Adversarial Robustness and Explainability of Machine Learning Models

Jamil Gafur
jamil-gafur@uiowa.edu
University of Iowa
Iowa City, Iowa, USA

Steve Goddard
steve-goddard@uiowa.edu
University of Iowa
Iowa City, Iowa, USA

William K.M. Lai
wkl29@cornell.edu
Cornell University
Ithica, New York, USA

ABSTRACT

The rapid advancement of machine learning has brought forth sophisticated neural network models harnessing computational prowess and vast datasets for diverse applications. Nonetheless, with the proliferation of these complex models, apprehensions have surfaced regarding their resilience, interpretability, and biases. To mitigate these concerns, we propose the "Adversarial Observation" framework, amalgamating explainable and adversarial methodologies for comprehensive neural network scrutiny. By integrating explainable techniques, users gain profound insights into the model's internal mechanisms, fostering transparency and facilitating bias identification. This framework aims to enhance the trustworthiness and accountability of neural network systems amidst their expanding utility.

CCS CONCEPTS

• **Computing methodologies** → **Machine learning**; *Explainable machine learning*; • **Computer systems organization** → Application program interfaces (APIs).

KEYWORDS

Machine Learning, eXplainable AI, Framework, packaging

ACM Reference Format:
Jamil Gafur, Steve Goddard, and William K.M. Lai. 2024. Adversarial Robustness and Explainability of Machine Learning Models. In *Practice and Experience in Advanced Research Computing (PEARC '24), July 21–25, 2024, Providence, RI, USA.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3626203.3670522

1 INTRODUCTION

Machine learning (ML) models, particularly neural networks, have gained widespread adoption owing to their exceptional performance in various domains. However, their increasing prevalence has brought forth concerns regarding their robustness and interpretability. Existing research highlights the pressing need for comprehensive methodologies and tools to address these challenges.

Kumar et al. [14] identified a significant gap in industry practitioners' capabilities to defend their systems against adversarial attacks and manipulation. They observed a lack of essential tools among many organizations, indicating a clear demand for guidance in this area. These findings are supported by seminal research studies [21, 22], underscoring the urgency of enhancing model robustness and interpretability.

The empirical evidence and theoretical insights provided by these studies underscore the necessity for the development of comprehensive methodologies and frameworks. Such measures are essential for addressing vulnerabilities and enhancing the explainability of ML models. Rigorous investigations conducted by these studies emphasize the critical importance of exploring robustness evaluation techniques and interpretability frameworks within the context of Artificial Intelligence (AI) systems.

Interpreting and explaining model predictions are crucial for establishing trust and promoting fairness and transparency in AI/ML systems [6, 7, 23]. This trust can be weakened through the use of Adversarial attacks [3]. These attacks generate model inputs that include noise that is imperceptible to human recognition but leads the model to classify the input incorrectly. By understanding the factors contributing to model decisions, stakeholders can have confidence in the system's reliability and accountability [2]. Interpretation methods also facilitate the detection and mitigation of biases, ensuring equitable treatment across demographics and minimizing discriminatory outcomes.

Explanations generated from interpretation methods not only empower end-users to understand the reasoning behind AI-generated predictions but also facilitate regulatory compliance and ethical decision-making processes. Advancements in interpretability research significantly contribute to the responsible development and deployment of AI technologies, bolstering public trust and fostering the adoption of AI systems in critical domains.

We introduce the "Adversarial Observation" framework, a Python-based toolkit designed to facilitate the implementation of adversarial attacks and Explainable AI (XAI) techniques during machine learning model training. This framework provides researchers and industry practitioners with an end-to-end set of tools to evaluate model robustness, interpretability, and bias.

Our framework implements two prominent adversarial methods: the Fast Gradient Sign Method (FGSM) [8] and the Adversarial Particle Swarm Optimization (APSO) technique [19]. These methods reliably generate adversarial noise, this noise can be used to generate adversarial noise that can be used to attack a model. By incorporating this noise into a training set, the model can become more robust against attacks.

In terms of XAI techniques, we have integrated activation mapping, which visualizes and analyzes significant input regions influencing model predictions. Additionally, we have modified the APSO algorithm to determine global feature importance and enable local interpretation. This analysis helps understand and mitigate biases

PEARC '24, July 21–25, 2024, Providence, RI, USA

ACM ISBN 979-8-4007-0419-2/24/07
https://doi.org/10.1145/3626203.3670522

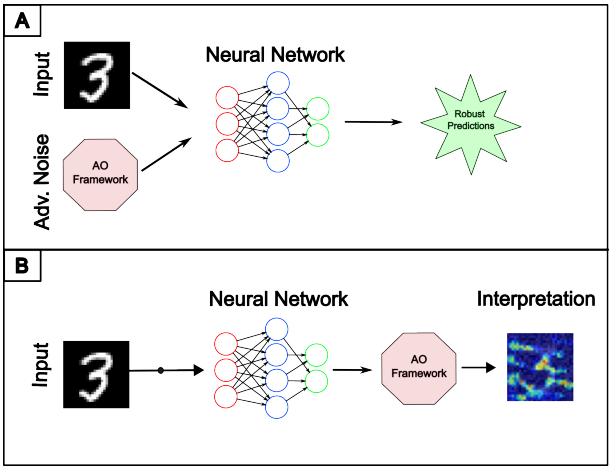

Figure 1: Two use cases for the framework: (A) generating adversarial images to secure the network [8, 27], and (B) generating interpretability of the model after training [12].

in the model, promoting the development of fair and unbiased AI systems [12, 15, 25].

Our research stands out by combining different techniques into a single framework, without losing their strengths. As shown in Fig. 1 A noise can be generated and incorporated into neural network training for more robust predictions, while Fig. 1 B shows that the framework can also be used to generate explanations of a model by using different XAI techniques. This integration enhances the overall performance and provides a more complete solution. We have reworked several XAI algorithms and adversarial approaches, bringing them together in a unified framework. This approach allows users to unravel the complex nature of neural networks more effectively and with greater clarity than ever before. Moreover, our framework includes a diverse set of attacks and XAI methods, accompanied by detailed explanations of each technique and how to implement them.

This paper is broken up into five distinct and complementary sections. Section 2 provides an overview of the implementations of adversarial attacks and XAI methods in our framework. Section 3 discusses the framework and its use cases. Section 4 and 5 concludes the paper with a discussion of potential applications and future research directions.

2 CURRENT STATE OF THE ART

In the realm of eXplainable Artificial Intelligence (XAI), various libraries stand out, namely Quantus [10], INNVestigate [1], and Captum [13]. Each library is tailored for enhancing model interpretability, yet they exhibit distinct characteristics.

Quantus, designed to seamlessly integrate quantitative XAI into existing software frameworks, requires minimal code modification. It prioritizes a comprehensive array of metrics and is compatible with both PyTorch and TensorFlow frameworks. Its primary focus lies in supporting a wide range of evaluation metrics, ensuring a versatile tool for model understanding.

INNVestigate, in a similar vein, offers a library of XAI techniques but employs a "wrapper" programming style. Through the instantiation of an analyzer object tethered to a TensorFlow model, it treats the analyzer as the model during training, thereby enabling the analysis of model predictions using various XAI techniques post-training. This approach provides flexibility in incorporating XAI into the training pipeline.

Captum, a creation of Facebook AI, assumes a pivotal role with its specialized focus on PyTorch models and attribution methods. It stands out for its emphasis on transparency and interpretability. Captum facilitates an in-depth understanding of the nuanced impact of individual input features on model predictions. Its distinctive feature is its targeted application to PyTorch models, providing a valuable tool for those working within that framework.

3 TECHNIQUES

Each of these libraries focuses on developing XAI techniques, and while they are under active development and have a wide range of techniques, they are not comprehensive. These techniques are used for different purposes, and while some are similar, they are not the same as ours.

In this section, we discuss common techniques employed to enhance model interpretability. Adversarial attack algorithms and model interpretability methods, utilize either model gradients or input gradients to achieve their goals. One prevalent algorithm in this domain of adversiral attacks is the Fast Gradient Sign Method (FGSM) [8]. Another notable approach is Activation Mapping, which reveals insights into model behavior by mapping its activations. Additionally, the Shapley Additive Explanations (ShAP) method [9] contributes to interpretability by providing additive explanations of how input influences model output.

While these algorithms serve as potent tools for interpretability, like all XAI approaches, they are not without limitations and biases. Instead of introducing a new orthogonal approach, we have chosen to develop an interoperable framework that incorporates all these techniques. This integration allows us to leverage the collective power of these algorithms, surpassing their individual capabilities and providing a more comprehensive solution.

Fast Gradient Sign Method: Adversarial Attack

The Fast Gradient Sign Method (FGSM) [8] is a well-known technique for generating adversarial examples. It strategically updates input data by leveraging the gradient of the loss function with respect to the input, inducing misclassification by the target model.

FGSM identifies influential features in the input data for the model's decision-making by analyzing the gradient's direction. It then updates the input data in the opposite direction of the gradient, with the magnitude determined by a small constant, ϵ, ensuring imperceptibility to the human eye.

The FGSM equation for generating adversarial examples is defined as follows:

$$\mathbf{x}_{\text{adv}} = \mathbf{x} + \epsilon \cdot \text{sign}(\nabla_{\mathbf{x}} J(\theta, \mathbf{x}, y)) \tag{1}$$

Here, $\mathbf{x}$ is the input data, ϵ is the perturbation magnitude, $\nabla_{\mathbf{x}} J(\theta, \mathbf{x}, y)$ is the gradient of the loss function, and $\text{sign}(\cdot)$ is the sign function.

FGSM is a one-shot attack, requiring only a single iteration for adversarial example generation [8].

Adversarial Particle Swarm Optimization: Adversarial Attack

The Adversarial Particle Swarm Optimization (APSO) [20] algorithm utilizes Particle Swarm Optimization (PSO) [11] to search for perturbations in neural networks, aiming to maximize misclassification.

PSO is a heuristic optimization technique where a swarm of particles navigates a high-dimensional search space. Each particle represents a potential solution, updated iteratively based on its best-known position and the global best-known position of the swarm. PSO's algorithmic formulation is expressed as follows:

$$\mathbf{v}_i^{t+1} = \omega \mathbf{v}_i^t + c_1 r_1 (\mathbf{p}_i^t - \mathbf{x}_i^t) + c_2 r_2 (\mathbf{p}_g^t - \mathbf{x}_i^t) \tag{2}$$

In this equation, $\mathbf{v}_i^t$ is the velocity of particle i at iteration t, ω is the inertia weight, $\mathbf{p}_i^t$ is the best-known position of particle i at iteration t, $\mathbf{x}_i^t$ is the current position of particle i at iteration t, and c_1, c_2, r_1, and r_2 are parameters.

APSO applies PSO to discover perturbations maximizing misclassification by optimizing the cost function. It iteratively explores the input space, updating positions and velocities to identify vulnerable regions in neural networks susceptible to adversarial attacks. Swarm algorithms like APSO have gained prominence for inducing misclassification in neural networks [18, 20].

Activation Mapping: Visualization & Feature Attribution

Activation Mapping facilitates the visualization of essential features used by machine learning models for prediction [25]. It analyzes the network's gradients for a given input, generating activation maps that assign higher scores to more significant features.

The Activation Mapping equation is expressed as:

$$\mathrm{A}_i^c = \sum_k \frac{\partial y^c}{\partial \mathbf{a}_i^k} \tag{3}$$

In this equation, $\mathbf{A}_i^c$ denotes the activation map for class c at layer i, y^c is the output score for class c, and $\mathbf{a}_i^k$ represents the activation of unit k at layer i. Activation Mapping aids in comprehending the significance of input features for model predictions.

Activation maps highlight pixels crucial to the model's predictions, assisting in tasks like object localization. They contribute to model debugging, diagnosis, and provide insights into complex model decision-making processes. Our implementation is based on the research presented in Simonyan et al. [25].

Shapley Additive Explanations (ShAP): Feature Attribution

Shapley Additive Explanations (ShAP) [4, 16] represents a feature attribution technique that quantitatively measures the importance of each feature in the input space for the model's output. Using Shapley values from game theory, ShAP estimates feature contributions by considering all possible subsets of features.

ShAP belongs to additive feature attribution methods [16], explaining the model's output as a sum of real values assigned to each input feature. It possesses three key properties: local accuracy, missingness, and consistency [4]. Model-agnostic, ShAP is applicable to any machine learning model, irrespective of architecture, size, or complexity, handling both classification and regression problems. Our implementation of ShAP is based on the work of Lundberg et al. [16].

4 ADVERSARIAL OBSERVATION FRAMEWORK

The Adversarial Observation framework provides a convenient means of accessing the methods discussed previously in a single framework. This adaptable framework serves as an interface for interacting with machine learning models, offering a user-friendly API for generating adversarial examples and extracting feature importance. Its design emphasizes modularity and extensibility, enabling the seamless integration of novel methods and algorithms as they are published. To our knowledge, this is the first time these algorithms have been leveraged together in a single framework.

We explore three distinct use cases that demonstrate the framework's potential and how the modules seamlessly interoperate with each other. The first use case involves adversarial visualization, wherein we generate deceptive examples to fool machine learning models by using the APSO and activation mapping. The second use case focuses on enhancing adversarial robustness with explainability, whereby the model's resilience is improved by subjecting it to adversarial examples with FGSM and ShAP. Lastly, the third use case revolves around feature importance, encompassing the extraction of crucial model features associated with a specific class with FGSM and activation mapping.

For the experiments, we employed a Convolutional Neural Network (CNN) that was trained on the widely used MNIST dataset [5]. The training details, including the model architecture and parameters, can be found in the examples folder of the GitHub repository[1]. The training process and model architecture was based on the official PyTorch MNIST documentation for Image Classification Using ConvNets[2].

4.1 Adversarial Visualization

In this section, we demonstrate a method for generating adversarial examples using the APSO algorithm and activation mapping. Specifically, we initialize the swarm with random noise and define the cost function as defined in Eq. 4.

The cost function $\mathrm{cost}(x)$ combines the prediction probability and the normalized activation to determine the overall cost associated with the image. For each iteration the swarm takes, we reduce the particles to a 2-D Uniform manifold approximation (UMAP) [17] space. This allows us to visualize how the particles move over time.

$$cost(x) = f(x)[3] * act(x) \tag{4}$$

where x represents the input MNIST image, $f(x)[3]$ denotes the prediction probability for the digit 3, $\mathrm{act}(x)$ represents the

[1] https://github.com/EpiGenomicsCode/Adversarial_Observation
[2] https://pytorch.org/examples/

activation of the model. This activation can come from any XAI technique within the framework. By utilizing XAI techniques we can optimize for features that most represent our output of interest.

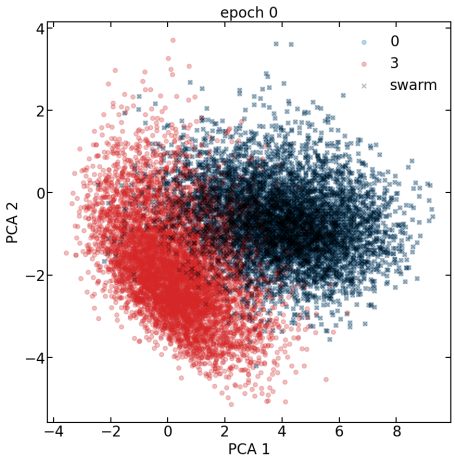

Figure 2: APSO initialization (Epoch 0)

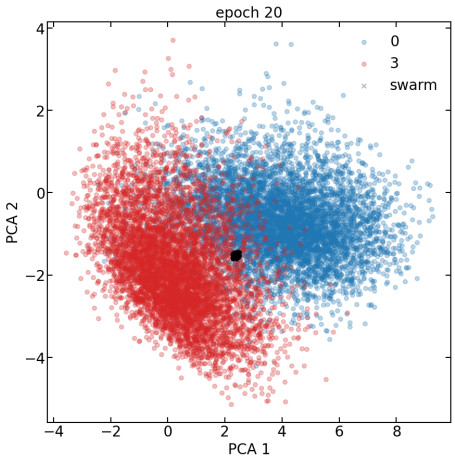

Figure 3: APSO after 20 epochs

Figure 4: Visualization of APSO initialization and convergence after 20 epochs.

The swarm iteratively searches for images with high predictive values for the "3" class that also have high activations for explainable visualization. We run the APSO algorithm for 20 epochs, with Fig. 2 depicting the initialized swarm and Fig. 3 showing the swarm after 20 epochs. For each epoch, we present the points with the highest confidence in classifying the image as "3", which are illustrated in figures 5 and 6.

Given the framework's broad utility, it can readily be customized to suit various purposes. Specifically, the implementation of the APSO algorithm can facilitate the generation of comparable outcomes to those reported in [15, 24].

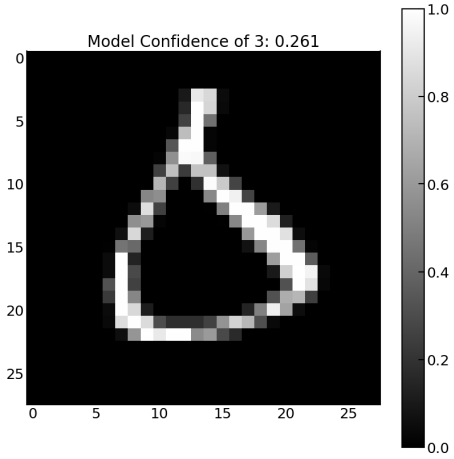

Figure 5: Initial Best (Epoch 1)

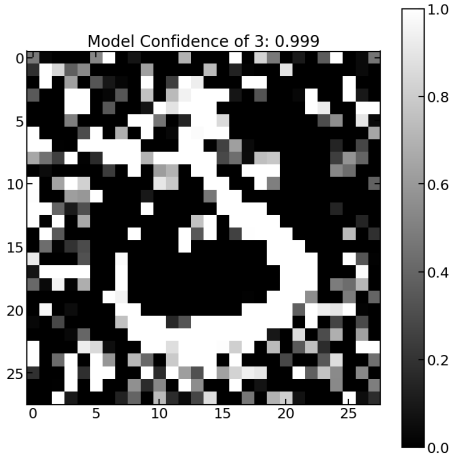

Figure 6: Best after 20 epochs

Figure 7: Visualization of the image with the highest confidence of "3" found by the swarm after 1 and 20 epochs.

4.2 Adversarial Robustness

This section showcases the capability of our framework in enhancing the robustness of a Neural Network against adversarial noise and the updated explainability based on ShAP values. We utilize the FGSM attack [8] to generate such noise and incorporate it into the training data as described in Mosli et al. [15]. Adversarial training has proven to be an effective approach in improving a model's resilience to future attacks [15, 26], which involves adding an extra neuron to the output layer to classify an "adversarial" or "unknown" class.

Our framework simplifies the generation and addition of adversarial noise to using the FGSM attack, offering an easy and efficient way to retrain the network. This technique can significantly enhance the robustness of a neural network with minimal additional effort. By incorporating adversarial noise into the training process,

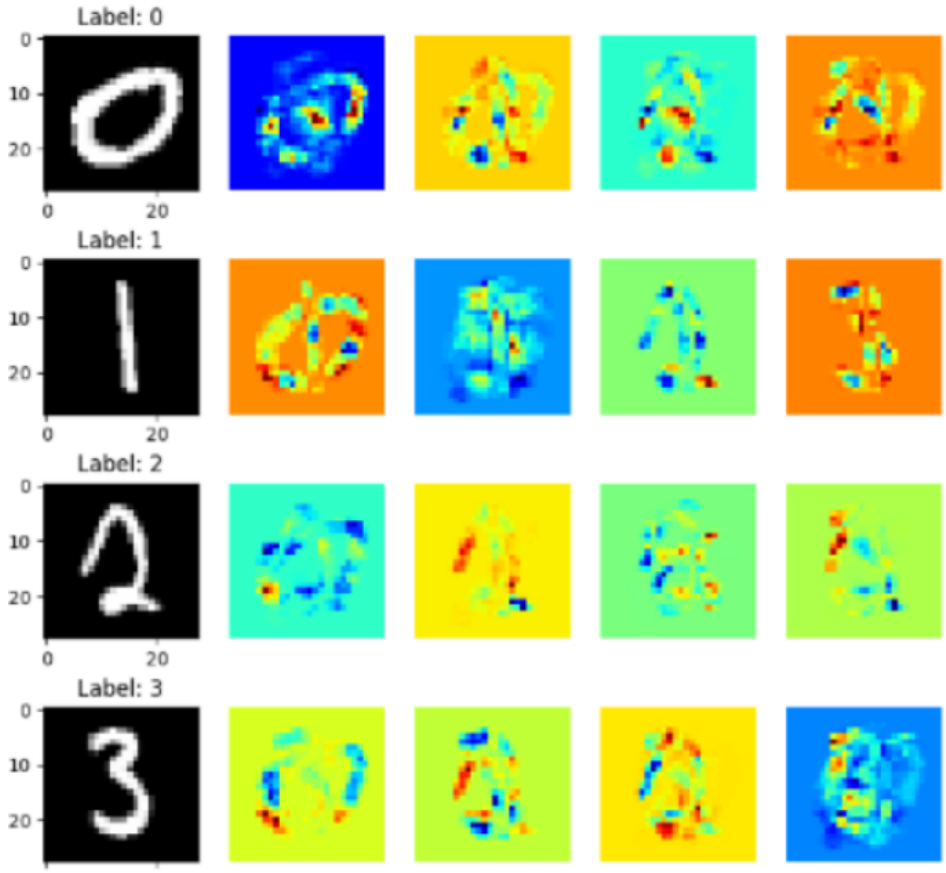

Figure 8: ShAP values for an adversarial secure network. The left column is the original validation image. All other columns are ShAP values for the same image with respect to different classifications (0-3)

the model becomes better prepared to handle adversarial attacks during deployment.

We then use the (SHapley Additive exPlanations) ShAP values to explain the model's new decision-making process to see if by applying this technique the model is still able to identify the correct features. ShAP values represent the change in the model's output when a specific feature is included compared to when it is excluded, taking into account all possible combinations of features. This approach enables a comprehensive understanding of the model's decision-making process, shedding light on the importance of each feature in influencing the final prediction.

We use the same model as in the previous section, take some validation data and run it through the ShAP algorithm. This is then compared to the ShAP values of the same data after the model has been retrained with adversarial noise. The results are shown in Fig 8.

4.3 Feature Importance of Adversarial Images

Our framework can also be used to generate feature importance maps for original data, and adversarial images. To do this we utilize the FGSM and the activation mapping. This approach enables us to visualize the most significant features that influence a model's decision-making process, and how adversarial noise affects what the model "sees". An example of this is shown in Fig 9. The noise shown in Fig 9 **C** is the noise multiplied by ϵ and added to the original.

The development of feature importance maps represents a significant contribution to the field of XAI. By enabling us to understand a model's decision-making process and how it changes under adversarial attack, we can increase accountability and build trust in machine learning systems. We show how the combination of activation mapping and the FGSM attack can be used to generate feature importance maps for adversarial images.

Our end-to-end framework enables the smooth implementation of the aforementioned approach, making it both feasible and straightforward to execute. In Fig. 10, we present the activation mapping of the original image alongside the adversarial image (Fig. 11). While these images exhibit visual similarity, a careful examination reveals a subtle alteration in the center-right region of both images. This observation underscores the significant impact that minor perturbations in the overall gradients of the model can exert on its prediction outcomes. Our innovative methodology combines adversarial attacks and activation mapping, thereby providing valuable insights into the susceptibility of the model to

Figure 9: Adversarial image generated using FGSM with $\epsilon = 0.001$.

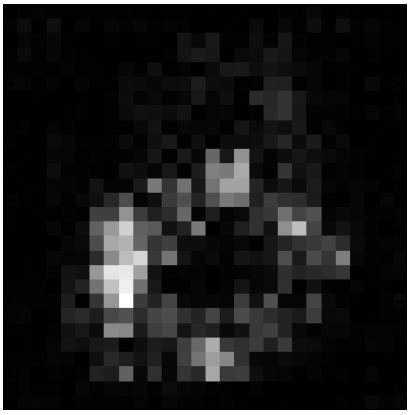

Figure 10: Activation mapping comparison.

Figure 11: Adversarial image generated using FGSM with $\epsilon = 0.001$.

such attacks. By analyzing the influence of adversarial noise on the decision-making process of the model, our approach enables the identification of vulnerabilities and the enhancement of the robustness of machine learning systems. The seamless integration of our proposed approach within the comprehensive end-to-end framework ensures its practicality and ease of implementation.

The utilization of feature importance maps and the identification of how gradients activate differently with malicious inputs represent a novel and effective means of enhancing the transparency and interpretability of machine learning models. Our framework simplifies the generation of these maps through activation mapping, presenting a fresh approach to comprehending the decision-making process of intricate models. Ultimately, this approach contributes to ensuring the responsible and accountable use of machine learning systems, rendering them more trustworthy and beneficial for society.

5 CONCLUSION

The Adversarial Observation framework has been specifically developed to meet the requirements of researchers and practitioners in the field of machine learning. This comprehensive framework offers a wide range of algorithms for adversarial attacks and explainable methods. We show how the framework can enable users to conduct thorough evaluations of adversarial resistance, seamlessly integrate adversarial training into their data, and visualize activation maps based on inputs. While these are only a few of the many capabilities of the framework, they demonstrate the framework's potential to enhance the robustness and reliability of machine learning models.

One of the key advantages of the Adversarial Observation framework lies in its ability to enhance model robustness through the combination of generative adversarial attacks and explainable AI techniques. The framework's advanced algorithms for generating adversarial noise provide researchers and practitioners with powerful tools to assess their model's resistance against attacks. This can be used to address the issues stated by [14].

The adoption of the Adversarial Observation framework brings significant advancements to the effectiveness and reliability of models. By introducing a novel approach to incorporating adversarial attacks and XAI techniques, this framework empowers researchers and practitioners to strengthen the robustness of their models while gaining deep insights into their inner workings.

6 FUTURE WORK

We aim to broaden the scope of our framework by testing it on diverse datasets, including CIFAR-10, DNA sequencing data, and text-generated data. This expansion will allow us to evaluate the generalizability and effectiveness of our framework across different domains and data types, providing valuable insights into its performance and applicability in real-world scenarios.

To enhance the transparency and interpretability of neural networks, we will integrate additional Explainable AI techniques into our framework. Specifically, we will incorporate the LIME (Local Interpretable Model-Agnostic Explanations) method [9]. By leveraging these techniques, our aim is to gain deeper insights into the decision-making processes of neural networks and identify the key features that contribute to their predictions. Integrating these techniques will provide interpretable explanations and strengthen the reliability of our framework.

Furthermore, we plan to explore the similarities and connections between the Adversarial Particle Swarm Optimization algorithm

and the ShAP method. This analysis will enable us to comprehensively understand the underlying principles and potential synergies between these techniques, thereby advancing our knowledge of their individual strengths and opening up new avenues for research in the field of XAI.

Our future research endeavors involve expanding dataset coverage, exploring the connections between APSO and Shap, and incorporating additional XAI techniques. By collaborating with domain experts, we aim to apply our framework in diverse scientific contexts, driving progress in the field of XAI and facilitating significant discoveries.

REFERENCES

[1] Maximilian Alber, Sebastian Lapuschkin, Philipp Seegerer, Miriam Hägele, Kristof T Schütt, Grégoire Montavon, Wojciech Samek, Klaus-Robert Müller, Sven Dähne, and Pieter-Jan Kindermans. 2019. iNNvestigate neural networks! *J. Mach. Learn. Res.* 20, 93 (2019), 1–8.

[2] Nicholas Carlini and David Wagner. 2017. Adversarial examples are not easily detected: Bypassing ten detection methods. In *Proceedings of the 10th ACM workshop on artificial intelligence and security*. 3–14.

[3] Anirban Chakraborty, Manaar Alam, Vishal Dey, Anupam Chattopadhyay, and Debdeep Mukhopadhyay. 2021. A survey on adversarial attacks and defences. *CAAI Transactions on Intelligence Technology* 6, 1 (2021), 25–45.

[4] Anupam Datta, Shayak Sen, and Yair Zick. 2016. Algorithmic transparency via quantitative input influence: Theory and experiments with learning systems. In *2016 IEEE symposium on security and privacy (SP)*. IEEE, 598–617.

[5] Li Deng. 2012. The mnist database of handwritten digit images for machine learning research. *IEEE Signal Processing Magazine* 29, 6 (2012), 141–142.

[6] Finale Doshi-Velez and Been Kim. 2017. Towards a rigorous science of interpretable machine learning. *arXiv preprint arXiv:1702.08608* (2017).

[7] Finale Doshi-Velez and Been Kim. 2018. Considerations for evaluation and generalization in interpretable machine learning. *Explainable and interpretable models in computer vision and machine learning* (2018), 3–17.

[8] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. 2014. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572* (2014).

[9] Alex Gramegna and Paolo Giudici. 2021. SHAP and LIME: an evaluation of discriminative power in credit risk. *Frontiers in Artificial Intelligence* 4 (2021), 752558.

[10] Anna Hedström, Leander Weber, Daniel Krakowczyk, Dilyara Bareeva, Franz Motzkus, Wojciech Samek, Sebastian Lapuschkin, and Marina M-C Höhne. 2023. Quantus: An explainable ai toolkit for responsible evaluation of neural network explanations and beyond. *Journal of Machine Learning Research* 24, 34 (2023), 1–11.

[11] James Kennedy and Russell Eberhart. 1995. Particle swarm optimization. In *Proceedings of ICNN'95-international conference on neural networks*, Vol. 4. IEEE, 1942–1948.

[12] Khaled Khalifa, Mona Safar, and M Watheq El-Kharashi. 2020. Verification of Neural Networks for Safety Critical Applications. In *2020 32nd International Conference on Microelectronics (ICM)*. IEEE, 1–4.

[13] Narine Kokhlikyan, Vivek Miglani, Miguel Martin, Edward Wang, Bilal Alsallakh, Jonathan Reynolds, Alexander Melnikov, Natalia Kliushkina, Carlos Araya, Siqi Yan, et al. 2020. Captum: A unified and generic model interpretability library for pytorch. *arXiv preprint arXiv:2009.07896* (2020).

[14] Ram Shankar Siva Kumar, Magnus Nyström, John Lambert, Andrew Marshall, Mario Goertzel, Andi Comissoneru, Matt Swann, and Sharon Xia. 2020. Adversarial machine learning-industry perspectives. In *2020 IEEE Security and Privacy Workshops (SPW)*. IEEE, 69–75.

[15] Qi Liu, Tao Liu, Zihao Liu, Yanzhi Wang, Yier Jin, and Wujie Wen. 2018. Security analysis and enhancement of model compressed deep learning systems under adversarial attacks. In *2018 23rd Asia and South Pacific Design Automation Conference (ASP-DAC)*. IEEE, 721–726.

[16] Scott M Lundberg and Su-In Lee. 2017. A unified approach to interpreting model predictions. *Advances in neural information processing systems* 30 (2017).

[17] Leland McInnes, John Healy, and James Melville. 2018. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426* (2018).

[18] Rayan Mosli, Matthew Wright, Bo Yuan, and Yin Pan. 2019. They might not be giants: Crafting black-box adversarial examples with fewer queries using particle swarm optimization. *arXiv preprint arXiv:1909.07490* (2019).

[19] Hyunjun Mun, Sunggwan Seo, Baehoon Son, and Joobeom Yun. 2022. Black-box audio adversarial attack using particle swarm optimization. *IEEE Access* 10 (2022), 23532–23544.

[20] Hyunjun Mun, Sunggwan Seo, Baehoon Son, and Joobeom Yun. 2022. Black-Box Audio Adversarial Attack Using Particle Swarm Optimization. *IEEE Access* 10 (2022), 23532–23544. https://doi.org/10.1109/ACCESS.2022.3152526

[21] Nicolas Papernot, Patrick McDaniel, Ian Goodfellow, Somesh Jha, Z Berkay Celik, and Ananthram Swami. 2017. Practical black-box attacks against machine learning. In *Proceedings of the 2017 ACM on Asia conference on computer and communications security*. 506–519.

[22] Nicolas Papernot, Patrick McDaniel, Arunesh Sinha, and Michael Wellman. 2016. Towards the science of security and privacy in machine learning. *arXiv preprint arXiv:1611.03814* (2016).

[23] Cynthia Rudin, Chaofan Chen, Zhi Chen, Haiyang Huang, Lesia Semenova, and Chudi Zhong. 2022. Interpretable machine learning: Fundamental principles and 10 grand challenges. *Statistic Surveys* 16 (2022), 1–85.

[24] K. G. Shreeharsha, Charudatta Korde, M. H. Vasantha, and Y. B. Nithin Kumar. 2021. Training of Generative Adversarial Networks using Particle Swarm Optimization Algorithm. *2021 IEEE International Symposium on Smart Electronic Systems (iSES) (Formerly iNiS)* (2021), 127–130.

[25] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. 2013. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034* (2013).

[26] Jiakai Wang. 2021. Adversarial Examples in Physical World.. In *IJCAI*. 4925–4926.

[27] Cihang Xie, Mingxing Tan, Boqing Gong, Jiang Wang, Alan L Yuille, and Quoc V Le. 2020. Adversarial examples improve image recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 819–828.