FALCONNet: A Multi-Defense Approach for Securing Few-Shot Learning Against Adversarial Attacks

Dr. Sivakumar P
School of Computer Engineering
Vellore Institute of Technology, Chennai, India
sivakumar.p@vit.ac.in

Aneek Kumar Saha
School of Electronics Engineering
Vellore Institute of Technology, Chennai, India
aneekkumar.saha2021@vitstudent.ac.in

Jervis Francis Lopes
School of Electronics Engineering
Vellore Institute of Technology, Chennai, India
jervisfrancis.lopes2021@vitstudent.ac.in

Shuban M S
School of Electronics Engineering
Vellore Institute of Technology, Chennai, India
shuban.ms2021@vitstudent.ac.in

***Abstract*—This paper presents a comparative analysis of few-shot learning models, focusing on the evaluation of Siamese and Prototypical Network architectures on the Omniglot dataset. Furthermore, we introduce and evaluate our novel model, FALCONNET (Few-Shot Adversarial Learning and Classification Network). We evaluate the performance of all the models against adversarial attacks, specifically PGD (Projected Gradient Descent) and FGSM (Fast Gradient Sign Method), and explore defense mechanisms to enhance model resilience. The Prototypical Network was configured for a 60-way 5-shot 5-query task, as was FalconNet. The Siamese Network followed a pairwise 5-shot 5-query setup. Data augmentation techniques were applied to improve generalization across all models. To counter adversarial impact, we implemented adversarial training, defensive distillation, and their combination. Experimental results demonstrate that FalconNet, incorporating these defense strategies, significantly enhances model accuracy and stability under adversarial conditions, outperforming both the standard Siamese and Prototypical Networks.**

***Index Terms*—Few-shot learning, Siamese Networks, Prototypical Networks, Omniglot dataset, adversarial attacks, PGD, FGSM, adversarial training, defensive distillation**

I. Introduction

Few-shot learning [1] has emerged as a critical research direction in machine learning, addressing the challenge of learning from limited labeled examples [2]. This technique is particularly useful in real-world applications where data is very limited, such as medical diagnosis, rare object recognition and security systems [3] like signature matching or facial recognition where data is often limited. Among various approaches, *Siamese Networks* [4] and *Prototypical Networks* [5] have demonstrated remarkable success in learning robust representations from minimal data. However, recent studies reveal that these models are highly vulnerable to *adversarial attacks* [6] [7], where perturbations in the images can significantly degrade their performance posing a serious limitation for real-world applications and uses.

To evaluate this vulnerability, we use the *Omniglot dataset* [8], a widely adopted benchmark containing 1,623 handwritten characters across multiple alphabets. Following standard few-shot learning protocols, we configure the *Prototypical Network* for a **60-way 5-shot 5-query task** and the *Siamese Network* for **pairwise 5-shot 5-query classification**. Additionally, we employ *data augmentation techniques* [9] [10] such as 90°, 180°, and 270° image rotations and Horizontal and Vertical image flips to enhance model generalization.

Our work makes three key contributions to robust few-shot learning:

- **Adversarial Vulnerability Analysis** – We analyze the impact of *Projected Gradient Descent (PGD)* [11] [12] and *Fast Gradient Sign Method (FGSM)* [13] [14] attacks on both network architectures, demonstrating their susceptibility to adversarial perturbations.
- **Defensive Strategies** – We evaluate *adversarial training* [15] [16], *defensive distillation*, and their combination to mitigate adversarial effects, showing that these approaches significantly improve model robustness.
- **Empirical Validation of Trade-offs** – We investigate the balance between standard accuracy and adversarial robustness [17], providing insights into the challenges of defending few-shot learning models.

Our builds upon standard few-shot evaluation protocols while increasing the number of classes making it increasingly challenging for the models to learn and also incorporating adversarial defense strategies. Our findings help us understand the drawbacks and advantages between *few-shot learning* and *adversarial robustness research*, offering practical insights for developing more resilient and trustworthy few-shot learning systems.

II. Literature Review

A. Foundations of Few-Shot Learning

Few-shot learning (FSL) addresses the challenge of learning from limited data by enabling models to generalize with minimal supervision. Unlike traditional deep learning, which requires extensive datasets, FSL methods leverage meta-learning, transfer learning, and metric-based learning to improve adaptability and stability [18]. Recent studies categorize FSL into three main approaches: optimization-based, metric-based, and memory-based methods. Metric-based models, such as Prototypical Networks and Siamese Networks, learn feature embeddings that facilitate classification with minimal examples. Additionally, data augmentation techniques, including transformations and synthetic data generation, enhance generalization in low-data regimes.

Despite progress, challenges remain in improving FSL robustness, scalability, and generalization, particularly under adversarial conditions. Ongoing research explores new strategies to strengthen FSL models for real-world deployment [19].

B. Adversarial Robustness in Classification Models

Deep learning models are highly effective but remain susceptible to adversarial perturbations—small, imperceptible modifications that can significantly degrade model performance [20]. Research has categorized adversarial attacks and defenses into three primary strategies: adversarial (re)training, gradient-based regularization, and certified robustness methods [21]. Among these, adversarial training, which incorporates adversarial examples during model training, is a widely adopted defense but often comes with trade-offs between robustness and generalization.

In the context of few-shot learning, existing approaches assume clean, well-labeled data. However, real-world datasets often contain outliers, making robust few-shot learning (RFSL) a crucial research direction. Recent studies introduce novel techniques to mitigate representation and label outliers, ensuring better model reliability in practical settings. These findings highlight the need for tailored adversarial defenses for few-shot learning architectures [22].

C. Defense Mechanisms and Their Limitations

Adversarial defenses primarily focus on mitigating vulnerabilities in deep learning models. Adversarial training improves performance by removing small perturbations in hidden representations, yet it remains computationally expensive and struggles to generalize against unseen attacks [23]. Defensive distillation reduces attack success by modifying output distributions, but adaptive adversaries can still bypass it [24]. Another approach stabilizes neuron sensitivities to enhance robustness, though its effectiveness varies across architectures [25]. While hybrid strategies combining these defenses show promise, they often introduce accuracy trade-offs and increased computational costs.

D. Critical Gaps in Current Research

Despite advances in few-shot learning and adversarial robustness, several key gaps remain:

- **Adversarial Robustness in Few-Shot Models**: Few-shot learning systems have received limited attention in adversarial contexts. Our work focuses on enhancing these models to handle adversarial attacks, crucial for security and medical applications.
- **Scalability Limitations**: Existing research often limits few-shot learning to small classification tasks like 20-way. Our approach expands this to 60-way through advanced data augmentation, addressing scalability challenges.
- **Hybrid Defense Mechanisms**: Few studies combine defenses like adversarial training and defensive distillation in few-shot models. We show that these hybrid defenses improve robustness without compromising performance.
- **Real-World Application**: While few-shot learning models often use standard datasets, we aim to apply these models to security-sensitive and medical fields, where few samples per class are common.

III. Methodology

In this section, we detail the methodology employed in our experiments, including the dataset and pre-processing steps, model architectures, training procedures, adversarial attack implementations, and defense mechanisms. The experiments were conducted using Python, with TensorFlow Keras and PyTorch for model development. The computational resources consisted of GPUs provided by Google Colab and Kaggle.

A. Dataset and Preprocessing

We utilized the Omniglot dataset, a standard benchmark for few-shot learning, which includes a large number of characters from various alphabets. The dataset was divided into support and query sets, as well as training and testing sets. An example from the Omniglot dataset's is shown in Figure 1 it consists of a single alphabet written with different strokes from a particular language.

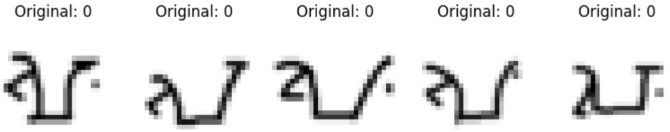

Fig. 1. Sample of Omniglot Alphabet Characters

Preprocessing involved normalizing and resizing the images to specific dimensions required by the model architectures. Specifically, the images were resized to 105×105 pixels for the Siamese network and 28×28 pixels for the Prototypical network.

Data augmentation techniques were applied during training. These techniques included rotations of 90, 180, and 270 degrees, as well as horizontal and vertical flips.

B. Model Architectures

We implemented two types of models: a Siamese network and a Prototypical network, each designed for few-shot learning tasks.

1) Siamese Network: The Siamese network consists of two identical sub-networks that share weights. Each sub-network processes one image from a pair, and the outputs are compared to determine if the images belong to the same class.

The base network, which forms the shared sub-network, comprises the following layers:

- Input layer: Accepts images of size 105×105 pixels.
- Convolutional layers: Four convolutional layers with 32, 64, 128, and 256 filters, respectively, each with a kernel size of 3×3, ReLU activation, and 'same' padding.
- Batch normalization: Batch normalization layers after each convolutional layer.
- Max pooling: Max pooling layers with a pool size of 2×2 after each batch normalization.
- Dropout: Dropout layers with a rate of 0.25 after each max pooling.
- Flatten layer: Flattens the output of the last convolutional layer.
- Dense layers: Two dense layers with 512 and 256 units, respectively. The first dense layer uses ReLU activation, and the second uses sigmoid activation to produce the embedding.

The similarity between the embeddings of the two input images is computed using the Euclidean distance. This distance is then passed through a sigmoid activation function to produce a binary output, indicating whether the images belong to the same class or different classes.

2) Prototypical Network: The Prototypical network embeds input images into a feature space and computes class prototypes by averaging the embedded vectors of support images for each class. The classification of query images is based on the Euclidean distance to the class prototypes.

The encoder network, which embeds the input images, comprises the following layers:

- Convolutional blocks: Four convolutional blocks, each consisting of a convolutional layer with 64 filters, kernel size 3×3, and padding, followed by batch normalization, ReLU activation, and max pooling with a pool size of 2×2.
- Flatten layer: Flattens the output of the last convolutional block.

The Prototypical network uses Euclidean distance as a metric of similarity to calculate the distance between the embedded query images and the class prototypes. The logarithmic of Softmax function is then used to compute the probabilities for each class, and the final classification is based on the class with the highest probability.

C. Training Procedure

The models were trained using the Adam optimization algorithm with a batch size of 32. The training continued for 15 epochs, and convergence was determined when there was no significant increase in validation accuracy, indicating a plateau.

D. Adversarial Attack Implementation

We implemented two types of adversarial attacks: Projected Gradient Descent (PGD) and Fast Gradient Sign Method (FGSM).

For the PGD attack, we used the following parameters: epsilon = 0.5, alpha = 0.5, and 10 iterations.

For the FGSM attack, we used an epsilon value of 0.5.

Both attacks were implemented as untargeted attacks, aiming to degrade the model's performance without targeting specific misclassifications.

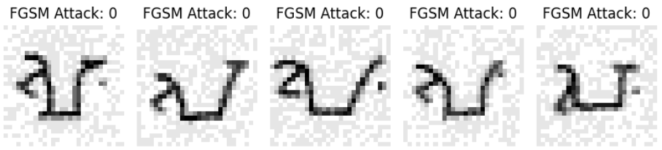

Fig. 2. Omniglot Character Attacked with FGSM

Figure 2 shows an Omniglot character that has been attacked using the Fast Gradient Sign Method (FGSM).

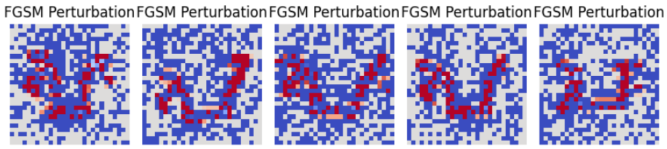

Fig. 3. Perturbation Introduced by FGSM Attack

Figure 3 visualizes the perturbation introduced by the FGSM attack on the Omniglot character.

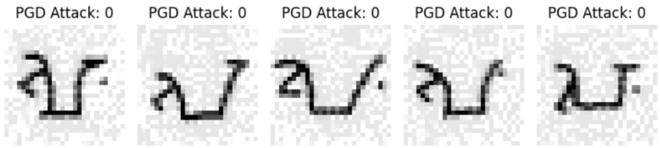

Fig. 4. Omniglot Character Attacked with PGD

Figure 4 shows an Omniglot character that has been attacked using the Projected Gradient Descent (PGD) method.

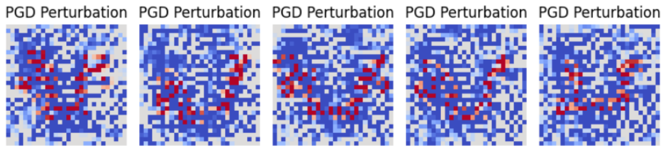

Fig. 5. Perturbation Introduced by PGD Attack

Figure 5 visualizes the perturbation introduced by the PGD attack on the Omniglot character.

The perturbation visualizations use color coding to represent the magnitude and direction of pixel changes:

- **Red**: Indicates a positive change in pixel value (increased intensity).
- **Blue**: Indicates a negative change in pixel value (decreased intensity).
- **Orange**: Represents intermediate changes in pixel values.

The intensity of the colors corresponds to the magnitude of the perturbation. Brighter colors signify larger pixel value changes, while darker colors indicate smaller changes. This visualization aids in understanding the nature and extent of the adversarial perturbations introduced by each attack.

E. Defense Mechanisms

We explored two defense mechanisms: Adversarial Training (AT) and Defensive Distillation (DD), mentioned below is how these two defense mechanisms work.

1) Adversarial Training (AT): Adversarial training involves adding the training dataset with adversarial examples, forcing the model to learn with such perturbations present in the training stage itself. We generated adversarial examples using the same attack methods (FGSM and PGD) as in the evaluation 50% from each, but with slightly weaker perturbations. This was achieved by reducing the epsilon parameter for both FGSM and PGD during the training phase from 0.5 to 0.3.

2) Defensive Distillation (DD): Defensive distillation is a technique that transfers the knowledge from a teacher model to a student model, making the student model less susceptible to adversarial attacks. This is achieved by training the student model on the soft labels produced by the teacher model, which are less sensitive to small input perturbations.

We first trained a teacher model on the clean dataset. Then, we generated soft labels by passing the training data through the teacher model with a temperature parameter of 5.0. The temperature parameter controls the smoothness of the soft labels; a higher temperature results in smoother or softer labels.

3) Combined Defense Mechanism: We considered the benefits of both adversarial training and defensive distillation, we implemented a combined defense mechanism. This involved training the model in two stages.

1) **Adversarial Training Stage**: We first trained the model using adversarial training, as described above.
2) **Defensive Distillation Stage**: After the adversarial training stage, we used the adversarially trained model as the teacher model for defensive distillation. We generated soft labels using this teacher model and trained a student model on these soft labels.

This staged approach allowed us to first improve the model's performance against adversarial examples through adversarial training and also smooth the model's decision boundaries using defensive distillation.

IV. Results and Discussion

A. Analysis of Network Performance

As shown in Table II, the Prototypical network generally achieved higher accuracy than the Siamese network. However, it was more susceptible to PGD attacks. Specifically, the Prototypical network's accuracy dropped from 0.96 to 0.39 under PGD, with a corresponding loss increase from 0.11 to 2.31 (Table I). The Siamese network showed more stable performance.

B. Impact of Adversarial Attacks

PGD attacks significantly reduced accuracy for both networks (Table II). Increased loss values (Table I) confirm the attack's disruptive impact, altering model representations. As visualized in Figure 6, the PGD attack leads to a notable decrease in accuracy across all models.

C. Effectiveness of Defense Mechanisms

Adversarial training (AT) improved robustness, particularly in training (Table I). For example, the Siamese network's PGD accuracy improved from 0.50 to 0.74 with AT. Defensive distillation (DD) had mixed results. The AT+DD combination generally performed best. The effectiveness of AT and the mixed results of DD can also be observed in Figure 6.

D. Training vs. Test Performance

The Prototypical network showed a larger training-test gap, indicating overfitting (Table I). The Siamese network demonstrated better generalization.

V. Conclusion

In this study, we evaluated the robustness of Siamese and Prototypical networks under adversarial attacks, specifically PGD and FGSM. While the Prototypical network demonstrated superior accuracy in clean data scenarios, it exhibited a significant vulnerability to PGD attacks, experiencing a dramatic drop in performance. The Siamese network maintained a more stable performance under adversarial perturbations. We explored the effectiveness of adversarial training (AT) and defensive distillation (DD) as defense mechanisms. The combination of AT and DD generally provided the best overall performance, highlighting the joint benefits of these defense strategies. The Prototypical network showed a larger gap between training and test performance, suggesting potential overfitting, while the Siamese network exhibited better generalization. These findings display the importance of considering adversarial defenses in few-shot learning models.

VI. Future Scope

This study opens several aspects for future research. Firstly, exploring more advanced defense mechanisms, such as input transformation techniques or certified defenses, could further enhance the performance of these networks. Investigating the impact of different attack parameters and exploring adaptive attack strategies would provide a better understanding of any model's weak points. Secondly, extending this analysis

TABLE I
PERFORMANCE COMPARISON OF SIAMESE AND PROTOTYPICAL NETWORKS UNDER ADVERSARIAL ATTACKS

Network	Eval. Type	Model	Accuracy			Loss		
			No Attack	PGD Attack	FGSM Attack	No Attack	PGD Attack	FGSM Attack
Siamese	Test	Base Model	0.82	0.50	0.50	5.78	6.90	6.50
		AT Model	0.79	0.63	0.68	5.95	5.20	5.15
		DD Model	0.80	0.48	0.58	5.82	7.24	6.30
		AT + DD Model	0.77	0.68	0.69	6.05	4.80	5.10
	Train	Base Model	0.84	0.53	0.56	5.46	6.80	6.20
		AT Model	0.82	0.74	0.74	5.60	4.90	4.70
		DD Model	0.83	0.55	0.63	5.50	6.70	6.17
		AT + DD Model	0.81	0.73	0.75	5.70	4.50	4.80
Prototypical	Test	Base Model	0.96	0.39	0.45	0.11	2.31	1.74
		AT Model	0.93	0.64	0.72	0.27	1.22	1.04
		DD Model	0.94	0.32	0.57	0.15	2.32	1.71
		AT + DD Model	0.92	0.73	0.71	0.25	1.02	1.15
	Train	Base Model	0.98	0.96	0.97	0.08	0.30	0.25
		AT Model	0.97	0.83	0.65	0.09	0.52	0.45
		DD Model	0.98	0.62	0.74	0.07	0.52	0.55
		AT + DD Model	0.97	0.83	0.89	0.10	0.68	0.63

Note: AT = Adversarial Training, DD = Defensive Distillation, AT + DD = Combination of Adversarial Training and Defensive Distillation.

TABLE II
KEY PERFORMANCE COMPARISON OF SIAMESE AND PROTOTYPICAL NETWORKS UNDER ADVERSARIAL ATTACKS (TEST SET)

Network	Model	No Attack Accuracy	PGD Attack Accuracy	FGSM Attack Accuracy
Siamese	Base Model	0.82	0.50	0.50
	AT Model	0.79	0.63	0.68
	DD Model	0.80	0.48	0.58
	AT + DD Model	0.77	0.68	0.69
Prototypical	Base Model	0.96	0.39	0.45
	AT Model	0.93	0.64	0.72
	DD Model	0.94	0.32	0.57
	AT + DD Model	0.92	0.73	0.71

Note: AT = Adversarial Training, DD = Defensive Distillation, AT + DD = Combination of Adversarial Training and Defensive Distillation.

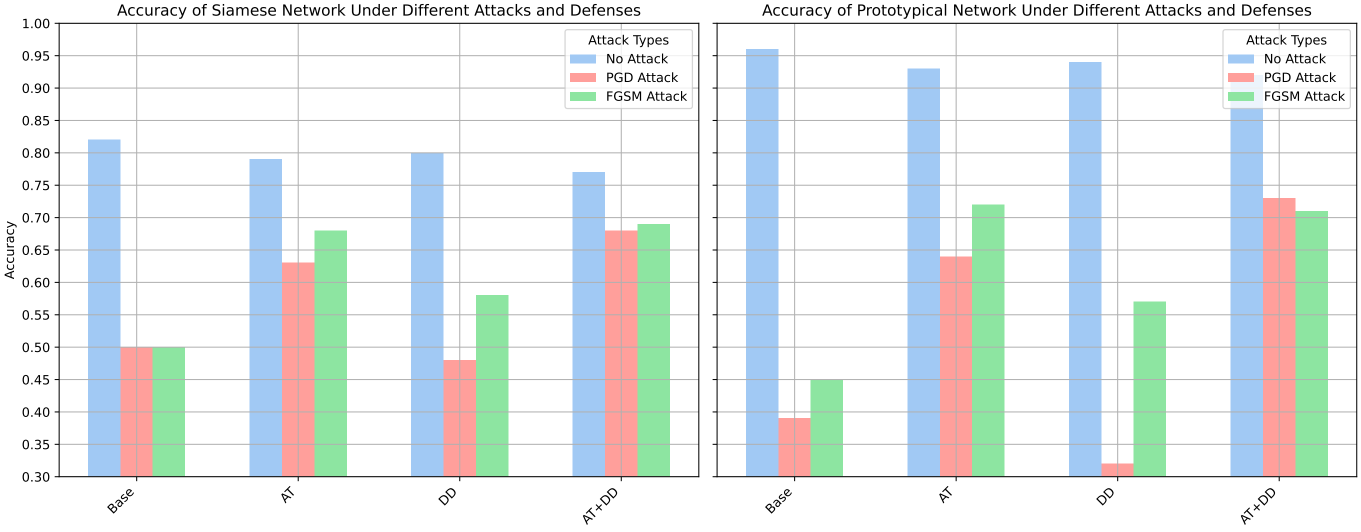

Fig. 6. Accuracy of Siamese and Prototypical Networks Under Different Attacks and Defenses

to larger and more complex datasets, as well as real-world applications where data is found in all kinds of format and is not available in abundance. Finally, investigating the interpretability of these models under adversarial attacks and developing methods to visualize and understand the internal representations that are affected would provide valuable insights into model behavior and help us develop more secure architectures.

References

[1] Y. Wang, Q. Yao, J. T. Kwok, and L. M. Ni, “Generalizing from a Few Examples,” ACM Computing Surveys, vol. 53, no. 3, pp. 1–34, Jul. 2020, doi: https://doi.org/10.1145/3386252.

[2] Archit Parnami and M. Lee, “Learning from Few Examples: A Summary of Approaches to Few-Shot Learning,” arXiv (Cornell University), Mar. 2022, doi: https://doi.org/10.48550/arxiv.2203.04291.

[3] T. Althiyabi, I. Ahmad, and M. O. Alassafi, “Enhancing IoT Security: A Few-Shot Learning Approach for Intrusion Detection,” Mathematics, vol. 12, no. 7, p. 1055, Mar. 2024, doi: https://doi.org/10.3390/math12071055.

[4] D. Chicco, “Siamese Neural Networks: An Overview,” Methods in Molecular Biology, pp. 73–94, Aug. 2020, doi: https://doi.org/10.1007/978-1-0716-0826-5_3.

[5] J. Snell, K. Swersky, and R. Zemel, “Prototypical Networks for Few-shot Learning,” Neural Information Processing Systems, 2017. https://proceedings.neurips.cc/paper_files/paper/2017/hash/cb8da6767461f2812ae4290eac7cbc42-Abstract.html

[6] A. Madry, Aleksandar Makelov, L. Schmidt, Dimitris Tsipras, and A. Vladu, “Towards Deep Learning Models Resistant to Adversarial Attacks,” Jun. 2017, doi: https://doi.org/10.48550/arxiv.1706.06083.

[7] A. Chakraborty, M. Alam, V. Dey, A. Chattopadhyay, and D. Mukhopadhyay, “A survey on adversarial attacks and defences,” CAAI Transactions on Intelligence Technology, vol. 6, no. 1, pp. 25–45, Mar. 2021, doi: https://doi.org/10.1049/cit2.12028.

[8] B. M. Lake, R. Salakhutdinov, and J. B. Tenenbaum, “The Omniglot challenge: a 3-year progress report,” Current Opinion in Behavioral Sciences, vol. 29, pp. 97–104, Oct. 2019, doi: https://doi.org/10.1016/j.cobeha.2019.04.007.

[9] C. Shorten and T. M. Khoshgoftaar, “A survey on Image Data Augmentation for Deep Learning,” Journal of Big Data, vol. 6, no. 1, Jul. 2019, doi: https://doi.org/10.1186/s40537-019-0197-0.

[10] L. Taylor and G. Nitschke, “Improving Deep Learning with Generic Data Augmentation,” IEEE Xplore, Nov. 01, 2018. https://ieeexplore.ieee.org/abstract/document/8628742

[11] Y. Deng and L. J. Karam, “Universal Adversarial Attack Via Enhanced Projected Gradient Descent,” Sep. 2020, doi: https://doi.org/10.1109/icip40778.2020.9191288.

[12] O. Bryniarski, N. Hingun, P. Pachuca, V. Wang, and N. Carlini, “Evading Adversarial Example Detection Defenses with Orthogonal Projected Gradient Descent,” arXiv (Cornell University), Jan. 2021, doi: https://doi.org/10.48550/arxiv.2106.15023.

[13] A. Naqvi, M. Shabaz, Muhammad Attique Khan, and Syeda Iqra Hassan, “Adversarial Attacks on Visual Objects Using the Fast Gradient Sign Method,” Journal of grid computing, vol. 21, no. 4, Sep. 2023, doi: https://doi.org/10.1007/s10723-023-09684-9.

[14] Sigit Wibawa, “Analysis of Adversarial Attacks on AI-based With Fast Gradient Sign Method,” International Journal of Engineering Continuity, vol. 2, no. 2, pp. 72–79, Aug. 2023, doi: https://doi.org/10.58291/ijec.v2i2.120.

[15] T. Bai, J. Luo, J. Zhao, B. Wen, and Q. Wang, “Recent Advances in Adversarial Training for Adversarial Robustness,” arXiv (Cornell University), Feb. 2021, doi: https://doi.org/10.48550/arxiv.2102.01356.

[16] A. Shafahi, M. Najibi, Z. Xu, J. Dickerson, L. S. Davis, and T. Goldstein, “Universal Adversarial Training,” Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, no. 04, pp. 5636–5643, Apr. 2020, doi: https://doi.org/10.1609/aaai.v34i04.6017.

[17] C. Shao, W. Li, J. Huo, Z. Feng, and Y. Gao, “Attention-based investigation and solution to the trade-off issue of adversarial training,” Neural Networks, vol. 174, p. 106224, Mar. 2024, doi: https://doi.org/10.1016/j.neunet.2024.106224.

[18] S. Jadon and A. Jadon, “An Overview of Deep Learning Architectures in Few-Shot Learning Domain,” arXiv (Cornell University), Jan. 2020, doi: https://doi.org/10.48550/arxiv.2008.06365.

[19] Y. Song, T. Wang, P. Cai, S. K. Mondal, and J. P. Sahoo, “A Comprehensive Survey of Few-shot Learning: Evolution, Applications, Challenges, and Opportunities,” ACM Computing Surveys, Feb. 2023, doi: https://doi.org/10.1145/3582688.

[20] W. Ruan, X. Yi, and X. Huang, “Adversarial Robustness of Deep Learning: Theory, Algorithms, and Applications,” arXiv (Cornell University), Oct. 2021, doi: https://doi.org/10.1145/3459637.3482029.

[21] S. H. Silva and P. Najafirad, “Opportunities and Challenges in Deep Learning Adversarial Robustness: A Survey,” arXiv:2007.00753 [cs, stat], Jul. 2020, Available: https://arxiv.org/abs/2007.00753

[22] J. Lu, S. Jin, J. Liang, and C. Zhang, “Robust Few-Shot Learning for User-Provided Data,” IEEE transactions on neural networks and learning systems, vol. 32, no. 4, pp. 1433–1447, Apr. 2021, doi: https://doi.org/10.1109/tnnls.2020.2984710.

[23] Zeyuan Allen-Zhu and Y. Li, “Feature Purification: How Adversarial Training Performs Robust Deep Learning,” arXiv (Cornell University), pp. 977–988, Feb. 2022, doi: https://doi.org/10.1109/focs52979.2021.00098.

[24] N. Papernot, P. McDaniel, X. Wu, S. Jha, and A. Swami, “Distillation as a Defense to Adversarial Perturbations Against Deep Neural Networks,” 2016 IEEE Symposium on Security and Privacy (SP), May 2016, doi: https://doi.org/10.1109/sp.2016.41.

[25] C. Zhang et al., “Interpreting and Improving Adversarial Robustness of Deep Neural Networks with Neuron Sensitivity,” IEEE Transactions on Image Processing, pp. 1–1, 2020, doi: https://doi.org/10.1109/tip.2020.3042083.