View metadata, citation and similar papers at core.ac.uk
brought to you by CORE
provided by UCL Discovery

Why the Failure? How Adversarial Examples Can Provide Insights for Interpretable Machine Learning

Richard Tomsett*
IBM Emerging Technology, IBM Research
Hursley, UK

Amy Widdicombe
Dept of Computer Science, UCL
London, UK

Tianwei Xing
Electrical & Computer Engineering Dept, UCLA
Los Angeles, USA

Supriyo Chakraborty
IBM Research
Yorktown, USA

Simon Julier
Dept of Computer Science, UCL
London, UK

Prudhvi Gurram
Booz Allen Hamilton Army Reseach Laboratory
Adelphi, USA

Raghuveer Rao
Army Research Laboratory
Adelphi, USA

Mani Srivastava
Computer Science Dept
Electrical & Computer Engineering Dept, UCLA
Los Angeles, USA

***Abstract*— Recent advances in Machine Learning (ML) have profoundly changed many detection, classification, recognition and inference tasks. Given the complexity of the battlespace, ML has the potential to revolutionise how Coalition Situation Understanding is synthesised and revised. However, many issues must be overcome before its widespread adoption. In this paper we consider two — *interpretability* and *adversarial attacks*. Interpretability is needed because military decision-makers must be able to justify their decisions. Adversarial attacks arise because many ML algorithms are very sensitive to certain kinds of input perturbations.**

In this paper, we argue that these two issues are conceptually linked, and insights in one can provide insights in the other. We illustrate these ideas with relevant examples from the literature and our own experiments.

***Index Terms*—interpretability, interpretable machine learning, deep learning, adversarial machine learning, adversarial examples, explainable AI, AI alignment, internet of battlefield things**

I. Introduction

Recent advances in machine learning (ML), particularly deep learning (DL), have begun to have a profound impact in many areas of decision-making [1]. Within military operations, ML has the potential to revolutionize the way in which Situational Awareness (SA) is developed and revised [2]: by fitting parameter values of flexible and general models directly to data, it is possible to create algorithms that can be far more accurate and capable than those using features engineered directly by humans. These advantages are extremely important in new war fighting concepts, such as the Internet of Battlefield Things (IoBT) in which the battlefield is populated by multiple agents [3], [4] which collect many types of hard and soft data.

However, before ML can be applied to IoBT and CSU, many challenges must be overcome. In this paper we consider two: *interpretability* and *adversarial examples*. Interpretability is required because military decision-makers must be able to provide reasoned justifications for their decisions. Therefore, the ML systems must a provide level of explanation to support this justification. Adversarial examples arise because many ML systems can exhibit sensitivities which means that a carefully crafted input can cause them to make mistakes [5]–[7].

Although interpretability and adversarial examples are not often considered together, we argue that they are conceptually linked, and that research into one has the potential to provide valuable insights into the other. The existence of adversarial examples illustrates that ML models do not learn input mappings and class boundaries that align with our intentions as model builders, despite the model performing well at the given task. Adversarial examples could thus be used to better understand a model's decision surfaces and feature representations. Improving interpretability will allow us to improve model alignment through a better understanding of how best to design and train the model, as well as helping to spot mistakes by providing explanations for model decisions. This should allow us to build models that are more robust to adversarial examples.

Our contributions are as follows: in Section II we introduce the motivating example of SA in a military coalition operation. In Sections III and IV, we survey the literature related to interpretability and adversarial examples respectively, developing specific ideas with reference to the coalition context. Following these discussions, we develop our central thesis which links the two concepts. In Section V we propose how adversarial examples could be employed to improve ML interpretability, while in Section VI we consider how interpretability techniques could be employed to improve defences against adversarial examples, illustrating our ideas with some preliminary experimental results. Conclusions are drawn in Section VII.

II. Coalition Situational Understanding in the Internet of Battlefield Things

The work in this paper is motivated by the need to develop Coalition Situational Understanding (CSU) in the Internet of Battlefield Things (IoBT). The IoBT vision is illustrated in Fig. 1, which shows three collaborating coalition partners

* rtomsett@uk.ibm.com

(blue, green and yellow). In the IoBT vision, the future battlefield is populated by multiple smart machines which can act as agents. Agents can be of different types. These include sensors, munitions, weapons, vehicles, robots, and human-wearable devices. They can sense, communicate, and collaborate with one other and with human warfighters [3], [4]. The data from these different agents must be combined to create SU. This SU must be formed at two levels: *within* each coalition partner, and *amongst* all the coalition partners. There are numerous challenges in achieving this. These include source bias, heterogeneous data, soft data, different policies for data sharing and access, and variable mutual trust impose *information flow constraints* and affect data quality, on which the ML models and SU are based [8]. Placing these challenges within a coalition setting makes these issues even harder, when different partners might not even agree on the ontological description of the battlefield.

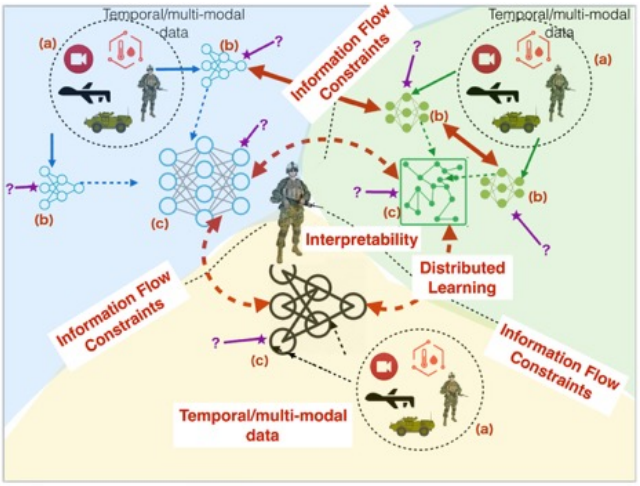

Fig. 1. Conceptual illustration of the Internet of Battlefield Things. The environment is populated by a large number of disparate agents. These include sensing and inference systems which continually distribute information between one another and human operators. Figure adapted from [8].

Military decision-makers must be able to provide reasoned justifications for their decisions; therefore, ML systems must provide a suitable level of explanation for their outputs to facilitate reasoning. In other words, the ML systems must be in some sense interpretable. In a military coalition, model interpretability between coalition partners is especially important for engendering trust; indeed, a specific level of interpretability may be required by the coalition's information-sharing policies [2].

The distributed setting of the IoBT also provides adversaries with opportunities to interrupt the operation of the network [3]. For example, in Fig. 1 the adversaries (purple stars) could attack a network by injecting fake sensor data, corrupt information flowing between agents, or even perform offline attacks by corrupting training data and classifiers. What makes this risk particularly important is that malicious actors can craft inputs that are explicitly designed to cause a ML model to make errors – so-called adversarial examples [5]–[7]. Adversarial examples are often imperceptible to human operators, but can cause an ML model to reach an incorrect decision, often with an arbitrarily high degree of confidence. These mistakes can affect all three levels in Endsley's SA model [9]. At Level 1 (perception), ML models might be used to recognize patterns or detect anomalies; mistakes at this level will impact Level 2 SA, comprehension – also referred to as Situational Understanding (SU) – as inference will be made using misleading inputs. ML systems may also be employed at this level, and could make mistakes even if the inputs from Level 1 contain none. The same reasoning applies to Level 3 SA (projection). Adversarial examples could thus prove hugely detrimental to SA.

In the rest of this paper, we explore the concept of interpretability and the use of adversarial examples, bearing in mind the IoBT conceptual model and how different methods and techniques will help towards our goal of AI alignment in coalition operations.

III. Interpretable Machine Learning

A. What is interpretability?

The past few years have seen a boom in ML interpretability research [10]. Despite this increased research activity, interpretability still does not have an agreed-upon (formal or informal) definition. Researchers tend to use their own intuitions to decide what interpretability means, or what an explanation should look like [11]. Papers that ostensibly address interpretability therefore tackle a diverse set of different problems.

Lipton proposed a taxonomy for interpretability to help address this issue [12]. This provides a vocabulary to assist in the comparison and evaluation of interpretable ML research. He proposed two high-level categories: techniques to improve model *transparency* (which "connotes some sense of understanding the mechanism by which the model works" [12]), and methods for providing *post-hoc explanations* for model decisions. Transparency is further divided into *simulatability* (whether a human can feasibly reproduce the model output given its input and knowledge of the model internals), *decomposability* (whether the model components and parameters are intuitively explicable), and *algorithmic transparency* (whether we understand why and how the learning algorithm works). Post-hoc interpretability is divided into *text explanation* (the model provides a textual description of why it made a particular decision), *visualization* (displaying what the model has learned visually), *localization* (explaining what a decision depends on in the vicinity of a particular input), and *explanation by example* (showing examples in the training data the model considers closest to the current input). Lipton's taxonomy is both intuitive and useful, and we have adopted it to help structure our prior work in this area [10], [13].

Doshi-Velez and Kim provided further insights into the notion of interpretability, arguing that "the need for interpretability stems from an incompleteness in the problem formalization, creating a fundamental barrier to optimization and evaluation"

[14]. They provide a complementary taxonomy to that of [12], focusing on how interpretability can be evaluated. We discuss the evaluation of model interpretability in the next section.

B. Metrics for interpretability

The performance of an ML model on some task is defined using a set of standard metrics. For example, classifiers can be judged on their ability to generalize by measuring their accuracy when classifying held-out test data. We can then estimate that the classifier will perform with that level of accuracy when deployed (assuming no distributional shift). No metrics are currently agreed upon for assessing a model's interpretability. This is unsurprising given the lack of a common definition of interpretability, but developing metrics and standardized tests can stimulate research progress by providing quantitative comparison points for different approaches, in a similar way to how large standardized datasets such as MNIST [15], CIFAR10/CIFAR100 [16], and ImageNet [17] spurred progress in computer vision research.

Doshi-Velez and Kim's "taxonomy of interpretability evaluation" [14] may help towards the definition of interpretability metrics. They take a human-centric approach to evaluation, defining tasks that measure human performance or judgments, either directly or indirectly:

- *Application-grounded evaluation*: evaluate model interpretability by assessing whether, and by how much, it improves human performance on the application task. This is the most difficult kind of assessment to conduct.
- *Human-grounded evaluation*: evaluate model interpretability using a simplified task, such as a binary forced-choice experiment (where users are asked to choose the better of two model explanations). This kind of assessment is easier to set-up and may be performed by non-experts, widening the pool of potential participants. However, it provides a context-free measure of interpretability, which may not transfer to a particular application.
- *Functionally-grounded evaluation*: evaluate model interpretability against a formal definition that acts as a proxy for human-based assessment. This kind of assessment is the easiest and cheapest to conduct, but relies on the availability of a useful, appropriate definition to test against.

The approaches suggested in [14] for performing relevant functionally-grounded evaluation rely on using formal interpretability definitions inferred from the results of previous human-subject experiments, so some element of human judgment is still built in to the evaluation. This is consistent with the authors' definition of interpretability: "the ability to explain or to present in understandable terms to a human." Dhurandhar et al. generalize this view by proposing a definition of interpretability that does not depend on human judgment, and so lends itself to formal evaluation via metrics [18]. They define interpretability relative to a target model (TM). This model could be a human, but crucially does not have to be. They call this *δ-interpretability*: a model is δ-interpretable if it "can somehow convey information to the TM that will lead to improving its performance ... for the task at hand" [18]. More formally, if, after receiving information I from procedure P_I, the expected error of the TM is less than or equal to δ times the TM's expected error prior to receiving I, then P_I is δ-interpretable. δ thus becomes a metric of interpretability. A δ of 1 implies that the procedure P_I adds nothing to the interpretability of the model it is attempting to explain to the TM, and interpretability improves as $\delta \to 0$. Using the taxonomy in [14], this is a kind of application-grounded evaluation if the TM is a human, and if we can formally define an error function on the human's task.

Defining interpretability in relation to non-humans could be extremely useful in any multi-agent setting, but particularly in the context of CSU in the IoBT. In this case, communication between agents from different coalition partners occurs under information flow constraints. These include structural aspects, such as data storage capacity and network coverage, bandwidth, and stability, as well as policy constraints that govern what data is allowed to be exchanged. An agent may pass information to another agent via a δ-interpretable process that conforms to coalition policies, and/or reduces data transfer requirements compared with directly transmitting training data. The effectiveness of knowledge-sharing between differently structured models – even models that perform different tasks – can be quantified and compared using δ-interpretability, stimulating progress in this challenging research area. δ-interpretability has the potential to provide a new approach for sensor and resource management as well.

C. Interpretability and uncertainty

Intuitively, a model's uncertainty in its output seems an important quantity, both for the model users (to make decisions based on its output), and for model trainers (to understand how well the model has characterized the problem space). In the former case, an analyst might make quite different actions depending on whether the model was certain or uncertain in its output (we could quantify whether sharing uncertainty information leads to better actions using the framework of δ-interpretability described above). In the latter case, high uncertainty in the output for a particular input region indicates to the model trainer that the system should be trained with more data points close to this input region. Kapoor et al. use this approach to improve classifier performance when only a small amount of training data is available [19].

In classification tasks, most models provide a numerical output between 0 and 1 for each known class, and a classification is made by selecting the class with the highest output. These values can be interpreted as the model's level of confidence that the input belongs to each class. However, for this interpretation to hold true, the model must be appropriately *calibrated*: the confidence scores should reflect the true likelihoods that the model is correct. For example, if a model predicted class A for 100 examples with confidence 0.8, we expect 80 of those classifications to be correct [20]. While some classes of model generally produce well-calibrated output probabilities

[21], recent work has shown that modern NN architectures with many/wide layers are generally poorly calibrated, tending towards over-confidence [20]. This has serious implications for the interpretation of confidence scores from such models: they do not mean what we intuitively think they mean. Simple methods are shown to improve DNN calibration [20], and should be implemented and the model calibration tested before presenting confidence scores to users.

Confidence scores provide the probability of class membership, given an input and the learned model. This still leaves us to account for uncertainty in the input data and uncertainty in the model parameters, both of which might be useful for improving model interpretability. If we know a model has only had access to highly uncertain data, or has only been trained on a small amount of data so is unsure of its parameter estimates, our interpretation of the model's output will likely be different than if we knew the model had been trained on large quantities of high quality data. Many popular ML models do not provide this information, including modern DNNs. Probabilistic methods such as Gaussian processes (GPs) are an obvious exception [22]. A GP learns a distribution over functions conditioned on the training data, and estimates the distribution mean and variance at a given test point. The variance naturally decreases around the training data, leaving high variance (i.e. high uncertainty) in regions of input space far away from the training data.

We illustrate this graphically in Fig. 2. We generated non-linearly separable data for a 2-class classification problem and trained three models using 200 training points: a shallow NN with 1 hidden layer made up of 2 $tanh$ units, a DNN with 3 hidden layers made up of 32, 16 and 8 rectified linear units respectively, and a Gaussian process classifier with radial basis function kernel. Each model is approximately as accurate as the other on held-out test data (accuracy 0.9), but they exhibit different confidence scores over the input space (the black-white gradient). The shallow NN has a region of low confidence (shaded grey) that appears unrelated to the actual data distribution – rather it is an artifact of the network architecture, and disappears as we increase the number of hidden units. The DNN outputs high confidence scores across input space, except very close to the decision boundary, while the GP's confidence varies from high to low over the input space depending on the distance from the training data. Because the GP also estimates variance, we can specify a confidence interval on its output and define decision boundaries on each side of this interval (see dashed lines in Fig. 2). Inputs that are assigned different labels by these two decision boundaries can then be rejected as belonging to an unknown class, or highlighted for further inspection by a human. We will return to this idea in section VI.

Even if a model provides confidence scores and uncertainty estimates, it is not immediately clear how best to present those values to humans as we are generally poor at reasoning with probabilities and randomness. Examples of the negative impacts of this trait are provided in [23], which describes how even highly trained medical professionals are liable to reason irrationally about probabilistic intervention outcomes. Suitable presentation of uncertainty information that genuinely improves model interpretability is therefore an important line of research.

IV. Adversarial Examples

A. What are adversarial examples?

Adversarial ML is the study of attacks on, and defenses for, ML systems. Such attacks are possible whenever an opponent has access to a model's input data. The field originally arose in the area of spam email filtering [24]–[26]: as spam classifiers became more successful at identifying junk emails, spammers started to change their email contents to include words or images that made them more likely to be classified as non-spam. More recently, concern has arisen regarding the potential to fool even highly accurate non-linear classifiers like DNNs. This concern follows from results on image classifiers showing that tiny alterations to the input images – often imperceptible to humans – can lead to incorrect classifications [5] [27] (such images are now often called "adversarial examples"). These results have serious implications for safety-critical systems that rely on ML.

Several different kinds of attack are possible on ML models, which Huang et al. [7] classify along three axes:

- *Influence*: the attack could manipulate the training data (a *causative* attack), or it could probe a trained model (an *exploratory* attack)
- *Security violation*: the attack could cause the system to wrongly classify an input (an *integrity* violation), could render the system useless or unavailable (an *availability* violation), or could obtain private information from the model (a *privacy* violation)
- *Specificity*: the attack could could be *targeted* towards a specific subset of inputs, or *indiscriminate* – designed to degrade performance on a wide range of inputs

This taxonomy provides a useful vocabulary for describing and grouping different adversarial ML studies. For instance, a denial-of-service attack on the infrastructure running a classifier is an exploratory attack causing an indiscriminate availability violation, while adversarial examples for image classifiers (as described above) can be causative (see e.g. [28] but are more usually exploratory attacks that cause integrity violations and can be targeted or indiscriminate. The recent explosion in research on adversarial examples in particular has led Yuan et al. to augment Huang et al.'s terminology, developing an additional taxonomy just for this subset of attacks [6].

Classifiers are not the only models susceptible to adversarial examples. Kos et al. demonstrate attacks on generative models that use perturbed inputs to manipulate the learned latent space, causing the model to produce poor quality input reconstructions [29]. Lin et al. demonstrate two attacks against reinforcement learning: strategically timed attacks that reduce an agent's reward using a low number of perturbations, and enchanting attacks that lure an agent towards a specified target

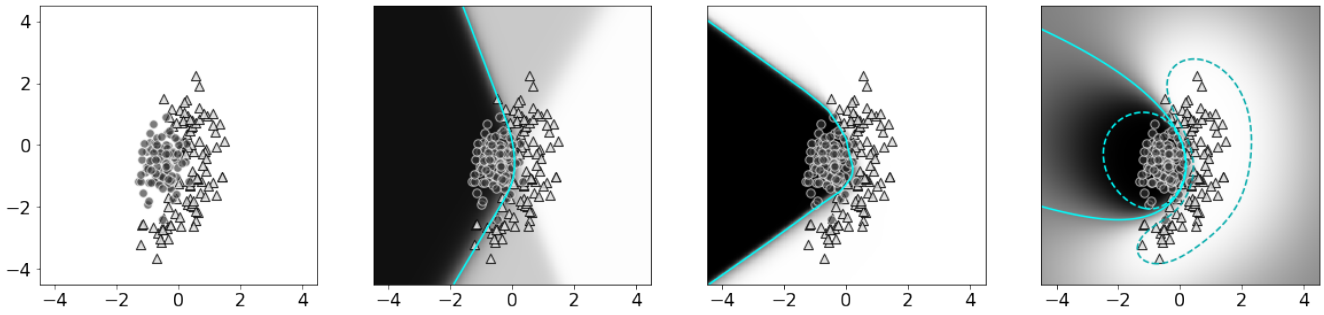

Fig. 2. Confidence and uncertainty information provided by different classifiers. *Left*: training data for a binary classification problem. Dark circles are in class 1, light triangles in class 2. *Middle-left*: shallow, narrow NN output after training, with decision boundary indicated as solid cyan line. Black/white gradient indicates output confidence score over input space (solid black: 100% likely to be class 1, solid white: 100% likely to be class 2). *Middle-right*: deep, wide NN output after training. *Right*: Gaussian process output after training. Dashed lines show decision boundaries estimated using the 95% confidence range for the output mean, calculated using the output variance of the Gaussian process over the input space.

state [30]. While attacks against other kinds of models are important to consider, we will focus the remainder of our discussion on adversarial examples for classifiers, as these have been the most widely studied.

B. Adversarial examples in the real world

Despite a plethora of proposed attacks using adversarial examples, it remains unclear how practical many of them would be to implement against real-world ML systems. Early work on email spam detection was motivated by actual spam attacks; however, adversarial examples against image classifiers were discovered while studying the sensitivity of DNNs to input perturbations, not their vulnerability to attacks [5], [26], so the practicality of the attacks is often not a consideration. Additionally, while spammers work with purely digital information that can be arbitrarily manipulated before being sent, many classifiers identify items that exist in the physical world via a digital representation of that item, such as a camera image or audio recording. Often the only way to implement an attack against a model will be to alter the item in the physical world, as the adversary will not have access to the model's digital input directly. Understanding the susceptibility of different ML systems to physical adversarial examples is thus particularly pertinent for the IoBT, where many ML models receive data from a wide variety of physical sensors.

Recent work has shown the success of adversarial examples in real-world computer vision systems. Evtimov et al. developed a method of fooling traffic sign classifiers, as might be implemented in self-driving vehicles, using either life-size printouts of adversarially perturbed signs, or by applying graffiti-like modifications to existing signs [31]. Their attacks fooled traffic sign classifiers even when using frames extracted from videos in drive-by tests at different speeds. A related attack against traffic sign classifiers was developed by Sitawarin et al., though their approach perturbs circular advertisements and logos to be wrongly identified as traffic signs [32].

Brown et al. take another approach, generating a circular 2D image (an "adversarial patch") that can be printed and attached to physical objects to trick classifiers [33]. The generated image is highly salient for a particular class, and is likely to fool a classifier even if it only takes up a small percentage of the total image, and even when disguised to look innocuous to humans. This approach does not attempt to minimize the image perturbation, but only considers the possibility of implementing the attack in the physical world. The patch is also tested for transferability, and shown to work reasonably effectively on classifiers it was not optimized to deceive. Athalye et al. demonstrate another impressive attack using 3D-printed objects with an adversarial texture applied, successfully fooling a classifier with pictures of the objects taken from a wide variety of angles, poses, and different lighting conditions [34].

Despite these worrying possibilities, there is some evidence that real-world attacks might be more difficult to implement against object detection models, as opposed to classifiers. Lu et al. showed that the technique described in [31] did not trick standard object detectors despite fooling classifiers, demonstrating that testing attacks on classifiers as proxies for object detection models is not valid in general [35]. Physically implementing robust attacks against object detectors is theoretically more difficult, as they need to be effective in the face of a broad range of parametric distortions [35]. We also note that, during (limited and preliminary) testing of publicly available image classification API demos (Google Vision [36], IBM Watson Visual Recognition [37], Microsoft Computer Vision [38]), we found it difficult to fool the default demo classifiers using the adversarial patch from [33] unless we covered a significant portion of the image, suggesting that the transferability of the attack may be limited. Further research is needed into the real-world feasibility of attacks, especially in domains other than vision such as audio/speech recognition [39].

C. Adversarial examples in the coalition context

The military coalition setting described above provides new avenues for attacks on ML systems (denoted by the star-headed, purple arrows in Fig. 1). Agents in the IoBT collect

data through a variety of sensors using different modalities (e.g. visible light, infrared, sound, vibration). Agents may share their data or model parameters to improve their collective performance, which opens them up to causative (aka poisoning) attacks if the shared information is tainted by adversarial perturbations. Tainted information may come from a malicious agent masquerading as a friendly one or from a friendly agent that has been compromised in some way (e.g. by malware).

New attacks or defenses may be possible by using multi-modal data to build models. Incorrect classification on perturbed inputs in one modality may be mitigated against by considering multiple modalities simultaneously. However, as is known from prior work on data/decision fusion, it is by no means assured that the use of multimodal data will always result in improved classification, and indeed attacks may be possible that rely on a model's use of multi-modal data specifically.

It is also conceivable that new attacks might exist in this setting – for example, using a causative attack on one agent's model such that their performance is not affected, but when they exchange knowledge with a second agent, that second agent's performance is degraded. One compromised agent could thus be used to poison many further agents, without itself noticing that it was compromised.

In addition to new attacks, new defenses against adversarial examples may arise that take advantage of the coalition's distributed architecture. For example, ensemble effects could be exploited to add robustness against adversaries. Distributed adversarial learning is, to our knowledge, only just beginning to be explored [40], [41], [42], so these and other related questions remain open.

V. Using Adversarial Examples to Improve Interpretability

The existence of adversarial examples is understandably concerning, particularly in cases where ML is heavily relied upon. However, when these model failures do occur, it may be possible to use what we learn from them to improve model interpretability. Examining a system's failures can often be more enlightening than studying its successes; looking at examples of failure could lead to an improved understanding of how a model works and why it fails, or at the very least give a better idea of its weaknesses and improve the ability to predict when it will fail. Indeed, the original study on adversarial examples for DNNs generated such examples to improve understanding of how DNNs responded to small input perturbations [5].

Exploring examples of when humans make mistakes as a way of better understanding how the brain works is a common approach in cognitive neuroscience, and various methods have been developed. Ritter et al. evaluated whether some of these methods can be applied to ML research [43]. In particular, they chose an analysis which is used to explain how children learn word labels for objects, and they applied this analysis to DNNs. They found that DNNs demonstrated a bias to categorizing objects by shape rather than by colour. This same bias has also been observed in humans. This work "leads the way to the study of artificial cognitive psycology" [43], and provides a case for using the study of "adversarial" examples in human behaviour (for example, visual illusions) to broaden how we study adversarial examples and interpretability in DNNs. However, this approach is limited to explaining ML models designed to replicate human capabilities.

Using adversarial examples to improve understanding of DNNs was studied more directly in [44], which uses generated adversarial images to explore internal representations of DNNs. In one experiment, for example, they show that high-level neurons that ostensibly represent high level concepts present in the training data also respond strongly to an array of different image contents in adversarially perturbed images. Additionally, they find that the high-level feature representations of adversarial images are detectably different from those of unperturbed images. Their findings from this method contradict previous conclusions about these internal representations, which demonstrates that using adversarial examples in the context of interpretability can lead to new understanding. They use this knowledge to develop an adversarial training method that improves the consistency of representations between real and adversarial images. They argue that their approach improves the interpretability of the trained DNN, as the network's representations are more closely aligned to high-level concepts due to the adversarial training.

Ross and Doshi-Velez developed a method to defend against adversarial examples that also has the effect of improving the DNN's interpretability [45]. They train DNNs with input gradient regularization, which reduces the amount that small changes in input can alter the network's output. Their method is effective against a wide variety of different attacks, but also has a side-effect: attacks designed specifically to fool DNNs trained with input gradient regularization are more likely to be rated as reasonable by humans than other attacks. In other words, such networks are still vulnerable, but the adversarial examples must appear more similar to the adversarial target class for them to be fooled.

VI. Improving Interpretability to Defend Against Adversarial Examples

Adversarial examples are typically generated by adding bounded noise over the entire input such that the perturbation is imperceptible to the human observer. The effect of the added noise is magnified as the input is projected onto the latent spaces, corresponding to the hidden layers in the model, leading to incorrect classification at the output layer. This motivates our hypothesis that interpretability, i.e., semantic visibility into the representations of the hidden layer, can inform the presence of adversarial perturbations in the input data.

To validate the hypothesis, we performed some simple experiments combining adversarial examples and state-of-the-art interpretability techniques. Saliency mapping methods are used to explain image classifier outputs in terms of their input pixels [46]–[48], but have not been designed with adversarial

examples in mind. We tested the robustness and sensitivity of these techniques (in particular deep Taylor decomposition [49]) to adversarial examples. We trained a convolutional NN for classification of the MNIST handwritten digit dataset [15]. We then perturbed the images by adding bounded noise generated by Carlini and Wagner's method [50] with different perturbation measurements ($l_0-norm, l_2-norm, l_\infty-norm$) to create adversarial examples. These examples were then used to generate heat maps using the deep Taylor decomposition technique [49]. Some examples from these initial experiments are shown in Fig. 3.

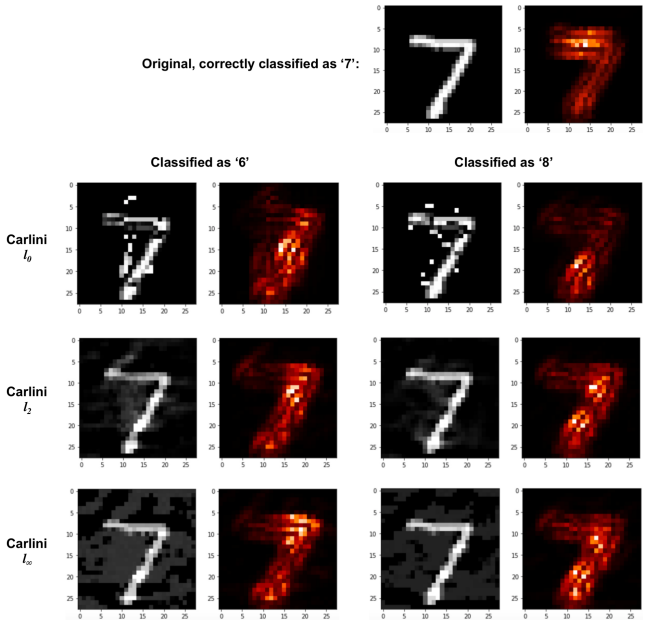

Fig. 3. Three types of adversarial perturbation on an MNIST digit. The original digit is a 7 (top-right). Top row: Carlini l_0, middle row: Carlini l_2, bottom row: Carlini l_∞, with corresponding deep Taylor decomposition heatmaps as explanations. Left-most and center-left columns show digits perturbed to be classified as a 6, and corresponding deep Taylor decomposition explanations, respectively. Center-right and right-most columns show digits perturbed to be classified as an 8, and corresponding deep Taylor decomposition explanations, respectively.

We observed that, besides the heat maps being quite diffused, there are no clear, specific anomalies that can be used to detect an attack. In other words, while the saliency map does change for each kind of attack, it does not appear to provide reliable visible markers differentiating between normal and adversarial examples. This could be because current saliency map generation techniques are not sensitive enough to detect the presence of the diffused noise in the adversarial examples, especially when focused solely at the input layer. In future, we would like to design interpretability techniques that could intercept the activation of neurons at the hidden layers of the network to detect representational anomalies indicative of adversarial examples. Visualizing the features that are being used by the NN for the decision, could identify irregularities and hence identify an attack. Furthermore, current interpretability techniques are not resilient to adversarial examples and need to be hardened to handle such attacks [51].

Finally, uncertainty information may be used to help defend against attacks. If an adversarially perturbed image is far away from the training data in feature space, as seems likely given the findings in [44], then the classifier's output uncertainty will be relatively high. Referring back to figure 2, an attack would likely push a data point across the decision boundary, but probably not past the 95% confidence decision boundary of the other class. Data points in this region would be classified differently by the decision boundaries on either side of the 95% confidence interval over the GP's output mean, and this discrepancy could be used as a rejection criterion, or to flag the data to a human for further inspection. These data points may not be adversarial – they could be outliers, or be from a class that the classifier was not trained on, so this approach should improve the general robustness of the classifier. This style of approach was explored for GPs in [52], Bayesian DNNs in [53] and hybrid DNN-GPs (a DNN with a GP instead of the standard softmax as the output layer) in [54]. In all three articles, the authors showed that the model output uncertainty for adversarial examples was higher than for unperturbed inputs. This indicates that models able to represent their own uncertainty are promising candidates for defending against adversarial examples.

VII. Conclusion

In this paper, we have described the problems of ML model interpretability and susceptibility to adversarial examples, why these problems are particularly pertinent for future military coalition operations, and why exploring the links between the two areas might prove fruitful for solving the problems posed in each. Some pioneering studies have begun to investigate these links, but we anticipate many further insights remain to be gleaned from the joint exploration of these problems.

Acknowledgment

This research was sponsored by the U.S. Army Research Laboratory and the UK Ministry of Defence under Agreement Number W911NF–16–3–0001. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the U.S. Army Research Laboratory, the U.S. Government, the UK Ministry of Defence or the UK Government. The U.S. and UK Governments are authorized to reproduce and distribute reprints for Government purposes notwithstanding any copy-right notation hereon.

References

[1] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016, http://www.deeplearningbook.org.

[2] S. Chakraborty, A. Preece, M. Alzantot, T. Xing, D. Braines, and M. Srivastava, "Deep Learning for Situational Understanding," in *2017 20th International Conference on Information Fusion (Fusion)*, pp. 1–8.

[3] A. Kott, A. Swami, and B. J. West, "The internet of battle things," *Computer*, vol. 49, no. 12, pp. 70–75, 2016.

[4] N. Suri, M. Tortonesi, J. Michaelis, P. Budulas, G. Benincasa, S. Russell, C. Stefanelli, and R. Winkler, "Analyzing the applicability of internet of things to the battlefield environment," in *Military Communications and Information Systems (ICMCIS), 2016 International Conference on*. IEEE, 2016, pp. 1–8.

[5] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," *arXiv:1312.6199*, 2013.

[6] X. Yuan, P. He, Q. Zhu, R. R. Bhat, and X. Li, "Adversarial examples: Attacks and defenses for deep learning," *arXiv:1712.07107*, 2017.

[7] L. Huang, A. D. Joseph, B. Nelson, B. I. Rubinstein, and J. D. Tygar, "Adversarial machine learning," in *Proceedings of the 4th ACM Workshop on Security and Artificial Intelligence*, ser. AISec '11. New York, NY, USA: ACM, 2011, pp. 43–58. [Online]. Available: http://doi.acm.org/10.1145/2046684.2046692

[8] S. Chakraborty, A. Preece, M. Alzantot, T. Xing, D. Braines, and M. Srivastava, "Deep learning for situational understanding," in *Information Fusion (FUSION), 2017 20th International Conference on*. IEEE, 2017.

[9] M. R. Endsley, "Toward a theory of situation awareness in dynamic systems," *Human factors*, vol. 37, no. 1, pp. 32–64, 1995.

[10] S. Chakraborty, R. Tomsett, R. Raghavendra, D. Harborne, M. Alzantot, F. Cerutti, M. Srivastava, A. D. Preece, S. Julier, R. M. Rao, T. D. Kelley, D. Braines, M. Sensoy, C. J. Willis, and P. Gurram, "Interpretability of deep learning models: a survey of results," *IEEE Smart World Congress*, 2017.

[11] T. Miller, "Explanation in artificial intelligence: Insights from the social sciences," *arXiv:1706.07269*, 2017.

[12] Z. C. Lipton, "The mythos of model interpretability," *CoRR*, vol. abs/1606.03490, 2016.

[13] D. Harborne, C. Willis, R. Tomsett, and A. Preece, "Integrating learning and reasoning services for explainable information fusion," in *International Conference on Pattern Recognition and Artificial Intelligence*, 2018 (to appear).

[14] F. Doshi-Velez and B. Kim, "Towards a rigorous science of interpretable machine learning," *arXiv:1702.08608*, 2017.

[15] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[16] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," 2009.

[17] J. Deng, W. Dong, R. Socher, L. jia Li, K. Li, and L. Fei-fei, "Imagenet: A large-scale hierarchical image database," in *In CVPR*, 2009.

[18] A. Dhurandhar, V. Iyengar, R. Luss, and K. Shanmugam, "A formal framework to characterize interpretability of procedures," *arXiv:1707.03886*, 2017.

[19] A. Kapoor, K. Grauman, R. Urtasun, and T. Darrell, "Active learning with gaussian processes for object categorization," in *2007 IEEE 11th International Conference on Computer Vision*, Oct 2007, pp. 1–8.

[20] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger, "On calibration of modern neural networks," in *Proceedings of the 34th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, D. Precup and Y. W. Teh, Eds., vol. 70, 06–11 Aug 2017, pp. 1321–1330.

[21] A. Niculescu-Mizil and R. Caruana, "Predicting good probabilities with supervised learning," in *Proceedings of the 22Nd International Conference on Machine Learning*, ser. ICML '05, 2005, pp. 625–632.

[22] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*. The MIT Press, 2006.

[23] P. T. Humphrey and J. Masel, "Outcome orientation: A misconception of probability that harms medical research and practice," *Perspectives in Biology and Medicine*, vol. 59, no. 2, pp. 147–155, 2016.

[24] N. Dalvi, P. Domingos, Mausam, S. Sanghai, and D. Verma, "Adversarial classification," in *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '04, 2004, pp. 99–108.

[25] D. Lowd and C. Meek, "Adversarial learning," in *Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery in Data Mining*, ser. KDD '05, 2005, pp. 641–647.

[26] B. Biggio and F. Roli, "Wild Patterns: Ten years after the rise of adversarial machine learning," *arXiv:1712.03141*, 2017.

[27] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," *arXiv:1412.6572*, 2014.

[28] P. W. Koh and P. Liang, "Understanding black-box predictions via influence functions," *arXiv preprint arXiv:1703.04730*, 2017.

[29] J. Kos, I. Fischer, and D. Song, "Adversarial examples for generative models," *arXiv:1702.06832*, 2017.

[30] Y.-C. Lin, Z.-W. Hong, Y.-H. Liao, M.-L. Shih, M.-Y. Liu, and M. Sun, "Tactics of adversarial attack on deep reinforcement learning agents," *arXiv:1703.06748*, 2017.

[31] I. Evtimov, K. Eykholt, E. Fernandes, T. Kohno, B. Li, A. Prakash, A. Rahmati, and D. Song, "Robust physical-world attacks on deep learning models," *arXiv:1707.08945*, 2017.

[32] C. Sitawarin, A. N. Bhagoji, A. Mosenia, P. Mittal, and M. Chiang, "Rogue signs: Deceiving traffic sign recognition with malicious ads and logos," *arXiv:1801.02780*, 2018.

[33] T. B. Brown, D. Man, A. Roy, M. Abadi, and J. Gilmer, "Adversarial patch," *arXiv:1712.09665*, 2017.

[34] A. Athalye, L. Engstrom, A. Ilyas, and K. Kwok, "Synthesizing robust adversarial examples," *arXiv:1707.07397*, 2017.

[35] J. Lu, H. Sibai, E. Fabry, and D. Forsyth, "Standard detectors aren't (currently) fooled by physical adversarial stop signs," *arXiv:1710.03337*, 2017.

[36] "Google Vision API Demo." [Online]. Available: https://cloud.google.com/vision/

[37] "IBM Watson Visual Recognition API Demo." [Online]. Available: https://www.ibm.com/watson/services/visual-recognition/demo/

[38] "Microsoft Computer Vision API Demo." [Online]. Available: https://azure.microsoft.com/en-gb/services/cognitive-services/computer-vision/

[39] M. Alzantot, B. Balaji, and M. Srivastava, "Did you hear that? adversarial examples against automatic speech recognition," *arXiv:1801.00554*, 2018.

[40] R. Zhang and Q. Zhu, "Secure and resilient distributed machine learning under adversarial environments," in *2015 18th International Conference on Information Fusion (Fusion)*, July 2015, pp. 644–651.

[41] ——, "A game-theoretic analysis of label flipping attacks on distributed support vector machines," in *2017 51st Annual Conference on Information Sciences and Systems (CISS)*, March 2017, pp. 1–6.

[42] Y. Chen, L. Su, and J. Xu, "Distributed statistical machine learning in adversarial settings: Byzantine gradient descent," *arXiv:1705.05491*, 2017.

[43] S. Ritter, D. G. T. Barrett, A. Santoro, and M. M. Botvinick, "Cognitive psychology for deep neural networks: A shape bias case study," in *Proceedings of the 34th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, D. Precup and Y. W. Teh, Eds., vol. 70. PMLR, 2017, pp. 2940–2949.

[44] Y. Dong, H. Su, J. Zhu, and F. Bao, "Towards interpretable deep neural networks by leveraging adversarial examples," *arXiv:1708.05493*, 2017.

[45] A. S. Ross and F. Doshi-Velez, "Improving the adversarial robustness and interpretability of deep neural networks by regularizing their input gradients," *arxiv:1711.09404*, 2017.

[46] K. Simonyan, A. Vedaldi, and A. Zisserman, "Deep inside convolutional networks: Visualising image classification models and saliency maps," *arXiv:1312.6034*, Dec. 2013.

[47] A. Binder, G. Montavon, S. Bach, K. Müller, and W. Samek, "Layer-wise relevance propagation for neural networks with local renormalization layers," *CoRR*, vol. abs/1604.00825, 2016.

[48] M. T. Ribeiro, S. Singh, and C. Guestrin, "Why should i trust you?: Explaining the predictions of any classifier," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2016, pp. 1135–1144.

[49] G. Montavon, S. Lapuschkin, A. Binder, W. Samek, and K.-R. Müller, "Explaining nonlinear classification decisions with deep taylor decomposition," *Pattern Recognition*, vol. 65, pp. 211–222, 2017.

[50] N. Carlini and D. Wagner, "Towards evaluating the robustness of neural networks," *arXiv:1608.04644*, 2016.

[51] P.-J. Kindermans, S. Hooker, J. Adebayo, M. Alber, K. T. Schutt, S. Dahne, D. Erhan, and B. Kim, "The (un)reliability of saliency methods," *CoRR*, vol. abs/1711.00867, 2017.

[52] K. Grosse, D. Pfaff, M. T. Smith, and M. Backes, "How wrong am I? Studying adversarial examples and their impact on uncertainty in gaussian process machine learning models," *arXiv:1711.06598*, 2017.

[53] M.-I. N. A. Rawat, M. Wistuba, "Harnessing model uncertainty for detecting adversarial examples," *NIPS Workshop on Bayesian Deep Learning*, 2017.

[54] J. Bradshaw, A. G. de G. Matthews, and Z. Ghahramani, "Adversarial examples, uncertainty, and transfer testing robustness in gaussian process hybrid deep networks," *arXiv:1707.02476*, 2017.