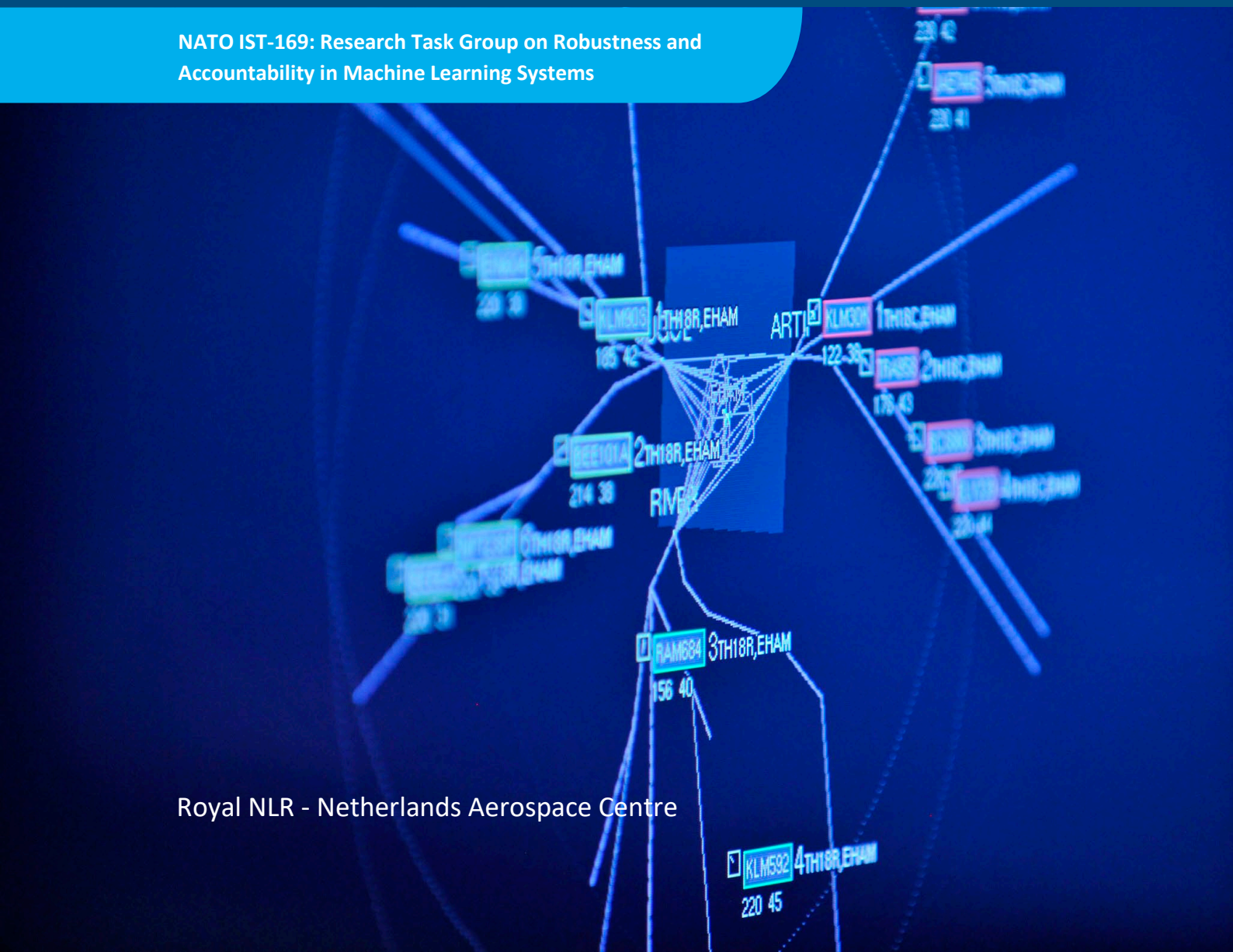
Dedicated to innovation in aerospace

NLR-TP-2021-484 | January 2022

Robustness of Artificial Intelligence for Hybrid Warfare

NATO IST-169: Research Task Group on Robustness and Accountability in Machine Learning Systems



Royal NLR - Netherlands Aerospace Centre

Dedicated to innovation in aerospace

Robustness of Artificial Intelligence for Hybrid Warfare

Problem area

Machine learning systems are a key element in virtually all decision support systems, autonomous systems, and other systems that are important in NATO operations. These systems will over time have the potential to influence both control and over vehicles, sensors and weapons, as well as the decisions made from sensor input.

In order to achieve trust in military systems using complex machine learning models and algorithms, the military needs to be able to prove both robustness and accountability. Robustness is important for the availability and integrity of any military system, with or without both sensors and effectors. Accountability is likely a future requirement for such systems, and the more complex a system becomes, the documentation of accountability will grow towards “non-human” complexity. Military decision makers must be able to document how the military decision making systems operates in order to show why their system recommended those specific actions based on the input from these specific sensors.

REPORT NUMBER
NLR-TP-2021-484

AUTHOR(S)
J. Sharp
J. Melrose
B. Madahar
M. Aktaş
N. Martinel
J.A. de Marchi
E. Solberg
D.S. Lange
G.O. Tanik
F. Kurth
L. Luotsinen

REPORT CLASSIFICATION
UNCLASSIFIED

DATE
January 2022

KNOWLEDGE AREAS
Aerospace Collaborative Engineering and Design

DESCRIPTORS
AI
artificial intelligence
machine learning
neural network
robustness

Description of work

- Determine the state-of-the-art in robustness and accountability for machine learning systems. Especially deep learning systems with complex and large models which are virtually impossible to manage by humans.

- Examine whether a methodology can be made to verify that commercial MLS, e.g. cloud MLS, comply with a set of criteria. Including what kind of criteria that may be possible, and what should be mandatory in a military setting.

Results and conclusions

Hybrid warfare suggests that there may be many systems and many models, so if the assumption is that AI will be used throughout a conglomeration of hybrid warfare systems, the impact of multiple sources of error has the significant potential to undermine AI's application in the military domain.

Applicability

Machine Learning components are building blocks that form part of an overall "AI" decision tree. If we are to propose a degree of confidence to the end recommendation resulting from that decision tree process, then every block therein contributes toward a degree of certitude, therefore each block therein must be able to output not merely a decision/recommendation, but erstwhile a confidence factor.

The ultimate decision is up to a human, therefore the confidence accumulated "factors" leading to that decision must be human-understandable. That's why further research into XAI (explainable artificial intelligence) is warranted.

Because the various blocks in the decision-making chain are implemented by algorithms, XAI demands that there is an element of human-understandable back-traceability. In so many words: the AI ought to be able to defend its decision-making on human-understandable terms.

GENERAL NOTE

This report is based on an article published as "Robustness of Artificial Intelligence for Hybrid Warfare", NATO publication STO-MP-IST-190-17. In the IST-190-RSY Symposium ''AI, ML and BD for Hybrid Military Operations (AI4HMO)', this article won the Best Paper Award.

Royal NLR
Anthony Fokkerweg 2
1059 CM Amsterdam, The Netherlands
p) +31 88 511 3113
e) info@nlr.nl i) www.nlr.nl

Dedicated to innovation in aerospace

NLR-TP-2021-484 | January 2022

Robustness of Artificial Intelligence for Hybrid Warfare

NATO IST-169: Research Task Group on Robustness and Accountability in Machine Learning Systems

AUTHORS:

J. Sharp	UK Defence Science and Technology Laboratory DSTL
J. Melrose	UK Defence Science and Technology Laboratory DSTL
B. Madahar	UK Defence Science and Technology Laboratory DSTL
M. Aktaş	ASELSAN Defence Systems Technologies Division
N. Martinel	University of Udine
J.A. de Marchi	NLR
E. Solberg	Norwegian Defense Research Establishment
D.S. Lange	US Naval Information Warfare Systems
G.O. Tanik	Turkish Aerospace Industries
F. Kurth	Fraunhofer FKIE
L. Luotsinen	Swedish Defence Research Agency FOI

Royal NLR - Netherlands Aerospace Centre

This report is based on an article published as “Robustness of Artificial Intelligence for Hybrid Warfare”, NATO publication STO-MP-IST-190-17.

WORKING GROUP	**NATO**
CONTRACT NUMBER	**IST-169**
OWNER	**NLR + partner(s)**
DIVISION NLR	**Aerospace Vehicles**
DISTRIBUTION	**Unlimited**
CLASSIFICATION OF TITLE	**UNCLASSIFIED**

APPROVED BY:		**Date**
AUTHOR	J.A. de Marchi	07-12-2021
REVIEWER	R.J. Rijken	25-01-2022
MANAGING DEPARTMENT	A.A. ten Dam	11-02-2022

Summary

There are many activities, projects and programs that look at manipulation of machine learning systems (MLS) and how specific systems can be influenced by creative input. But there is too little activity in machine learning research to look at how we can create more robust systems and how such systems might require a fundamental change in training, testing, validation and/or product phases.

One problem might be that commercial MLS may be trained in ways that cannot be verified through the product. Can the products contain back doors in the system, much like software in general, only made by creatively crafting the input/training data? E.g. is it possible to train a missile detection system, that is trained to report no detection on one specific type of missile, and that this manipulation cannot be detected because the machine learning model is too large and complex? This RTG will look into methods for how such training can take place, how training can take place which will avoid these types of challenges, and how systems must be documented in order to avoid being the victim for such solutions as a customer.

Data from military sensors are being fed directly into systems for fast analysis and decisions. Robustness in training phase is only one step towards a more robust overall system. Military systems also need sensor input to be unpredictive enough that the analysis will not be compromised with fake data. Robustness in operations will also be an important area of research.

Another problem might be the accountability of using MLS when decisions have been made. How can the decision be documented at the time of the event in a way that later can be verified was correct with information currently available. This accountability will require machine learning systems, especially dynamic MLSs, to have major changes from todays “take it or leave it” output.

With the extreme growth of machine learning systems into military equipment, it is important to cover the potential problems listed above. In order to achieve trust in military systems using complex machine learning models and algorithms, the military needs to be able to prove both robustness and accountability. Robustness is important for the availability and integrity of any military system, with or without both sensors and effectors. Accountability is likely a future requirement for such systems, and the more complex a system becomes, the documentation of accountability will grow towards “non-human” complexity. Military decision makers must be able to document how the military decision making systems operates in order to show why their system recommended those specific actions based on the input from these specific sensors.

Contents

Abbreviations

ACRONYM	DESCRIPTION
AI	Artificial Intelligence
BNN	Binary Neural Network (a type of NN)
C2	Command and Control
CIFAR	Canadian Institute for Advanced Research (a research institute)
CNN	Convolutional Neural Network (a type of NN)
DARPA	Defense Advanced Research Projects Agency (a research branch of US DoD)
DSTL	Defence Science Technology Laboratory (a research branch of UK MoD)
DoD	US Department of Defense
DNN	Deep Neural Network (a type of NN with multiple internal layers)
FFNN	Feed-Forward Neural Network (a type of NN)
FGSM	Fast Gradient Sign Method (a type of optimization algorithm)
GAN	Generative Adversarial Network (a type of NN that generates datasets)
GNOME	GNU Network Object Model Environment (a computer desktop environment)
GNU	GNU's Not Unix!
GP	Gaussian Process (a type of NN output probability distribution)
GPT	Generative Pre-trained Transformer (an NN-based language prediction model)
IR	Information Retrieval
IST	Information Systems Technology panel (an expert panel within STO)
LIDAR	Light Detection and Ranging (a type of spatial sensor)
ML	Machine Learning
MLS	Multi-Level Scaling (a method for NN input data normalization)
MNIST	Modified National Institute of Standards and Technology (a training dataset)
MoD	UK Ministry of Defence
NATO	North Atlantic Treaty Organization
NLR	Royal NLR - Netherlands Aerospace Centre (a research institute)
NN	Neural Network(s)
PBN	Projected-Belief Network (a type of FFNN)
RADAR	Radio Detection and Ranging (a type of spatial sensor)
SAR	Synthetic Aperture RADAR (a type of spatial sensor)
SGD	Stochastic Gradient Descent (a method for training NNs)
STO	Science and Technology Organization (a branch of NATO)
UK	United Kingdom
Unix	Unix is not an acronym; it is a pun on "Multics", a multi-user operating system
US	United States
VAE	Variational Auto-Encoder (a type of NN architecture)

1 Introduction

Modern hybrid warfare, which it encompasses not just traditional warfare but also political and cyber warfare (amongst others), is becoming increasingly dependent on Artificial Intelligence (AI) to perform in an ever increasing complex environment. Much of modern AI implementations are constructed using Machine Learning (ML) techniques, using data meant to represent the anticipated situations. This means that:

- Most current AI building blocks are built-to-purpose, and whilst intended to generalise to be supportive of real-world inputs, are not always capable of handling unfamiliar situations (inputs). They are 'black-box' designs that perform complex decision-making or environmental interpretations (classifications) in or at near real-time, but which can typically only produce reliable answers for familiar inputs.
- The AI building blocks are typically easy to fool and confuse if provided with information that they have never seen before, or through attacks, which may be imperceptible to humans.

Essentially, what we're dealing with is a matter of susceptibility: modern ML solutions, and indeed other AI solutions, are inherently susceptible to being easily fooled using data unfamiliar to them [2] [3]. This makes, for example, the Command and Control (C2) decision-tree logic that depends on them vulnerable to failure. And of course, we would like to know how to protect ourselves against such vulnerabilities by ensuring that C2 utilising AI is robust to failure.

In summary:

- Many ML approaches are inherently susceptible to environmental changes and attack;
- Ergo, AI relying on MLs (predominately Neural Network (NN) based) are inherently vulnerable;
- Ergo, hybrid warfare relying on AI must be made robust.

1.1 Robustness

Both the training and the operation of ML approaches are based on the following aspects:

1) input data,
2) internal structure, and
3) learning algorithm.

The fragility of ML may be caused by a number of factors. For the purposes of this paper we are assuming that the internal structure of the network is static, sufficiently capable, and secure, and whilst there are a number of other factors, we consider two of the main susceptibilities,

a) poor training data, and
b) previously unrepresented operational data.

Our focus is thus on the input data to the ML solution. To this end, "poor" input data could mean any of the following (noting that this is not an exhaustive list: erroneous, inconsistent, unrepresentative/uncharacteristic, insufficiently diverse, out-of-range/out-of-context, false, incorrect, or intentionally duplicitous.

Naïvely, we assume that ML approaches (particularly NNs) are trained using high-quality ("good") input data: a selective representation of the range of inputs AI might be expected to deal with during operation. The idea being that

later, during operation, AI can produce the "right" decisions for operational data that is similar to the ones against which it was trained. In other words, the AI must be able to interpolate and also, to some extent, extrapolate its reasoning.

Poor training data will lead, at worst, to ML that is unfit for purpose, or at best, a 'dumbed-down' model; that is, one that can only make vague decisions with high uncertainty. However there are also dangers at the other end of the data quality spectrum, because whereas "good" training data might yield a model that can make very accurate decisions, it may only be able to do so using a narrow range of input data. Of course what we desire is ML that meets its performance requirements, but also able to adapt to new circumstances for which it was not originally trained; that is, be able to deal with novelty.

Thus, an important goal in ML is to construct a capability that generalizes well. In narrow applications, we want to ensure that models that have been trained on a sample of the environment work as advertised for the whole environment. Ultimately we want an AI's capability for the environment to be considered all of reality, or at least all of reality, as perceived by humans. In one sense, there are no novel situations. If we view the Newtonian universe and had an impossible amount of memory, all situations could be predicted from current data. But as our ability to model the universe is severely limited, novelty arises, perhaps frequently. There is no possibility of training models for complex environments that will not result in novel situations appearing when those models are introduced into the real world.

Characterising the robustness of the model is therefore challenging, requiring different aspects of a model to be considered with regards to its robustness. Whilst there are many definitions for robustness available, a distinction should be made between the definition used for traditional software robustness, such as that defined by IEEE 24765[4], and the definition when used in relation to an AI model. Not wishing to add a further definition of robustness to the plethora already available, instead in this paper we use the definition provided in ISO CD22989 [5]: Robustness is "the ability of a system to maintain its level of performance under any circumstances. Robustness properties demonstrate the ability (or inability) of the system to have comparable performance on new data as on the data on which it was trained or the data of typical operations."

1.1.1 Robustness metrics

Having defined the term robustness, and since the focus of this paper is on robustness metrics, we will now define the term metric when applied to robustness. To inform our thought process in writing a definition, it is helpful to identify the various purposes a robustness metric could have and who the associated stakeholders might be. Since the purpose and requirements of a robustness metric will depend on the lifecycle phase of the ML model, we analyse the purpose with respect to lifecycle phases.

Although many ML models will be based on NNs, our analysis expands to cover different variants of ML type and architecture, noting that the predominant variants of ML are: NNs, Decision Trees, and Reinforcement Learning.

During the ML model design and development phase, developers will be experimenting with model designs and adjusting the model's architecture and parameters to optimise the model's performance. At this phase, the purpose of a robustness metric could be both to provide a means of measuring improvements in robustness as these changes are made, and to describe how the robustness is manifested by the model. Additionally, a metric agreed between developers of different models will allow reliable comparisons to be made between model designs.

At the system design phase, in selecting off-the-shelf ML models for incorporation within the whole system, a robustness metric will inform decisions made by system designers on the choice of model by providing a means of comparing both the level and nature of the robustness of one model against another.

Prior to the deployment phase, a robustness metric will be used by security practitioners to inform security risk assessment of systems incorporating ML. Specifically, the metric will inform a vulnerability analysis of the ML model with a low level of robustness representing a vulnerability which an attacker could exploit.

Finally, during the deployment phase, an overall system robustness metric derived from the robustness metrics of the individual ML components will support an end user's trust and confidence in the system's output or behaviour.

Given the above spectrum of uses and associated stakeholders, for the purposes of this paper we will leave the definition of the term metric intentionally broad. Our definition extends beyond the pure act of measurement or quantification to include how we might describe or characterise the robustness of ML in any particular setting. We therefore base the remainder of this paper on the following definition:

> *A robustness metric is a measure or characterisation of the robustness of a ML model against the variety of challenges that it may face during its lifecycle. The precise nature of a particular metric will depend on the type of ML model, the task the model is designed to accomplish, and the phase of the model's lifecycle.*

1.2 Methodology & Paper Structure

When considering robustness metrics, we started the analysis by asking the question "robustness in the face of… what?" This generated a list of situations which ML models could be faced with in which their robustness might be challenged. We call these the "in the face of" conditions:

- Uncertainty in training and operational data
- Inputs that are different from the training set, yet consistent with the training population statistically or semantically
- Inputs that are outside the training population
- Learning with limited data
- Novel situations, different from how learned policies and classifiers were developed
- Adversarial action

Our literature search rendered many pieces of prior research on robustness, and for each of these we attempted to identify into which of the in the face of categories they fitted. Although this was not always obvious, it seemed to represent a logical way of structuring the analysis. In the following paragraphs therefore, an attempt has been made to classify the individual pieces of research from the literature search in this way.

For each of the in the face of categories we describe the nature and surrounding detail of each robustness challenge followed by the type of metrics used to quantify the robustness. Although the examination of robustness within this paper does not include a running hybrid warfare example, what is discussed is applicable to hybrid warfare approaches.

2 Challenges and Metrics

2.1 Uncertainty in training and operational data

Being able to deal with uncertainty in training and operational data is of paramount importance for AI and it represents a key component of current ML systems, especially those which are exploited under critical situations, like the military domain.

2.1.1 Challenges

In ML the aim is to learn the parameters of a model that best fits the training data, given a considered cost function. Then, the model is deployed to get predictions on new and unseen data. As a result of the training process, any learned model comes with uncertainty since its generalisation capabilities are necessarily based on a process of induction, i.e., replacing specific observations with general models of the data-generating process [6]. Despite the many efforts by the research community no existing ML model has ever been proven correct, since any possible experiment relies heavily on hypothetical assumptions, hence every current ML model output remains uncertain when subject to previously unseen input data.

Uncertainty has a long history within the statistical field and, since the beginning, it has often been linked and perceived as a concept likened to standard probability and probabilistic predictions. In the recent past, however, due to the current hype in ML and to the fact that nowadays systems based on such solutions are controlling our everyday lives, there is an increasing interest in such a concept by the community. This has first and foremost been motivated by safety requirements, for which new methodologies are required.

Among the different methodologies in the existing literature that have discussed uncertainty in ML, it is possible to cluster them accordingly to the type of uncertainty that is considered. Most of the current works address either the aleatoric or the epistemic uncertainty.

2.1.1.1 Aleatoric and Epistemic Uncertainty

The traditional way to model the uncertainty in ML is through the application of probability theory. Such probabilistic modelling generally handles a single probability distribution, thus neglecting the importance of distinguishing between the aleatoric and epistemic uncertainty [7] [8].

Aleatoric uncertainty: we can refer to it as statistical uncertainty, which derives from the notion of randomness in the variability of the outcome of an experiment. In simple words, when referring to aleatoric uncertainty we implicitly refer to the uncertainty that cannot be reduced even in the presence of any additional source of information. Let us describe this by considering a very basic example: suppose we want to model the probability outcome of flipping a coin. We can define a probabilistic model that is able to provide the probability of *heads* or *tails*, but not the guaranteed outcome. This uncertainty defines the irreducible part of the total uncertainty.

Epistemic uncertainty: also known as systematic uncertainty, this is the part of the total uncertainty that is determined by the ignorance/lack of knowledge. This uncertainty is due to the epistemic state of the ML system, and it can be reduced through additional information. For example, let us assume that we have an ML model learning a new language and it is given a new word, and it should guess if it means head or tail. The agent is equally uncertain about the correct answer as it would be in predicting the flip of a coin, however by including additional information into the situation (i.e., providing a synonym or explaining the correct meaning of the word) we can cancel out any uncertainty in the answer. It should be therefore clear enough that, as opposed to the aleatoric, the epistemic uncertainty defines the reducible part of the total uncertainty.

Now that we have defined the aleatoric and epistemic uncertainties, we take into consideration supervised ML algorithms and how these two distinct types of uncertainties are represented in ML.

In supervised learning setting, we have access to a training set $\boldsymbol{D} = \{(x_1,y_1),\ldots,(x_n,y_n)\}$ composed by n tuples (x_i, y_i) where x_i is the i^{th} sample containing the features (i.e., measurements) belonging to the instance space $\boldsymbol{X}$, and y_i is the associated target variable coming from the set of possible outcomes $\boldsymbol{Y}$.

Under such a scenario, an ML algorithm comes with three sources of uncertainty:

- Aleatoric uncertainty: in general, the dependency between $\boldsymbol{X}$ and $\boldsymbol{Y}$ is not deterministic. As a result, we can have more than one possible outcome for a given input x_i . Even in presence of full information, uncertainty exists about the actual outcome y_i .
- Model uncertainty: the model chosen to tackle a given problem can be far from being the best one for the task. This is due to the uncertainty on the correctness of the model and the correctness of the hypothesis.
- Approximation uncertainty: the model parameters learned through the optimization process are only an estimate of the true hypothesis. The quality of such an estimation is due to the lack of fidelity in data used during the learning process.

Both the model and approximation uncertainty represent the epistemic uncertainty.

It should be noticed that, for ML algorithms, the aleatoric and the epistemic uncertainty are heavily dependent on the context. For instance, by permitting the learning process the possibility of changing the initially defined setting, it could be possible to reduce the aleatoric uncertainty in favour of epistemic uncertainty; that is, the aleatoric uncertainty in the original context has not changed, but is altered by changing the context (akin to weighting one side of a coin in the coin toss example). In contrast, if we take into consideration a fixed initial setting, we know that the epistemic uncertainty (i.e., the lack of knowledge of the ML algorithm) depends on the amount of data (how many observations) used during the learning process. As the number of training samples tends to infinity, the ML system is able to completely reduce the approximation uncertainty.

2.1.2 Machine learning methods for representing uncertainty

Different ML methods for representing uncertainty have different capabilities, which can be clustered according to:

i. the way in which uncertainty is represented
ii. if dealing with both or just one among the two types of uncertainty (aleatoric and epistemic)
iii. if they provide any solution that can be exploited to provide a rough estimate of the amount of uncertainty.

2.1.2.1 Gaussian Processes

Gaussian Processes (GPs) [9] are a generic modelling tool for supervised learning. They can be applied to generalise the Bayesian inference of a multivariate random variable and inference of functions. In the case of classification, GPs have discrete outcomes, and the difficulty on the definition of uncertainty lies in the representation of the knowledge that is then identified as the epistemic uncertainty of the model, as in the Bayesian approaches. In the case of regression, it is possible to distinguish between the aleatoric uncertainty – that is the variance of the error term – from the epistemic uncertainty.

2.1.2.2 Maximum likelihood estimation and Fisher Information

In ML, the principle of maximum likelihood estimation plays a key role. Indeed, if a model can get "very close" to the maximum of the likelihood function, it means that small changes of the data are likely to have a limited effect on the estimation. If the likelihood function is flat, it might be a good indicator that there is a high level of uncertainty about the estimate that might be due to the many parameters' configurations having similar likelihoods.

In ML, we often exploit the Fisher matrix [10] to represent the amount of epistemic uncertainty [11].

2.1.2.3 Generative models

Generative models can be used to quantify epistemic uncertainty. Given the probabilistic nature of these approaches, which aim to model the density of the data distribution, by establishing if a given datum lies in a region with high or low density, such models implicitly provide information about the epistemic uncertainty. The most relevant works in this category are based on kernel density estimation or Gaussian mixtures, with some recent advancements in deep autoencoders [12].

Density estimation is a key ingredient for methods tackling anomaly and outlier detection where the latter is nothing but a classification problem, where an example is considered out the distribution when it lies in low-density regions. Such works instead capture aleatoric uncertainty.

In general, generative models tackle a very challenging problem, need a lot of data to properly work, and usually have high uncertainty.

2.1.2.4 Deep neural networks

An artificial deep neural network (DNN) is inherently a probabilistic classifier, and we can define the procedure of training a DNN as performing maximum likelihood inference. This results in a model that is able to produce probabilistic estimates of a given input datum, but cannot provide details and information about its probabilities' confidences: aleatoric uncertainty is captured, epistemic is not. Despite this, the latter is generally referred to as the uncertainty in the model parameters. In the literature there have been recent works [13] [14] trying to model this kind of epistemic uncertainty by introducing Bayesian extensions to DNN.

2.1.2.5 Model Ensembles

Common examples of the Model Ensembles class of models are bagging or boosting. Such approaches are very popular because they can provide significant accuracy improvement of point prediction by producing a set of predictions instead of a single hypothesis [15]. The most relevant works that can be included in such a category are the random forests models [16]. Methods in this category are mostly concerned with the aleatoric part of the overall uncertainty.

2.1.2.6 Credal sets and classifiers

The credal set is a set of probability distributions and it is the basis for a generalisation of Bayesian inference where each single prior distribution is replaced by a credal set of candidate priors. Works [17] [18] investigating how to define uncertainty for credal sets, and related representations, define two types of uncertainty that are present in credal sets: “conflict” due to the randomness, and “non-specificity”. These directly correspond to aleatoric and epistemic uncertainty; it is common to use the Hartley function [19] as a standard uncertainty measure; [20] also defines a tool that can be used to assess the robustness of an ML system in the face of uncertainty in training and operational data. This function, The Hartley function, can be used to evaluate uncertainty if we know that the unknown value of a given random variable is within a given finite set. Further, extensions to infinite sets have been proposed through Hartley-like [80] and generalized Hartley [81] measures.

2.2 Inputs that are different from the training set, yet consistent with the training population statistically or semantically

During operation, a classifier assigns a class label to each sample of input data. Considering the above definitions of robustness, the intra-class variability, i.e., the possible variation between all samples assigned to the same class, is implicitly contained in the set of training data used for learning the classifier.

2.2.1 Robustness with respect to semantic data variations

Using a more constructive approach to defining robustness helps to better model the user’s expectation towards classifier performance. To this end, we will for the moment call a classifier robust if it is invariant with respect to all meaningful variations of the input data. Obviously, the set of all meaningful variations depends on the application scenario and is in general very difficult to describe. However, for many classification problems, such meaningful variations can be divided into two categories:

i. physical modifications (e.g., noise addition, mix with distortions, cropping, rotation, scaling), and
ii. semantical modifications (e.g. different ways of pronouncing a spoken word) of the input sample.

Figure 1(1) illustrates those two categories of possible variations for the example of handwritten digit classification. We consider different variations of writing the digit ‘9’. Whereas (as depicted in Figure 1) noise addition (a) and mix with distortions (b) can be regarded to fall into the first category, the third type (c), adding a small arc to the digit ‘9’,

is a meaningful (syntactic) variation, particular to local culture in different countries, which leaves the semantics of the symbol ("nine") intact.

2.2.1.1 Physical robustness

Robustness of AI/ML with respect to the first category of variations, yet not solved satisfactorily, has been addressed to a considerable extent in recent years. In many publications dealing with robustness to the first category of variations, underlying data samples are modelled as vectors in Euclidean vector space. Distortions are then modelled by adding norm-bounded vectors to the data samples. Here, usually Lebesgue-type norms (LP norms) are used (in particular L1, L2, and L∞). In a widely cited paper [20] it was shown that such L2-norm-bounded "adversarial attacks" can be used to cause misclassifications in neural network-based classifiers. Subsequently, much work was done in the field of both adversarial attacks and corresponding protection methods (discussed in further detail later in the paper). Is was shown that attacks are in many cases difficult to detect and that for then state-of-the art methods, detection could be bypassed [21]. Obviously, robustness in this context requires protection from adversarial attacks. Many ways of defining robustness in such an adversarial attack setting can be captured under a common framework, as shown in [22].

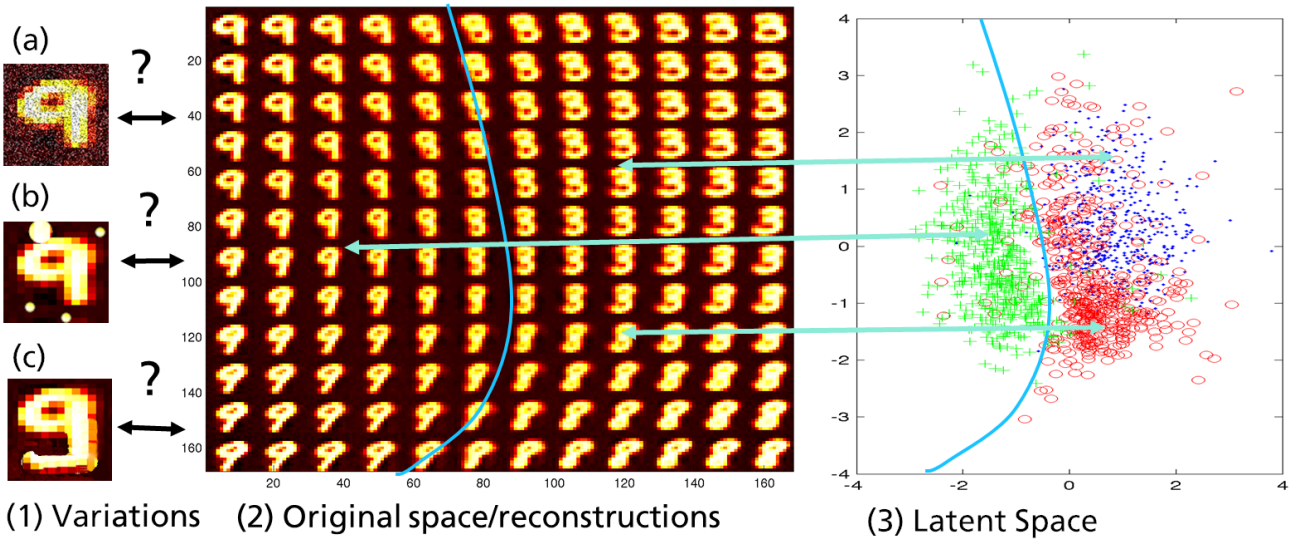


Figure 1: (1) Possible data variations of handwritten digit 9, (2) Space of digits 3, 8, 9 reconstructed using a Variational Auto Encoder (VAE) trained on the respective digits from MNIST corpus, (3) corresponding latent space representation with colours

2.2.1.2 Semantic robustness

The second category, semantically meaningful variations of data samples, leads to substantial challenges which are largely unsolved to date. Correspondingly, in [68], robustness to so called perceptual perturbations is referred to as an open research problem. Although modern AI-based classifiers, and in particular deep neural networks, achieved record-breaking improvements on the well-known public classification challenges, their discriminative nature, in contrast, does not naturally lead to easy interpretability of classification results. In recent years, a whole branch of

research has focused on explainable AI, i.e., on ways to formally, or even semantically, characterise the sets of samples mapped to the same classes by a given classifier.

An important approach towards understanding semantics of a classifier is to combine successful discriminative classifiers with generative models. The advantage of the generative approach is that examples from the original (sample) space can be generated using those models. A successful approach combining classifiers and generative models is that of Generative Adversarial Networks (GANs) [24].

A generative model that can also be adapted for classification is the (Variational) Autoencoder (VAE) [25]. The basic idea of an auto encoder is to learn a compact representation of the original data by training a deep neural network having full dimensional (with respect to the original data) layers at both ends and a sparse “bottleneck” layer in the middle. Figure 1 (2) and (3) illustrates how a VAE can be used for “understanding” the classes learned by the network: (2) shows a representative set of reconstructions obtained by the generative part of a VAE trained to classify the digits ‘3’, ‘8’ and ‘9’ of the MNIST dataset. Thus, in a sense, (2) summarises what the classifier is prepared to recognise. On the right of Figure 1, (3) shows a latent-space representation of the input samples (i.e., MNIST digits) obtained from the classifier branch of the VAE. Colours encode the three digits. Correspondences between latent space points and reconstructed samples are shown as arrows. In blue, curves separating the manifold of 9's from the other digits are sketched to indicate the learned classification boundary. Considering this example we notice that the above variation (c) is not well-represented in the reconstruction part (2) - which is not surprising considering that the corpus is biased by the North American style of writing digits. Hence, to make the classifier robust to the variation (c), additional measures would have to be applied such as augmenting or adding to the training data.

Following this motivation of using generative models, Buzhinsky et al. [26] propose several metrics to measure robustness of classifiers to “natural” adversarial examples. To this end they propose a set of six performance metrics working in the latent space and subsequently show connections between the above classical adversarial robustness and “latent adversarial robustness”, i.e., robustness to perturbations in the latent space. The interesting aspect of the latter is that latent space perturbations for several examples have been shown to correspond to semantically meaningful changes in the original sample space.

We note that classical adversarial robustness can already be used to obtain “certified” robustness of AI-based classifiers with respect to small norm-bounded perturbations. However, semantic robustness is more difficult to formalise and also strongly connected with properly understanding and modelling the target classes. To this end, generative models are an important tool. Novel concepts such as Projected Belief Networks (PBNs), i.e., layered generative models based on a feed-forward neural network structure, having the advantage of possessing a tractable likelihood function, are very promising in this area [27].

A recent piece of work [75] concerns a form of ML called Complex Event Processing, in which multimodal inputs with spatial and temporal relationships from multiple sensors are fused to allow a deep learning model to infer a particular type of event e.g. a gunshot or an explosion. Such events are termed “complex events”. As such, the concept of robustness applies not to the model itself, but to the overall system of components which the machine learning functionality comprises. The research claims that the combination of

a) human logic in pre-defining complex events based on patterns and sequences, with
b) the deep learning inferences from individual sensors, improves the robustness of the system against misclassification.

2.3 Inputs that are outside the training population

In [78] Ashmore et al. identify a set of definitions with respect to the input domain, and subsets thereof:

> ***I,*** the input domain space – the set of inputs a model can accept;
> ***O***, the operational domain space – the set of inputs that a model may be expected to receive when used within the intended operational domain
> ***F***, the failure domain space – the set of inputs a model may receive if there are failures elsewhere in the system; and
> ***A***, the adversarial domain space – the set of inputs that a model may receive if it is being attacked by an adversary,

where ***O***, ***F***, and ***A*** are all subsets of ***I***. These definitions are useful when thinking not just about inputs outside of the training population (that could be drawn from ***O***, ***F***, or ***A***), but more generally when reasoning about the inputs to a model.

Small, pixel-space, perturbations, which may be imperceptible to a human, often with the perturbation magnitude measured using LP norms, are a justifiable method of assessing the robustness of a model (and thus discussed later in Section 2.6); particularly in a hybrid warfare domain where the potential for an adversarial attack is higher. However, when considering assessing the robustness of a model, these small perturbations are not necessarily applicable outside of Ashmore's attack domain space (***A***). Recent, separate, work [79] [80] has begun to investigate perturbing the inputs to a model away from the much discussed, and researched, small perturbation approaches, instead generating what are deemed contextually relevant, and human distinguishable, perturbations: these perturbations look to introduce sheer, blur, or haze, etc., over the input (which could reasonably be representative of inputs from either ***F*** or ***O***).

Further, in [80], the authors propose to introduce meaningful perturbations to images that are semantically relevant, but which may not have been incorporated within the models training set; for example, introducing a flock of geese to a scene where the model is identifying the number of vehicles in a car park. Whilst this last category of meaningful perturbations is clearly part of Ashmore's input domain space (***I***), arguably, if the training dataset was insufficient, these semantically relevant perturbations could also be considered part of the operational domain space (***O***). Interestingly, [80] also identifies that when increasing the robustness of a system to small perturbations, the models may become less robust to dealing with semantically meaningful perturbations, thus it is clearly important to consider assessing a models robustness to both these perturbation types.

In order to assess the levels to which a model is robust to such semantically meaningful, or contextually relevant, perturbations, the authors of [80] propose a titration method for introducing the perturbation, such that one measures, incrementally, the level to which a perturbation can be introduced before the accuracy of the model becomes suspect (eg,, by its confidence, or a change in classification, of a known ground truth). This provides a further metric by which to assess the robustness of a model, when considering its application within the intended operational domain space.

2.4 Learning with limited data

It is well known that modern AI using deep learning requires large quantities of data to learn complex tasks. If the training data is too small, the model will overfit and its generalisation capabilities will be poor. Unfortunately, acquiring high-quality training data is difficult and expensive as it often requires human labelling efforts. For instance, the fine grained Cityscapes dataset took on average 1.5h to label per sample [28]. Furthermore, unlike datasets developed for academic purposes (proof-of-concept, evaluation, benchmarking, etc.), military datasets must also contain data representing the large number of edge cases that are likely to occur, but hard to observe or even predict, in the real-world. Without such training data the military model will be of limited practical value when it may matter the most, or when conditions unexpectedly change as a result of adversarial actions.

The data acquisition challenge for military applications is significant, yet critical to address, to ensure that models will be robust when deployed in the real-world. Fortunately, many transfer learning techniques [29] [30] [31] have been proposed that leverage the fact that deep neural networks seem to learn general features that are transferable and, hence, can be reused by other similar tasks [32]. Pre-training combined with fine-tuning are commonly used to learn with little/limited data, while at the same time avoiding expensive re-training of large scale models (e.g., GPT-3) that may require specialized hardware to learn. The main idea is to:

1) copy parts of the pre-trained source model into a target model;
2) add one or more randomly initialized (untrained) layers to the target model such that the last layer now matches the target's label space; and finally,
3) train the model using labelled target domain data.

However, these techniques cannot be used in cases where military data originates from special sensors (e.g. LIDAR, IR, SAR and hyperspectral) where pre-trained models rarely exists, or are too sensitive to share even among allies.

Unsupervised domain adaptation is another transfer learning technique that, although it has been studied for decades in shallow learning, recently also received a lot of attention in deep learning [33]. Using this technique labelled training data from a source domain can be used to train a model using unsupervised data from the target domain. The approach assumes that labelled data is cheap and easy to acquire for the source.

The idea is intriguing from a military perspective because the source data could potentially be synthetic. That is, simulators or other generative models that already exist could potentially be adapted to generate, not only perfectly labelled source data, but also data representing edge cases that are otherwise difficult or even impossible to acquire. The simulation-based approach would completely eliminate the human labelling efforts that may otherwise result in incorrect, biased and incomplete datasets that would also transfer into the model when trained. Closing the "simulation to real" gap (sim2real) using unsupervised domain adaptation is actively being pursed [34] [35] using a variety of techniques, many of which rely on using adversarial approaches such as domain loss functions [36] [37], and generative adversarial networks (GANs) [38] [39].

2.5 Novel situations, different from how learned policies and classifiers were developed

To be useful in complex environments AI must exhibit robustness to novelty. Demonstrations by DeepMind [41] have shown that ML can be used to develop policies that result in superhuman play in rigid games. The game "Go", provides a complex environment that exceeds our limits for storing the possible states of the game, therefore providing the situation that is discussed earlier concerning our limits to modelling the Newtonian universe. Yet, if the rules of the game are changed, the agents generated can become brittle or fail completely. In [42] this type of result was demonstrated in a much simpler environment allowing experiments to illuminate how different changes affected the robustness of the agent.

But novelty is not simply a situation where a data point is not included in the training set for ML. In an attempt to bring together research on novelty, [43] proposes a framework to describe novelty. Figure 2 provides an illustration of how one can look at novelty in a way that can allow measurement for both the novelty and the agent's response. Critical to this view of novelty is that novelty can be considered with regard to the world and relative to the agent's experience. It is also clear that novelty that has an impact on the agent's task affects robustness differently than novelty in the world that has no effect on the tasks. This is also a finding demonstrated in Chao [42].

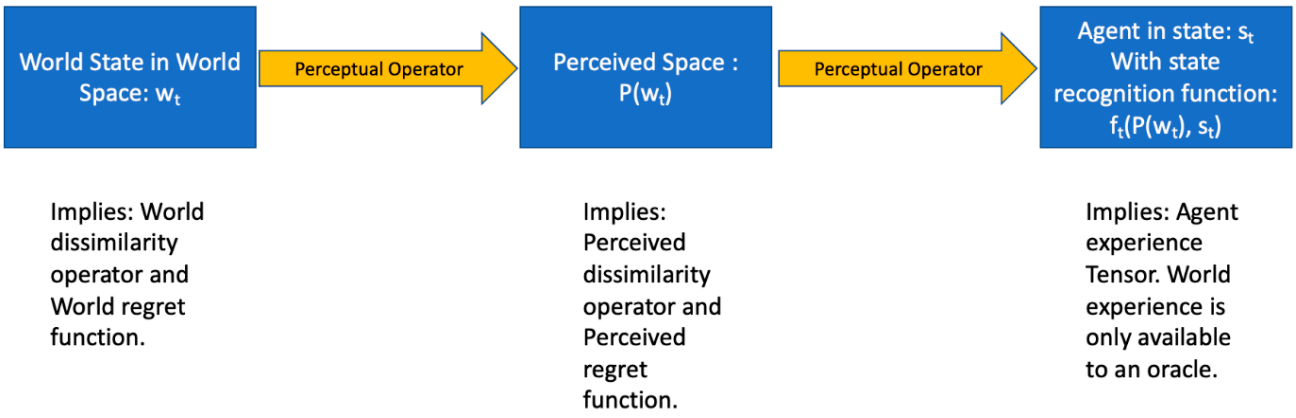


Figure 2: Framework for Novelty. After [43]

2.5.1 The DARPA SAIL-ON Program

One method for experimentation in novelty that is being employed in the DARPA SAIL-ON program [40] is through games. The DARPA SAIL-ON program postulates that agents may have the following four elements:

- A performance element that uses known expertise to pursue tasks and achieve goals (e.g., finding and collecting underwater objects with desired features) through mechanisms for perception, inference, planning, and control;
- A monitoring element that compares observations with expectations to detect anomalies in both the environment (e.g., sonar unreliable, unfamiliar predators) and in the agent's own behaviour (e.g., vehicle veering to right);

- A diagnostic element that localizes problems in expertise, generates hypotheses about the causes (e.g., non-reflective surfaces, cross currents, misaligned propeller), evaluates alternative candidates, and selects among them; and
- A repair element that revises those facets of expertise deemed to be responsible for the performance problems and corrects them (e.g., updated sonar equations, current-sensitive controller, or a new propeller model).

As mentioned in the introductory section on novelty above, much of this research began with the realization that the methods used by DeepMind to solve the games of Go, Chess, Shogi, and Starcraft, were not robust to changes in the rules of the games. An example is the Gnome framework developed at University of Southern California (USC) and published through GitHub .

NIWC Pacific has worked with USC to develop a version of the UK DSTL developed the “Hunting of the Plark” game using the GNOME framework. This will allow experimentation in the effects of novelty on agents trained to play this game, which was the focus of a Turing Institute Study Group. Further experimentation is planned on decision support tools developed using ML where we can not only work with simulated situations, but take out on live experiments with the US Navy.

2.5.2 Detection of Novelty

One can be robust to novelty without ever knowing that a change in the world’s situation has occurred. Most likely this is due to the novelty not being important to the task being performed, or at least is a change in an areas where there is less sensitivity. However, one strategy for dealing with novelty is to at least detect that an agent is in a novel situation, even if the agent won’t know how to work in the novel environment other than quitting or alerting others to the situation.

The fundamental question for the agent is: has the environment changed or is the data being analysed simply on one of the tails of the previous distributions? For much of ML, currently, it may be sufficient simply to recognise that data are out of sample. ML that can at least recognize its own limitations is a step forward in many instances. The classic adversarial example demonstrations are often brought up in this regard: agents are often very confident of their wrong answer in these experiments [44].

In a planning system, recognition might be based on a dynamic evaluation of the progress of a task. If a plan isn’t working, one possibility is that the world has changed in a manner that is not reflected in the models. Early detection might prevent catastrophic results, but that is not guaranteed. Indeed, one can envision scenarios from which there is no recovery (making a turn past the event horizon of a black hole is an extreme example).

2.5.3 Robust Response to Novelty

The task of providing a robust response was defined by [45] as the following:

- Given: An agent architecture that uses expertise to operate in a class of environmental situations;
- Given: Expertise that supports acceptable agent performance in this class of environments;

- Given: Limited experience with an environment in which sudden, unannounced changes degrade performance;
- Find: When environmental changes occurred, and what revised expertise will support acceptable performance.

The type of response to novelty is related to the type of task being performed. In classifiers, a system may need to adjust its model to allow not only a change to the answer it provides, but also an explanation as to what the difference means. For example, imagine a perceptual agent that determines the presence of obstacles for a robot. A change to the camera system, such as a fly landing on the lens might create a new novel situation for the system. If the system is able to adapt and decides that no obstacle exists, an explanation of the situation will be necessary to justify the answer.

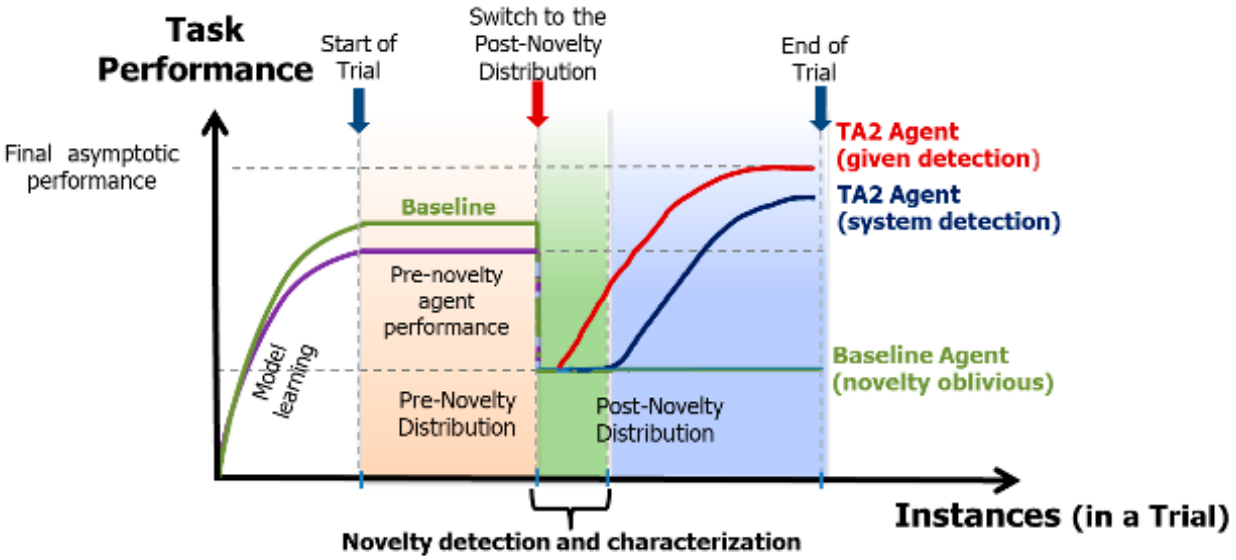


Figure 3: SAIL-ON Novelty Metric Assumptions. Note TA2 agents in the program are those reacting to novelty in the environment

For planning systems, novelty might take the form of working with new actions or finding that actions cost a different amount than before; the goals might drastically change. Planning systems may have to adapt their knowledge, and recompute previous tasks, utilising experience to alter their computation. The assumptions in Figure 3 above, illustrate an environment for measurements. Learning and operation may proceed for a while before novelty appears in the environment. Agents that are not yet robust to that particular change drop off in performance and must find a way to detect that novelty has occurred, determine what has changed and account for it in operation.

2.6 Adversarial Action

Over the past few decades, it has been shown that machine learning models based on deep learning techniques can achieve and even surpass human-level performance in a variety of tasks. On the other hand machine learning models are often vulnerable to perturbation of their input and can easily be fooled to yield incorrect output [53] [54]. These types of manipulations are referred to as adversarial attacks and the performance of machine learning models against these attacks are measured as adversarial robustness [55]. Adversarial robustness is investigated in two different camps. In the first camp, the researchers try to find a method to generate adversarial attacks to decrease the

robustness of the models the most [56] [57] [58] [59] [48]. The researchers in the second camp try to find better training or defensive methods that make the network architectures more robust to such adversarial attacks [60] [61] [62] [63] [64]. In this section, we survey the methods for adversarial attacks and defences and we define the metrics and measurement methods of the adversarial robustness from current literature.

2.6.1 Adversarial Attack

The adversarial attack was defined in [54] for a machine learning system ***M*** and input sample ***C***, which is called clean example as follows:

> *"Assuming that sample* **C** *is correctly classified by the machine learning system, i.e.* **M**(**C**) = y. *It's possible to construct an adversarial example* **A** *which is perceptually indistinguishable from* **C** *but is classified incorrectly, i.e.* **M**(**A**) ≠ y."

Based on this definition, the aim of the adversarial attack is to modify the model input to result in incorrect model output such that it cannot be distinguished by the human observer. Indistinguishability criteria comes with some limitations on the perturbation that can be applied to input, which is referred to as LP norm in literature, i.e.,

$$\left|\left|\boldsymbol{C}-\boldsymbol{A}\right|\right|_p \leq \epsilon,$$

where ϵ is the maximum allowable perturbation. The most commonly used norms are L2 and L∞.

Considering this limitation, several methods are proposed to generate adversarial examples [65] [55] [48]. Generating adversarial examples follows mainly two different approaches, i.e., black-box and white-box. In black-box approaches, the user has no knowledge of the model and can only access the predicted probabilities or just the predicted class for a given input. On the other hand, the model and its parameters are assumed to be completely known in a white-box approach [47].

The white-box attacks are more effective in fooling the models than the black-box attack, and widely investigated, with different approaches, in the literature [56] [57] [58] [48]. White-box attacks are predominantly gradient-based attack methods: they usually construct a loss function that can cause the improvement of the perturbation attacking ability and decreasing of the perturbation magnitude, which then optimizes the loss function through gradients to generate adversarial examples [66]. Using the gradient of the loss function to determine the adversarial perturbations can be performed in a single step like in Fast Gradient Sign Method (FGSM) [65] for fast generation adversarial examples. In order to improve the effect and reduce the perturbation, instead of taking a single step in the direction of the gradient, multiple smaller steps are taken in iterative gradient-based attacks [54][48].

Adversarial attacks can also occur as part of the training activity. The setting for some recent work [46] was a peer-to-peer network in which each peer has a copy of a neural network model to create a distributed learning set-up, which does not rely on the existence of a central co-ordinating node. Such a machine learning architecture is well suited to a military coalition scenario with multiple partners. Initially, each peer has a subset of the total training dataset and as training of the models progresses, model parameters are shared between peers at each training iteration.

Rather than attempting to improve the robustness of the peer-to-peer ML, the motivation for this experiment, based on the Fashion-MNIST dataset, was to measure and optimise the effectiveness of the poisoning technique in causing peers to misclassify. The metric for efficacy of poisoning was how quickly, in terms of the number of training iterations, the malicious peer could reliably poison the benign peers. However, we believe the same metric could be used to infer the robustness of the ML to this type of poisoning: the higher the number of iterations required to achieve misclassification, the higher the robustness.

2.6.2 Adversarial Defence

Methods have been proposed for guaranteeing robustness towards norm-bounded adversarial attacks under particular conditions. For example, Wong and Kolter [67] propose provable defences for ReLU-based classifiers using the concept of adversarial polytopes. Furthermore, an efficient and complete robustness verifier for piecewise-linear neural networks was proposed in [68]. In that paper, an algorithm is proposed yielding certified bounds on the adversarial error based on the maximum (L^∞-) norm.

One of the most successful methods to obtain robust deep neural networks is through adversarial training. The main motivation of adversarial training is to cast both attacks and defences into a common theoretical framework, naturally encapsulating most prior work on adversarial examples [55]. In this method, instead of feeding samples from the original dataset directly into training, the adversarial attack is allowed to perturb the input first and then the perturbed examples are fed into training. Adversarial training has been augmented in different ways, such as changing the attack procedure, loss function or model architecture [69] [50].

The performance of the adversarial training highly depends on the loss function and adversarial attack method used in generating augmented training dataset, and it takes much longer when compared with clean training due to the need for adversarial examples generation. In [73] it has been demonstrated that the performance of state-of-the-art adversarial training methods can be improved more easily using classical adversarial training with early stopping. This shows that our understanding about adversarial training is limited. The effects of adversarial training on robustness is analysed in [74], and they conclude that during the clean training process using (stochastic) gradient descent, neural networks will accumulate, in all features, some “dense mixture directions” that have low correlations with any natural input, but are extremely vulnerable to (dense) adversarial perturbations. During adversarial training, such dense mixtures are “purified” to make the model more robust.

2.6.2.1 Implicit generative modelling of random noise during training improves adversarial robustness

Recent work [70] carried out looked specifically at the approaches identified above. Indeed the work aimed to make deep neural networks more robust to adversarial inputs by introducing random noise into the training inputs and optimizing it with Stochastic Gradient Descent (SGD), while minimizing the overall cost function over the training data. The effect was that the input noise, which is randomly initialized at the beginning, is gradually learnt during the training process. As a result, the noise approximately models the input distribution to effectively maximize the likelihood of the class labels given the inputs.

The authors [70] evaluated their approach on classification tasks such as MNIST, CIFAR10 and CIFAR100 and showed that models trained in such a way are more adversarially robust. The way in which the noise and the clean images were combined was found to have a major impact on accuracy with multiplication achieving far higher accuracy than addition. A direct metric for robustness did not evolve, but rather robustness was quantified as a function of accuracy as the level of perturbation was increased.

2.6.2.2 Discretization-based solutions against adversarial attacks

Following on from the theme of adversarial training, [72] shows that the robustness of an image classification deep neural network to adversarial inputs can be improved by discretization of both the input space and the model's parameter space with minimal loss of accuracy. In the experiments using MNIST, CIFAR10, CIFAR100, and ImageNet datasets, discretization of the input space involved reducing the number of pixel intensities from 256 (2^8) to 4 (2^2) and discretization of the parameter space involved training the model with low precision weights and activations such as Binary Neural Networks (BNN). Further, combining these two discretization techniques greatly improves the model's robustness. This combined scheme can be seen as an alternative way of improving robustness compared to the more expensive process of adversarial training (i.e. training the model using adversarial examples). In each experiment, a measure of robustness is achieved by comparing the accuracy of the classification whilst the amount of adversarial perturbation ϵ is progressively increased. In effect, the metric for robustness in this work would appear to be the degree of perturbation that can be tolerated whilst retaining a given accuracy.

2.6.2.3 Mitigating adversarial examples in neural networks

In a final example, a relatively simple piece of recent work [71] was undertaken. Preconditioning of the input to an image classifier was achieved by feeding the input through a Gaussian kernel, whose effect was equivalent to a smoothing low-pass filter where the level of smoothing depended on the kernel's standard deviation parameter. The experiment was conducted using the MNIST dataset and measured accuracy for varying combinations of smoothing and various levels of adversarial noise. The results showed that to optimise accuracy for a given level of adversarial noise, there existed an optimum level of smoothing. In this case, a metric used for robustness was the percentage of successful attacks for a given amount of adversarial noise. This metric allowed for a direct comparison of performance with, and without, smoothing.

2.6.3 Measuring Adversarial Robustness

Adversarial robustness can be measured as the model accuracy for the inputs perturbed by the adversarial attacks [47]. Since the evaluation depends on the applied adversarial attack, it is hard to measure the actual adversarial robustness of the model.

Most of the works in literature reported adversarial robustness of their approaches by using the same or similar adversarial attack method and loss function that are used in their training phase. It has been shown in [48] that by changing the loss function and the method to generate adversarial examples, lower adversarial robustness than reported in the original papers can be achieved. Indeed, it is stated in [48] that the change in robustness is larger than 10% in 13 out of 49 cases and larger than 30% in eight (8) cases.

A similar evaluation is performed in [49] by comparing the performance of several deep neural networks against human observers for different types of manipulations. In this work it has been shown that deep neural networks can achieve human-level performance only when the applied manipulations are known in the training phase. For unknown manipulations, the performance of the deep neural networks decreases dramatically. Further, many of the defence strategies proposed in the literature were broken by stronger adversaries [48] [50]. Hence, the comparison of robustness obtained under different methods should be performed carefully to make sure the evaluation is as strong as possible [47].

The adversarial robustness is reported as the model accuracy on worst-case inputs taken from perturbed sets.
In addition to the accuracy, two types of performance metrics can also be measured for evaluating the robustness of the model. The first metric is adversarial frequency, which measures how often the model fails to be robust [51].
The second one is adversarial severity, which is used to measure the expected minimum distance from original input to an adversarial example [51] [52], i.e., how easily the model can be fooled. Indeed, quoting [51]:

> *"The frequency and severity capture different robustness behaviours. A neural net may have high adversarial frequency but low adversarial severity, indicating that most adversarial examples are [a very small] distance away from the original point. Conversely, a neural net may have low adversarial frequency but high adversarial severity, indicating that it is typically robust, but occasionally severely fails to be robust. Frequency is typically the more important metric, since a neural net with low adversarial frequency is robust most of the time. Indeed, adversarial frequency corresponds to the accuracy on adversarial examples used to measure robustness. Severity can be used to differentiate between neural nets with similar adversarial frequency."*

3 Conclusions from here

Hybrid warfare suggests that there may be many systems and many models, so if the assumption is that AI will be used throughout a conglomeration of hybrid warfare systems, the impact of multiple sources of error has the significant potential to undermine AI's application in the military domain.

The criteria and survey of current techniques above therefore are all relevant in understanding the potential weaknesses in applying AI & ML into a hybrid military domain, and thus where considerations pertaining to the robustness of AI & ML, are involved there is a clear need to ensure a wide reaching assessment going forward. It is clear that there is a significant area of considerations, and available metrics. However, as previously posed in Section 2, these metrics are applicable to different stakeholders, for different models, and potentially different tasks.

Thus, the ongoing question is how to determine and find the right kinds of metrics for the specific models to obtain the required level of confidence in hybrid warfare systems. IST-169 intends to progress this initial survey to do just that. We believe that developing a pictorial representation of the various types of robustness, with their applicable phases to the different types of AI, would benefit a holistic understanding of the AI robustness landscape. This would enhance the move toward a more rigorous approach to the development and use of AI applications.

4 References

[1] Steps Toward Robust Artificial Intelligences, AAAI President's Address. AAAI 2016. Phoenix, AX. February 14, 2016. Accessed on 13 April 2021 at http://web.engr.oregonstate.edu/~tgd/talks/dietterich-aaai-presidents-address-final.pdf

[2] Tom B. Brown and Dandelion Mané and Aurko Roy and Martín Abadi and Justin Gilmer, Adversarial Patch, CoRR, abs/1712.09665, 2018

[3] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian J. Goodfellow, and Rob Fergus. Intriguing properties of neural networks. CoRR, abs/1312.6199, 2013

[4] ISO/IEC/ IEEE 24765 Systems and software engineering — Vocabulary, Second Edition, 2017

[5] ISO CD22989

[6] Hüllermeier E., Waegeman W. (2021). Aleatoric and epistemic uncertainty in machine learning: an introduction to concepts and methods. Machine Learning, 110, 457-506.

[7] Hora S. (1996). Aleatory and epistemic uncertainty in probability elicitation with an example from hazardous waste management. Reliability Engineering and System Safety, 54(2–3), 217–223

[8] Der Kiureghian A., Ditlevsen, O. (2009). Aleatory or epistemic? Does it matter? Structural Safety, 31, 105–112

[9] Vanhatalo, J., Riihimäki, J., Hartikainen, J., Jylänki, P., Tolvanen, V., & Vehtari, A. (2013). GPstuff: Bayesian modeling with Gaussian processes. Journal of Machine Learning Research, 14(Apr), 1175-1179

[10] Frieden, B. (2004). Science from Fisher information: A unification. Cambridge: Cambridge University Press

[11] Sourati, J., Akcakaya, M., Erdogmus, D., Leen, T., & Dy, J. (2018). A probabilistic active learning algorithm based on Fisher information ratio. IEEE Transactions on Pattern Analysis and Machine Intelligence, 40(8), 2023–2029

[12] Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep learning. Cambridge, MA: The MIT Press

[13] Denker, J., & LeCun, Y. (1991). Transforming neural-net output levels to probability distributions. In Proceedings of NIPS, advances in neural information processing systems

[14] Neal, R. (2012). Bayesian learning for neural networks (p. 118). Berlin: Springer

[15] Martinel, N., Piciarelli, C., & Micheloni, C. (2016). A supervised extreme learning committee for food recognition. Computer Vision and Image Understanding, 148, 67–86

[16] Breiman, L. (2001). Random forests. Machine Learning, 45(1), 5–32

[17] Yager, R. (1983). Entropy and specificity in a mathematical theory of evidence. International Journal of General Systems, 9, 249–260

[18] Kolmogorov, A. (1965). Three approaches to the quantitative definition of information. Problems of Information Transmission, 1(1), 1–7

[19] Hartley, R. (1928). Transmission of information. Bell Labs Technical Journal, 7(3), 535–563

[20] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian J. Goodfellow, and Rob Fergus. Intriguing properties of neural networks. CoRR, abs/1312.6199, 2013

[21] Carlini and Wagner. Adversarial Examples Are Not Easily Detected: Bypassing Ten Detection Methods, 2017. https://arxiv.org/abs/1705.07263

[22] T. Dreossi, S. Ghosh, A. Sangiovanni-Vincentelli, and S. A. Seshia. A formalization of robustness for deep neural networks, 2019. https://arxiv.org/abs/1903.10033

[23] Eric Wong, J. Zico Kolter, Provable defenses against adversarial examples via the convex outer adversarial polytope, 2018, https://arxiv.org/abs/1711.00851

[24] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, Yoshua Bengio: Generative Adversarial Networks. In: NIPS. 2014

[25] Diederik P. Kingma, Max Welling, An Introduction to Variational Autoencoders, CoRR, 2019

[26] Igor Buzhinsky, Arseny Nerinovsky, Stavros Tripakis, Metrics and methods for robustness evaluation of neural networks with generative models, https://arxiv.org/abs/2003.01993

[27] Paul M. Baggenstoss, The Projected Belief Network Classfier : both Generative and Discriminative, EUSIPCO 2020, Amsterdam 2021

[28] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016.

[29] S. J. Pan and Q. Yang. A survey on transfer learning. IEEE Transactions on Knowledge and Data Engineering, 22(10):1345–1359, 2010

[30] Karl Weiss, Taghi M Khoshgoftaar, and DingDing Wang. A survey of transfer learning. Journal of Big data, 3(1), 2016

[31] Chuanqi Tan, Fuchun Sun, Tao Kong, Wenchang Zhang, Chao Yang, and Chunfang Liu. A survey on deep transfer learning. In International Conference on Artificial Neural Networks (ICANN), pages 270–279, 2018

[32] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? In Proceedings of the Advances in Neural Information Processing Systems (NIPS), volume 27, 2014

[33] Garrett Wilson and Diane J. Cook. A survey of unsupervised deep domain adaptation. ACM Transactions on Intelligent Systems and Technology, 11(5), 2020

[34] Tuan-Hung Vu, Himalaya Jain, Maxime Bucher, Matthieu Cord, and Patrick Perez. Advent: Adversarial entropy minimization for domain adaptation in semantic segmentation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), June 2019

[35] Ajay Kumar Tanwani. Domain invariant representation learning for sim-to-real transfer. In Proceedings of the Conference on Robot Learning (CoRL), 2020

[36] Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. In Proceedings of the 32nd International Conference on International Conference on Machine Learning (ICML), volume 37, pages 1180–1189, 2015

[37] Minghao Chen, Shuai Zhao, Haifeng Liu, and Deng Cai. Adversarial-learned loss for domain adaptation. Proceedings of the AAAI Conference on Artificial Intelligence, 34(04):3521–3528, 2020

[38] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), pages 2242–2251, 2017

[39] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. Image-to-image translation with conditional adversarial networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 5967–5976, 2017

[40] T. Senator, Science of AI and Learning for Openworld Novelty (SAIL-ON). Presented at the Proposers' Day Meeting. DARPA: Arlington, VA. https://www.darpa.mil/attachments/SAIL-ON Proposers Day Distro A no notes.pdf

[41] Silver, D., Schrittwieser, J., Simonyan, K. et al. Mastering the game of Go without human knowledge. Nature 550, 354–359 (2017)

[42] J. Chao, J. Sato, C. Lucero, D. Lange, Evaluating Reinforcement Learning Algorithms For Evolving Military Games, AAAI Symposium on The 2nd Workshop on Deep Models and Artificial Intelligence for Defense Applications: Potentials, Theories, Practices, Tools, and Risks 2020

[43] T. E. Boult, P. A. Grabowicz, D. S. Prijatelj, R. Stern, L. Holder, J. Alspector, M. Jafarzade, Towards a Unifying Framework for Formal Theories of Novelty, AAAI 2021

[44] Carlini and Wagner. Adversarial Examples Are Not Easily Detected: Bypassing Ten Detection Methods, 2017. https://arxiv.org/abs/1705.07263

[45] P. Langley, "Open-World Learning for Radically Autonomous Agents", AAAI 2020

[46] Tomsett, Chan and Chakraborty. Investigating the Robustness of Peer-to-peer Machine Learning. Annual Fall Meeting of the DAIS ITA, 2019

[47] Nicholas Carlini, Anish Athalye, Nicolas Papernot, Wieland Brendel, Jonas Rauber, Dimitris Tsipras, Ian Goodfellow, Aleksander Mądry and Alexey Kurakin, On Evaluating Adversarial Robustness, arXiv:1902.06705v2

[48] Francesco Croce and Matthias Hein, Reliable Evaluation of Adversarial Robustness with an Ensemble of Diverse Parameter-free Attacks, arXiv:2003.01690v2

[49] Robert Geirhos, Carlos R. Medina Temme, Jonas Rauber, Heiko H. Schütt, Matthias Bethge and Felix A. Wichmann 'Generalisation in humans and deep neural networks', 32nd Conference on Neural Information Processing Systems (NeurIPS 2018), Montréal, Canada

[50] Sven Gowal, Chongli Qin, Jonathan Uesato, Timothy Mann and Pushmeet Kohli, Uncovering the Limits of Adversarial Training against Norm-Bounded Adversarial Examples, arXiv:2010.03593v3

[51] Bastani, O., Ioannou, Y., Lampropoulos, L., Vytiniotis, D., Nori, A., Criminisi, A., Measuring neural net robustness with constraints, Advances in Neural Information Processing Systems, pp. 2613–2621 (2016)

[52] Igor Buzhinsky, Arseny Nerinovsky and Stavros Tripakis, Metrics And Methods For Robustness Evaluation Of Neural Networks With Generative Models, arXiv:2003.01993v2

[53] Nilesh Dalvi, Pedro Domingos, Sumit Sanghai, Deepak Verma, et al. Adversarial classification. In Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 99–108. ACM, 2004

[54] Alexey Kurakin, Ian J. Goodfellow and Samy Bengio, Adversarial Machine Learning At Scale, arXiv:1611.01236v2

[55] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras and Adrian Vladu, Towards Deep Learning Models Resistant to Adversarial Attacks, arXiv:1706.06083v4

[56] Anish Athalye and Ilya Sutskever, Synthesizing robust adversarial examples, Int. Conf. Mach. Learn., 2018

[57] Logan Engstrom, Andrew Ilyas, and Anish Athalye, Evaluating and Understanding the Robustness of Adversarial Logit Pairing, arXiv preprint arXiv:1807.10272, 2018

[58] Jonathan Uesato, Brendan O'Donoghue, Aaron van den Oord, and Pushmeet Kohli, Adversarial Risk and the Dangers of Evaluating Against Weak Attacks, Int. Conf. Mach. Learn., 2018

[59] Felix Assion, Peter Schlicht, Florens Greßner, Wiebke Gunther, Fabian Huger, Nico Schmidt and Umair Rasheed, The Attack Generator: A Systematic Approach Towards Constructing Adversarial Attacks, arXiv:1906.07077v1

[60] Nicolas Papernot, Patrick McDaniel, Xi Wu, Somesh Jha, and Ananthram Swami, Distillation as a defense to adversarial perturbations against deep neural networks, IEEE Symposium on Security and Privacy, 2016

[61] Jiajun Lu, Hussein Sibai, Evan Fabry, and David Forsyth, NO Need to Worry about Adversarial Examples in Object Detection in Autonomous Vehicles, IEEE Conf. Comput. Vis. Pattern Recog., 2017

[62] Harini Kannan, Alexey Kurakin, and Ian Goodfellow, Adversarial Logit Pairing, arXiv preprint arXiv:1803.06373, 2018

[63] Guanhong Tao, Shiqing Ma, Yingqi Liu, and Xiangyu Zhang, Attacks Meet Interpretability: Attributesteered Detection of Adversarial Samples, Adv. Neural Inform. Process. Syst., 2018

[64] Haichao Zhang and Jianyu Wang, Defense Against Adversarial Attacks Using Feature Scattering-based Adversarial Training, Adv. Neural Inform. Process. Syst., 2019

[65] Goodfellow, I. J., Shlens, J., and Szegedy, C. Explaining and harnessing adversarial examples, In ICLR, 2015

[66] Hongying Liu, Zhenyu Zhou, Fanhua Shang, Xiaoyu Qi, Yuanyuan Liu and Licheng Jiao, Boosting Gradient for White-Box Adversarial Attacks, arXiv:2010.10712v1

[67] Eric Wong, J. Zico Kolter, Provable defenses against adversarial examples via the convex outer adversarial polytope, 2018, https://arxiv.org/abs/1711.00851

[68] Vincent Tjeng, Kai Xiao, Russ Tedrake, Evaluating Robustness of Neural Networks with Mixed Integer Programming, 2019. https://arxiv.org/abs/1711.07356

[69] Minghao Guo, Yuzhe Yang, Rui Xu, Ziwei Liu and Dahua Lin, When NAS Meets Robustness: In Search of Robust Architectures against Adversarial Attacks, arXiv:1911.10695v3

[70] Panda, Roy. Implicit Generative Modeling of Random Noise during Training Improves Adversarial Robustness. arXiv:1807.02188v4 [cs.LG] 31 May 2019

[71] Alzantot, Chakraborty and Srivastava. Mitigating Adversarial Examples in Neural Networks. 1st Annual Fall Meeting of the DAIS ITA, 2017

[72] Panda, Chakraborty and Roy. Discretization Based Solutions For Secure Machine Learning Against Adversarial Attacks. IEEE Access 7 (2019): 70157-70168

[73] Leslie Rice, Eric Wong, and J. Zico Kolter. Overfitting in adversarially robust deep learning. Int. Conf. Mach. Learn., 2020

[74] Zeyuan Allen-Zhu and Yuanzhi Li, Feature Purification: How Adversarial Training Performs Robust Deep Learning, arXiv:2005.10190v2

[75] Xing, Vilamala, Garcia, Cerutti, Kaplan, Preece and Srivastava. DeepCEP: Deep Complex Event Processing Using Distributed Multimodal Information. IEEE International Conference on Smart Computing (SMARTCOMP) 2019

[76] George J. Klir (2011) A note on the Hartley-like measure of uncertainty, International Journal of General Systems, 40:2, 217-229

[77] Andrey G. Bronevich and Igor N. Rozenberg (2021). Generalized Hartley Measures on Credal Sets. Proceedings of Machine Learning Research, 147:32-41

[78] Rob Ashmore, Radu Calinescu, Colin Paterson. Assuring the Machine Learning Lifecycle: Desiderata, Methods, and Challenges. ACM Comput. Surv. 54(5): 111:1-111:39 (2021)

[79] Colin Paterson, Haoze Wu, John Grese, Radu Calinescu, Corina S. Pasareanu, Clark W. Barrett. DeepCert: Verification of Contextually Relevant Robustness for Neural Network Image Classifiers. CoRR abs/2103.01629 (2021)

[80] Isaac Dunn, Hadrien Pouget, Daniel Kroening, Tom Melham. Exposing previously undetectable faults in deep neural networks. ISSTA 2021: 56-66

Dedicated to innovation in aerospace

Royal NLR - Netherlands Aerospace Centre

NLR operates as an objective and independent research centre, working with its partners towards a better world tomorrow. As part of that, NLR offers innovative solutions and technical expertise, creating a strong competitive position for the commercial sector.

NLR has been a centre of expertise for over a century now, with a deep-seated desire to keep innovating. It is an organisation that works to achieve sustainable, safe, efficient and effective aerospace operations.
The combination of in-depth insights into customers' needs, multidisciplinary expertise and state-of-the-art research facilities makes rapid innovation possible. Both domestically and abroad, NLR plays a pivotal role between science, the commercial sector and governmental authorities, bridging the gap between fundamental research and practical applications. Additionally, NLR is one of the large technological institutes (GTIs) that have been collaborating over a decade in the Netherlands on applied research united in the TO2 federation.

From its main offices in Amsterdam and Marknesse plus two satellite offices, NLR helps to create a safe and sustainable society. It works with partners on numerous programmes in both civil aviation and defence, including work on complex composite structures for commercial aircraft and on goal-oriented use of the F-35 fighter. Additionally, NLR helps to achieve both Dutch and European goals and climate objectives in line with the Luchtvaartnota (Aviation Policy Document), the European Green Deal and Flightpath 2050, and by participating in programs such as Clean Sky and SESAR.

For more information visit: **www.nlr.org**

Postal address
PO Box 90502
1006 BM Amsterdam, The Netherlands
e) info@nlr.nl **i)** www.nlr.org

Royal NLR
Anthony Fokkerweg 2
1059 CM Amsterdam, The Netherlands
p) +31 88 511 3113

Voorsterweg 31
8316 PR Marknesse, The Netherlands
p) +31 88 511 4444

Chamber of Commerce No. 41150373. VAT No. NL002760551B01