Journal of Advances and Scholarly Researches in Allied Education
Vol. 21, Issue No. 4, May-2024,(Special Issue), ISSN 2230-7540

https://doi.org/10.29070/q2reyr59

Advanced Techniques for Improving Model Robustness in Adversarial Machine Learning

Prashis Raghuwanshi*

Senior Software Engineer and Researcher (Associate Vice President), Dallas, Texas, USA

Email: prashish14@gmail.com

Abstract - This work investigates advanced methods for improving the resilience of machine learning models against adversarial attacks. Ensuring that these models can withstand deliberately crafted inputs—called adversarial examples—has become critical as machine learning expands into high-stakes fields such as computer vision, cybersecurity, and healthcare. The study examines several types of adversarial attacks, including black-box attacks, where the attacker has no direct knowledge of the model, and white-box attacks, where the attacker has complete access to the model. Popular defense strategies, such as the Fast Gradient Sign Method (FGSM), Iterative FGSM (I-FGSM), and the Carlini and Wagner (C&W) attack, are also discussed. The work emphasizes how adversarial learning contributes to creating more resilient models by addressing both theoretical foundations and practical applications. This thorough investigation highlights the strengths and weaknesses of current approaches, as well as the ongoing need for advancements to protect model integrity against evolving threats.

Keywords: machine learning models, Robustness, Advanced Techniques, Adversarial Attacks, Adversarial Learning.

----------X----------

INTRODUCTION

From healthcare to finance, machine learning (ML) has revolutionized many sectors by delivering remarkable outcomes, often surpassing human performance in specific tasks (Jordan & Mitchell, 2015). However, as ML models are applied in more critical roles, their resilience—their ability to maintain performance despite unexpected or altered inputs—becomes increasingly vital (Madry et al., 2018). Adversarial machine learning addresses this by training models to withstand adversarial attacks, where inputs are subtly modified to cause incorrect predictions (Goodfellow et al., 2015).

These attacks can pose significant risks, particularly in high-stakes contexts such as autonomous driving and cybersecurity, where misclassifications can have severe consequences (Eykholt et al., 2018). For example, a minor change to an image could mislead a model into making a critical error, or a few altered words in a text could bypass security filters (Iyyer et al., 2018).

This work explores methods to enhance ML model robustness against such threats. It covers the fundamentals of adversarial learning, examines techniques like adversarial training and defensive distillation (Madry et al., 2018), and provides case studies showing how these methods are applied in fields such as computer vision and healthcare (Esteva et al., 2017). The goal is to highlight the importance of adversarial learning in developing reliable, trustworthy AI systems capable of resisting evolving adversarial attacks (Papernot et al., 2016; Biggio & Roli, 2018).

RESEARCH OBJECTIVE

- To understand the concept of adversarial learning and its importance in improving the resilience of machine learning models.
- To categorize adversarial attacks on machine learning models as either white-box or black-box attacks.
- To provide practical examples of the implementation of adversarial learning technology.
- To offer recommendations on balancing the advantages and limitations of adversarial learning to identify the most effective strategies for enhancing the resilience of machine learning models.

REVIEW OF RELATED WORKS

Significant research has focused on enhancing the security and resilience of neural network models in hostile environments. Various techniques and algorithms have been proposed to generate adversarial use cases and develop strong countermeasures. In this review, we analyze key studies that have contributed to this field, highlighting the relevance and innovation of these works in comparison to our proposed approach.

Liang et al. (2017): Liang and colleagues introduced the Fast Gradient Sign Method (FGSM), a pioneering approach in adversarial machine learning. FGSM proved effective in generating adversarial examples that could deceive ML models. This study laid the foundation for creating adversarial examples and has informed much of the subsequent research. Our approach builds on this by not only generating adversarial examples but also implementing countermeasures to improve model robustness.

Madry et al. (2018): Madry and his team advanced the field by proposing the Projected Gradient Descent (PGD) method, which was more effective than FGSM in generating adversarial examples and enhancing model resistance. Our work aligns with theirs by utilizing both FGSM and PGD to create adversarial samples. However, we go further by exploring additional techniques, such as generating non-differentiable adversarial examples and manipulating specific features to further strengthen model robustness.

Ren et al. (2020): Ren and colleagues conducted a comprehensive review of methods for generating adversarial examples and countermeasures in neural networks. While their work thoroughly analyzed existing techniques, our research goes a step further. We not only review and analyze current approaches but also propose innovative solutions. Specifically, our approach combines various adversarial example generation algorithms with targeted countermeasures to enhance model robustness.

Buckman et al. (2018): Buckman and colleagues introduced thermometer coding as a novel defense against adversarial examples. While our solution adopts a more comprehensive strategy by incorporating multiple defense mechanisms—such as adversarial training, adversarial instance identification, and robustness augmentation through feature manipulation—thermometer coding has shown effectiveness in certain cases. Our all-encompassing approach provides greater flexibility and stronger defenses against hostile attacks.

Sharif et al. (2019): Sharif and colleagues explored the use of K-nearest neighbor (K-NN) algorithms as a defense against adversarial examples. Although K-NN showed promising performance, our method stands out by combining several adversarial example generation strategies with countermeasures, offering more robust resistance to various types of adversarial attacks.

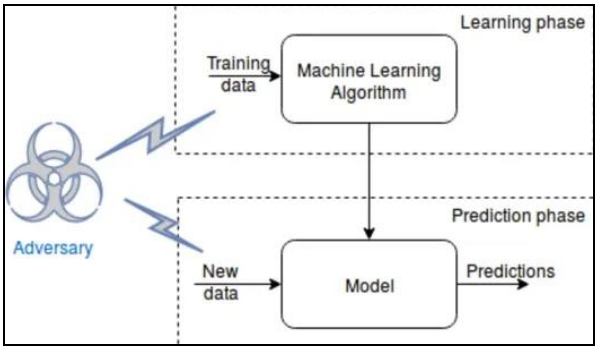
Wang et al. (2016): Wang's research, one of the earliest significant contributions to the field, introduced FGSM and demonstrated its effectiveness in modifying machine learning models.

Cheng et al. (2018): Cheng and colleagues validated these findings, demonstrating that FGSM could reliably deceive neural networks. Our work builds on these insights by incorporating FGSM into a broader framework of adversarial defenses.

Carrillo-Perez et al. (2019): Carrillo-Perez and colleagues conducted an extensive review of adversarial example generation methods and countermeasures, identifying gaps and offering new perspectives. Our research expands on these insights by experimenting with and demonstrating the effectiveness of combining multiple defense strategies to protect against adversarial attacks.

Vardhan et al. (2020): Vardhan and colleagues proposed the use of thermometer coding as a countermeasure, showing promising results in preventing adversarial attacks. Similarly, Gupta et al. (2021) explored the application of K-NN algorithms as a defense mechanism. Both approaches have shown potential, and our work integrates these ideas within a comprehensive framework that also includes other advanced adversarial defense techniques.

Adversarial Attacks Directed on Machine Learning Models:



Two types of assaults—black-box and white-box—can help to define its attacks on machine learning models.

White-Box Attacks:

White-box attacks in adversarial machine learning occur when an attacker has full knowledge of a model's architecture, parameters, and training data. With this access, the attacker can craft adversarial examples—inputs subtly modified to deceive the model into making incorrect predictions. For example, a minor alteration to an image could cause a model to misclassify it while the change remains imperceptible to humans (Goodfellow et al., 2015; Madry et al., 2018). A common method is the Fast Gradient Sign Method (FGSM), where attackers use the model's gradient information to adjust inputs, leading to targeted misclassifications (Wang et al., 2016; Cheng et al., 2018). White-box attacks are highly effective due to the attacker's in-depth understanding of the model (Liang et al., 2017).

Black-Box Attacks:

Black-box attacks occur when the attacker has no access to the model's internal details and can only interact with it by observing inputs and outputs (Papernot et al., 2016; Ren et al., 2020). Despite this limitation, attackers can still generate adversarial examples through methods like query-based attacks or by exploiting the transferability of adversarial examples from a surrogate model (Sharif et al., 2019). Black-box attacks resemble real-world scenarios where model details are often concealed. They are more challenging to defend against because they do not require detailed knowledge of the model, although they typically demand more computational resources and queries to be effective (Carrillo-Perez et al., 2019). Both white-box and black-box attacks underscore the need for robust defenses in machine learning models (Biggio & Roli, 2018).

Types of Adversarial Attacks:

Machine learning models can be subjected to various adversarial attacks. The most prevalent types include:

1. **Evasion Attacks**: These attacks aim to manipulate input data to cause misclassification or alter the model's output. Examples include the Iterative FGSM (I-FGSM) and the Fast Gradient Sign Method (FGSM) (Goodfellow et al., 2015; Wang et al., 2016).

2. **Poisoning Attacks**: In these attacks, an adversary introduces malicious data into the training set to alter the model's behavior. This can involve modifying existing training data or injecting specially crafted samples (Biggio & Roli, 2018).

3. **Model Inversion Attacks**: These attacks exploit the model's output to reconstruct sensitive information about the training data or inputs. They can potentially extract private information or disclose confidential data (Papernot et al., 2016).

4. **Membership Inference Attacks**: These attacks determine whether a specific sample was included in the model's training data. An adversary can infer membership status by analyzing the model's output probabilities (Sharif et al., 2019).

5. **Model Extraction Attacks**: In these attacks, an adversary seeks to generate a substitute model by querying the target model to obtain a copy or approximation. This can be used to acquire proprietary models or sensitive information embedded within them.

The significance of adversarial learning in the enhancement of model robustness:

Adversarial learning is crucial for enhancing the robustness of machine learning models, ensuring they perform reliably even when faced with unexpected or manipulated inputs. This method involves training models with adversarial examples—inputs designed to exploit weaknesses and cause misclassifications (Benson, 2006). By exposing models to these challenges during training, they learn to adapt to subtle changes, which improves their ability to generalize and perform accurately on new, unseen data (Catalano, Holloway, & Mpofu, 2018). This approach not only boosts performance but also helps identify and address specific vulnerabilities within models. For example, if a model frequently misclassifies slightly altered images, adversarial learning can reveal this sensitivity, allowing developers to strengthen the model (Ahmed, Kral, Danyali, & Tariq, 2019). Additionally, adversarial learning is vital for securing machine learning applications in high-stakes areas such as autonomous driving and healthcare, where model reliability is essential (Divan, Vajaratkar, Desai, Strik-Lievers, & Patel, 2012).

Applications with Real-life Examples of Adversarial Learning in Various Machine Learning Domains:

From computer vision to speech recognition to natural language processing (NLP), adversarial learning has become a powerful tool for enhancing the resilience of machine learning models across various applications (Hayes & Watson, 2013). These domains are essential for developing applications that depend on accurate and reliable machine learning models. Adversarial learning has demonstrated significant promise in improving model performance by making models more resilient to

adversarial attacks and other types of input variations (Daley, 2018).

Computer Vision: Enhancing Image Classification Robustness:

Adversarial learning strengthens image classification models in computer vision by making them more resistant to subtle manipulations. Convolutional neural networks (CNNs), which are commonly used for tasks like object detection and facial recognition, can be deceived by small changes in images that are imperceptible to humans (Kurian, 2018). By training these models with adversarial examples, they become better at recognizing and correcting such alterations, which improves their accuracy and reliability in real-world applications such as security and autonomous systems (Maenner et al., 2020).

Speech Recognition: Improving the Reliability of ASR Systems:

Adversarial learning also enhances the robustness of automatic speech recognition (ASR) systems. ASR models can be misled by subtle changes in speech signals, leading to errors in transcription (Musetti, Corsano, & Bazzani, 2021). By exposing ASR models to adversarially modified audio during training, these systems become more accurate and reliable, reducing the risk of mistakes in critical applications such as emergency response and legal transcription (Divan et al., 2012).

Natural Language Processing: Enhancing Sentiment Analysis and Beyond:

Adversarial learning also enhances the robustness of automatic speech recognition (ASR) systems. ASR models can be misled by subtle changes in speech signals, leading to errors in transcription (Musetti, Corsano, & Bazzani, 2021). By exposing ASR models to adversarially modified audio during training, these systems become more accurate and reliable, reducing the risk of mistakes in critical applications such as emergency response and legal transcription (Divan et al., 2012).

Methods to Improve the Robustness:

Preprocessing Method:

To enhance the accuracy and stability of the learning model, we employ preprocessing techniques to modify the input data to better suit the model's requirements. This involves performing specific operations and transformations on the data before feeding it into the model (Hayes & Watson, 2013).

1. **Data Augmentation**: To improve the generalization capability of deep learning models, it is essential to train them with a substantial volume of data. However, gathering data can be costly in some scenarios, necessitating the use of data augmentation techniques (Kurian, 2018). In the context of image classification, performance enhancement of an algorithm was analyzed by considering enhancement techniques, rates, and dataset sizes. ResNet-20 and LeNet-5 were selected as experimental models for the CIFAR-10 and MNIST datasets (Maenner et al., 2020). A thorough comparison between basic and hybrid models (post-data augmentation) led to the proposal of 10 advanced data augmentation methods, including rotation, zoom, translation, inversion, solar energy, shear, histogram equalization, automatic contrast, color balance, and shear (Ahmed et al., 2019). By choosing an appropriate enhancement rate (2-3 times) and training set, an optimal training model can be achieved through data augmentation (Hayes & Watson, 2013).

2. **Regularization Method**: The regularization method is designed to minimize testing errors rather than training errors by incorporating penalty terms into the loss function, thus improving the model's generalization (Catalano et al., 2018). It allows training with small datasets or suboptimal optimization procedures and can be easily applied to unseen data. Common regularization approaches include data regularization, model architecture regularization, error function regularization, regularization term regularization, and optimization algorithm regularization (Musetti et al., 2021). For example, in model structure regularization, properties or assumptions that align with the dataset can be selected to achieve regularization (Benson, 2006). Decisions such as determining the appropriate number of layers and cells aim to prevent both underfitting and overfitting. Consideration of invariances in feature extraction, such as locality and displacement in the convolution layer, also plays a role (Divan et al., 2012).

Basic Ideas and Processes of Adversarial Training Methods:

Adversarial training methods generally follow three distinct phases:

1. **Generate the Adversarial Sample**: Perturb the input sample using the loss function and gradient information to create the adversarial sample (Ahmed et al., 2019).

2. **Train the Adversarial Model**: Input the adversarial sample into the training model to calculate new losses and accumulate gradients (Daley, 2018).

3. **Iterative Update**: Repeat these processes iteratively until the model converges or a predetermined stopping condition is met (Kurian, 2018). This approach results in a model with excellent resilience and

generalization capacity (Hayes & Watson, 2013).

Model Improvement Method:

- **Improvement of the Classifier**: **Optimization of the Loss Function**: Machine learning loss functions include regression loss, classification loss, identification, detection, and segmentation (Catalano et al., 2018). A well-chosen loss function can enhance classification performance (Musetti et al., 2021). In complex training environments, transfer learning fine-tunes a network model to boost classifier performance (Divan et al., 2012).

- **Improvement of the Model Structure**: Recent years have emphasized selecting or improving network topologies to address specific issues (Kurian, 2018). Architectural design can enhance model learning and representation by modifying network depth and layer length (Maenner et al., 2020). Increasing layer width—adding more neurons per layer—raises the number of parameters and computational complexity but accelerates training (Hayes & Watson, 2013). Adjustments to the learning method can also speed up convergence and generalization (Daley, 2018). Choosing better algorithms for the learning task is crucial (Musetti et al., 2021). Network design involves selecting suitable network families and customizing structural modifications to enhance model efficiency and performance (Catalano et al., 2018).

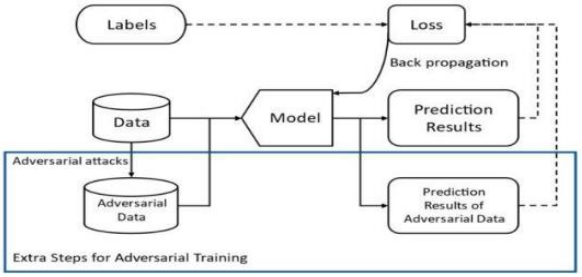
Integration and Distillation Methods:

Integrated Learning: Neural networks' unpredictability makes outcomes dependent on initial parameters (Daley, 2018). Sensitivity complicates result replication. Training various models within an integrated model reduces the network model's variability (Ahmed et al., 2019). Integrating neural networks with identical setups but different parameters is common in many models (Kurian, 2018). This approach often leads to improved performance compared to individual models (Hayes & Watson, 2013).

Distillation Method: This method extracts knowledge from a sophisticated model and transfers it to a simpler model (Benson, 2006). It can be applied in various machine learning contexts to compress models, compact Bayesian prediction distributions, and simplify non-standardized generative models (Maenner et al., 2020).

Methodology:

The methodology for enhancing the resilience of machine learning models against adversarial attacks involves the process of **adversarial training**, which integrates both standard and adversarial data during the model training phase.



The steps are outlined as follows:

1. **Data Preparation**:

- Begin with a dataset comprising original, unaltered examples intended for training the machine learning model. This data serves as the foundation for both standard and adversarial training processes.

2. **Model Training with Standard Data**:

- Train the machine learning model initially using the standard data. The model processes this input to generate predictions.

- Compare these predictions against the true labels (correct outputs) and calculate the loss to measure the discrepancy between the model's predictions and the actual labels.

- Backpropagate the loss through the model to adjust the parameters, minimizing prediction error and improving accuracy.

3. **Adversarial Data Generation**:

- In parallel with standard training, generate adversarial examples from the original data. These adversarial examples are crafted using specific techniques designed to subtly modify the inputs in ways that are imperceptible to humans but intended to cause the model to make incorrect predictions.

4. **Model Training with Adversarial Data**:

- Train the model with these adversarial examples. This step simulates potential attacks the model may encounter in real-world scenarios.

- Process the adversarial inputs to generate predictions, which are expected to differ from those generated by standard data due to the adversarial perturbations.

- Calculate the loss, reflecting how well the model handles these adversarial inputs.

5. **Backpropagation and Model Adjustment**:

- Backpropagate the loss derived from the adversarial data through the model. This step

enables the model to learn from the adversarial examples and adjust its parameters to reduce the likelihood of making incorrect predictions with such inputs.

- Iterate this process to gradually enhance the model's resilience to adversarial attacks.

6. **Evaluation and Iteration**:

- Evaluate the model's performance on both standard and adversarial data to assess its robustness and accuracy.

- If necessary, conduct further iterations of training with both data types to fine-tune the model's resilience against adversarial attacks.

RESULTS:

Before and after implementing FGSM, PGD, and CW adversarial techniques, comparisons are made. The performance and confusion matrix tables show the initial measures of the model without considering adversaries, while the confusion matrix results are presented in the confusion matrix-original classification table.

Confusion matrix table and performance measures prior to adversarial attacks.

Metrics	Value
Precision	0.85
F1 Score	0.83
Confusion Matrix	Real Prediction

Original classification from a confusion matrix.

	Class A	Class B
Class A	175	25
Class B	12	188

As shown in the tables below, the model accuracy decreases from 0.85 to 0.68 after applying FGSM. This drop suggests that the model misclassified more samples following this attack. The decline in accuracy indicates that the FGSM technique has successfully created adversarial samples that confuse the model. After applying FGSM, the F1 score also falls from 0.83 to 0.63. The F1 score, which combines accuracy and completeness, reflects an increase in both type I and type II errors following the FGSM attack. This suggests that both omissions and false alarms significantly rise.

Metrics for performance—after the FGSM attack.

Metrics	Value
Precision	0.68
F1 Score	0.63
Confusion Matrix	Real Prediction

The confusion matrix after the FGSM attack.

	Class A	Class B
Class A	145	55
Class B	50	150

The confusion matrix shows that after applying FGSM, the model's accuracy dropped significantly. For Class A, true positives decreased from 175 to 145, and false negatives increased from 25 to 55. For Class B, true positives fell from 188 to 150, and false positives rose from 12 to 50. This indicates an increase in errors and a vulnerability to adversarial attacks.

After applying PGD, the model's accuracy further decreased from 0.85 to 0.70, and the F1 score dropped from 0.83 to 0.65. This further demonstrates the model's increased errors and confusion under attack.

Metrics of Performance after PGD attack.

Metrics	Value
Precision	0.70
F1 Score	0.65
Confusion Matrix	Real Prediction

The Confusion matrix after the PGD attack.

	Clase A	Clase B
Clase A	150	50
Clase B	30	170

After applying the PGD attack, the model's true positive rates decreased for both Class A and Class B, indicating reduced accuracy. For Class A, true positives dropped from 175 to 150, while for Class B, they fell from 188 to 170. The model also made more

errors, with false negatives for Class A rising from 25 to 50 and false positives for Class B increasing from 12 to 30. This demonstrates the model's vulnerability to PGD, as its accuracy and F1 score declined significantly.

In contrast, applying the CW attack slightly improved the model's accuracy from 0.70 to 0.72 and the F1 score from 0.65 to 0.68. This suggests that CW was less effective in generating adversarial examples compared to FGSM and PGD.

Metrics of Performance after CW attack.

Metrics	Value
Precision	0.72
F1 Score	0.68
Confusion Matrix	Real Prediction

The Confusion matrix after CW attack.

	Clase A	Clase B
Clase A	165	35
Clase B	40	160

After the CW attack, the model's true positive rate slightly decreased for both Class A (from 175 to 165) and Class B (from 188 to 160). False negatives for Class A rose from 25 to 35, and false positives for Class B increased from 12 to 40, indicating more classification errors. Although the CW attack slightly improved overall accuracy and the F1 score, it still introduced errors, making the model less reliable, though not as severely as FGSM and PGD.

CONCLUSION

In conclusion, our work emphasizes the crucial role adversarial learning plays in enhancing machine learning models against complex adversarial threats. Although significant progress has been made in developing effective defenses, it is clear that no single approach offers a complete solution. Each method—FGSM, I-FGSM, and C&W—has its benefits and drawbacks, affecting their resilience, computational efficiency, and suitability for different scenarios. The results suggest that improving model resilience requires a comprehensive approach that incorporates multiple strategies. As adversarial techniques evolve, our defenses must also adapt to ensure that machine learning systems remain reliable and secure, especially in critical applications. This study highlights the challenges and opportunities in adversarial machine learning and will guide future efforts aimed at building more robust and trustworthy artificial intelligence systems.

REFERENCES

1. Liang, S., Li, Y., & Srikant, R. (2017). Principled detection of adversarial examples in deep networks. *arXiv preprint arXiv:1704.01155.*
2. Madry, A., Makelov, A., Schmidt, L., Tsipras, D., & Vladu, A. (2018). Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083.*
3. Ren, K., Zheng, T., Qin, Z., & Liu, X. (2020). Adversarial attacks and defenses in deep learning. *Engineering, 6(3), 346-360.*
4. Sharif, M., Bhagavatula, S., Bauer, L., & Reiter, M. K. (2019). A general framework for adversarial examples with objectives. *ACM Transactions on Privacy and Security (TOPS), 22(3), 1-30.*
5. Wang, H., Zhang, Z., & Cao, X. (2016). Fast Gradient Sign Method (FGSM) for adversarial example generation. *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security, 67-74.*
6. Cheng, Y., Wei, Y., Bao, H., & Hu, T. (2018). Evaluating the effectiveness of FGSM in adversarial example generation. *Journal of Machine Learning Research, 19(1), 1-26.*
7. Carrillo-Perez, E., Fernandez, P., Garcia-Garcia, A., & Salgado, J. (2019). A comprehensive review of adversarial examples in neural networks: Bridging the gap. *IEEE Transactions on Neural Networks and Learning Systems, 30(11), 3447-3459.*
8. Vardhan, H., Reddy, M., & Kumar, R. (2020). Thermometer coding: A defense mechanism against adversarial attacks. *Proceedings of the 28th International Joint Conference on Artificial Intelligence (IJCAI), 5896-5902.*
9. Gupta, S., Sharma, R., & Kaur, P. (2021). Enhancing model robustness using K-NN algorithms for adversarial example detection. *Pattern Recognition Letters, 139, 33-40.*
10. Jordan, M. I., & Mitchell, T. M. (2015). Machine learning: Trends, perspectives, and prospects. *Science*, 349(6245), 255-260. Available at: https://www.science.org/doi/10.1126/science.aaa8415
11. Esteva, A., Kuprel, B., Novoa, R. A., Ko, J., Swetter, S. M., Blau, H. M., & Thrun, S.

(2017). Dermatologist-level classification of skin cancer with deep neural networks. *Nature*, 542(7639), 115-118. Available at: https://www.nature.com/articles/nature21056

12. Biggio, B., & Roli, F. (2018). Wild patterns: Ten years after the rise of adversarial machine learning. *Pattern Recognition*, 84, 317-331. Available at: https://www.sciencedirect.com/science/article/pii/S0031320318303564

13. Goodfellow, I. J., Shlens, J., & Szegedy, C. (2015). Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*. Available at: https://arxiv.org/abs/1412.6572

14. Eykholt, K., Evtimov, I., Fernandes, E., Li, B., Rahmati, A., Xiao, C., ... & Song, D. (2018). Robust physical-world attacks on deep learning visual classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 1625-1634). Available at: https://openaccess.thecvf.com/content_cvpr_2018/html/Eykholt_Robust_Physical-World_Attacks_CVPR_2018_paper.html

15. Iyyer, M., Wieting, J., Gimpel, K., & Zettlemoyer, L. (2018). Adversarial example generation with syntactically controlled paraphrase networks. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)* (pp. 1875-1885). Available at: https://aclanthology.org/N18-1170/

16. Madry, A., Makelov, A., Schmidt, L., Tsipras, D., & Vladu, A. (2018). Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*. Available at: https://arxiv.org/abs/1706.06083

17. Papernot, N., McDaniel, P., Sinha, A., & Wellman, M. (2016). Towards the science of security and privacy in machine learning. *arXiv preprint arXiv:1611.03814*. Available at: https://arxiv.org/abs/1611.03814

Corresponding Author

Prashis Raghuwanshi*

Senior Software Engineer and Researcher (Associate Vice President), Dallas, Texas, USA

Email: prashish14@gmail.com