Detection of adversarial attacks on machine learning systems

Matthew Judah, Jen Sierchio, Michael Planer

Matthew Judah, Jen Sierchio, Michael Planer, "Detection of adversarial attacks on machine learning systems," Proc. SPIE 12538, Artificial Intelligence and Machine Learning for Multi-Domain Operations Applications V, 125380D (12 June 2023); doi: 10.1117/12.2664015

Event: SPIE Defense + Commercial Sensing, 2023, Orlando, Florida, United States

Detection of adversarial attacks on machine learning systems

Matthew Judah
Jen Sierchio
Michael Planer

BAE Systems, Inc., matthew.judah@baesystems.com, jennifer.sierchio@baesystems.com, michael.planer@baesystems.com

April 17, 2023

ABSTRACT

One of the major issues limiting the adoption of machine learning (ML) in applications where accuracy is critical is failure of an otherwise accurate system. Recent work has developed tools to independently measure the competency of machine learning models and the conditions that drive these competency differences. The purpose of this paper is to explore ways to detect and mitigate adversarial attacks by using the same tools to assess competency. We will introduce BAE System's MindfuL™ software which assesses ML competency under varying environmental conditions. We then consider a few different types of adversarial attacks and describe detection experiments. We examine the predicted performance and strategy and use that information to detect adversarial attacks. We will present the results of these experiments and discuss the implications of this work and potential future directions for this research.

1. INTRODUCTION

Machine learning techniques are applicable to a wide range of scenarios.[1] In many cases, the failure of these machine learning or deep neural networks can cause safety or security concerns in applications where accuracy is critical.[2] For example, convolutional neural networks (CNN) in self-driving cars may miscalculate the distance to an upcoming turn during a rain shower. Additionally, deep learning models are potentially vulnerable to failures due to perturbations of input data that are unseen by human observers but drastically reduce the confidence of the machine learning system in what are called adversarial examples.[3] Thus it is essential to train accurate and robust models prior to their deployment in real-world applications.

Overcoming and attempting to correctly classify adversarial examples has become an active area of research.[4] Unfortunately, most defenses typically fail when encountering novel or stronger types of adversarial attacks.[5–7] Therefore, detection-based defenses have become increasingly popular as a solution with techniques that include introducing an extra classifier class to contain adversarial examples,[8] using additional classifiers to determine if an input is adversarial,[9] or performing regularization on the output feature vectors of the neural network.[10] While these solutions mitigate certain aspects of the problem, malicious agents can craft targeted adversarial attacks to defeat any defense.[7] The difficulty addressing the vulnerability of deep learning models to adversarial attacks has led to additional concerns of how trustworthy deep learning technologies can be in real world conditions where changing scenarios or malicious attackers can potentially exploit these vulnerabilities.

Recent work has sought to improve the understanding of how ML systems perform by monitoring the network's competency. ML competency provides a comprehensive summary of the operation of the ML system, including when it is operating in an unfamiliar context and then passing that information on to a human operator. While competency encompasses the typical performance of the ML model, like accuracy, it primarily focuses on providing operators an insight into the ML system, such as identifying the current strategy (approach the ML agent takes to accomplish a task) or conditions (characteristics of input data) that may impact the system's performance or strategy. These external conditions could include: environmental factors that enhance or impede performance (rain showers impacting a self-driving car) or other factors that contribute to data drift, and intentional manipulation such as adversarial attacks. This paper explores the capability of competency-measuring ML systems to detect and mitigate adversarial attacks on the ML system, based upon the DARPA Competency-Awareness ML (CAML) program. This paper explores the use of BAE System's MindfuL™ software suite under

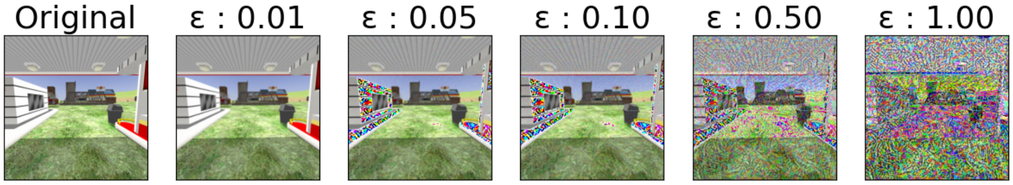

Figure 1: An example image used as an input to the CNN during adversarial example testing of the competency-awareness agent. This single image is shown with an increasing perturbation parameter (ϵ) for the FGSM adversarial attack as labeled above each image.

an adversarial attack and examines the predicted performance of its ability to detect the presence of adversarial examples on its underlying deep neural network classification system. The rest of the paper is structured as follows. In Section 2 we discuss simulated autonomous vehicle path datasets used for this study and the technique used to generate the adversarial examples. Section 3 provides details on the structure of the MindfuL™ software suite and the techniques used to identify the adversarial examples. Results follow in Section 4, with a concluding discussion in Section 5.

2. DATA AND EXPERIMENTAL SAMPLES

An 80,000 image dataset of simulated scenarios that an autonomous vehicle could encounter was simulated with the Gazebo simulator.[11] Each scenario includes obstacles, such as buildings, bridges, or other vehicles, and one of ten environmental parameters including 'rain', 'day', and 'bad pixels'. An example image from this dataset is shown as the top left image in Figure 1. Each scenario was annotated and labeled with distances from the observer to each obstacle and a descriptor of the environment. An ML agent based on the AlexNet CNN[12] was trained using half of the dataset evenly split across the 10 environmental parameters, to classify whether or not the autonomous vehicle would need to perform an avoiding action to maneuver to prevent collision with an obstacle. Due to the turning radius of the vehicle and the extent of obstacles, a value of 9 meters from an obstacle was used as the threshold distance for a maneuver to be required.

Many studies have attempted to detect adversarial examples.[8–10] In this paper, we study the ability of a competency-awareness agent to correctly classify or detect adversarial examples. Because our scenario is a binary classification task, we chose the simple and well-known Fast Sign Gradient Method (FSGM) attack, as implemented in the Adversarial Robustness Toolkit,[13] is utilized to produce adversarial examples for testing. FSGM is a one-step attack, which creates adversarial examples via:

$$x' = x + \epsilon \left(\nabla_x L\left(x, y\right)\right), \tag{1}$$

where x' is the adversarial example, x is the original input, ϵ is a perturbation factor, $L(x, y)$ is the loss function for the input-label pair (x, y). Five adversarial datasets consisting of 20,000 images, were produced by varying the perturbation factor, ϵ, to one of four values, either 0.01, 0.05, 0.10, or 1.00. The impact of these perturbation parameters on the input data is illustrated in Figure 1, showing the same simulated scenario with increasing perturbations applied going from left to right through the images. Each adversarial dataset used within the following experiments has an even split among the various environmental parameters.

The performance of the AlexNet classifier on the test datasets with varying ϵ perturbation factors is shown in the left plot of Figure 2. At values of $\epsilon < 0.10$, there is a slight decrease in classification accuracy. At $\epsilon = 0.10$, the classifier's accuracy is substantially degraded by adversarial attack. Counterintuitively, the classifier accuracy at $\epsilon = 0.5$ and 1.0 is about 50%. This is due to the classifier labelling input images as blocked greater than 95% of the time when it was trained on data in which the path is free half of the time. In other words, at these larger ϵ values, the classifier is simply defaulting to classifying the path as blocked. Normally, this would be caused by either a lack of training data or from using too simplistic a network structure. As the point of a

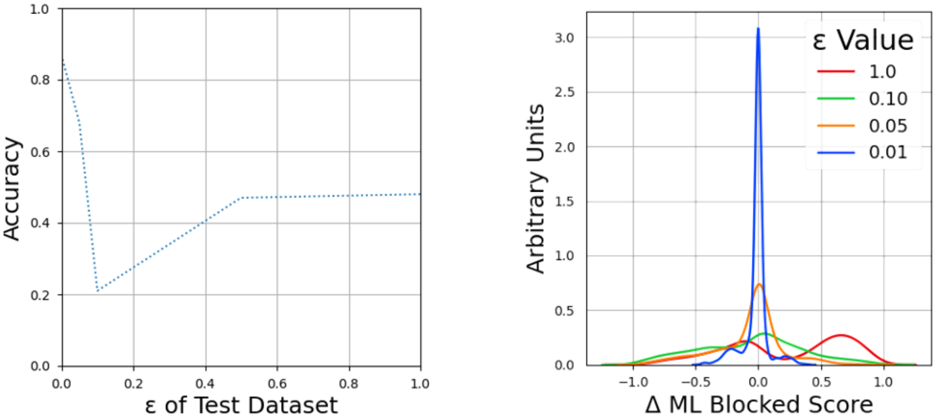

Figure 2: (Left)Accuracy of the AlexNet classifier on a test dataset as a function of applied ϵ perturbation factor of the FSGM attack. (Right) The difference of the AlexNet classifer's blocked score between images with $\epsilon = 0.01, 0.05, 0.10,$ or 1.00with the same image with $\epsilon = 0.0$ as indicted in the legend.

competency-awareness agent is not to improve network accuracy, but to evaluate it and understand what affects it, the network was neither retrained nor changed in structure in any way.

To illustrate the impact of the adversarial examples on the AlexNet classifier, the ML outputs at $\epsilon = 0.0$ are compared to those of the adversarially shifted examples in the right plot of Figure 2. The original classification accuracy, or $\epsilon = 0$ is almost 90% as shown in the figure.

3. METHODS

3.1 Competency-Awareness Agent Overview

There are two primary aspects of competency assessment that can be applied to ML systems: performace in which users have traditionally been concerned with metrics such as accuracy, precision and recall, and F1 score, and "strategy". Strategy is a generalization of the ML system's behavior as it completes its task given a set of input parameters. In the context of ML agent's obstacle detection, strategy is determined using the output activation patterns of the AlexNet CNN for given input images (as in Figure 1) and grouping activation patterns into meaningful high-level strategies.

The MindfuL™ software suite, graphically depicted in Figure 3, was used to assess and evaluate the performance of the AlexNet CNN classification network. Images were input into the MindfuL™ Experience Encoder (MEE), which examines all input data and performs topic modeling via hierarchical Dirichlet processes (HDP)[14] over generic features extracted from the image and metadata (eg. ORB features[15]). In addition to the input images, the HDPs are trained using competency-related inputs like environmental conditions to allow for the development of topics that are more strongly correlated with contextual assessments relevant to human users. The topic distributions are used as a compressed, lightweight input into all downstream applications of the MindfuL™ suite. In addition to the HDPs, the suite also relates historical competency information from training to the current data stream. These components work together to provide real-time assessments of the network's competency.

The HDP topics offer the potential for the identification of adversarial examples by a competency-awareness agent. It is important to note that the HDP topic model was trained only on the unperterbed data and has no knowledge of the adversarial dataset. This was intentional to demonstrate the applicability of using a pretrained comeptency-awareness system to detect adversarial perturbations of a type we may not have training data to identify. Figure 4 depicts the difference between HDP topic outputs for the adversarially shifted examples and the same images with $\epsilon = 0.0$. The $\epsilon = 1.0$ distribution shows a large shift in almost all HDP topic outputs. The HDP topics of the other adversarial shifts provide more subtle variations from the non-adversarial sample but these minor differences and the correlations between the HDP topics provide a potential avenue for the detection of the adversarial images.

3.2 Detecting Adversarial Examples

Two ML approaches are used to distinguish adversarial examples from the standard simulated dataset, the use of a Random Forest classifier[16] and a Forest Deep Neural Network.[17] Each technique relies on the output of the AlexNet CNN and the 21 topic HDPs extracted by the MEE, with the goal of finding a set of features that can be used to classify an image as adversarial.

3.2.1 Random Forest

Random forests[16] (RF) are an ensemble learning method for classification that is able to determine robust feature representations with a straightforward learning method. These models are composed of multiple decision trees which are in turn composed of internal nodes and leaves. The leaves are assigned to one of the potential labels, adversarial or non-adversarial, while the internal nodes form a path defined by the input features. Because of their straightforward learning method, these are used as the baseline performance for testing MindfuL™'s ability to identify adversarial samples. Several RFs were trained, each using an ensemble of 150 decision trees with a maximum of 10 internal nodes and differing in their training datasets. The five different training datasets were $\epsilon = [Normal : 0.00, Adversarial : 0.01]$, $\epsilon = [0.00, 0.05]$, $\epsilon = [0.00, 0.10]$, $\epsilon = [0.00, 1.00]$, and $\epsilon = [Normal : 0.00; Adversarial : 0.01, 0.05, 0.10, and 1.00]$. Each value of ϵ contributed 18,000 images from the adversarially-perturbed (or normal) datasets to the training dataset. Training was performed with class reweighting applied. Multiple trained models allowed for a description of MindfuL™'s ability to identify adversarial examples changes as a function of ϵ.

3.2.2 Forest Deep Neural Network

Because the training dataset is relatively small compared to the needs of a DNN, the techniques described by Kong and Yu[17] were used to develop a forest deep neural network (fDNN). The model consists of two components, a random forest which acts as a feature detector to learn sparse representations from the raw HDP topics and AlexNet classifier inputs and a DNN that predicts outcomes using the forest's feature representations. Two input RFs were trained following the same methods as described in Section 3.2.1, one to separate non-adversarial from $\epsilon = 0.01$ and 0.05, and the other to separate non-adversarial from $\epsilon = 0.1$ and 1.0. The predictions from each tree in the forests are fed into a fully-connected DNN, that consists of three hidden layers with 64, 10, and 2 neurons,

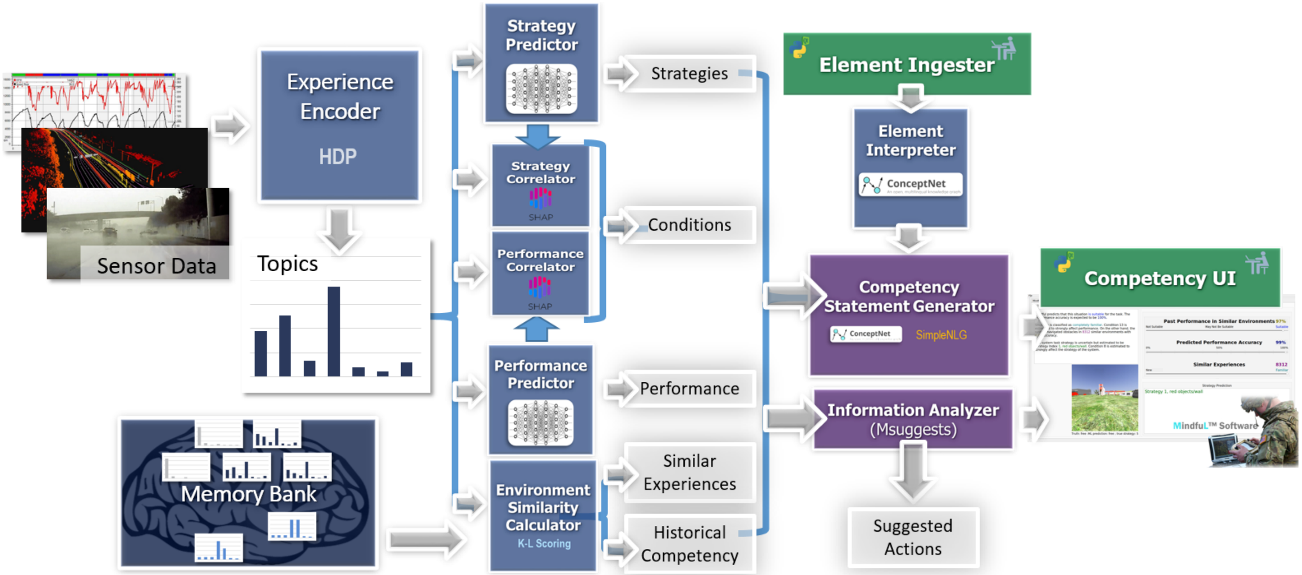

Figure 3: Graphical overview of the MindfuL™ software suite.

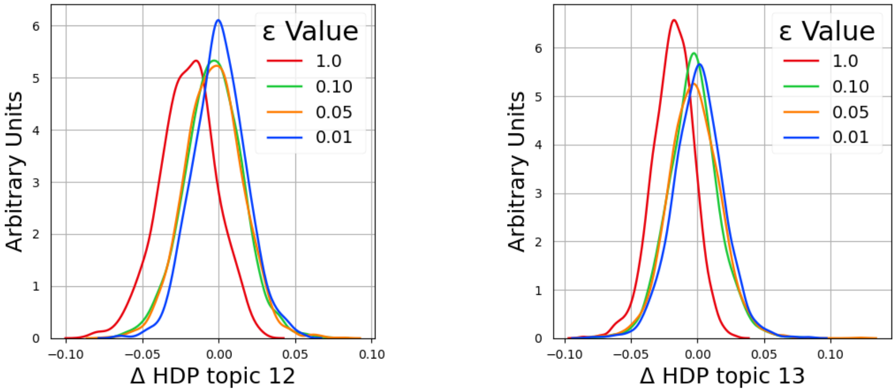

Figure 4: The difference in the MEE's output HDP topics between images with $\epsilon = 0.01, 0.05, 0.10,$ or 1.00 and the same image with $\epsilon = 0.0$ as indicated by the legend. (Left) Shows the difference in HDP Topic 12. (Right) Shows the difference in HDP Topic 13.

a flattening layer, followed by a final hidden layer with 10 neurons. Each hidden layer uses a ReLU activation function[18] with L_2 kernal regularization with $\lambda = 0.01$. The fDNN was trained using the same training dataset as the two input RFs.

4. RESULTS

This section discusses the results of the various classification methods covered in Section 3.2 on test samples of the adversarial datasets.

The performance of the RF and fDNN was evaluated by determining the accuracy in the detection of adversarial examples and a non-reaction to non-adversarial images. Classification accuracy on a test samples consisting of 2,000 images derived from each ϵ class are summarized in Table 1. The ability to correctly identify non-adversarial examples increases as the difference between the ϵ values used in training increases. Identification of adversarial examples at $\epsilon = 0.01$ is difficult, the ML performance is only slightly impacted and both most RFs and the fDNN are unable to detect a substaintial shift from the nominal unperturbed dataset. The ability to identify adversarial examples tends to increase as ϵ of the test dataset increases. This is due to a group of HDP topic features that are common across all values of ϵ and become enhanced at higher values of ϵ. It is interesting to note that although these common features exist the $\epsilon = [0.00, 0.10]$ and $\epsilon = [0.00, 1.00]$ classifiers do not interpolate back to the low non-zero values ϵ due to the overall magnitude of the shift in these HDP topics being much smaller at lower ϵ values.

As shown in Table 1, the fDNN shows minor improvements over the RF trained on all values of ϵ. Figure 5 shows the confusion matrix for the subset of the adversarial datasets where the ΔML between images matched in the adversarial and unperturbed dataset's classification score was changed by less than 0.20. This subset consists of approxiamtely 80% of the test datasets with $\epsilon = 0.01$ and 0.05 where the adversarial examples are identified relatively accurately by the AlexNet classfier desipte the applied perturbations. An improvement of approximately 10% is seen for this particularly difficult to identify subsample of adversarial images using the fDNN. This can be contrasted to subset of data with $\Delta\text{ML} > 0.40$ which completely ruin the predictions of the AlexNet classifier. It is observed that both the RF and fDNN are able to identify this particular subset of adversarial examples with greater than 94% accuracy even at the lower values of ϵ.

A final test of this approach was to determine if the fDNN classifier could perform well (or interpolate) to an untrained value of ϵ. To test this, a new sample of adversarial examples was created using $\epsilon = 0.50$ as the perturbation factor. The fDNN was able to correctly classify this samples with a 94% accuracy. This is comparable to a network purposfully trained to identify this adversarial sample which achieved a 95% classification accuracy.

Table 1: Classification accuracy results of the RF and fDNN classifiers trained using differing combinations of adversarial and non-adversarial exaples as described in Section 3.2. The performance is shown using test datasets of varying values of ϵ in a FSGM attack as indicated in the first column.

	Classification Accuracy (%)					
	RF Classifier Training Dataset					fDNN
Test Dataset ϵ Parameter	$\epsilon = [0.00, 0.01]$	$\epsilon = [0.00, 0.05]$	$\epsilon = [0.00, 0.10]$	$\epsilon = [0.00, 1.00]$	$\epsilon = [0.00, 0.01, 0.05, 0.10, 1.00]$	
0.0	42	51	83	97	66	70
0.01	76	60	23	2	47	48
0.05	85	88	46	7	73	74
0.10	96	94	86	10	94	94
1.00	93	91	61	97	99	99

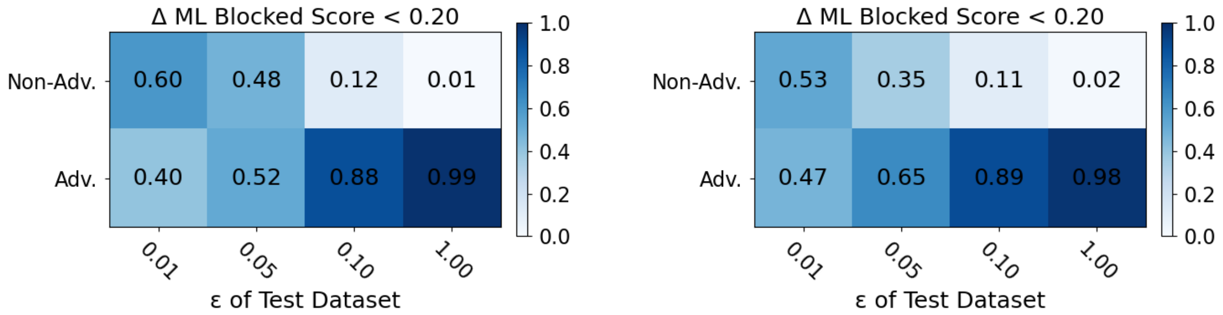

Figure 5: Confusion matrix for the adversarial images with Δ ML blocked (or free) classification score < 0.2 as classified by the RF (all ϵ trained) (Left) and fDNN (Right) as described in Section 3.2.

Adversarial examples are detectable using the HDP topics generated by MindfuL™. A natural question that arises is if a subset of the HDP topics exists that provides high predictive power for adversarial attacks. This was tested using SHAP,[19] which is one of the methods used in the MindfuL™ Performance and Strategy Correlators as shown in Figure 3, to rank variable importance according to the average impact each feature contributes to the prediction. This verified that a subset of HDP topics in conjunction with the AlexNet classifier's output could be used to identify adversarial examples regardless of the ϵ perturbation parameter applied. SHAP was not used to explore the fDNN due to an inability to transform from the encoded RF feature space used as inputs to the fDNN to the HDP topic weights. The RF SHAP importance was verified against the RF feature importance calculated using a permutation importance algorithm. Figure 6 deptics the variable importance for the RF trained using the $\epsilon = 0.00$ and 1.00 by their average impact on model output as calculated by the Shapely value.

To further explore this line of research, the features of Figure 6 with an average impact greater than 5% were identified and subjected to two tests: 1) removing these variables from the training of each of the RFs; 2) use only these variables for the training of each of the RFs. The results of the first test show that the classification accuracy of the $\epsilon = 0.01$ and 0.10 samples was reduced by greater than 15% with the removal of these seven variables identified by the $\epsilon = 1.00$ RF classifier. The $\epsilon = 0.05$ and 1.00 classification accuracy was reduced by less than 2% in this test due to correlations between the remaining HDP topics that accounted for the loss of the information provided by those removed. The second test, using only the top seven variables, resulted in RF classifiers with reduced test dataset classification accuracies varying from a 3% reduction at $\epsilon = 0.00$ to 1% at $\epsilon = 1.00$ compared to the RF classifier accuracies shown in Table 1. These experiments show that there is a set of features shared across the adversarial examples, which is how the MindfuL™ extracted HDP topics are able to be used to interpolate to ϵ values despite their abscense during training.

5. DISCUSSION

The ongoing effort to independently measure the competency of machine learning models in the face of real-world threats like adversarial attacks is becoming increasingly important as AI/ML systems become more prevalent.

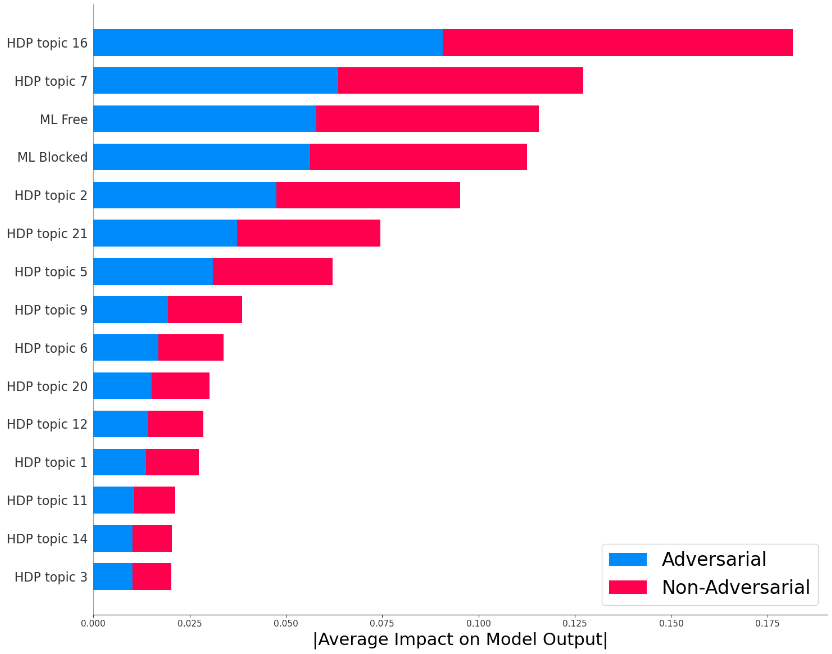

Figure 6: SHAP feature importance as measured by mean absolute Shapley values for the RF $\epsilon = [0.00, 1.00]$ model. The feature is shown for both Adversarial and Non-Adversarial classification scores.

This work accomplishes proof of the concept that MindfuL™ system's HDP topics provide useful insight into the identification of adversarial images without exposure to adversarial data during training. Straightforward ML methods were able to detect FSGM attacks, showing success at large levels of adversarial perturbation and moderate success at low levels of adversarial perturbation. As these methods rely on detecting adversarial attacks with HDPs derived from ORB feature distributions, only adversarial attacks that can change classification category without disturbing the ORB features are expected to be undetectable. As adversarial attacks are unrelated to the HDP algorithm or its underlying features, we expect this approach to provide robust detection of attacks beyond the FSGM approach. Future research into this work will include studying both black- and white-box adversarial attacks of varying strength and complexity to determine how robust the HDP topics remain for adversarial detection. Additional studies could also answer the question of whether the MindfuL™ experience encoder HDP topic's can be adversarially attacked in conjunction with the underlying AlexNet.

This paper shows the promise of adversarial detection using existing competency-awareness systems without the need for any changes to the underlying methods for competency estimation. Several different methods were explored to highlight potential avenues for exploration in the future. This study provides more motivation for further study of the applicability of competency-awareness agents to adversarial detection or identification of other out-of-domain samples, as well as some of the challenges of using these methods to tackle these problems. We expect that combining this approach with other adversarial detection methods could increase its effectiveness. Other future work includes testing how the competency-awareness agent responds to different adversarial methods. Most detection techniques can be defeated through targeted adversarial attacks,[7] but the presence of a competency-awareness agent, like MindfuL™, acting in conjunction with the underlying ML system may provide another layer of security in the detection of adversaries or detecting that the dataset has been shifted in some fundamental way. Each of these avenues of research pose interesting questions in the context of competency-awareness systems.

6. ACKNOWLEDGMENTS

This material is based upon work supported by the Defense Advanced Research Projects Agency (DARPA) under Contract No. HR001120C0033. Approved for Public Release; Distribution Unlimited "Not export controlled per ES-FL-041223-0057"

REFERENCES

[1] Sarker, J. I. and Ahad, W., "Coverage of the kashmir conflict in bangladeshi media: A content analysis," *Dicle Academi Dergisi* **1**(2), 1–20 (2021).

[2] Kurakin, A., Goodfellow, I. J., and Bengio, S., "Adversarial examples in the physical world," in [*Artificial intelligence safety and security*], 99–112, Chapman and Hall/CRC (2018).

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

[4] Qiu, H., Zeng, Y., Zhang, T., Jiang, Y., and Qiu, M., "Fencebox: A platform for defeating adversarial examples with data augmentation techniques," *arXiv preprint arXiv:2012.01701* (2020).

[5] Biggio, B., Corona, I., Maiorca, D., Nelson, B., Šrndić, N., Laskov, P., Giacinto, G., and Roli, F., "Evasion attacks against machine learning at test time," in [*Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2013, Prague, Czech Republic, September 23-27, 2013, Proceedings, Part III 13*], 387–402, Springer (2013).

[6] Dang, H., Liu, F., Stehouwer, J., Liu, X., and Jain, A. K., "On the detection of digital face manipulation," in [*Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern recognition*], 5781–5790 (2020).

[7] Carlini, N. and Wagner, D., "Adversarial examples are not easily detected: Bypassing ten detection methods," in [*Proceedings of the 10th ACM workshop on artificial intelligence and security*], 3–14 (2017).

[8] Grosse, K., Manoharan, P., Papernot, N., Backes, M., and McDaniel, P., "On the (statistical) detection of adversarial examples," *arXiv preprint arXiv:1702.06280* (2017).

[9] Gong, Z., Wang, W., and Ku, W.-S., "Adversarial and clean data are not twins," *arXiv preprint arXiv:1704.04960* (2017).

[10] Pertigkiozoglou, S. and Maragos, P., "Detecting adversarial examples in convolutional neural networks," *arXiv preprint arXiv:1812.03303* (2018).

[11] Koenig, N. and Howard, A., "Design and use paradigms for gazebo, an open-source multi-robot simulator," in [*2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566)*], **3**, 2149–2154 vol.3 (2004).

[12] Krizhevsky, A., Sutskever, I., and Hinton, G. E., "Imagenet classification with deep convolutional neural networks," in [*Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1*], *NIPS'12*, 1097–1105, Curran Associates Inc., Red Hook, NY, USA (2012).

[13] Nicolae, M.-I., Sinn, M., Tran, M. N., Buesser, B., Rawat, A., Wistuba, M., Zantedeschi, V., Baracaldo, N., Chen, B., Ludwig, H., Molloy, I. M., and Edwards, B., "Adversarial robustness toolbox v1.0.0," (2019).

[14] Teh, Y. W., Jordan, M. I., Beal, M. J., and Blei, D. M., "Hierarchical dirichlet processes," *Journal of the American Statistical Association* **101**(476), 1566–1581 (2006).

[15] Rublee, E., Rabaud, V., Konolige, K., and Bradski, G., "Orb: An efficient alternative to sift or surf," in [*2011 International Conference on Computer Vision*], 2564–2571 (2011).

[16] Breiman, L., "Random forests," *Machine learning* **45**, 5–32 (2001).

[17] Kong, Y. and Yu, T., "A deep neural network model using random forest to extract feature representation for gene expression data classification," *Scientific reports* **8**(1), 16477 (2018).

[18] Agarap, A. F., "Deep learning using rectified linear units (relu)," *arXiv preprint arXiv:1803.08375* (2018).

[19] Lundberg, S. M. and Lee, S.-I., "A unified approach to interpreting model predictions," in [*Advances in Neural Information Processing Systems 30*], Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R., eds., 4765–4774, Curran Associates, Inc. (2017).