Applications of Certainty Scoring for Machine Learning Classification in Multi-modal Contexts

Alexander M. Berenbeim, David A. Bierbrauer, Iain J. Cruickshank, Robert H. Thomson, Nathaniel D. Bastian, *Member, IEEE*

Abstract—Quantitative characterizations and estimations of uncertainty are of fundamental importance for machine learning classification, particularly in safety-critical settings such as the military battlefield where continuous real-time monitoring requires explainable and reliable scoring. Reliance on the maximum a posteriori principle to determine label classification can obscure a model's certainty of label assignment. We develop quantitative scores of *certainty* and *competence* based on predicted probability estimates as an effective tool for inferring the verity of positives across different data modalities and architectures. Our theoretical results establish that competent models have distinct distributions of certainty for true and false positives. Our empirical results bear out that there are distinct distributions of certainty scores on training and holdout data, as well as data that is a priori out-of-distribution. Further, we find that the most reliable test for out-of-distribution data is to compare the global True positive certainty score distribution against test data. At least 92.3% of out-of-distribution are successfully identified this way across our two experimental modalities at the tranche level. Further, 100% of the out-of-context images are identified as out-of-distribution using the stochastic form of our out-of-distribution detection test across all five stochastic variants of the ResNet models. Consequently, we find that the use of our certainty framework provides a robust means of detecting out-of-distribution inputs, while also serving as a reliable mechanism for comparing model quality of accurately distinguishing between true and false positives, particularly in safety-critical contexts.

Index Terms—Machine Learning Assurance, Uncertainty Quantification, Network Intrusion Detection, Image Classification.

1 Introduction

As machine learning systems are becoming increasingly embedded in everyday life, particularly within safety-critical settings, there is a growing need to develop robust techniques. These techniques should assure model quality in the face of uncertainty and enable human agents to identify and prevent the misapplication of automated decision-making. Traditional approaches to building intelligent applications and autonomous systems primarily rely on knowledge representations and symbolic reasoning whose implementations include programming condition-based rules, stateful logic encoded in finite state machines, and physics-based dynamics of environments and objects [12], [32], [46].

Although rule-following provides a form of assurance and can be readily subject to human scrutiny, they fare poorly when used in production autonomy applications when dealing with real-world uncertainty and high dimensional sensory data, which is necessary for perception and situation-understanding applications. The rule-set and stateful logic in these settings is often incomplete and challenged against encompassing an ever-evolving set of situations. Hybrid solutions inspired by human cognition benefit from both rule-like and statistical (sub-symbolic) properties. However, they are not necessarily scalable [11] (for some instances of attempts to increase scalability see [2], [5], [13]) and suffer when data is noisy and high dimensional [14].

Uncertainty quantification (UQ) techniques have provided an invaluable means towards this end, not only in providing a broad framework for articulating the forward propagation of uncertainty in models [28], [55], but also for providing frameworks for studying model and parameter uncertainty [7], [51]. In particular, these so-called inverse problems help with bias correction between experimental and mathematical models, as well as enable model developers a means to perform parameter calibration. Furthermore, UQ provides a means for assessing discrepancies between the data that models are trained on and the often highly divergent data production models encounter.

We are primarily interested in this latter use of UQ to establish the trustworthiness of machine learning models in real-world safety-critical settings where tasks span multiple modalities, such as those present on the modern military battlefield. In establishing trustworthiness of models, we seek to establish resilience in light of shifts in data, both natural and adversarial, which are intentional efforts to thwart model development and impair decision making.

Our work addresses the general challenge of providing machine learning assurance that when encountering novel data, classification models are certain in their predictions, and with giving human agents a means of detecting if inputs are out-of-distribution relative to the training and validation data that a model has relied on. In particular, our work aims to address the relative paucity of resilient UQ methods in two safety-critical settings with relevance to the modern military battlefield: zero-day network intrusion detection, and out-of-context image classification for battlefield sensors.

A. Berenbeim, D. Bierbrauer, I. Cruickshank, R. Thomson and N. Bastian are with the United States Military Academy, West Point, NY, 10996, USA. Emails: {alexander.berenbeim, david.bierbauer, iain.cruickshank, robert.thomson, nathaniel.bastian}@westpoint.edu.

Advancements in distributed computing and small form factor devices have the capacity and capability to revolutionize warfare, with many sensors and other devices interconnected across multiple warfighting domains. Known as the Internet of Battlefield Things (IoBT), this future battlefield will rely on intelligent systems to properly transport heterogeneous, multi-modal data across networks to the point of impact to enable successful military operations [24].

Some of the artificial intelligence (AI) tasks relevant to the IoBT include securing the network, image recognition and detection, and bringing intelligence to edge sensors. In order to ensure the models used in the IoBT are robust and resilient against novel adversarial methods – including zero-day attacks – methods to detect out-of-distribution inputs for multi-modal data must be developed. Current machine learning methods produce deterministic outputs and train well on existing datasets, but often stumble in real-world applications when confronted with novel attacks [37], [44].

Another important IoBT task is the proper detection and classification of images from these interconnected sensors, which are used to establish *context*. In order to reduce harm and minimize adversarial advantage, detection of novel inputs that are out-of-context is necessary to inform human decision makers. Specifically, images that are out-of-context are those that are out-of-distribution (OOD).

Given that the IoBT will be an adversarial environment with degraded signal and adversarial poisoning of information, the need for robust UQ methods to provide assurance in the open-world setting [63] motivated our work to develop a mathematical framework that provides resilient guarantees across modalities, tasks, architectures, and statistical frameworks (particularly, between deterministic/frequentist and stochastic/Bayesian frameworks) in safety-critical settings. Further, applying uncertainty quantification techniques to Bayesian deep learning models has enabled out-of-distribution detection on individual inputs, particularly when identifying adversarial attacks [47], [54].

Having identified the importance of UQ for OOD for these two safety-critical domains, and the ability of applying UQ to Bayesian models in order to perform OOD detection on individual inputs, we make the following contributions:

- A mathematically rigorous theoretical framework of *certainty*, *competence*, and *doubt* that provides intrinsic scores of *any* classification model architecture's degree of certainty for a given prediction, and a groundwork for penalty-functions for future model training that optimizes around competence;
- A non-parametric statistical test for out-of-distribution detection test using the *distribution* of *certainty scores*, with reliability guarantees derived from the above theoretical framework;
- Significant computational experiments demonstrating the robustness of this framework for out-of-distribution detection across multiple domains, modalities, tasks, architectures, datasets, and statistical frameworks;
- Evidence that even minor Bayesian sampling on top of pre-trained deterministic models can be used to build ensemble models with improved accuracy and competence, while also enabling OOD at the edge.

Section 2 provides a detailed literature review while Section 3 describes our mathematical contributions to the theory of uncertainty quantification for machine learning assurance. Section 4 details the data, modeling architectures, computational experimentation, and evaluation metrics used to assess performance of our proposed certainty scoring approach. Section 5 provides and discusses the results of our experiments, while Section 6 summarizes key impacts and proffers future directions for this research.

2 Literature Review

2.1 Uncertainty Quantification

Uncertainty quantification has seen a rise in use for various types of deep learning applications. Uncertainty estimation forms a significant component of software testing for software containing deep learning elements. Previous research has found various techniques from uncertainty estimation effective in reducing defects and vulnerabilities of software containing deep neural networks [52], [65]. The main methodology for employing uncertainty quantification techniques for reducing defects in software containing deep neural networks is to use a *supervisor* module in a deep neural network software system that evaluates whether a given prediction by the deep neural network should be trusted or not. Recent empirical work has found measures developed for uncertainty quantification to work well for the supervisor module [65]. As such, these uncertainty estimation techniques have been thoroughly empirically validated for their use in software engineering [65], and a Python package is even available for developers to implement many of these uncertainty quantification techniques as part of their software development [53].

One particular uncertainty estimation technique used in the software engineering of deep learning-enabled systems is the Prediction Confidence Score [42]. Zhang et al. use the intuition that a more certain prediction for a machine learning classification is one that has a large gap in SoftMax scores between the most probable and second most probable classes. They use this score to successfully sort benign from adversarial examples. This technique is noteworthy in the context of our research, as we build upon a similar instantiation but from a different perspective and for a different objective. Namely, we derive the same measure based on the geometric properties of SoftMax scores as well as use this intuition to derive fully new measures for machine learning models, which can be used for out-of-distribution detection. Overall, these studies have shown that machine learning-enabled software has successfully used uncertainty quantification to develop more robust and resilient software.

Our contributions to UQ aim to inform model development from a decision-theoretic perspective and are intended for the same level of general use as above. Whereas earlier literature relied on low confidence scores to detect out-of-distribution data [35], our UQ scores and corresponding out-of-distribution detection test compares the distribution of these scores on unlabeled inputs against the prior distributions gathered from training and validation data. One immediate advantage of this test is that it avoids mistakenly identifying as OOD low confidence inputs when a model correctly classifies a predicted label with low confidence,

or otherwise overfits within a label and determines similar scores to both true and false positives.

2.2 Network Intrusion Detection

The security of any network, including the IoBT, should be considered as a first principle. Network Intrusion Detection Systems (NIDS) play a critical role in network security. These systems are engineered to detect unauthorized use of computer and network systems by both internal and external users. NIDS has seen recent and successful applications of artificial intelligence and machine learning techniques, with deep learning techniques such as Variational Autoencoders, Deep Neural Networks, Convolutional Neural Networks, Recurrent Neural Networks, and Belief Neural Networks accounting for the most notable recent successes [10], [45], [49]. Although these models often produce high-accuracy predictions, they are primarily deterministic [20], [34], and few have been evaluated on their ability to detect OOD data [62]. As it stands, there is a paucity of research into uncertainty quantification for NIDS despite the risks posed by emerging cyber threats. Most contemporary NIDS researchers rely on the following simulated Netflow datasets to train their models: KDD Cup'99, Kyoto 2006+, NSL-KDD, UNSW-NB15, CIC-IDS2017, and CSE-CIC-IDS2018 [21], [23], [31], [34], [36], [45].

NIDS fall into two main categories: signature-based and anomaly-based [22]. Signature-based systems identify potentially harmful network traffic by cross-referencing a database of known malicious signatures and attack strategies. In contrast, anomaly-based systems strive to establish a baseline of normal behavior and flag any activity that strays from this standard. Given the dynamic nature of cybersecurity threats, adaptive security measures that combine these methodologies are the preferred choice [3], [38].

Packet capture data provides a granular view of a network by providing raw datum containing header information and payload at a standardized size. Due to the frequency of packets through a network and the granularity of the datum, models trained on packet capture data may be able to detect cyber threats in-time when compared to those trained on Netflow data, which requires communication to end. In particular, recent research using deep learning for NIDS with packet-level data has produced promising results with high accuracy [35], [56], including edge network environments [61]. However, this research on packet-level data has not investigated detecting out-of-distribution inputs.

In order to ensure the models used in the IoBT are robust and resilient against novel adversarial methods – including zero-day attacks – methods to detect out-of-distribution inputs for multi-modal data must be developed. Our contributions to this space are novel because OOD inputs at packet capture level is not presently in the literature, and can be used to identify OOD inputs, such as zero-days.

2.3 Image Classification

Generative AI models are becoming increasingly capable of producing adversarial input data such as images for either positive [43] or nefarious [40], [48], [50] intent. This evolution necessitates the creation of more resilient models that can provide assurance and remain effective against the emergence of new threats. Object detection and recognition models invariably depend on the implicit contextual information of an image, in the sense that the objects identified in an image co-vary with each other and other scene properties (i.e., the *context*) across images. Most image detection and identification systems are trained on images in a natural context, like a plate of food on a restaurant table. When an image of a plate of food is superimposed in an out-of-context environment, such as underwater, the surrounding visual information can negatively affect a model's ability to detect or identify the out-of-context image. Adversaries can exploit this weakness in order to make hostile objects harder to detect or properly identify.

Deep neural networks have acquired a reputation for processing visual information at scale, with convolutional neural network block architectures such as AlexNet providing a successful research direction for computer vision problems such as image detection and classification [19].

The ResNet architectures developed in [15] won the 2015 ImageNet competition, outperforming the 2012 winner AlexNet. The ImageNet project is an image database organized by the WordNet hierarchy, and the challenges issued by the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) from 2010 to 2017 have been instrumental in organizing and providing open source data for the development of computer vision algorithms [18].

When considering the application of context integration, ResNet appears to be a popular architecture for implementation. A survey from Wang and Zhu [64] shows ResNet is one of the most used architectures for both image- and video-based context approaches. Thus, the performance, popularity, and availability of pre-trained ResNet models make them suitable for the baseline testing of our approach in the deterministic and stochastic settings discussed in Section 4.

Networks trained with natural object datasets such as ImageNet implicitly rely on the labeled objects occurring in appropriate contexts, and networks trained on these datasets have repeatedly failed to identify objects that are placed out-of-context [8], [25], [27], [29], [33], [41]. Additionally, Zhang et al. [41], with support from human psychophysics experimental results that provide a benchmark for the computational models– including for ResNet architectures, identify ten properties of where, when, and how context modulates image recognition. Acharya et al. [58] trained a ResNet50 network for feature extraction as part of their large-scale OOC experiments for detecting OOC objects using contextual clues. This latter research broadly informs our experimental work, not the least by providing the COCO OOC dataset.

Our contributions in part ameliorate the issues posed by out-of-context images for visual network development by providing an architecture independent means of assessing when data diverges from the training and test distributions used in model development.

3 Theoretical Contributions

In full, our theoretical contributions are:

- The theoretical framework of *certainty* and *competence* as intrinsic features of predictors $\mathbf{p}$ in multi-class classification decision problems;
- The *competence hierarchy*, which furnishes a qualitative characterization of a model $\mathbf{p}$'s performance at approximating $\mathbf{q}$, and quantitative bounds on $\mathbf{p}$'s performance at distinguishing the distributions of True and False positives, both with respect to the means of distributions, as well as with respect to their medians if models are relatively α-expert;
- A two stage non-parametric out-of-distribution test for determining if an input sample distribution belongs to the underlying distribution of true positives on which $\mathbf{p}$ was trained;
- Establishing that HMC for Bayesian inference within our certainty framework enabled the extension of our out-of-distribution test to out-of-distribution detection at the level of individual outputs.

3.1 Classification Certainty and Competence

Given a random variable $X : \Omega \to \mathbb{R}^n$ subject to a probability distribution $q(x)\,dx$, where dx is the Lebesgue measure on $\mathbb{R}^n$, a *learning machine* is a conditional probability density function $p(\mathbf{x}|\omega)$ with $\mathbf{x} \in \mathbb{R}^n$ and $\omega \in A \subseteq \mathbb{R}^W$, where A denotes a parameter space of weights, which attempts to approximate the true density function q [4]. When given a set of random samples $D_N = \{X_1, X_2, \ldots, X_N\}$, a statistical learning machine is a function approximating the true probability density function drawn from the sample set D_N.

In multi-class classification decision problems on d distinct labels, we assume the presence of an underlying learning machine $f((\mathbf{x}, y); \omega)$, with input features $\mathbf{x} \in \mathbb{R}^n$, weights $\omega \in \mathbb{R}^W$, and discrete label $y \in [d]$. Upon receiving $\mathbf{x}$, the learning machine outputs a vector of likelihoods for each label. These likelihoods are then transformed into a probability vector $\mathbf{p} \in \Delta^d$ following the application of a normalizing function, such as SoftMax.The learning machine generates a parametric model $p(y|\mathbf{x}; \omega)$. We specifically explore situations where the decision maker is a neural network $f : \mathbb{R}^n \to \mathbb{R}^d$, with input features drawn from $\mathbb{R}^n$, and whose weights represent the parameters of the neural network trained according to our loss function. This discussion can also be generalized to encompass classical multi-classification problems.

The concepts of certainty, competence and doubt were introduced in [60], and were developed with multi-class classification decision problems in mind. Here, the decision maker is conceived as a neural network, but this concept can be readily generalized to arbitrary discriminant functions $\gamma : \mathcal{X} \to [d]$ as discussed in decision science literature [26]. Across d distinct labels, we assume that we have a decision maker $f(\mathbf{x}; \omega)$ and corresponding predictor $\mathbf{p}_\omega : \mathbb{R}^n \to \Delta^d$ determined by weights ω, whichupon receiving input $\mathbf{x} \in \mathbb{R}^n$ will output a probability vector in Δ^d. We encode the *certainty* of our predictor by first computing the following skew-symmetric matrix:

$$\mathbf{C}^o(\mathbf{p}) := \mathbf{1}\mathbf{p}^\top - \mathbf{p}\mathbf{1}^\top \tag{1}$$

and then setting the *certainty* of a probability vector $\mathbf{p}$ to be the following matrix:

$$\mathbf{C}(\mathbf{p}) = \mathbf{I} + \mathbf{C}^o(\mathbf{p}). \tag{2}$$

Equations 1 and 2 can be understood as *component certainty* and *complete component certainty* respectively, with the completion with respect to the certainty we have when comparing a choice with itself. In particular, in Equation 2 each matrix coordinate score describes the *pairwise certainty*, defined by Equation 3, between labels with repeats, as:

$$\mathbf{C}_{ij}(\mathbf{p}) = \begin{cases} 1 & i = j \\ (\pi_j - \pi_i)(\mathbf{p}) \equiv p_j - p_i & i \neq j \end{cases} \tag{3}$$

From pairwise certainty, we then set the *certainty score* for probability vector $\mathbf{p}$ by Equation 4

$$\begin{aligned} \varsigma(\mathbf{p}) &:= \min_i \{\mathbf{C}_{ij}(\mathbf{p}) \colon j = \arg\max_k \sum_i \mathbf{C}_{ik}(\mathbf{p})\} \\ &:= \min_{i \in [d]} \{|\delta_{ij} + p_j - p_i| \colon p_j = \max_{k \in [d]} \{\pi_k(\mathbf{p})\}\} \\ &:= \mathbf{p}_{\hat{j}} - \mathbf{p}_{\check{j}} \\ &:= \pi_{\hat{j}}(\mathbf{p}) - \pi_{\check{j}}(\mathbf{p}), \end{aligned} \tag{4}$$

where $\mathbf{p}_{\hat{j}}$ denotes the highest probability in $\mathbf{p}$, and $\mathbf{p}_{\check{j}}$ the next highest probability so that the certainty score of a sample prediction is the difference between the probability of the label predicted by the maximum a posteriori (MAP) principle and the next greatest probability.

The penultimate equivalent definition of certainty score appears earlier [42] under the name Prediction Confidence Score, although our work was derived independently and with respect to the full *certainty* of a probability vector on which we derive our proposed cost function in suggested in Section 6.

The following is a collection of straightforward properties of certainty:

Proposition 1. *For all* $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$,

1) $\mathrm{Tr}(\mathbf{C}(\mathbf{x})) = d$;
2) $\mathbf{C}(\mathbf{x}) \in \mathrm{GL}_d(\mathbb{R})$;
3) $\langle \mathbf{x}, \mathbf{C}(\mathbf{y})\mathbf{x} \rangle = \|\mathbf{x}\|^2 \geq 0$.

From this, we can conclude that *certainty* is a mapping sending $\mathbf{C} : \Delta^d \to \mathrm{GL}_d(\mathbb{R})$.

Geometrically understood, ς is a map from $\Delta^d \to [0, 1]$ whose zero locus is geometrically understood as the subsimplices passing through the barycenter of Δ^k and the barycenters of the boundary components, $\partial\Delta^k$.

With D_N denoting the set of N samples $(\mathbf{x}, y)$ where $\mathbf{x}$ is drawn from sample space Ω, and y drawn from $[d] = \{1, \ldots, d\}$, for a predictor $\mathbf{p} : \Omega \to \Delta^d$, we denote by $\mathbf{p}^i$ the probability vector of $\mathbf{p}(\mathbf{x}_i)$, the i^{th} sample of D_N. Further, we denote by $\hat{p}^i := \arg\max_j \pi_j(\hat{p}(\mathbf{x}^i))$, and by $\mathcal{Y}(i)$ the true label for the i^{th} sample.

The following proposition follows immediately from the definitions.

Proposition 2. For all $\mathbf{p} \in \Delta^d$,

$$0 \leq \varsigma(\mathbf{p}) \leq \max_{k \in [d]} \{\pi_k(\mathbf{p})\}.$$

Our notion of certainty and certainty scoring in turn led to a scoring system to evaluate the *competence* of a

learning machine f. In particular, we define *component competence* with respect to the probability vectors $\mathbf{p}$ derived from $f(\cdot;\omega)$, and $\mathbf{q}$, the probability vector derived from the empirical distribution of label assignments from $D_N := (\mathcal{X}, \mathcal{Y}) \in (\Omega \times [d])^N$ in Equation 5.

$$\begin{aligned}\mathcal{CC}(\mathbf{p},\mathbf{q};D_N) &= \frac{1}{dN}\sum_{i\in[N]} \mathbf{1}^\top \mathbf{C}^o(\mathbf{p}(\mathbf{x}^i))\mathbf{q}(\mathbf{x}^i) \\ &= -\frac{1}{d} + \frac{1}{N}\sum_{i\in[N]} \langle \mathbf{p},\mathbf{q}\rangle(\mathbf{x}^i)\end{aligned} \tag{5}$$

and similarly, using Equation 2 and complete component certainty, we define *complete component competence* in Equation 6 as:

$$\begin{aligned}\mathcal{CCC}(\mathbf{p},\mathbf{q};D_N) &= \frac{1}{dN}\sum_{i\in[N]} \mathbf{1}^\top \mathbf{C}(\mathbf{p}(\mathbf{x}^i))\mathbf{q}(\mathbf{x}^i) \\ &= \frac{1}{N}\sum_{i\in[N]} \langle \mathbf{p},\mathbf{q}\rangle(\mathbf{x}^i)\end{aligned} \tag{6}$$

with the second equivalent definition following from unfolding

$$\mathbf{1}^\top \mathbf{C}^o(\mathbf{p})\mathbf{q} = \langle \mathbf{1}, \mathbf{1}\mathbf{p}^\top\mathbf{q} - \mathbf{p}\mathbf{1}^\top\mathbf{q}\rangle.$$

Given these identities, $\mathcal{CCC}$ is the expected probability that a random assignment of $\mathbf{x}$ according to $\mathbf{p}$ correctly predicts the random assignment of $\mathbf{x}$ according to $\mathbf{q}$, which we denote by $\mathbf{p} \equiv \mathbf{q}$; consequently, $\mathcal{CC}$ is the improvement of $\mathbf{p}$ over relying on a uniform distribution of labels.

We observe that in principle $\mathcal{CCC}(\mathbf{q},\mathbf{q};D_N) = 1$ whenever we are trying to approximate a deterministic process with our learning machine. However, when trying to learn a stochastic process, $\mathcal{CCC}(\mathbf{q},\mathbf{q};D_N) < 1$ for sufficiently large N. Whenever $\mathcal{CCC}(\mathbf{p},\mathbf{q}) \geq \mathcal{CCC}(\mathbf{q},\mathbf{q})$, we say that $\mathbf{p}$ has *mastered* $\mathbf{q}$. Without loss of generality, we will omit $\mathbf{p}$ and $\mathbf{q}$ when discussing competence, using $\mathcal{QC}$ to denote $\mathcal{CCC}(\mathbf{q},\mathbf{q})$.

Component competence is used in contrast with *empirical competence* (see Equation 7), which we define with respect to the observed certainty scores on our training data. We defined empirical competence to score and compare the quality of model predictions by penalizing models with high certainty false positives. Specifically, empirical competence is the difference between the sum of certainty scores of the true positives and the certainty scores of the false positives, divided by the total number of predicted positives. We formally define empirical competence as:

$$\begin{aligned}&\mathcal{K}(\mathbf{p};D_N) = \\ &\frac{1}{N}\sum_{i\in[N]} \varsigma(\mathbf{p}^i)\left[1\{j\colon \mathcal{Y}(j) = \hat{\mathbf{p}}^j\} - 1\{j\colon \mathcal{Y}(j) \neq \hat{\mathbf{p}}^j\}\right](i).\end{aligned} \tag{7}$$

We use the order relationship between $\mathcal{K}$, $\mathcal{CC}$, and $\mathcal{CCC}$ to define the following competence hierarchy relative to $\mathbf{q}$ and samples D_N. Let $\alpha \in [0,1)$.

If $\mathcal{CC} \leq \alpha$, then $\mathbf{p}$ is *relatively α-uninformed*; otherwise, we say $\mathbf{p}$ is *relatively α-informed*. Whenever $\mathbf{p}$ is relatively α-informed, then the average estimated probability for the true label of $\mathbf{x}^i$ is greater than the uniform probability plus α. Whenever $\mathbf{p}$ is relatively α-informed, we have the following *competence hierarchy with respect to empirical competence* such that:

- Whenever $\mathcal{K} < 0$, we say $\mathbf{p}$ is *incompetent*;
- When $0 \leq \mathcal{K} < \alpha$, we say $\mathbf{p}$ is *relatively α-amateur*.
- Whenever $\alpha \leq \mathcal{K} < \mathcal{CC}$, we say $\mathbf{p}$ is *relatively α-competent*.
- Whenever $\mathcal{CC} \leq \mathcal{K} < \mathcal{CCC}$, we say $\mathbf{p}$ is *relatively α-expert*.
- Whenever $\mathcal{CCC} \leq \mathcal{K}$, we say $\mathbf{p}$ is *relatively prescient*.

We defined empirical competence to rely on certainty scores given their utility when the task is to approximate a d-coloring of Ω. Further, in the empirical setting $\mathcal{CCC}$ is the average probability assigned by the learning machine to the observed label.

Our use of certainty scoring was motivated by the need for machine learning model assurance, model explainability, and to perform out-of-distribution testing. In particular, we find that the distribution of certainty scores within labels and across the entire domain can be used to infer when data has been mislabelled by a learner when models have sufficiently high competence, and moreover, can be used to perform out-of-distribution detection on datum whenever sufficiently high competence has been established.

The following theorem establishes the key inferential relations between the α-competence hierarchy and the distribution of certainty scores:

Theorem 1. *Suppose that $\mathbf{q}$ is an almost everywhere continuous d-coloring of Ω, i.e. for all $\mathbf{x} \in \pi_\Omega(D_N)$, there is an $i \in [d]$ such that $\mathbf{q}(\mathbf{x}) = \mathbf{e}_i$ so that $\langle \mathbf{p}^i, \mathbf{q}^i\rangle = \pi_i(\mathbf{p})$.*

Further, suppose that for some $\alpha \in [0,1)$ that $\mathbf{p}$ is α-informed. Let $C^T_{\mathbf{p}} : \Omega \to [0,1]$ and $C^F_{\mathbf{p}} : \Omega \to [0,1]$ denote the random variable representing the certainty score of $\mathbf{p}$ when $\mathbf{p}$ predicts a true positive and false positive respectively with probability distributions ρ, ν respectively defined accordingly from $\mathbf{p}$ and $\mathbf{q}$ with sample averages $\bar{\mu}_\rho$ and $\bar{\mu}_\nu$ (respectively). Further, let $\check{X}$ denote the bounded random variable of the second-greatest alternate probability. Finally let $N^T \in [N]$ denote the number of samples correctly predicted by $\mathbf{p}$ and $N^F = N - N^T$ the number of incorrect prediction. Then :

1) *Whenever $\mathbf{p}$ is relatively α-competent, either $\bar{\mu}_\rho > \bar{\mu}_\nu + \alpha$, or otherwise*

$$\left(\frac{N}{N^T}\right)\alpha \leq \bar{\mu}_\rho \leq \min\left\{1, \left(\frac{1}{N^F}\right)\frac{N^T}{N}\right\}$$

and

$$\left[\frac{N^F}{N^T}\right]\bar{\mu}_\rho \leq \bar{\mu}_\nu \leq \min\left\{1, \left[\frac{N^T}{N^F}\bar{\mu}_\rho - (\frac{N}{N^F})\alpha\right]\right\};$$

2) *Whenever $\mathbf{p}$ is relatively α-expert, and with $\bar{\check{X}} =: \frac{1}{N^T}\sum \mathbf{p}^i_{\hat{j}}$, either:*

   a) *$\bar{\check{X}} < \frac{1}{d}$ and for all $a > 0$,*

$$\mathbb{P}\{\check{X} \geq \frac{1}{d} + \frac{a}{4}\} \leq \frac{1}{1+a^2}.$$

   or

   b) *$\bar{\check{X}} \leq \frac{N}{dN^T}$ and for all $a > 0$*

$$\mathbb{P}\{\check{X} \geq \frac{N}{dN^T} + \frac{a}{4}\} \leq \frac{1}{1+a^2}.$$

3) $\mathbf{p}$ *relatively prescient implies* $\mathbf{p}$ *must have* $N^T = N$ *on* D_N. *Further,* $\mathbf{p}$ *will be* α*-informed for all* $\alpha \in [0, 1)$.

Theorem 1 is intended to capture intuitions regarding hypothesis testing given the sample averages of the certainty score of true and false positives. In particular, we identify that while high competence is desirable on our sample data, for the purposes of inference, a trade-off may need to be made between accuracy and the α hyperparameter indicating a degree of competence.

Immediately, every relatively α-informed, α-competent $\mathbf{p}$ will either separate the expected values of the certainty score for true and false predictions, or otherwise, the number of false positives from our sample must be bounded above by $\lfloor \min \left\{ \frac{\left(N^T/N\right)^2}{\alpha}, N^T(\bar{\mu}_\rho - \alpha) \right\} \rfloor$, which can be confirmed by manipulating the two intervals bounding $\bar{\mu}_\rho$ and $\bar{\mu}_\nu$.

Whenever α doesn't separate $\bar{\mu}_\rho$ and $\bar{\mu}_\nu$, and $\mathbf{p}$ is α-competent, then for any $\alpha \geq \frac{1}{30}$, we effectively must have too few false positives for the purposes of inferring if the certainty of $\mathbf{p}$ on an arbitrary $\mathbf{x} \in \Omega$ will be distributed according to either the true or false positives of $\mathbf{p}$. In this respect, whenever selecting α as a target threshold for competence, we must weigh the trade-off between sample accuracy and separating true and false positives on our sample by a distance sufficient for the purposes of significance testing. α-expert $\mathbf{p}$ are desirable because in addition to the separation property of α-competence, the model will have high certainty relative to the number of alternative choices, with at most approximately 20 percent of samples having certainty scores close to zero.

3.2 Doubt

When certainty is low, *doubt* about the given model's prediction is high. The connection between certainty and doubt is realized in the geometric context of the real projective line, as discussed in [60].

We define the doubt of $\mathbf{p}$ as:

$$\mathbf{D}(\mathbf{p}) = \mathrm{Inv}(\mathbf{C}(\mathbf{p})) - \mathbf{I}$$

where Inv is the element-wise inverse function defined on $\hat{\mathbb{R}}^{d\times d}$ such that 0 and ∞ are assigned as reciprocals.

When working with sparse or otherwise large matrices, the doubt scores can be computed pairwise by computing *pairwise doubt*, defined in Equation 8 where $j = \arg\max_i \pi_i(\mathbf{p})$.

$$\delta_i(\mathbf{p}) = \begin{cases} 0 & i = j \\ \dfrac{1}{\pi_j(\mathbf{p}) - \pi_i(\mathbf{p})} & i \neq j \end{cases} \tag{8}$$

In [60], we present the argument that the projective line characterization of certainty/doubt can provide a differentiable function to be composed with the outputs of SoftMax as part of the optimization process.

The projections of the iterated Segre map are with respect to the components fixed by the columns of the certainty of $\mathbf{p}$ and the final component, ie $[\prod_i \mathbf{C}_{ij}(\mathbf{p}) : 1]$. The intuition behind this is that minimizing doubt will be guaranteed to maximize certainty, and vice versa.

Finally, we define *raw certainty* and *raw doubt* with respect to the logits themselves (see Equations 9 and 10 respectively).

$$\xi_{ij}(f) = \begin{cases} 1 & i = j \\ \pi_j(f(\mathbf{x})) - \pi_i(f(\mathbf{x})) \equiv (\pi_j - \pi_i)(f(\mathbf{x})) & i \neq j \end{cases} \tag{9}$$

$$\rho_{ij}(f) = \begin{cases} 0 & i = j \\ \dfrac{1}{\pi_j(f(\mathbf{x})) - \pi_i(f(\mathbf{x}))} & i \neq j \end{cases} \tag{10}$$

3.3 Loss Functions Shaped By Certainty and Doubt

We propose that a cost function for optimizing in our certainty framework be define by the minimum angle determined by applying the stereographic diffeomorphism to projections of an iterated Segre map (see [9] for further details).

For our cost function, the two natural candidates for assigning penalty per sample k are the doubt-cost (Equation 11) and the raw-doubt cost (Equation 12).

$$\theta_{cost}(\mathbf{p}^{(k)}) = \min_j \arcsin \left(\frac{1 - (\prod_i \mathbf{C}_{ij}(\mathbf{p}^{(k)}))^2)}{1 + (\prod_i \mathbf{C}_{ij}(\mathbf{p}^{(k)})^2)} \right), \tag{11}$$

$$\theta_{cost}(f(\mathbf{x}^{(k)})) = \min_j \arcsin \left(\frac{1 - (\prod_i \xi_{ij}(f(\mathbf{x}^{(k)}))^2)}{1 + (\prod_i \xi_{ij}(f(\mathbf{x}^{(k)}))^2)} \right). \tag{12}$$

The primary difference between Equations 11 and 12 stems from a decision whether to apply softmax before taking the differences to determine θ or take the difference of the logits produced by f to determine θ. In the former case, $\theta \in [0, \frac{\pi}{2}]$ while in the latter case $\theta \in (-\frac{\pi}{2}, \frac{\pi}{2}]$ following whether we have unbounded raw-doubt scores, which only occur as certainty approaches 1.

For the multi-class classification decision problem with loss function

$$L(f(\mathbf{x}), y) = 1\{f_y(\mathbf{x}) \leq f_i(\mathbf{x}) \text{ for some } i \neq y\},$$

the goal of minimizing L-risk is equivalent to minimizing the misclassification probability and thus, when using the raw-doubt cost, we could minimize the expected loss from the adjusted loss function

$$D(f(\mathbf{x}), y) = \chi(\theta_{cost}(f(\mathbf{x})))L(f(\mathbf{x}), y),$$

where χ is a penalty-function that monotonically decreases on $\theta_{cost}(f(\mathbf{x}))$ so that we realize greater losses when we misclassify with greater certainty than with less certainty.

Although the doubt-cost and raw-doubt cost functions naturally arise from considering the underlying determinantal variety of the iterated Segre embedding, they suffer the drawback of being neither convex nor concave functions without further restriction. Since the squared Euclidean matrix norm is convex, one computationally tractable convex penalty function defined with respect to the entire certainty is:

$$\chi(\mathbf{p}) = \frac{\|\mathbf{C}(\mathbf{p})\|_2^2}{d^2} + \|\mathbf{p}\|_2^2$$

with corresponding adjusted loss function

$$D(f(\mathbf{x}), y) = \chi(\sigma(f(\mathbf{x})))L(f(\mathbf{x}), y),$$

where σ is the SoftMax function. We may refer to such a penalty function $\chi(\mathbf{p})$ as the *gross certainty* of $\mathbf{p}$.

Although gross certainty is strictly convex when differentiating with respect to coordinates from Δ^d, the Hessian of $\chi(f(\mathbf{x}))$ is not necessarily positive-semi definite with respect to a coordinate system in terms of $\mathbf{x}$ or ω. For this reason, it would be worthwhile to explore the empirical trade-offs when training models using the different penalty functions and corresponding family of loss functions vis a vis model performance and competence.

3.4 Out-of-Distribution Detection with Two Stage Mann-Whitney U-Test and ECDF Tests

We developed an out-of-distribution detection test using our certainty framework that compares the distribution of certainty scores of test data against the distribution scores of the training and validation data. This test has two stages and two variants; the first is intended for application to test data which is randomly drawn, while the second is intended to be applied to individual data from which we have formed a Bayesian sample, such as the ones formed using Hamiltonian Monte Carlo in our stochastic models.

3.4.1 Mann-Whitney U Test

When test samples are randomly drawn, our out-of-distribution detection test uses the Mann-Whitney U test, which is a nonparametric test whose null hypothesis is that random draws from samples X and Y have an equal likelihood of dominating one another, with the alternate hypothesis being that one distribution stochastically dominates the other. The general formulation assumes that:

1) The distribution under the null hypothesis is known;
2) Observations from both samples are independent;
3) The sample populations are linearly ordered;
4) Under the null hypothesis the H_0, the distributions of X and Y are identical,
5) Under the alternative hypothesis, the distributions are not identical, such that the following test statistic is consistent only when one distribution dominates the other.

Each assumption is satisfied in our general context when working with certainty scores, provided we assume the distribution under the null hypothesis to be the empirical certainty score distribution of all samples (either globally, or within a given label). From Theorem 1, sufficiently competent models will have distinct TP and FP distributions both globally, and provided sufficient competence and sample sizes within labels, locally as well.

Finally, the two stage, global-local test we developed can substitute other general non-parametric tests for the U-test, such as Kolmogorov-Smirnov.

3.4.2 Empirical Cumulative Distribution Function Test

For our Bayesian models, in-lieu of Mann-Whitney U (MWU) test, we developed the following test using parameters $\beta, \gamma, \delta \in (0, 1)$ such that $\gamma < \delta$.

With control data $\mathcal{D}$, generate D many Bayesian samples with corresponding empirical CDFs $\mathbb{P}_i$ describing the image of functions f_i applied to $\mathcal{D}$. Given test data $t \in \mathcal{T}$, the Bayesian samples produce D many corresponding sample points $f_i(t)$ for each $t \in \mathcal{T}$. Modulo conditions on $\mathcal{D}$, we define an indicator function for $t \in \mathcal{T}$ by

$$\mathcal{E}_{\mathcal{D}}(t) = \sum_{i \in [D]} \mathbf{1}\{\gamma \leq \mathbb{P}_i\{X_i \leq f_i(t)\} \leq \delta\} \tag{13}$$

and

$$\mathcal{A}_{\mathcal{D}}(t) := \mathbf{1}\{\beta D \leq \mathcal{E}_{\mathcal{D}}(t)\}. \tag{14}$$

We refer to Equations 13 and 14 as the Empirical Cumulative Distribution Function (ECDF) test-statistic and the Empirical Cumulative Distribution Function test respectively. Whenever $\mathcal{A}(t) = 1$, we consider t *likelier in-distribution relative to* $\mathcal{D}$, otherwise we consider t as *likelier out-of-distribution relative to* $\mathcal{D}$. In the use cases that we have considered, f_i is the corresponding certainty score on input t from the i^{th} Bayesian sample, and $\mathbb{P}_i$ is the corresponding empirical cumulative probability distribution of certainty scores from the i^{th} Bayesian sample drawn from our common input $\mathcal{D}$.

Further, one can adjust β or (γ, δ) respect to target significance α so that

$$1 - \alpha \leq \sum_{i \geq \lceil \beta D \rceil}^{D} \binom{D}{i} p^i (1-p)^{D-i}.$$

In doing so, we assume as a prior that the projection of a random variable X onto its individual Bayesian samples X_i will be independent of its other projections, and that the probability that $\mathcal{E}_{\mathcal{D}}(t)$ is a Binomial variable with the number of draws being the number of Bayesian samples and the probability p being the length of the interval from γ to δ. When adjusting, $(1 - \alpha)$ will be the prior probability that a random variable distributed according to $\mathcal{D}$ will be such that at least βD of the D many Bayesian samples were likely to have been drawn with respect to the interval (γ, δ).

As discussed later in Sections 5 and 6, the ECDF test performs wildly depending on input data and the target interval. Future development of the ECDF test ought to apply the posterior distribution drawn from the in-distribution data upon calculating the marginal likelihood against the prior, Binomial distribution, and then apply this new posterior distribution to samples.

3.4.3 Global/Local Out-of-Distribution Detection Test

Our two stage test is a multi-valued classification that we have prepared in two-forms: a *strong* form and a *weak* form. In both forms, the three possible over-arching classifications are: In-Distribution, Out-of-Distribution, and Indeterminate.

Our two stage test compares a test distribution of certainty scores against the pooled training and test certainty score distributions. In the first stage, these distributions are with respect to the global certainty scores and their respective predictive category. In the local stage, they are with respect to the individual labels themselves.

We then apply the MWU test for tranches or the ECDF test with respect to Bayesian samples for the global/local label distribution modulo predictive status. We then aggregate the results of these tests to determine if a test distribution is out-of-distribution. See the appendix for a more in-depth breakdown of the logic of our out-of-distribution test.

It is assumed that the input distribution is derived from an input stream belonging to a single target category. In practice this is achieved by using the certainty distributions from our Bayesian samples for an individual input in the stochastic case, and for tranches of data from distributed sensors that are capturing the same data simultaneously, or a sensors data taken sequentially, or otherwise, data that was pooled together prior to application of the test by some similarity score, in the deterministic cases.

3.5 Hamiltonian Monte Carlo Simulation

As BNNs and Bayesian sampling are an active area of research, there are many inference schemes currently deployed to learn the posterior distribution over a BNN's model parameters. For our research, we sampled directly from the posterior via Markov Chain Monte Carlo (MCMC) using HMC [1] as implemented in `hamiltorch` [39].The benefit of using `hamiltorch` is that it became feasible to run HMC on a single GPU, allowing us to run multiple experiments in parallel, and the ability to use HMC for computationally tractable Bayesian sampling allowed us to explore an additional point of comparison for the robustness of test.

HMC sampling occurs in two parts: first, as an approximate Hamiltonian dynamics simulation based on numerical integration, followed by a correction by performing a Metropolis acceptance step.

Starting from the unnormalized log posterior, which is defined via the likelihood $p(\mathbf{Y}|\mathbf{X},\omega)$ and the prior $p(\omega)$, the samples are generated as follows:

$$p(\omega|\mathbf{Y},\mathbf{X}) \propto p(\mathbf{Y}|\mathbf{X},\omega)\, p(\omega).$$

Likelihood is a function of the parameters, $\omega \in \mathbb{R}^D$, which in our case will be the weights of the Bayesian neural network models. The Bayesian model function is then transformed into a Hamiltonian system by introducing a momentum variable $\mathbf{p} \in \mathbb{R}^D$, such that we now have a log joint distribution, $\log[p(\omega,\mathbf{p})] = \log[p(\omega|\mathbf{Y},\mathbf{X})\, p(\mathbf{p})]$, which is proportional to the Hamiltonian, $H(\omega,\mathbf{p})$.

Letting $p(\mathbf{p}) = \mathcal{N}(\mathbf{p}|\mathbf{0},\mathbf{M})$, where the covariance $\mathbf{M}$ denotes the mass matrix, the Hamiltonian can be written as:

$$H(\omega,\mathbf{p}) = \underbrace{-\log[p(\mathbf{Y}|\mathbf{X},\omega)\, p(\omega)]}_{\substack{\text{Potential Energy}\\ U(\omega)}} + \underbrace{{}^1\!/_2\, \mathbf{p}^\top \mathbf{M}^{-1}\mathbf{p}}_{\substack{\text{Quadratic Kinetic Energy}\\ K(\mathbf{p})}}. \tag{15}$$

Equation 15 consists of a potential energy term, which is our original Bayesian model, and a quadratic kinetic energy term derived from the log probability distribution of a Gaussian.

We simulate the dynamics of the Hamiltonian system using the leapfrog integration scheme to sample from this model. The details of this scheme are to be found in [6], which provides a general overview, and [39], which gives the specific symmetric scheme we used.

4 Methodology

4.1 Data

4.1.1 Network Traffic Data

As mentioned in Section 2.2, NIDS researchers often use the same simulated Netflow datasets to train their machine learning models, with the benchmark datasets being KDD Cup'99, Kyoto 2006+, NSL-KDD, and CSE-CICIDS-2018 [21], [23], [31], [34], [36], [45]. These datasets are created by simulating a network and attacks on the network in an artificial environment. Papers have discussed similar methods for creating and prepossessing packet-level data, converting each byte into an integer from 0 to 255 and filling in missing bytes with 0s to feed into a neural network [35], [59]. For our experiments, we relied on two popular datasets for NIDS deep learning: CICIDS-2017 [30] and UNSW-NB15 [17]. Following [61], we chose these data sets because they included both raw packet capture data and labeled network flows. The Canadian Institute for Cybersecurity generated the CICIDS dataset containing five days of network traffic with various attacks. The CyberRange Lab at University of New South Wales - Canberra generated the IXIA PerfectStorm tool that was used to generate the raw network packets in the UNSW dataset.

In order to label the packet capture data from the CICIDS and UNSW data sets, we used Payload-Byte, which is a standardized method using metadata to label raw packet captures for NIDS datasets [59]. Specifically, data points from our modified data sets contained 1500 payload bytes with values of 0-255 before they were further normalized between 0 and 1, a label, and header information, which was dropped to focus solely on packets and labels. Whenever packets was smaller than 1500 bytes, the rest of the bytes were filled in with zeros, and whenever packets exceeded 1500 bytes, they were truncated. The CICIDS data contained 15 classes, with one benign class and 14 attack types. The UNSW data contained ten classes, one benign class, and nine attack types..

4.1.2 ImageNet and COCO OOC Data

Computer vision tasks in the context of machine learning vary in complexity from image detection and classification of stills to higher-order inferences drawn from video composed of multiple still images. Understanding visual scenes, the *context* in which objects appear in images, is one of the common overarching tasks in computer vision. Towards that end, Microsoft COCO introduced the Common Objects in Context (COCO) dataset to address three core research problems to understand scenes: detecting non-iconic views of objects, contextual reasoning between objects, and precise 2D localization of objects [16]. The COCO dataset contains 91 common object categories with 82 categories having more than 5000 labeled instances, with the total dataset having 2.5 million labeled instances across 328000 images. This is in contrast to the ImageNet dataset, which has more hand labeled categories (over 20000) but fewer instances per category. For our experiments, we restricted ourselves to the the 2012 ILSVRC validation set consisting of 50000 validation images across 1000 categories.

With the overarching goal of building models that capture contextual clues to improve detection of in-context

objects and identify inconsistent with the scene context, the COCO OOC dataset was formed by placing out of context COCO objects to other COCO images in order to detect and identify images out-of-context (OOC) [57], [58]. The COCO OOC dataset has the following eight broad out-of-context image scenarios with over 150000 images:

- Animal-in-indoor
- Appliance-in-outdoor
- Electronic-in-outdoor
- Food-in-outdoor
- Indoor-Big
- Outdoor-Big
- Outdoor-object-indoor
- Vehicle-in-indoor

Finally, the ground truth labeling of the out-of-context images was not included in the metadata for the COCO OOC dataset, and establishing a dictionary between the ImageNet categories, the COCO categories, and the COCO OOC data set was beyond the scope of this project.

4.2 Deep Learning Model Architectures

4.2.1 NIDS Model Architectures

The NIDS model architectures used for our experiments were developed in [66] and [61]. They consist of a fully connected neural network architecture (FcNN) and a 1-dimensional convolutional neural network (1dCNN) architecture implemented in both deterministic and stochastic settings with two models each trained on the UNSW and CICIDS data sets respectively (see Appendix for more details).

4.2.2 ResNet Model Architectures

Our Imagenet and COCO OOC Data experiments compared the five pre-trained ResNet models that are included in the torch.models repository, with a sampling of the final, dense layer of each model using HMC implemented in hamiltorch (see Figure 4 in Appendix).

This required turning each pre-trained model into Sequential model, and then partitioning this Sequential model into a function to be applied in a custom built data loader and a single layer neural network to be sampled by hamiltorch. Our baseline performance was established on the ImageNet dataset on which the ResNet models were trained, and the out-of-distribution detection experimented was conducted by sampling from the COCO OOC dataset.

4.3 Computational Experiments

When comparing our deterministic models against their corresponding Bayesian counterparts, we note that the weights of the former are optimized using stochastic gradient descent to determine a single model $\mathbf{f}(\cdot;\omega)$ with parameters ω, such that on a given input $\mathbf{x}^*$, the deterministic model outputs a prediction $\hat{\mathbf{y}} = \mathbf{f}(\mathbf{x}^*;\omega)$. For our Bayesian neural networks, samples were generated from the posterior distribution resulting in a set of samples $\{\omega_s\}_{s\in S} \sim p(\omega|\mathbf{X},\mathbf{Y})$ such that the predictive distribution can be approximated via multiple networks draws for each test image $\hat{\mathbf{y}}_s = \mathbf{f}(\mathbf{x}^*;\omega)$.

Our general aim was to establish the viability of our certainty scoring regime and out-of-distribution detection test in both deterministic and stochastic settings across different tasks and modalities. Towards that end, and given the limitations of each data modality, we devised different experiments for each modality, aiming to capture what a relevant form of OOD detection would look like.

Additionally, our experiments were set up to investigate the viability of using HMC with hamiltorch in lieu of a deterministic model to assess the performance trade offs that enable OODD for individual datum. In particular, when assessing the OODD detection abilities for our deterministic model, we conceptualized the OODD test as applying to data that was either held out of a data set as in the case of the packet data, or otherwise, data that is deliberately designed to be out-of-context, as in the image detection case.

4.3.1 Network Traffic Data

Our deterministic model architectures were implemented in Tensorflow, and our stochastic models were implemented in hamiltorch, with architectures specified using Pytorch. We implemented HMC with No U-Turn sampling (NUTS) in order to optimize our step-size parameters for each of our stochastic model, and restricted our training samples to approximately 8000 samples of the CICIDS and UNSW data sets.

Our experiments sought to achieve the following goals:

1) Verify that empirical competence, either globally or within labels indicated that TP certainty scores stochastically dominated the FP certainty scores;
2) Verify that our out-of-distribution detection test could identify that holdout data was out-of-distribution relative to models trained on the remaining dataset whenever the corresponding model was competent;
3) Verify that our out-of-distribution detection test could identify that individual hold out samples were out-of-distribution relative to the Bayesian models trained on the remaining dataset whenever the corresponding model was competent.

The first goal was reached by training and evaluating the performance of our different models, specifically running multiple forms of the Mann Whitney U test on the TP and FP distributions, in addition to plotting the distributions of certainty scores by their predictive status. The second goal was reached by first running our out-of-distribution test on the entire tranche of omitted data at the 0.05% and 0.001% significance level for both our deterministic models, and the ensemble model formed for our stochastic models. The final goal was reached by running our out-of-distribution test on the individual packet certainty distributions generated by our stochastic models. Each goal was tested in two different experimental contexts against the baseline model.

Our first experiment dropped any attack class from the dataset that appeared in less than 1% of the data set, before training the corresponding models. In the UNSW dataset, we dropped the attack class: worms. For the CICIDS dataset, we dropped the attack classes: Web Attack - Sql Injection, Portscan, Bot, Web Attack - Brute Force, and Heartbleed.

Our second experiment omitted all Denial of Service (DoS) labeled packets from the training samples. In the UNSW dataset, we only dropped the attack class DoS.

For the CICIDS data set, we dropped the following attack classes: DoS Hulk, DDoS, DoS GoldenEye, DoS Slowloris, and DoS Slowhttptest. The dropped attack classes account for approximately 4.25% of the UNSW dataset and 58.23% of the CICIDS data set.

4.3.2 ImageNet and COCO OOC Data

For our stochastic ResNet models, we transferred the pre-trained weights of each ResNet model to its corresponding stochastic model and sampled the weights from the final, dense layer. Each stochastic model applied the convolutional layers with corresponding pre-trained weights to each normalized image before sampling the weights from the final layer of the network. We trained our stochastic models on the 2012 ImageNet validation dataset that consisted of 50000 images, with a train-test split of 75%-25%.

We compared the performance of five pre-trained ResNet architectures in both a deterministic and stochastic setting against COCO OOC images drawn from eight different categories. Further, we gathered baseline performance of each architecture and setting against the ImageNet 2013 validation dataset to establish both accuracy and competence.

In the deterministic setting, we applied each of our five pre-trained ResNet models to images with a shared common object out of context. In total, there were 479 distinct common objects forming distinct tranches of images of common objects out-of-context across the eight categories. Each tranche has between 500 and 1200 unique images wherein the object was out of context with respect to the background for our deterministic models.

In the stochastic setting, we sampled from the final fully connected layer of the deterministic model architecture applied to the output of the penultimate layer of the deterministic model applied to our preprocessed image data. We trained the stochastic models using HMC implemented in hamiltorch on 75% of the ImageNet 2013 validation dataset, and tested each model against the remainder of the dataset to gather the baseline statistics across 300 samples.

Finally, for the purposes of performing out of distribution detection on the COCO OOC images, we randomly selected 64 baseline images across the eight categories, and applied our stochastic model to the corresponding tranche.

4.4 Evaluation Metrics

The primary experimental task was to determine the efficacy of our certainty framework to determine when inputs were out-of-distribution with respect to a given model. We report the empirical competence for both data modalities. When evaluating the out-of-distribution detection capability of our OODD test on both modalities, we used the *strong form* of our OODD test, making sure to report both the global and local proportions of the samples (tranches) which the OODD test identified as OODD. However, we gathered different scores for our two tasks given the size and scope differences between our two tasks, the number of local labels in each dataset, and the experimental design to generate out of distribution examples.

Additionally, for the NIDS data, we also ran an experiment to determine what proportion of tranche based TPs and Bayesian TPs were labeled as out-of-distribution by our test. This experiment compared the performance of the OODD test at determining what proportion of TP data relative to our model was incorrectly flagged as out-of-distribution. This experiment was applied to both the tranche data across both the deterministic and stochastic models, as well as to the individual data for the stochastic models.

When evaluating our NIDS models, in addition to empirical competence scores and OODD test performance scores, we gathered the following scores to compare models: multi-class classification accuracy, binary classification accuracy, false omission rate, misclassified positive rate, and F1 score. The multi-class classification accuracy refers to a model's ability to correctly classify packet labels. Binary classification accuracy refers to a model's ability to correctly classify packets as benign or malicious, and is computed as the sum of True Positive malicious packets, and True Negative benign packets, divided by the total number of packets.

In contrast with the scores gathered for our NIDS models, we focused on category accuracy and certainty scores in order to directly compare pre-trained ResNet model architectures with their corresponding stochastic variants that sampled parameters for the final, fully connected layer. This is in contrast with the common ILSVRC contest metric of gathering whether the models in question were able to accurately place the object identified in its top five predictions; because we are not proposing new architectures, but only trying to establish the efficacy of certainty scoring for OODD, placement in the top five is irrelevant to the use of a certainty score. We also gathered the two primary competence metrics, component and empirical competence of our baseline models, in addition to the performance of the strong form of our out-of-distribution detection test.

5 Results and Discussion

Our overarching goals are to evaluate our certainty scoring framework with respect to:

1) the quality of model performance;
2) the ability to perform out-of-distribution detection.

This necessitated the experiments described in Section 4 on our two different data modalities, each with different experiments conducted. We found that proposed certainty framework allows us to provide assurance about model performance and to perform out of distribution detection.

The Pearson correlation coefficient between multi-class accuracy and empirical competence was -0.034, suggesting that competence and accuracy are independent. To ground our intuitions, consider Table 1. The Deterministic 1d CNN had low multi-class accuracy, but high empirical competence, indicating that whenever the model was wrong, it did so with extremely low certainty, whereas when the model was correct, it did so with high certainty. This suggests that using certainty scores allows for inference about the quality of model prediction.

Additionally, we found that the global distribution of TP certainty scores were a reliable means of checking if a test data set is in-distribution or out of distribution, using both deterministic tests using Mann Whitney U scores, and our novel ECDF test, across modalities and tasks. As presently

implemented, the local stage of the test is inherently biased towards rejecting inputs as out-of-distribution.

In the first NIDS experiment with tranche data, the worst performing models was the deterministic FcNN, which only globally rejected 93.9% of all tranches on the CICIDS data at 0.1% significance, and the stochastic 1dCNN which only globally rejected 99.4% of the sampled tranches at 0.1% significance. In the second NIDS experiment, every DoS tranche was globally rejected as out of distribution with the exception of the Deterministic 1dCNN, which only identified 92.3% of the DoS tranches as out-of-distribution at the global stage with 0.1% significance.

In contrast, the ResNet experiment models performed well, with the worst performance coming from the deterministic ResNet50 and ResNet101 models, both of which identified 99.6% of OOC data as out-of-distribution at 5% significance. We discuss the respective experimental results in greater detail in the following subsections.

5.1 Network Traffic Experiments

Our two NIDS experiments consisted of comparing baseline model performance against models trained on datasets with omitted data. To determine the baseline performance, we first gathered scores on in-distribution data, which are displayed in Table 1. Model scores demonstrate high binary accuracy for in-distribution data across all models, with higher multi-classification accuracy for the fully connected neural networks being demonstrated consistently across both datasets.

The stochastic models had a greater misclassified positive rate, and the false omission rate increased for the UNSW data but decreased for the CICIDS. Overall, the FcNNs outperformed the 1dCNN in terms of accuracy across both datasets and implementations (deterministic or stochastic), and relatedly in terms of empirical competence.

UNSW	D FcNN	S FcNN	D 1dCNN	S 1dCNN
Multi-Class Accuracy	**.804**	.700	.188	.636
Binary Accuracy	**.984**	.941	.979	.902
Misclassified Positive Rate	**.258**	.326	.260	.362
False Omission Rate	.039	.158	**.025**	.256
F1 Score	**.999**	.959	.986	.930
Empirical Competence	**.649**	.503	.619	.269
CICIDS				
Multi-Class Accuracy	**.776**	.729	.174	.688
Binary Accuracy	.920	**.981**	.920	.932
Misclassified Positive Rate	.182	.339	**.181**	.463
False Omission Rate	.446	**.035**	.441	.172
F1 Score	.957	**.987**	.957	.954
Empirical Competence	.577	.507	**.580**	.353

TABLE 1
Baseline Deep Learning Model Performance Scores. Top scores in **Bold**

Furthermore, empirical competence for each model was sufficiently high enough that we were able to attain empirical validation for Theorem 1 of the competence hierarchy for nontrivial α, confirming the stochastic dominance of TP distributions of certainty scores by predictive status within our framework by informedness and competence.

We observed that the TP and FP distributions were sufficiently separated, because each model was deemed sufficiently competent globally, and similarly, we observed separations between the two distributions within categories where models were sufficiently competent, as seen globally in Figure 5 (see Appendix), and locally in Tables 2 and 3. The latter table is truncated only to show instances across architectures where the p-value was at least .001, and left blank when the model otherwise failed to predict *both* True and False positives for the listed category. We also observed that the CICIDS models were generally amateur or incompetent at identifying the various non-DDoS DoS labeled packets, as seen in Figure 6 (see Appendix).

We found that the stochastic models showed a greater divergence between the certainty of true positives from false positives. Given that the deterministic models showed higher empirical competence, and since the deterministic models had higher accuracy, we find that this indicated that there was greater certainty when falsely classifying, even though mistaken classification occurred less frequently. This would suggest that more information can be gleaned about the predictive status of a sample with a low certainty score in a stochastic model than one from a deterministic model.

	generic	exploits	normal	analysis	fuzzers	shellcode	dos	recon	backdoor	worms
D FcNN	0	0	0		0	0	0		.200	.040
S FcNN	0	0	0	.374	0	0	.125			
D 1dCNN		0	0	.926	0	0	0	0	1.0	
S 1dCNN	0	0	0		0	.538	1.0	0		

TABLE 2
Baseline UNSW Mann-Whitney p-values for TP vs. FP Certainty Distributions. Cells left blank when models failed to have both TP and FP distributions to compare. Higher p-values indicative of greater tendencies towards incompetence within category by model.

In each of our baseline models we generally found high competence at a global level, although each model varied with respect to empirical competence in-label, per Tables 2 and 3. Specifically, every UNSW model fared poorly with respect to at least one attack category, with *analysis*, *dos*, and *backdoor* being incompetently identified by the Bayesian FcNN & Deterministic 1dCNN, the Bayesian FcNN & Bayesian 1dCNN respectively, the Deterministic FcNN and Deterministic 1dCNN respectively. The CICIDS models all generally performed competently, excepting the DoS categories, and the FcNN models on WA-XSS. In particular, the deterministic models were highly incompetent with respect to the *DoS Slowloris* category while the Bayesian models were incompetent with respect to the *DoS GoldenEye* category, and the *DoS Hulk* category to a lesser degree. In part, our experiment withholding the DoS data was motivated by the widespread incompetence at identifying these categories, as well as our interest in evaluating model performance where DDoS was withheld.

	DoS Slowloris	WA-XSS	DoS GoldenEye	DoS Slowhttptest	Heartbleed	DoS Hulk
D FcNN	1	.521	0	.980		0
S FcNN	0	.051	.457			.022
D 1dCNN	1	.005	.021	.003		0
S 1dCNN	.031		.760	.020	.292	.037

TABLE 3
Insignificant Baseline CICIDS Mann-Whitney p-values for TP vs. FP Certainty Distributions. Cells left blank when models failed to have both TP and FP distributions to compare. Higher p-values indicative of greater tendencies towards incompetence within category by model.

In order to perform our out-of-distribution test experiments, we needed to retrain each model on datasets omitting the holdout data. We contrast the baseline model performance in Table 1 with our experimental results in Tables 4 and 8.

UNSW	D FcNN	S FcNN	D 1dCNN	S 1dCNN
Multiclass Accuracy	**.829**	.706	.206	.698
Binary Accuracy	**.986**	.943	.976	.967
Misclassified Positive Rate	.269	.321	**.268**	.300
False Omission Rate	**.035**	.160	.070	.094
F1 Score	**.990**	.960	.983	.982
Empirical Competence	**.635**	.509	.630	.233
CICIDS				
Multiclass Accuracy	.722	**.726**	.219	.674
Binary Accuracy	.907	**.983**	.901	.972
Misclassified Positive Rate	**.210**	.349	.229	.403
False Omission Rate	.153	**.038**	.444	.053
F1 Score	.950	**.989**	.946	.981
Empirical Competence	.482	.455	**.487**	.397

TABLE 4
Experiment #1 Deep Learning Model Performance Scores

Notably in our first experiment, we found that scores roughly remained the same, likely as the omitted samples appeared infrequently enough when training as to not significantly impact model weights, even on the final layers, which were contingent on the number of output labels.

UNSW	Tranche	% OOD	% Globally OOD	% Locally OOD	% Indeterminate
D FcNN	1	0	1	0	1
S FcNN	2	1	1	1	0
D 1dCNN	1	1	1	1	0
S 1dCNN	2	.5	1	.5	.5
CICIDS					
D FcNN	1000 (2883)	.572	.937	.575	.368
S FcNN	1000 (1030)	1	1	1	0
D 1dCNN	1000 (2883)	.483	1	.483	.517
S 1dCNN	1000 (1030)	.549	.997	.551	.450

TABLE 5
Experiment #1 Hold-out at 5% Significance. Parentheses represent total possible tranches whenever sampling was capped at 1000.

In general, we found that these results still held, with the worst performing models in both cases belonging to the deterministic CICIDS models, as seen in Table 5 at 5% significance and in Table 6 at 0.1% significance.

UNSW	Tranches	% OOD	% Globally OOD	% Locally OOD	% Indeterminate
D FcNN	1	1	1	1	0
S FcNN	2	1	1	1	0
D 1dCNN	1	1	1	1	0
S 1dCNN	2	1	1	1	0
CICIDS					
D FcNN	1000 (2883)	.193	.212	.28	.106
S FcNN	1000 (1030)	.99	.99	1	.01
D 1dCNN	1000 (2883)	.364	.777	.487	.536
S 1dCNN	1000 (1030)	.938	.993	.943	.06

TABLE 6
Experiment #1 Hold-out at 0.1% Significance. Parentheses represent total possible tranches whenever sampling was capped at 1000.

In contrast, Table 7 shows the performance of the Bayesian models on individual packets using two different intervals: the default interval of $(.25, .75)$ and an expanded interval of $(.05, 1)$, and assuming a Binomial distribution of the number of Bayesian samples where the corresponding ECDF evaluated at their certainty score would fall within the target interval.

Immediately we see varying the target interval lead to different outcomes depending on the dataset, and find support that when further developing the ECDF test, we ought to use the posterior distribution function in lieu of a binomial distribution before determining the likelihood that a test sample is not in-distribution.

As a result of dropping a significant portion of each data set, the second experiment saw more noticeable changes in model performance, as seen in Table 8.

UNSW & target interval	Samples	% OOD	%Globally OOD	%Locally OOD	% Indeterminate
S FcNN (.25,.75)	92	.696	.696	1	.304
S FcNN (.05,1)	92	.283	.283	1	.717
S 1dCNN (.25,.75)	92	.696	.696	1	.304
S 1dCNN (.05,1)	92	.283	.283	1	.717
CICIDS& target interval					
S FcNN (.25,.75)	3000(31960)	1	1	1	0
S FcNN (.05,1)	3000(31960)	1	1	1	0
S 1dCNN (.25,.75)	3000(31960)	.724	.724	1	.276
S 1dCNN (.05,1)	3000(31960)	1	1	1	0

TABLE 7
Experiment #1 Out-of-distribution detection at individual-packet level by Bayesian model, using ECDF test with β adjusted according to $\alpha = .05$, $\gamma_1 = .25$, $\delta_1 = .75$ (and alternatively $\gamma_2 = .05$ and $\delta_2 = 1$),and sample size; each FcNN has 360 Bayesian samples per sample, and each 1dCNN has 260 Bayesian samples per sample.

UNSW	D FcNN	S FcNN	D 1dCNN	S 1dCNN
Multi-Class Accuracy	**.805**	.698	.213	.698
Binary Accuracy	**.986**	.967	.973	.967
Misclassified Positive Rate	.260	**.212**	.264	.299
False Omission Rate	**.010**	.041	.075	.094
F1 Score	**.990**	.984	.982	.982
Empirical Competence	**.685**	.550	.680	.197
CICIDS				
Multi-Class Accuracy	**.996**	.966	.437	.945
Binary Accuracy	.907	.979	.901	**.980**
Misclassified Positive Rate	.210	**.032**	.229	.092
False Omission Rate	.153	.015	.444	**.014**
F1 Score	.950	**.978**	.976	.975
Empirical Competence	.981	.899	**.987**	.743

TABLE 8
Experiment #2 Deep Learning Model Performance Scores

The UNSW models performed similarly to their baseline counterparts other than small decreases in false omission rate for the stochastic models. However, the CICIDS models each had a significant increase in multi-class accuracy scores and decreases in misclassified positive and false omission rates. We attribute the significant increase in performance of the CICIDS models over the UNSW models to the significant portion of the dataset being omitted by dropping the five DoS-labeled classes in the CICIDS dataset. Other than DDoS attack class, the baseline models generally struggled to correctly classify DoS attacks with high competence.

UNSW	Tranches	% OOD	Globally	Locally	% Indeterminate
D FcNN	36	.722	1	.722	.278
S FcNN	109	.817	1	.817	.184
D 1dCNN	36	.639	1	.639	.361
S 1dCNN	109	.624	1	.624	.376
CICIDS					
D FcNN	1000 (8741)	.124	1	.124	.876
S FcNN	95	.674	1	.674	.326
D 1dCNN	1000 (8741)	.081	1	.081	.919
S 1dCNN	190	.463	1	.463	.537

TABLE 9
Experiment #2 Hold-out at 5% Significance. Parentheses represent total possible tranches whenever sampling was capped at 1000.

For both experiments, we found the test certainties were distributed differently from TP and FP in all models with the exception of the Stochastic FcNN, which had a p-value of .128 between the test certainty distribution, TP and FP certainty distributions. This generally held within label assignments, as we failed to reject the null hypothesis that the test and FP data were drawn from the same distribution for only one label per model in the second experiment.

We examine the performance of the Bayesian models on individual packets in Table 11. Here, we found that altering the size of the target interval improved out-of-distribution detection for the UNSW data, but otherwise did not impact

UNSW	Tranches	Tranche % Identified OOD	Globally	Locally	% Identified Indeterminate
D FcNN	36	1	1	1	0
S FcNN	109	1	1	1	0
D 1dCNN	36	1	1	1	0
S 1dCNN	109	.890	.945	.945	.11
CICIDS					
D FcNN	1000 (8741)	.126	1	.126	.874
S FcNN	95	.421	1	.421	.579
D 1dCNN	1000 (8741)	1	1	.126	.874
S 1dCNN	190	.258	.742	.258	.741

TABLE 10
Experiment #2 Hold-out at 0.1% Significance. Parentheses represent total possible tranches whenever sampling was capped at 1000.

the performance of the CICIDS models.

UNSW& target intervals	Samples	% ID'd OOD	Globally	Locally	% ID'd Indeterminate
FcNN (.25,.75)	3000(3396)	.898	.898	1	.102
FcNN (.05,1)	3000(3396)	1	1	1	0
1dCNN (.25,.75)	3000(3396)	.724	.724	1	.276
1dCNN (.05,1)	3000(3396)	1	1	1	0
CICIDS& target intervals					
FcNN (.25,.75)	2946	1	1	1	0
FcNN (.05,1)	2946	1	1	1	0
1dCNN (.25,.75)	3000 (5891)	1	1	1	0
1dCNN (.05,1)	3000 (5891)	1	1	1	0

TABLE 11
Experiment #2 Out-of-distribution detection at individual-packet level by Bayesian model, using ECDF test with β adjusted according to $\alpha = .05$, $\gamma_1 = .25$, $\delta_1 = .75$ (and alternatively $\gamma_2 = .05$ and $\delta_2 = 1$),and sample size; each FcNN has 360 Bayesian samples per sample, and each 1dCNN has 260 Bayesian samples per sample.

In order to justify using our OODD test, we also sought guarantees that the test would be capable of distinguishing TP from FPs, while also ensuring that random samples of the TP distribution would be recognized as belonging to the TP distribution. Towards that end, we gathered the performance of our models across the experiments at OODD on the TP data at 0.1% significance in Table 12 and Table 13, for tranches of 33 samples per tranche and Bayesian samples for individual packets respectively.

UNSW	Tranches	% ID'd OOD	Globally	Locally	% ID'd Indeterminate
Baseline DFcNN	640	0	0	1	1
Baseline SFcNN	181	0	0	1	1
Baseline D1dCNN	632	.002	.002	1	.998
Baseline S1dCNN	161	0	0	1	1
Ex1 DFcNN	638	0	0	1	1
Ex1 SFcNN	181	0	0	1	1
Ex1 D1dCNN	630	.003	.003	1	.997
Ex1 S1dCNN	169	0	0	1	1
Ex2 DFcNN	635	0	0	1	1
Ex2 SFcNN	180	0	0	1	1
Ex2 D1dCNN	631	0	0	1	1
Ex2 S1dCNN	180	.006	.006	1	.994
CICIDS					
Baseline DFcNN	109	0	0	1	1
Baseline SFcNN	16	0	0	1	1
Baseline D1dCNN	109	0	0	1	1
Baseline S1dCNN	2	0	0	1	1
Ex1 DFcNN	82	0	0	1	1
Ex1 SFcNN	2	0	0	1	1
Ex1 D1dCNN	82	0	0	1	1
Ex1 S1dCNN	2	0	0	1	1
Ex2 DFcNN	293	0	0	1	1
Ex2 SFcNN	4	0	0	1	1
Ex2 D1dCNN	293	.007	.007	1	.993
Ex2 S1dCNN	4	0	0	1	1

TABLE 12
Proportion of tranche data of TP (in-distribution) data by Dataset and Experimental Model that was determined as OOD by OODD test at 0.1% significance cutoff, with 33 samples per tranche.

With respect to the deterministic results, we found that the tranches of in-distribution data were near uniformly indeterminate with respect to status. Nonetheless, the global form of the test reliably characterized tranches of in-distribution data as being in-distribution across datasets and architectures.

Tables 13 and 14 displays the power of our OODD test with target intervals (.25,.75) and (.05,1) respectively using

UNSW	Samples	% ID'd OOD	Globally OOD	Locally OOD	% ID'd Indeterminate
Baseline SFcNN	3000(7989)	.576	.576	1	.424
Baseline S1dCNN	3000(7989)	.552	.552	1	.448
Ex1 SFcNN	3000(7979)	.592	.592	1	.408
Ex1 S1dCNN	3000(7979)	.638	.638	1	.362
Ex2 SFcNN	3000(7649)	.838	.838	1	.162
Ex2 S1dCNN	3000(3396)	.999	.999	1	.001
CICIDS					
Baseline SFcNN	3000(70513)	.594	.594	1	.406
Baseline S1dCNN	3000(14103)	.600	.600	1	.400
Ex1 SFcNN	3000(31960)	1	1	1	0
Ex1 S1dCNN	3000(31960)	1	1	1	0
Ex2 SFcNN	3000(2946)	1	1	1	0
Ex2 S1dCNN	3000(5891)	1	1	1	0

TABLE 13
Proportion of individual data (up to 3000 samples) of in-distribution data by Dataset and Experimental Bayesian Model that was determined as OOD using ECDF test with β adjusted according to $\alpha = .05$, $\gamma = .25$, $\delta = .75$, and sample size; each FcNN has 360 Bayesian samples per sample, and each 1dCNN has 260 Bayesian samples per sample.

the ECDF tests and a β cut-off determined by a target significance of 0.05. In contrast with the tranche based models, the individual Bayesian samples were found to be dispersed across the three possible categories of *In-Distribution*,*Out-of-Distribution*, and *Indeterminate*. The ability of the ECDF test to reliably characterize in-distribution data varied wildly between datasets, and architectures, and target intervals.

Presently, when comparing with the performance of the Bayesian samples in Tables 7 and 11, we see that there is a complete collapse of the ability to discern individual packets as in-distribution and out-of-distribution for the experimental CICIDS models, and for the second experiment UNSW models. The baseline CICIDS models performed comparably to the UNSW Baseline and first experiment. Additionally, expanding the interval improves the ECDF test performance at recognizing in-distribution data from out-of-distribution data, although it did not improve the performance for the CICIDS experimental models.

UNSW	Samples	% ID'd OOD	Globally OOD	Locally OOD	% ID'd Indeterminate
Baseline SFcNN	3000(7989)	.256	.256	.910	.644
Baseline S1dCNN	3000(7989)	.452	.452	1	.548
Ex1 SFcNN	3000(7979)	.364	.364	1	.636
Ex1 S1dCNN	3000(7979)	.675	.675	1	.325
Ex2 SFcNN	3000(7649)	1	1	1	0
Ex2 S1dCNN	3000(3396)	1	1	1	0
CICIDS					
Baseline SFcNN	3000(70513)	.481	481	1	.519
Baseline S1dCNN	3000(14103)	.114	.114	1	.886
Ex1 SFcNN	3000(31960)	1	1	1	0
Ex1 S1dCNN	3000(31960)	1	1	1	0
Ex2 SFcNN	2946	1	1	1	0
Ex2 S1dCNN	2946	1	1	1	0

TABLE 14
Proportion of individual data in-distribution data by Dataset and Experimental Bayesian Model that was determined as OOD using ECDF test with β adjusted according to $\alpha = .05$, $\gamma = .05$, $\delta = 1$, and sample size; each FcNN has 360 Bayesian samples per sample, and each 1dCNN has 260 Bayesian samples per sample.

Overall, we found that the MWU test performed better on the UNSW data than it did on the CICIDS data, whereas the ECDF test on the individual packets was more reliable on the CICIDS data than the UNSW data. However, this is most likely due to the use of the Binomial prior, which biased the ECDF test towards rejecting input data as out-of-distribution, as seen in Tables 13 and 14.

Across all experiments, we found that using the global stage of the OODD test was sufficient for determining if a tranche or Bayesian samples were out-of-distribution. Finally, in all cases, the local stage was incapable of correctly identifying in-distribution data as in-distribution in both statistical frameworks.

5.2 ImageNet and COCO OOC Experiments

Tables 15 and 16 describe the performance of our ResNet models. While we intended the stochastic ResNet model as a proof of concept for our out-of-distribution tests, they can be developed for production once provided with an appropriate weighted function that will yield a model with better performance than the pre-trained counterpart, see Figure 7 (see Appendix). We observe that there were tradeoffs made between accuracy and competence for our ensemble model, as each ensemble models at one point attained greater accuracy than its deterministic counterpart, before declining.

Deterministic Model	ResNet18	ResNet34	ResNet50	ResNet101	ResNet152
Accuracy	.648	.697	.796	.802	.810
Component Competence	.674	.733	.374	.700	.682
Empirical Competence	.368	.435	.276	.488	.499

TABLE 15
Performance Scores for Pre-trained ResNet models

Further, the inversion of the negative log-likelihood curve suggests that the hyperparameters should be vastly different for each of these models. While we attempted to account for this running these models with No U-turn Sampling to improve step-size performance, the inversion of the NLL curve still appeared, which suggests that we should optimize with respect to step-length.

Ensemble Stochastic Model	ResNet18	ResNet34	ResNet50	ResNet101	ResNet152
Accuracy	.570	.622	.781	.798	.801
Component Competence	.526	.579	.627	.703	.711
Empirical Competence	.267	.335	.498	.567	.577

TABLE 16
Performance Scores for Bayesian ResNet models

Within our eight categories and across all five baseline architectures, using our OODD test, we globally reject with p-values below 1e-50, and locally reject in every case with significance at 0.05. Further, per Tables 5.2 and 5.2, we found that while in the deterministic case that adding additional layers led to marginal decrease in the ability to detect that tranches were globally out-of-distribution.

Rather than directly contrast the deterministic and stochastic ResNet models, our tests for the stochastic case were able to be run on individual images as opposed to tranches of the same image and background subject to 2-dimensional transformations to change the context. A practical accounting of the deterministic OODD test is to consider its application to detecting an out-of-context object being captured by multiple simultaneous sensors, while the stochastic OODD test is drawn against the Bayesian samples for a single source image from a single sensor. Computational considerations and a preference to explore

Deterministic Model (N=475)	ResNet18	ResNet34	ResNet50	ResNet101	ResNet152
Estimated Out-of-Distribution	1	1	.996	.996	.998
Estimated Global Out-of-Distribution	1	1	.996	.996	.998
Estimated Local Out-of-Distribution	1	1	1	1	1
Estimated Indeterminate	0	0	.004	.004	.002

TABLE 17
Out-of-distribution Detection Performance For Pre-trained ResNet models with sampled tranches selected by common out-of-context image at 5% significance.

the OODD powers of the Bayesian approach led us to conduct the experiment by randomly selecting images from across the eight categories and then looking at the certainty score distributions drawn from the sample distributions we gathered from running `hamiltorch`. In contrast with the deterministic case, according to our OODD test, each stochastic model estimated less than 100% of the sampled images to be out-of-distributionBoth tranche sampling by common out-of-context image and the Bayesian sampling for individual inputs were determined to be locally out-of-distribution at least 99.6% of the time.

Ensemble Stochastic Model (N=640)	ResNet18	ResNet34	ResNet50	ResNet101	ResNet152
Estimated Out-of-Distribution	1	1	1	1	1
Estimated Global Out-of-Distribution	1	1	1	1	1
Estimated Local Out-of-Distribution	1	1	1	1	1
Estimated Indeterminate	0	0	0	0	0

TABLE 18
Out-of-distribution Detection Performance For Bayesian ResNet models on individual out-of-context images with ECDF test and adjusted $\beta = 0.927$ using significance .05, $\gamma = 0.025$, $\delta = .975$, and 267 Bayesian samples per Sample

6 Conclusion

Reiterating our overarching motivation, we wished to find if our certainty framework could be used: i) to evaluate the quality of model predictions; ii) for out-of-distribution detection across multiple modalities and architectures.

For the first goal, we found empirical support across all modalities, datasets, architectures, and tasks that we could reliably use our certainty and competence framework to distinguish TP from FPs within distributions, enabling us to conclude that outputs with low certainty were much more likely to be false positives than they were likely to be true positives, and similarly that predictions with high certainty scores were more likely to be True positives. Further, models with greater empirical competence demonstrated a wider spread between the medians of the certainty distributions for their global TPs and FPs. For the second goal, we found empirical support that the global TP distribution can be used to distinguish in-distribution and out-of-distribution data reliably across modalities, task, and statistical paradigms.However, we found that the ECDF test without updating the binomial prior had substantially worse performance than the tranche based sample method. Finally, we observed consistent failure of the local stage of our OODD test as presently constructed.

One point of failure for the local stage is that relying on uniform significance values for all labels will bias the local stage of the test towards determining that the test distributions are out-of-distribution. This underscores the need for using optimized parameters level per category, especially as whenever $\beta > \frac{1}{2}$, the local stage of the test will be biased towards determining any distribution is locally out-of-distribution whenever the corresponding partition of a datum's distribution by predicted label occurs less than β many times.

That our findings held independent of architecture, data set, task, and modality suggests that in addition to the theoretical nature of certainty as an intrinsic feature of probability manifolds, our certainty framework and use of certainty scoring distributions provides a robust means of uncertainty quantification for assurance and evaluation

of model quality, along with statistical inference. Since the certainty framework captures an intrinsic feature of a probability model, we can apply it to models presently in production, including those in safety-critical settings. This allows for guidance about model resilience at present, in addition to help guide further model development. There is the further implication that use of our certainty scoring framework can supplement training to enhance robustness by pulling out poor training examples, as well as help with data exploration by helping to identify mislabeled examples under certain theoretical conditions.

We also found evidence suggesting our certainty framework can supplement training in order to enhance robustness by helping to identify and pull out poor training examples or mislabeled examples. Specifically, when working with CICIDS data, model performance and competence can generally be improved by pooling the various DoS categories into a single category.

One goal for future research is to refine the test to ensure that in-distribution data does not become classified as out-of-distribution within tolerable bounds that depend on the informedness parameter α and a model's related placement in the competence hierarchy. Future work developing the OODD and ECDF test on Bayesian samples should first focus on improving performance on in-distribution data. We propose this should be done by compute the marginal likelihood, and then deriving the posterior distribution, which we then should apply to test data.

One final additional goal for future research with the COCO OOC datasets will be to develop better preprocessing of the COCO OOC datasets to extract and establish ground truth of the out-of-context objects in point of comparison with the ImageNet classification when working with the pre-trained ResNet models. Additionally, it would be worthwhile to train ResNet models on the COCO data and classification system and contrast the performance of these models on the COCO OOC data. Such future research aligns with the goal of developing models that can both:

- correctly identify an out-of-context object within bounding boxes,
- correctly classify the out-of-context image within the bounding box,
- produce a score indicating the confidence of these assessments to aid human agents making decisions.

Finally, a promising future direction of research is to determine the degree to which of the proposed cost functions in Section 3 can improve model competence and lead to better inferential properties, particularly for models to be deployed in an open-world environment.

ACKNOWLEDGMENTS

This work was supported in part by the U.S. Army Combat Capabilities Development Command (DEVCOM) Army Research Laboratory under Support Agreement No. USMA 21050 and the Defense Advanced Research Projects Agency (DARPA) under Support Agreement No. USMA 23004. The views expressed in this paper are those of the authors and do not reflect the official policy or position of the U.S. Military Academy, the U.S. Army, the U.S. Department of Defense, or the U.S. Government.

REFERENCES

[1] R. M. Neal, "Bayesian Learning for Neural Networks," Ph.D. dissertation, University of Toronto, 1995.

[2] S. Sanner, J. R. Anderson, C. Lebiere, and M. C. Lovett, "Achieving efficient and cognitively plausible learning in backgammon," in *Proceedings of the Seventeenth International Conference on Machine Learning*, 2000, pp. 823–830.

[3] K. Hwang, M. Cai, Y. Chen, and M. Qin, "Hybrid intrusion detection with weighted signature generation over anomalous internet episodes," *IEEE Transactions on dependable and secure computing*, vol. 4, no. 1, pp. 41–55, 2007.

[4] S. Watanabe, *Algebraic geometry and statistical learning theory*. Cambridge university press, 2009, vol. 25.

[5] D. Reitter and C. Lebiere, "Accountable modeling in act-up, a scalable, rapid-prototyping act-r implementation," in *Proceedings of the 10th international conference on cognitive modeling (iccm)*, Citeseer, 2010, pp. 199–204.

[6] R. M. Neal *et al.*, "MCMC using Hamiltonian dynamics," *Handbook of Markov chain Monte Carlo*, vol. 2, no. 11, p. 2, 2011.

[7] A. Capaldi, S. Behrend, B. Berman, J. Smith, J. Wright, and A. L. Lloyd, "Parameter estimation and uncertainty quantication for an epidemic model," *Mathematical biosciences and engineering*, p. 553, 2012.

[8] M. J. Choi, A. Torralba, and A. S. Willsky, "Context models and out-of-context objects," *Pattern Recognition Letters*, vol. 33, no. 7, pp. 853–862, 2012.

[9] J. Harris, *Algebraic geometry: a first course*. Springer Science & Business Media, 2013, vol. 133.

[10] R. Srivastava and V. Richhariya, "Survey of Current Network Intrusion Detection Techniques," *Journal of Information Engineering and Applications*, vol. 3, no. 6, 2013, ISSN: 2225-0506.

[11] R. Thomson and C. Lebiere, "Constraining bayesian inference with cognitive architectures: An updated associative learning mechanism in act-r," in *Proceedings of the Annual Meeting of the Cognitive Science Society*, vol. 35, 2013.

[12] B. Khayut, L. Fabri, and M. Abukhana, "Knowledge representation, reasoning and systems thinking under uncertainty," in *2014 UKSim-AMSS 16th International Conference on Computer Modelling and Simulation*, 2014, pp. 163–169. DOI: 10.1109/UKSim.2014.56.

[13] A. Oltramari, N. Ben-Asher, L. Cranor, L. Bauer, and N. Christin, "General requirements of a hybrid-modeling framework for cyber security," in *2014 IEEE Military Communications Conference*, IEEE, 2014, pp. 129–135.

[14] R. Thomson, C. Lebiere, and S. Bennati, "Human, model and machine: A complementary approach to big data," in *Proceedings of the 2014 Workshop on Human Centered Big Data Research*, 2014, pp. 27–31.

[15] K. He, X. Zhang, S. Ren, and J. Sun, *Deep residual learning for image recognition*, 2015. arXiv: 1512.03385 [cs.CV].

[16] T.-Y. Lin, M. Maire, S. Belongie, *et al.*, *Microsoft coco: Common objects in context*, 2015. arXiv: 1405.0312 [cs.CV].

[17] N. Moustafa and J. Slay, "Unsw-nb15: A comprehensive data set for network intrusion detection systems (unsw-nb15 network data set)," in *2015 military communications and information systems conference (MilCIS)*, IEEE, 2015, pp. 1–6.

[18] O. Russakovsky, J. Deng, H. Su, *et al.*, "ImageNet Large Scale Visual Recognition Challenge," *International Journal of Computer Vision (IJCV)*, vol. 115, no. 3, pp. 211–252, 2015. DOI: 10.1007/s11263-015-0816-y.

[19] K. Simonyan and A. Zisserman, *Very deep convolutional networks for large-scale image recognition*, 2015. arXiv: 1409.1556 [cs.CV].

[20] S. Potluri and C. Diedrich, "Accelerated deep neural networks for enhanced Intrusion Detection System," in *2016 IEEE 21st International Conference on Emerging Technologies and Factory Automation (ETFA)*, Sep. 2016, pp. 1–8. DOI: 10.1109/ETFA.2016.7733515.

[21] M. Almseidin, M. Alzubi, S. Kovacs, and M. Alkasassbeh, "Evaluation of machine learning algorithms for intrusion detection system," in *2017 IEEE 15th International Symposium on Intelligent Systems and Informatics (SISY)*, IEEE, 2017, pp. 000 277–000 282.

[22] E. Hodo, X. Bellekens, A. Hamilton, C. Tachtatzis, and R. Atkinson, "Shallow and deep networks intrusion detection system: A taxonomy and survey," *arXiv preprint arXiv:1701.02145*, 2017.

[23] J. Kim, N. Shin, S. Y. Jo, and S. H. Kim, "Method of intrusion detection using deep neural network," in *2017 IEEE International Conference on Big Data and Smart Computing (BigComp)*, ISSN: 2375-9356, Feb. 2017, pp. 313–316. DOI: 10.1109/BIGCOMP.2017.7881684.

[24] T. Abdelzaher, N. Ayanian, T. Basar, *et al.*, "Will distributed computing revolutionize peace? the emergence of battlefield iot," in *2018 IEEE 38th International Conference on Distributed Computing Systems (ICDCS)*, 2018, pp. 1129–1138. DOI: 10.1109/ICDCS.2018.00112.

[25] S. Beery, G. Van Horn, and P. Perona, "Recognition in terra incognita," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 456–473.

[26] J. Duchi, K. Khosravi, and F. Ruan, "Multiclass classification, information, divergence and surrogate risk," *The Annals of Statistics*, vol. 46, no. 6B, pp. 3246–3275, 2018. DOI: 10.1214/17-AOS1657. [Online]. Available: https://doi.org/10.1214/17-AOS1657.

[27] N. Dvornik, J. Mairal, and C. Schmid, "Modeling visual context is key to augmenting object detection datasets," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 364–380.

[28] O. Eriksson, A. Jauhiainen, S. Maad Sasane, *et al.*, "Uncertainty quantification, propagation and characterization by Bayesian analysis combined with global sensitivity analysis applied to dynamical intracellular pathway models," *Bioinformatics*, vol. 35, no. 2, pp. 284–292, Jul. 2018, ISSN: 1367-4803. DOI: 10.1093/bioinformatics/bty607. eprint: https://academic.oup.com/bioinformatics/article-pdf/35/2/284/48962822/bioinformatics_35_2_284.pdf. [Online]. Available: https://doi.org/10.1093/bioinformatics/bty607.

[29] R. Geirhos, P. Rubisch, C. Michaelis, M. Bethge, F. A. Wichmann, and W. Brendel, "Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness," *arXiv preprint arXiv:1811.12231*, 2018.

[30] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization.," *ICISSp*, vol. 1, pp. 108–116, 2018.

[31] N. Shone, T. N. Ngoc, V. D. Phai, and Q. Shi, "A deep learning approach to network intrusion detection," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 2, no. 1, pp. 41–50, 2018. DOI: 10.1109/TETCI.2017.2772792.

[32] Z. Sun, E. Ambrosi, A. Bricalli, and D. Ielmini, "Logic computing with stateful neural networks of resistive switches," *Advanced Materials*, vol. 30, 2018. [Online]. Available: https://api.semanticscholar.org/CorpusID:51922522.

[33] W. Brendel and M. Bethge, "Approximating cnns with bag-of-local-features models works surprisingly well on imagenet," *arXiv preprint arXiv:1904.00760*, 2019.

[34] H. Liu and B. Lang, "Machine Learning and Deep Learning Methods for Intrusion Detection Systems: A Survey," en, *Applied Sciences*, vol. 9, no. 20, p. 4396, Jan. 2019, Number: 20 Publisher: Multidisciplinary Digital Publishing Institute, ISSN: 2076-3417. DOI: 10.3390/app9204396. [Online]. Available: https://www.mdpi.com/2076-3417/9/20/4396 (visited on 02/08/2023).

[35] Y. Zhang, X. Chen, L. Jin, X. Wang, and D. Guo, "Network intrusion detection: Based on deep hierarchical network and original flow data," *IEEE Access*, vol. 7, pp. 37 004–37 016, 2019.

[36] M. Almiani, A. AbuGhazleh, A. Al-Rahayfeh, S. Atiewi, and A. Razaque, "Deep recurrent neural network for IoT intrusion detection system," en, *Simulation Modelling Practice and Theory*, Modeling and Simulation of Fog Computing, vol. 101, p. 102 031, May 2020, ISSN: 1569-190X. DOI: 10.1016/j.simpat.2019.102031. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1569190X19301625 (visited on 02/08/2023).

[37] M. Arjovsky, "Out of distribution generalization in machine learning," Ph.D. dissertation, New York University, 2020.

[38] A. N. Cahyo, A. K. Sari, and M. Riasetiawan, "Comparison of hybrid intrusion detection system," in *2020 12th International Conference on Information Technology and Electrical Engineering (ICITEE)*, IEEE, 2020, pp. 92–97.

[39] A. D. Cobb and B. Jalaian, "Scaling Hamiltonian Monte Carlo Inference for Bayesian Neural Networks with Symmetric Splitting," *arXiv preprint arXiv:2010.06772*, 2020.

[40] H. Xu, Y. Ma, H.-C. Liu, *et al.*, "Adversarial attacks and defenses in images, graphs and text: A review," *International Journal of Automation and Computing*, vol. 17, pp. 151–178, 2020.

[41] M. Zhang, C. Tseng, and G. Kreiman, "Putting visual object recognition in context," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 12 985–12 994.

[42] X. Zhang, X. Xie, L. Ma, *et al.*, *Towards characterizing adversarial defects of deep learning software from the lens of uncertainty*, 2020. arXiv: 2004.11573 [cs.SE].

[43] A. Aggarwal, M. Mittal, and G. Battineni, "Generative adversarial network: An overview of theory and applications," *International Journal of Information Management Data Insights*, vol. 1, no. 1, p. 100 004, 2021.

[44] Z. Ahmad, A. Shahid Khan, C. Wai Shiang, J. Abdullah, and F. Ahmad, "Network intrusion detection system: A systematic study of machine learning and deep learning approaches," *Transactions on Emerging Telecommunications Technologies*, vol. 32, no. 1, e4150, 2021.

[45] Z. Ahmad, A. Shahid Khan, C. Wai Shiang, J. Abdullah, and F. Ahmad, "Network intrusion detection system: A systematic study of machine learning and deep learning approaches," *Transactions on Emerging Telecommunications Technologies*, vol. 32, no. 1, e4150, 2021. DOI: https://doi.org/10.1002/ett.4150. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/ett.4150. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/ett.4150.

[46] A. D. Cobb, B. A. Jalaian, N. D. Bastian, and S. Russell, "Robust decision-making in the internet of battlefield things using bayesian neural networks," in *2021 Winter Simulation Conference (WSC)*, IEEE, 2021, pp. 1–12.

[47] Z. Deng, X. Yang, S. Xu, H. Su, and J. Zhu, "Libre: A practical bayesian approach to adversarial detection," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 972–982.

[48] S. Komkov and A. Petiushko, "Advhat: Real-world adversarial attack on arcface face id system," in *2020 25th International Conference on Pattern Recognition (ICPR)*, IEEE, 2021, pp. 819–826.

[49] S. Kumar, S. Gupta, and S. Arora, "Research Trends in Network-Based Intrusion Detection Systems: A Review," *IEEE Access*, vol. 9, pp. 157 761–157 779, 2021, Conference Name: IEEE Access, ISSN: 2169-3536. DOI: 10.1109/ACCESS.2021.3129775.

[50] X. Ma, Y. Niu, L. Gu, *et al.*, "Understanding adversarial attacks on deep learning based medical image analysis systems," *Pattern Recognition*, vol. 110, p. 107 332, 2021.

[51] V. Volodina and P. Challenor, "The importance of uncertainty quantification in model reproducibility," *Philosophical Transactions of the Royal Society A*, vol. 379, no. 2197, p. 20 200 071, 2021.

[52] M. Weiss and P. Tonella, "Fail-safe execution of deep learning based systems through uncertainty monitoring," in *2021 14th IEEE Conference on Software Testing, Verification and Validation (ICST)*, IEEE, 2021, pp. 24–35.

[53] M. Weiss and P. Tonella, "Uncertainty-wizard: Fast and user-friendly neural network uncertainty quantification," in *2021 14th IEEE Conference on Software Testing, Verification and Validation (ICST)*, IEEE, 2021, pp. 436–441.

[54] M. Wicker, L. Laurenti, A. Patane, Z. Chen, Z. Zhang, and M. Kwiatkowska, "Bayesian inference with certifiable adversarial robustness," in *International Conference on Artificial Intelligence and Statistics*, PMLR, 2021, pp. 2431–2439.

[55] D. Ye, L. Veen, A. Nikishova, *et al.*, "Uncertainty quantification patterns for multiscale models," *Philosophical Transactions of the Royal Society A*, vol. 379, no. 2197, p. 20 200 072, 2021.

[56] L. Yu, J. Dong, L. Chen, *et al.*, "Pbcnn: Packet bytes-based convolutional neural network for network intrusion detection," *Computer Networks*, vol. 194, p. 108 117, 2021.

[57] T. Abdelzaher, N. D. Bastian, S. Jha, L. Kaplan, M. Srivastava, and V. V. Veeravalli, "Context-aware collaborative neuro-symbolic inference in iobts," in *MILCOM 2022-2022 IEEE Military Communications Conference (MILCOM)*, IEEE, 2022, pp. 1053–1058.

[58] M. Acharya, A. Roy, K. Koneripalli, S. Jha, C. Kanan, and A. Divakaran, "Detecting out-of-context objects using contextual cues," *arXiv preprint arXiv:2202.05930*, 2022.

[59] Y. A. Farrukh, I. Khan, S. Wali, D. Bierbrauer, J. A. Pavlik, and N. D. Bastian, "Payload-byte: A tool for extracting and labeling packet capture files of modern network intrusion detection datasets," pp. 58–67, 2022. DOI: 10.1109/BDCAT56447.2022.00015.

[60] A. M. Berenbeim, I. J. Cruickshank, S. Jha, R. H. Thomson, and N. D. Bastian, "Measuring classification decision certainty and doubt," *arXiv preprint arXiv:2303.14568*, 2023.

[61] D. A. Bierbrauer, M. J. De Lucia, K. Reddy, P. Maxwell, and N. D. Bastian, "Transfer learning for raw network traffic detection," *Expert Systems with Applications*, vol. 211, p. 118 641, 2023.

[62] T. Bradley, E. Alhajjar, and N. D. Bastian, "Novelty detection in network traffic: Using survival analysis for feature identification," in *2023 IEEE International Conference on Assured Autonomy (ICAA)*, 2023, pp. 11–18. DOI: 10.1109/ICAA58325.2023.00010.

[63] J. Parmar, S. Chouhan, V. Raychoudhury, and S. Rathore, "Open-world machine learning: Applications, challenges, and opportunities," *ACM Computing Surveys*, vol. 55, no. 10, pp. 1–37, 2023.

[64] X. Wang and Z. Zhu, "Context understanding in computer vision: A survey," *Computer Vision and Image Understanding*, vol. 229, p. 103 646, 2023, ISSN: 1077-3142. DOI: https://doi.org/10.1016/j.cviu.2023.103646. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1077314223000267.

[65] M. Weiss and P. Tonella, "Uncertainty quantification for deep neural networks: An empirical comparison and usage guidelines," *Software Testing, Verification and Reliability*, e1840, 2023.

[66] J. A. Wong, A. M. Berenbeim, D. A. Bierbrauer, and N. D. Bastian, "Uncertainty-quantified, robust deep learning for network intrusion detection," *Proceedings of the 2023 Winter Simulation Conference*, 2023.

APPENDIX

Proofs

Proof. (of Proposition 1)

1) Immediate.
2) A proof by induction starting with $d = 2$ can be used to establish that $\det(\mathbf{C}(\mathbf{x})) = 1+(d-1)\|\mathbf{x}\|^2 - 2\sum_{i<j} x_i x_j = 1 + \sum_{i<j}(x_i - x_j)^2$. For all real values, it is clear that this will always be non-zero, whence $\mathbf{C}(\mathbf{p})$ will always be invertible.
3) Expanding $\mathbf{C}(\mathbf{y})$, we have

$$\langle \mathbf{x}, \mathbf{C}(\mathbf{y})\mathbf{x}\rangle = \langle \mathbf{x}, \mathbf{x}\rangle + \langle \mathbf{y}, \mathbf{x}\rangle\langle \mathbf{x}, \mathbf{1}\rangle - \langle \mathbf{x}, \mathbf{y}\rangle\langle \mathbf{1}, \mathbf{x}\rangle$$

which reduces to $\langle \mathbf{x}, \mathbf{x}\rangle$ as desired.

□

Proof. (of Theorem 1)

1) $\mathcal{K} \geq \alpha$ by the hypothesis that $\mathbf{p}$ is relatively α-competent. Whenever $\mathcal{K} < \bar{\mu}_\rho - \bar{\mu}_\nu$, it immediately follows that $\bar{\mu}_\rho > \alpha + \bar{\mu}_\nu$, so suppose that $\mathcal{K} \geq \bar{\mu}_\rho - \bar{\mu}_\nu$.
Let S_T and S_F denote the sum of certainty scores for the true and false positives respectively from $\mathbf{p}$ on sample D_N. Whenever $\mathbf{p}$ is α-competent and $\mathcal{K} \geq \bar{\mu}_\rho - \bar{\mu}_\nu$, we seek to prove the claim that $S_F \in \left[\left(\frac{N_F}{N_T}\right)^2 S_T, (N - N^T)\right]$. That $S_F \leq (N - N^T)$ is immediate since the sum of certainty scores for false positives is at most the number of false positives.
To prove the claim regarding the lower bound, we rearrange

$$\mathcal{K} = \frac{1}{N}(S_T - S_F) \geq \bar{\mu}_\rho - \bar{\mu}_\nu = \frac{1}{N^T}S_T + \frac{1}{N_F}S_F.$$

After algebraic manipulation, we have $S_F \geq \left(\frac{N^F}{N^T}\right)^2 S_T$.
From our bounds on S_F, we have

$$\frac{1}{N}S_T - \frac{1}{N}\left(\frac{N^F}{N^T}\right)^2 S_T \geq \mathcal{K} \geq \frac{1}{N}S_T - \frac{N^F}{N}.$$

By algebraic manipulation,

$$\frac{N^F}{N} \geq \left(\frac{(N^F)^2}{N^T}\right)\bar{\mu}_\rho.$$

Since $\bar{\mu}_\rho \leq 1$, this simplifies to

$$\bar{\mu}_\rho \leq \min\{1, \frac{N^T}{N}\left(\frac{1}{N^F}\right)\}.$$

The lower-bound for $\bar{\mu}_\rho$ follows straightforwardly from $\frac{N^T}{N}\bar{\mu}_\rho \geq \mathcal{K} \geq \alpha$.
To find the bounds for $\bar{\mu}_\nu$, immediately from our analysis above, $\bar{\mu}_\nu \in [\frac{N-N^T}{N^T}\bar{\mu}_\rho, 1]$. We sharpen this upper bound via algebraic manipulation of $\frac{1}{N}(S_T - S_F) = \mathcal{K} \geq \alpha$, isolating S_F on the right-hand side of the inequality, and appropriate scaling.
2) Supposing that $\mathbf{p}$ is α-expert, then $\mathcal{K} \geq \mathcal{CC} > \alpha$, and so the results above apply.

Expanding the definition of α-expert, we have

$$\begin{aligned}\mathcal{K} &= \frac{1}{N}[S_T - S_F] \\ &\geq \mathcal{CC} \\ &= \frac{1}{N}\sum_{i\in[N]}(\langle \mathbf{p}^i, \mathbf{q}^i\rangle - \frac{1}{d}) \\ &> \alpha \geq 0\end{aligned}$$

which we scale by N, deriving

$$S_T \geq S_T - S_F \geq \sum_{i\in[N]}(\langle \mathbf{p}^i, \mathbf{q}^i\rangle - \frac{1}{dN}) > 0.$$

Now, let ϑ_d be given by $\mathbf{p} \mapsto \max_i \pi_i(\mathbf{p}) - \frac{1}{d}$. Clearly $\vartheta_d \geq 0$, as the maximum probability with respect to d distinct labels will always be $\geq \frac{1}{d}$.
Next, partition $[N] = T \bigcup F$ into indices with true positive predictions and false positive predictions. With $\mathbf{p}^i_{\hat{j}}$ denoting the probability of the predicted index and $\mathbf{p}^i_{\tilde{j}}$ the next-highest probability value in $\mathbf{p}^i$, when $\hat{\mathbf{p}}^i = \hat{\mathbf{q}}^i$, we find that $\langle \mathbf{p}^i, \mathbf{q}^i\rangle - \frac{1}{d} = \vartheta_d(\mathbf{p}) \geq 0$. Split the sum over N into sums over subsets T and F, respectively and then substitute in ϑ_d for the T sum:

$$S_T \geq \sum_{i\in T}\vartheta_d(\mathbf{p}^i) + \sum_{i\in F}(\langle \mathbf{p}^i, \mathbf{q}^i\rangle - \frac{1}{d}) > 0$$

Since $\vartheta_d(\mathbf{p}^i) \geq 0$ for all $i \in T$, we consider two cases with respect to the F sum. In the case where $\sum_{i\in F}\langle \mathbf{p}^i, \mathbf{q}^i\rangle - \frac{1}{d} \geq 0$, we may drop the F sum immediately without loss of generality since $\sum_{i\in N}\langle \mathbf{p}^i, \mathbf{q}^i\rangle - \frac{1}{d} \geq \sum_{i\in T}\vartheta_d(\mathbf{p}^i) \geq 0$.
We then rearrange and simplify to find

$$\begin{aligned}S_T - \sum_{i\in T}\vartheta_d(\mathbf{p}^i) &= \sum_{i\in T}\frac{1}{d} - \mathbf{p}^i_{\hat{j}} \\ &> 0.\end{aligned}$$

We then normalize using $\frac{1}{N^T}$ to find

$$\frac{1}{d} > \frac{1}{N^T}\sum_{i\in T}\mathbf{p}^i_{\hat{j}},$$

as desired.
Continuing to finish the proof with respect to the first case, we establish the probability bound as follows:
Since $\check{X} \in [0, \frac{1}{2}]$, we apply Popoviciu's inequality to find that at most $\sigma_{\check{X}} \leq \frac{1}{4}$. Independent of the assumption that the sample average of alternatives and the mean of alternatives agree when defined with respect to sample D_N, by the one-sided Chebyshev inequality derived from Cantelli's inequality,

$$\begin{aligned}\mathbb{P}\{\check{X} \geq \frac{1}{d} + \frac{a}{4}\} &\leq \mathbb{P}\{\check{X} - \mu_\varpi \geq \frac{a}{4}\} \\ &\leq \mathbb{P}\{\check{X} - \mu_\varpi \geq a\sigma_{\check{X}}\} \\ &\leq \frac{1}{1+a^2}.\end{aligned}$$

As a technical aside, if we treat the true distribution as separate from the sampled distribution, then

the one-sided Chebyshev inequality with respect to the sampled average can be made slightly tighter inequality using the bias correction term $\frac{N}{N-1}$, e.g.

$$\mathbb{P}\{\check{X} \geq \frac{1}{d} + \frac{a}{4}\} \leq \frac{1}{1 + \frac{N}{N-1}a^2}$$

since this inequality is made with respect to the worst-case (sample) variance.
In the second case, we have $\sum_{i \in F} \langle \mathbf{p}^i, \mathbf{q}^i \rangle - \frac{1}{d} < 0$. In this sense, we will have an α-expert, α-informed $\mathbf{p}$ that in the average case where it makes a mistake assigns a probability for the correct label below *worse* that of a uniform guess. In the worst-case scenario, the assigned probability is effectively zero. With this in mind, we rearrange the inequality first to read:

$$\begin{aligned} S_T - \sum_T \vartheta_d(\mathbf{p}^i) &\geq S_F + \sum_F (\langle \mathbf{p}^i, \mathbf{q}^i \rangle - \frac{1}{d}) \\ &\geq \frac{-N^F}{d} \end{aligned}$$

This simplifies to

$$\frac{N^F}{d} + \frac{N^T}{d} = \frac{N}{d} \geq \sum_{i \in T} \mathbf{p}^i_j,$$

which we normalize to read:

$$\frac{N}{dN^T} \geq \bar{\check{X}}.$$

From here, we use the same analysis as above to derive the variation of the one-tailed Chebyshev inequality with respect to our new bound on $\bar{\check{X}}$.

3) If $\mathbf{p}$ is prescient, then

$$\sum_{i \in [N]} \varsigma(\mathbf{p}^i)[1\{j \colon y^j = \hat{\mathbf{p}}^j\} - 1\{j \colon y^j \neq \hat{\mathbf{p}}^j\}] \geq \sum_{i \in [N]} \langle \mathbf{p}^i, \mathbf{q}^i \rangle.$$

We can write this to read:

$$\sum_{i \in [N] : y^i = \hat{\mathbf{p}}^i} \varsigma(\mathbf{p}^i) \geq \sum_{i \in [N]} p^i_{\hat{j}} + \sum_{i \in [N] : y^i \neq \hat{\mathbf{p}}^i} \varsigma(\mathbf{p}^i)$$

and since $\varsigma(\mathbf{p}^i) = \min_k \{|p^i_{\delta_{j,k} - \hat{j}} - p^i_k|\} \leq p^i_{\hat{j}}$, it follows that the

$$\sum_{i \in [N]} \varsigma(\mathbf{p}^i) \leq \sum_{i \in [N]} p^i_{\hat{j}},$$

requiring that $\hat{\mathbf{p}}^i = y^i$ for all $i \in [N]$ and further, for equality to obtain, we need $\mathbf{p}^i = \mathbf{q}^i$ for all i. That is, $\mathbf{p}$ needs to both correctly identify the label for every label, but do so with total certainty on all samples. It follows that $\mathcal{CCC} = 1$, and thus $\mathcal{CCC} > \alpha$ for all $\alpha \in [0, 1)$.

□

Full Out-of-Distribution Detection Test

The first stage compares the distributions at a *global* level, comparing the input distribution against the True Positive (TP) distributions and False Positive (FP) distributions. If the Mann-Whitney U test statistic has a p-value below our significance threshold, then we reject the null hypothesis that the global TP distribution and the test distribution are the same. Similarly for the FP distribution. For the ECDF test, we accept if we are above the β threshold.

In the global form of the test, we either conclude that a test distribution belongs or does not belong to the TP and FP distributions respectively. When we conclude that the test distribution globally belongs, we *globally accept*, otherwise we *globally reject*.

The second stage compares the distributions at *local* levels, i.e within the label categories that are predicted by the test distribution. We gather the Mann Whitney U test statistics or ECDF test results in for both TP and FP cases as before. In the second stage, whenever we fail to reject the null hypothesis, we consider the projection of the test distribution into that label to be within the respective distribution. There are five possible local classifications: In-Distribution, Mixed - TP, Mixed, In FP, and Out-of-Distribution. If a test distribution has one label's TP distribution that it is said to belong to, then it is In-distribution. Otherwise, we consider the following subcases local labels:

- If there are more than one local projections of the test distribution that can be identified with the respective reference distributions local label's TP distribution, then we consider this distribution to be Mixed. If we only identify with local TP distributions, we say it is Mixed - TP, to indicate that the test distribution can be partitioned into multiple subsamples which would look identical to the TPs of the reference distribution. Otherwise, we classify the test distribution as Mixed to indicate that it resembles data on which the reference distribution classifies correctly and incorrectly by category.
- If there are no local projections of test distribution that are identified with the local projections of the reference distribution's TP distributions, but it is identified with FPs, then we say it is In FP, to indicate that the test distribution at best resembles data that the reference distribution falsely classifies.
- If there are no local projections of the test distribution that can be identified with local projections of the reference distribution to either TP or FP, we determine that the test distribution is Locally Out-of-Distribution.

Our two-stage evaluation counts accept/reject, global accept/reject, and local decision as follows:

- We always accept as in-distribution the test distributions whose global TP distribution is identified as In-Distribution with the reference global TP distribution and that are locally determined to be In-Distribution.
- We always reject as out-of-distribution the test distributions identified as Out-of-Distribution with respect to both global TP and FP distributions, and that

are locally determined to be Out-of-Distribution in the strong sense..

- If we use the *weak-form* of our test, anything not always identified as in-distribution or out-of-distribution, is labeled as Indeterminate. We also gather the respective intermediate local stage evaluation in order to provide a greater description to the proportion of the Indeterminate data to aid human decision making.
- If we use the *strong-form* of our test, we say anything that is not globally identified with the reference global TP distribution is globally Out-of-Distribution, and is globally rejected, including test distributions that are identified as In-Distribution with respect to global FPs. Further, any local status which is not identified as locally In-Distribution is counted as a local reject. If we reject both globally and locally, then we reject the test distribution as out-of-distribution. Otherwise, the test distribution status is Indeterminate.

Network Architectures

Our FcNN consisted of four dense layers, pictured in Figure 1. The first and third layers used a ReLU activation function, while the second layer used a LeakyReLU activation function. The final layer applied a SoftMax activation function for the output. Further, the final two layers' sizes depended on the number of classes in the data set. We set the output in each model's penultimate layer by dividing the output dimension of the first layer by the number of classes in the data set. The final output layer of each model was sized depending on the number of classes in each data set. All the layers used random normal distribution for the kernel initializer. We compiled the deterministic FcNN using sparse categorical cross-entropy and an Adam optimizer.

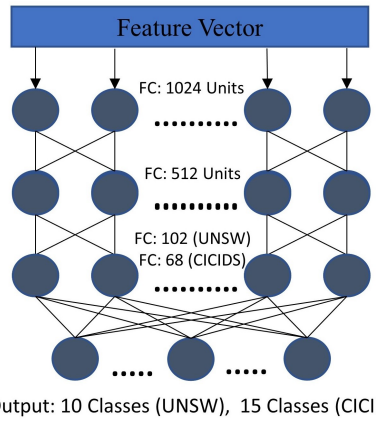

Fig. 1. FcNN Model Architecture

Our 1dCNN architecture, pictured in Figure 2, contained three 1D convolutional layers and three max pooling layers. All convolutional and max pooling layers utilized the same padding and a stride size of one. After flattening the data, the model used two dense layers whose size depended on the number of classes in the data set. We determined the number of units in the first dense layer by multiplying the number of classes in the models' respective data sets by five, and we implemented a ReLU activation function for the first dense layer. The size of the final output layer was equal to the number of classes in the data set and used a SoftMax activation function. The deterministic 1dCNN models were compiled using sparse categorical cross-entropy and an Adam optimizer.

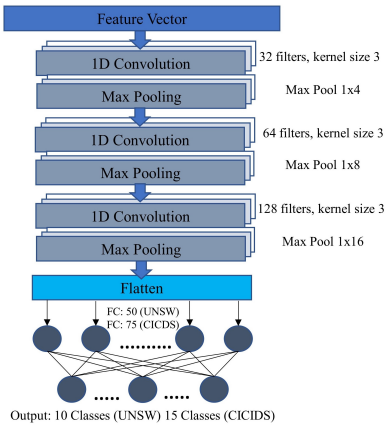

Fig. 2. 1dCNN Model Architecture
The 1D convolutional layers used a ReLU activation function. The filter size started at 32 for the first layer and was doubled for each subsequent layer. The pool size for the max pooling layer doubled with each subsequent layer, beginning at four.

For our COCO and COCO OOC experiments, we used the five pre-trained ResNet models available in the Pytorch library. Residual learning works by reparametrizing a subnetwork of stacked layers and lets the parameter layers represent a residual function. Letting $H(x)$ denote the function performed by the subnetwork with input x, and the parameter layer representing a residual function $F(x) = H(x) - x$. The subnetwork output is referred to as a residual block, and ResNet models are formed by stacking residual blocks (see Figure 3).

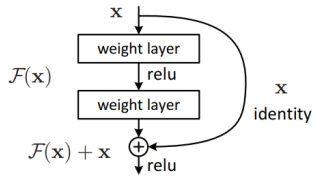

Fig. 3. Example Residual Block and Residual Connection Diagram

The primary advantage of introducing the identity mappings is that they facilitate signal propagation in both forward and backward paths, with backward propagation addressing the vanishing gradient problem[15].

Experimental Figures

Figure 5 displays the box-plot distributions of TP and FP certainty scores for the 8 baseline NIDS models we considered, in every case demonstrating the separation of the distributions that we expected given the competence of each model. In contrast, Figure 6 shows the local (in)competence of the baseline CICIDS models with respect to the DoS data.

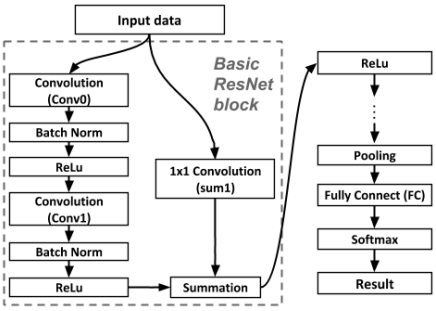

Fig. 4. ResNet Block Structure

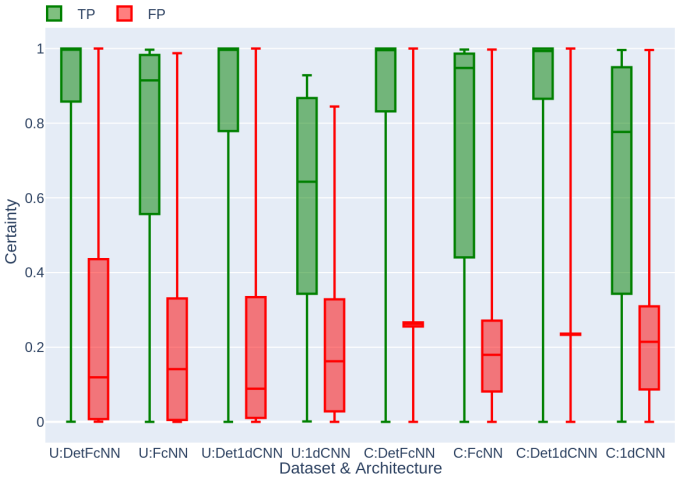

Fig. 5. Certainty Distributions for Baseline Models.

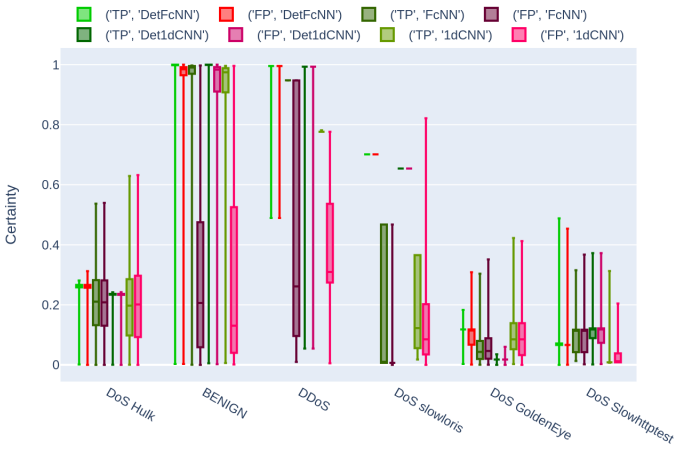

Fig. 6. Certainty Distributions on CICIDS Benign and DoS Data. Every Baseline CICIDS model demonstrated low certainty scores on DoS data, and inconsistent competence on DDoS data, consistent with findings that each CICIDS trained model was generally incompetent with respect to DoS data.

The relative incompetence here motivated our decision to drop DoS data as part of our second NIDS experiment.

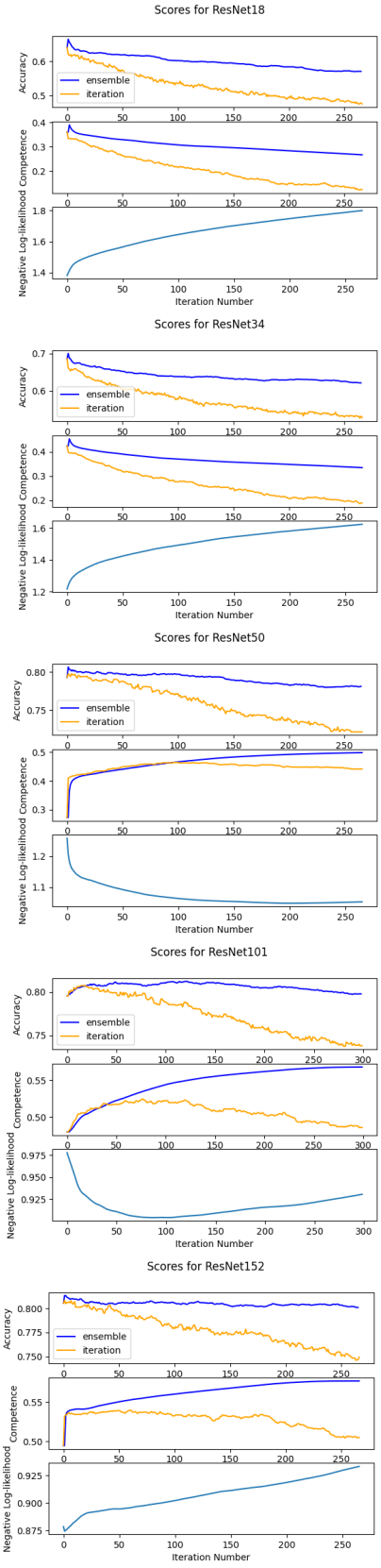

Fig. 7. Accuracy, Empirical Competence, and Negative Log-likelihood of Bayesian ResNets