pm 30.35\%$ )
CNN		dense1	0.7325 ( $\pm 0.0633$ )	4.22% ( $\pm 8.86\%$ )	20.71% ( $\pm 23.84\%$ )
FNN	✓	linear3	0.9193 ( $\pm 0.0093$ )	26.15% ( $\pm 13.82\%$ )	93.88% ( $\pm 5.38\%$ )
FNN		linear4	0.9657 ( $\pm 0.0160$ )	78.91% ( $\pm 28.70\%$ )	98.29% ( $\pm 0.88\%$ )
Transformer	✓	linear1	<b>0.9957 (<math>\pm 0.0008</math>)</b>	<b>100.00% (<math>\pm 0.00\%</math>)</b>	<b>100.00% (<math>\pm 0.00\%</math>)</b>
Transformer		linear1	0.9717 ( $\pm 0.0188$ )	85.06% ( $\pm 23.15\%$ )	99.23% ( $\pm 2.32\%$ )
<b>Infiltration</b>					
CNN	✓	dense3	0.9734 ( $\pm 0.0163$ )	<b>88.62% (<math>\pm 14.60\%</math>)</b>	98.17% ( $\pm 2.09\%$ )
CNN		dense3	0.9455 ( $\pm 0.0267$ )	71.56% ( $\pm 16.54\%$ )	89.31% ( $\pm 7.37\%$ )
FNN	✓	linear4	0.9270 ( $\pm 0.0205$ )	42.94% ( $\pm 18.52\%$ )	90.21% ( $\pm 5.49\%$ )
FNN		linear5	0.9548 ( $\pm 0.0233$ )	75.39% ( $\pm 11.27\%$ )	91.21% ( $\pm 7.31\%$ )
Transformer	✓	linear2	<b>0.9742 (<math>\pm 0.0149</math>)</b>	86.69% ( $\pm 15.44\%$ )	<b>99.66% (<math>\pm 0.97\%</math>)</b>
Transformer		linear1	0.8951 ( $\pm 0.1352$ )	42.96% ( $\pm 22.82\%$ )	86.34% ( $\pm 25.29\%$ )
<b>SSH-Patator</b>					
CNN	✓	dense2	0.8117 ( $\pm 0.0638$ )	16.01% ( $\pm 6.63\%$ )	51.92% ( $\pm 14.64\%$ )
CNN		dense3	0.7010 ( $\pm 0.0357$ )	20.29% ( $\pm 7.06\%$ )	35.71% ( $\pm 6.86\%$ )
FNN	✓	linear3	0.9504 ( $\pm 0.0036$ )	56.52% ( $\pm 4.39\%$ )	98.89% ( $\pm 1.36\%$ )
FNN		linear4	0.9570 ( $\pm 0.0043$ )	68.97% ( $\pm 6.26\%$ )	97.78% ( $\pm 1.79\%$ )
Transformer	✓	linear1	<b>0.9921 (<math>\pm 0.0014</math>)</b>	<b>100.00% (<math>\pm 0.00\%</math>)</b>	<b>100.00% (<math>\pm 0.00\%</math>)</b>
Transformer		linear1	0.9028 ( $\pm 0.0528$ )	41.19% ( $\pm 22.36\%$ )	78.76% ( $\pm 19.19\%$ )

# Identifying Concept Drift

- We can also alert cybersecurity operators to potential changes in network traffic.
- Multiple types of safeguards perform similarly well, allowing users to choose methods based on computational considerations.

Layer	AU ROC (↑)	TPR (TNR 95%) (↑)	TPR (TNR 85%) (↑)		
Gaussian Kernel Density					
CNN	✓	dense2	0.9263 (±0.0139)	61.61% (±4.44%)	83.14% (±4.44%)
CNN		dense1	0.8840 (±0.0135)	46.80% (±2.62%)	68.15% (±5.61%)
FNN	✓	linear4	0.9079 (±0.0109)	56.06% (±5.32%)	80.31% (±2.68%)
FNN		linear4	0.8698 (±0.0119)	43.02% (±7.04%)	67.41% (±3.14%)
Transformer	✓	linear2	<b>0.9636 (±0.0126)</b>	<b>79.23% (±8.25%)</b>	<b>95.73% (±2.68%)</b>
Transformer		linear1	0.8157 (±0.0308)	36.81% (±7.10%)	56.46% (±6.31%)
Normalizing Flows					
CNN	✓	dense2	0.9263 (±0.0063)	55.98% (±1.29%)	83.03% (±2.72%)
CNN		dense2	0.8800 (±0.0083)	47.96% (±2.82%)	66.47% (±2.49%)
FNN	✓	linear4	0.8963 (±0.0112)	45.10% (±8.46%)	77.66% (±3.28%)
FNN		linear4	0.8620 (±0.0136)	29.48% (±6.46%)	65.24% (±5.88%)
Transformer	✓	linear3	<b>0.9583 (±0.0090)</b>	<b>77.71% (±4.49%)</b>	<b>94.44% (±2.24%)</b>
Transformer		linear1	0.8000 (±0.0509)	31.86% (±10.60%)	49.08% (±16.72%)

# Why Not Exclusively Use Transformers?

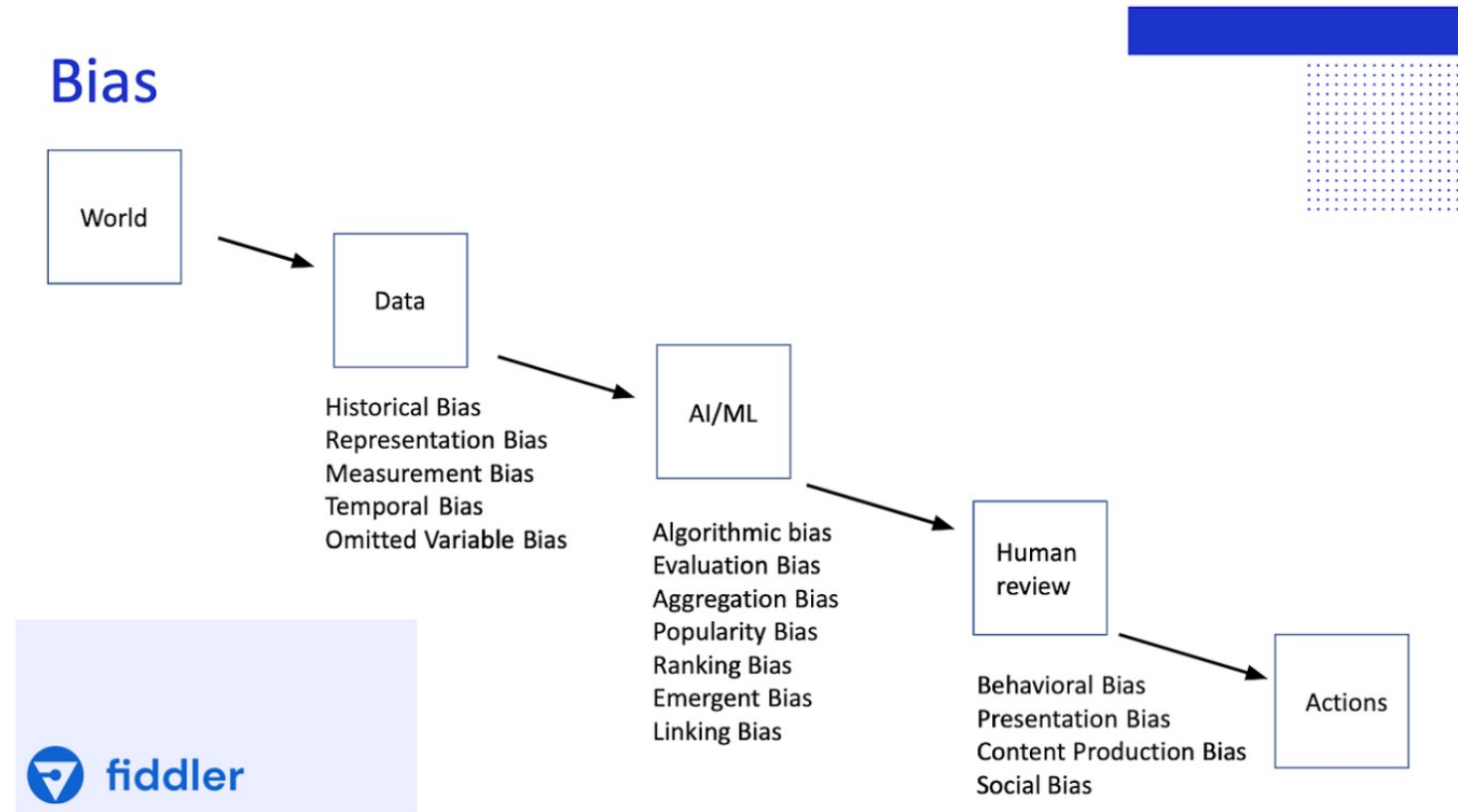
- Transformers outperform CNNs and FNNs on in-distribution accuracy and the ability for us to learn the feature representations for out-of-distribution detection.
- However, these successes come at a high computational cost.
- A real-world system should differentiate traffic based on importance of potential target and decide on the appropriate ML model to use.

Architecture	Context	Latency (↓)
CNN	✓	172.58 $\mu$ sec
CNN		129.86 $\mu$ sec
FNN	✓	144.60 $\mu$ sec
FNN		128.50 $\mu$ sec
Transformer	✓	2418.02 $\mu$ sec
Transformer		2420.89 $\mu$ sec

# High-Assurance Machine Learning

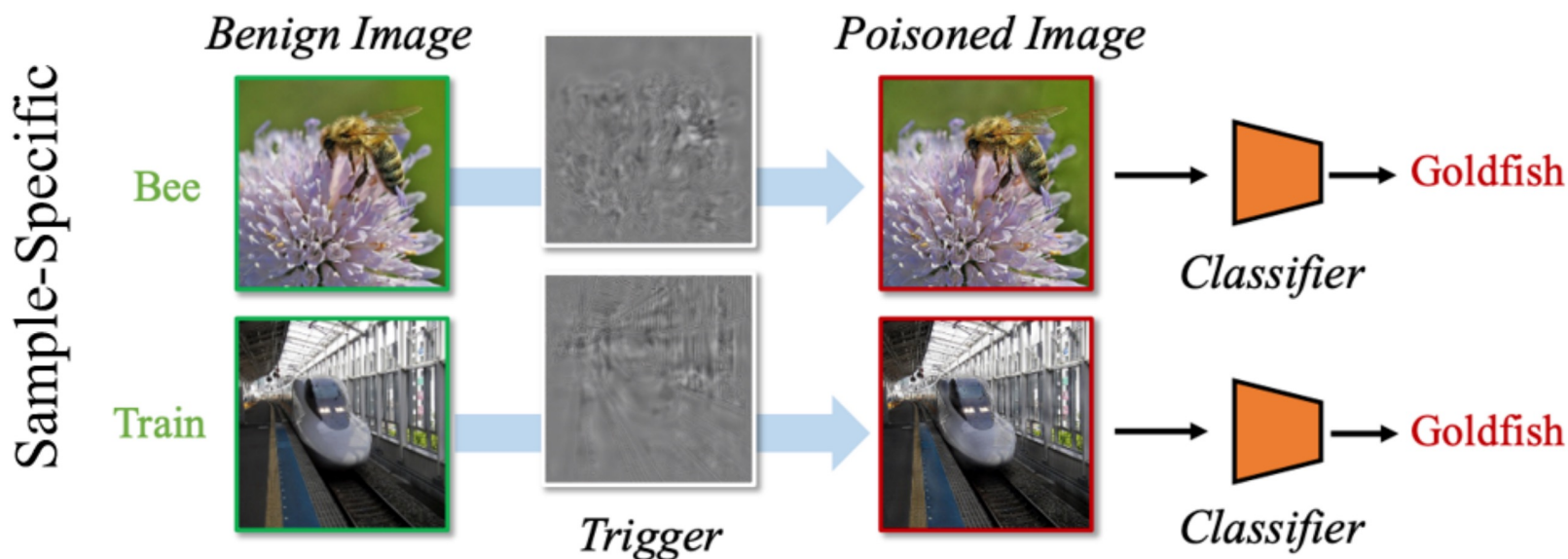
# Algorithmic Fairness

- Biases in the training data will cause ML models to exhibit the same biases.
- There are moral and can even be legal implications to using a biased ML system.



# Trojan Attacks

- With the cost of generating training data, many people rely on published neural network weights from outside sources.
- However, malicious agents can introduce Trojan backdoors to the models.
- These Trojans cause the model to misclassify otherwise trivial inputs.



# Large Language Model Hallucinations

- While impressive, large language models often hallucinate and produce factually incorrect statements.
- These models ingested nearly the entire internet during training, but do not have a knowledge base to draw on.



QUESTION OF THE DAY

## Why did Stanford take down its Alpaca AI chatbot?

Answer: "Hallucinations," among other things.

Thank You!



Questions?