JOURNAL OF
EMERGING INVESTIGATORS

Article

Evaluating need for adversarial training data given algorithmic defense methods against adversarial attacks

Braden Yian[1], Clayton Greenberg[2]
[1] Palos Verdes Distance Learning Academy, Palos Verdes, California
[2] Inspirit AI, San Diego, California

SUMMARY

An adversarial attack is a modification to the pixels of an image for the purpose of making a machine learning system misclassify the image. The foremost defense against adversarial attacks is adversarial training: a process in which the machine learning system trains on the already attacked images. But, this is not the only kind of defense. There are also algorithmic defense methods, which work to modify the learning process to be resilient to adversarial attacks without involving attacked examples. In this study, we considered three algorithmic defense settings: no algorithmic defense, defensive distillation, and gradient masking. Then we evaluated the role of adversarial training as part of defending machine learning models from adversarial attacks. Specifically, we set up a baseline image classifier for images of digits (MNIST dataset) and attacked the images using the fast gradient sign method. We hypothesized that introducing adversarial training for this classifier would significantly improve downstream classification accuracy in all three algorithmic defense settings. We found that, for all algorithmic defense settings applied to neural networks with between one and six convolutional layers, adding adversarial training consistently resulted in a statistically significant increase in accuracy. While these findings are limited by the specific data, parameters, and algorithms explored, our results suggest that implementing adversarial training within all lines of defense against adversarial attacks would be beneficial. We believe that this insight increases awareness of cybersecurity threats such as adversarial attacks and lowers the barrier of entry to protect against them.

INTRODUCTION

An image classifier is a machine learning model (most commonly a neural network) that predicts labels for given images. These models are being incorporated into daily operations across multiple industries. In the national defense industry, the United States military has already incorporated computer vision algorithms in Iraq and Syria operations to detect people and objects of interest (1). This poses a significant security risk if the image recognition model is exploited by opposing military forces. For example, the opposing force could perform adversarial attacks on the images. An adversarial attack is a collection of changes to the colors of the pixels of an image. They work by anticipating how the model will interpret the image and directing the model towards an incorrect classification. Researchers have demonstrated that the introduction of adversarial attacks can cause a system to misclassify a turtle for a rifle 82% of the time (2). This type of misidentification in real-life military computer vision algorithms could lead to devastating consequences during military operations. Goodfellow et al. proposed one of the first adversarial attacks, called the fast gradient sign method (FGSM), which makes subtle changes in the pixel values (perturbation) of the original image (3). These perturbations, often imperceptible to the human eye, resulted in an image of a panda being labeled as a gibbon with 99.3% confidence (3). Initially, adversarial attacks were viewed as only theoretical threats. However, researchers have demonstrated the physical real-world consequences by adding certain patterns to stop signs that caused self-driving vehicles to misclassify stop signs 100% of the time (4). Furthermore, Ilyas et al. successfully demonstrated the real-life threat of adversarial attacks on commercial classifiers such as Google Cloud Vision AI; thus, there is a clear need for robust defense mechanisms against adversarial attacks (5).

Goodfellow *et al.* first proposed adversarial training for defense against adversarial attacks, in which models are trained on combinations of clean data and adversarial perturbed examples. This approach of altering the model's training data helps the model to generalize its understanding of the data, rendering it more resilient to adversarial attacks (3). Subsequent studies have substantially built on this defense method since Goodfellow *et al.*, and it has since become one of the most widely used and effective defenses to this day. Kolter and Madry framed adversarial training as a robust optimization problem to minimize model parameters and maximize perturbations (3). This allowed users to minimize classification error of both the clean and attacked set on the Modified Institute of Standards and Technology Database (MNIST) to less than 6%. They also highlighted that its accuracy begins to falter for larger and more detailed real-life colored datasets, such as the CIFAR-10 dataset compared to the MNIST dataset. Previous work also explained how to evaluate adversarial training as well as other defenses to ensure they are best properly suited for the real world (6). They provided a checklist of best practices which we incorporated into our work, such as strong step sizes, reporting the clean and adversarial accuracy, code releases, and selecting the strongest and most adaptive attacks (6). For over 5 years, adversarial training has continued to be one of the most promising defensive methods (7). There are several disadvantages for adversarial training such as reductions of clean accuracy, overfitting to adversarial examples, and higher computational cost (7).

Concurrently with improvements to adversarial training,

https://doi.org/10.59720/25-181

researchers developed "algorithmic defense methods", which alter the model's training process via manipulation of the gradients and probability distributions. For example, defensive distillation, proposed by Papernot *et al.*, "distills down" the network's output probabilities, producing softer and less confident predictions (8). This would for instance drop the confidence of a label prediction from 99.9% to 80%. By introducing an element of ambiguity and uncertainty into the training process, the model becomes more resilient to subtle changes or perturbations in the data. Papernot also proposed gradient masking to reduce a model's vulnerability to adversarial attacks by injecting noise into the gradients used by the model (7). The gradients control how the weights in the neural network update during training. With the introduction of randomness (noise) into the gradients, it becomes more difficult to pinpoint specific regions in the images to which the models are especially sensitive.

While the benefits of adversarial training as a defense against adversarial attacks on image classification systems have been explored widely, as described above, it was unclear whether there was still a need for adversarial training given algorithmic defense methods such as defensive distillation and gradient masking. We aim to fill the research gap via a systematic comparison of the impact of adversarial training over and above defensive distillation and gradient masking. We hypothesized that introducing adversarial training significantly improves downstream classification accuracy in three algorithmic defense settings: (no algorithmic defense, defensive distillation, and gradient masking). Indeed, we found a significant increase in accuracy for all three algorithmic defense settings. Accordingly, we suggest that adversarial training be used in all defenses systems within which it can be reasonably implemented.

RESULTS

We hypothesized that the inclusion of adversarial training increases the accuracy for an image classification system with no algorithmic defense, one with defensive distillation, and one with gradient masking. To test this, the study utilized the MNIST dataset (9), a collection of 70,000 28x28 grayscale images of handwritten versions of the numbers 0-9, with the standard train-test split (60,000 training images and 10,000 test images). We selected this dataset for its simplicity, allowing us to focus on the adversarial attacks rather than intricacies in the images. Performing image classification on this dataset is based on optical character recognition, i.e. the task of the classifier is to identify which of the numbers appears in each picture. An adversarial attack would attempt to modify the pictures slightly so that for a given picture, a human would see one number and the classifier would see a different one. To defend against such attacks, we implemented three algorithmic defense settings (no defense, defensive distillation, and gradient masking) on neural networks containing between one and six convolutional layers. With more convolutional layers, we expected the network to capture larger and more complex visual patterns. We wanted to explore the robustness of the effect of adversarial training across a range of different network sizes and algorithmic defense settings. Each of the 18 networks were trained with and without adversarial training on the MNIST dataset, attacked with FGSM. Therefore, the study had a 3-by-6-by-2 design.

In this study, when we trained a model with adversarial training, we implemented an FGSM attack on each of the 60,000 images in the training data and appended the attacked versions to the training dataset. We trained and evaluated baseline CNN models of different sizes on the clean (not attacked) images. These models achieved an average accuracy of approximately 98% across all model sizes. Our results show an upward trend in accuracy as the size of the network (number of convolutional layers) increases (**Figure 1** and **Table 1**).

Deeper convolutional neural networks consistently outperformed shallower ones, and the distance between the adversarial training curve and the no adversarial training curve was smallest at six convolutional layers (**Figure 1**). So, we used McNemar test results on fully-trained models of this size to evaluate whether these distances were statistically significant. We used the McNemar test (chi-squared test) because we were evaluating the predictions of two distinct models on the same data For the no algorithmic defense setting with six convolutional layers, the network with adversarial training was significantly more accurate than the network without adversarial training (ΔAccuracy = 10.4%,

No. of Layers	None	None+AT	DD	DD+AT	GM	GM+AT
1	9.38%	34.74%	16.11%	31.80%	21.70%	31.99%
2	19.48%	86.75%	32.82%	80.53%	40.82%	78.75%
3	86.74%	97.27%	88.62%	94.93%	91.12%	96.27%
4	85.95%	97.71%	94.64%	98.00%	96.29%	98.39%
5	86.76%	98.36%	94.37%	98.11%	95.65%	98.09%
6	87.41%	97.84%	93.53%	98.13%	95.92%	98.13%

Table 1. All accuracies per convolutional layer. All three algorithmic defense settings (no algorithmic defense "None", defensive distillation "DD", and gradient masking "GM") with and without adversarial training tested and their respective accuracies per convolutional layer amount within the model architecture. Models trained for 7 training epochs per convolutional layer.

https://doi.org/10.59720/25-181

McNemar $\chi^2(1) = 894$, $p < 0.001$, **Table 2**). For the defensive distillation setting with six convolutional layers, the network with adversarial training was significantly more accurate than the network without adversarial training (ΔAccuracy = 4.6%, McNemar $\chi^2(1) = 356$, $p = < 0.001$, **Table 3**). For the gradient masking setting with six convolutional layers, the network with adversarial training was significantly more accurate than the network without adversarial training (ΔAccuracy = 2.21%, McNemar $\chi^2(1) = 145$, $p = 0.001$, **Table 4**).

DISCUSSION

Our research question explored whether adversarial training improves a system for defending against adversarial attacks, even when also employing algorithmic defense methods. Since the models with adversarial training consistently outperformed the models without adversarial training, we observed strong evidence that a marginal benefit in classification accuracy against attacks attributable to adversarial training was present in all cases.

Additionally, our results show the minimum appropriate network size to be effective on the attacked MNIST dataset. However, because the accuracies plateaued near the ceiling of 100% at around five convolutional layers, we inferred that there would be no substantial increase in accuracy if we used more than six convolutional layers or a pre-trained model such as ResNet. Our results aligned with previous

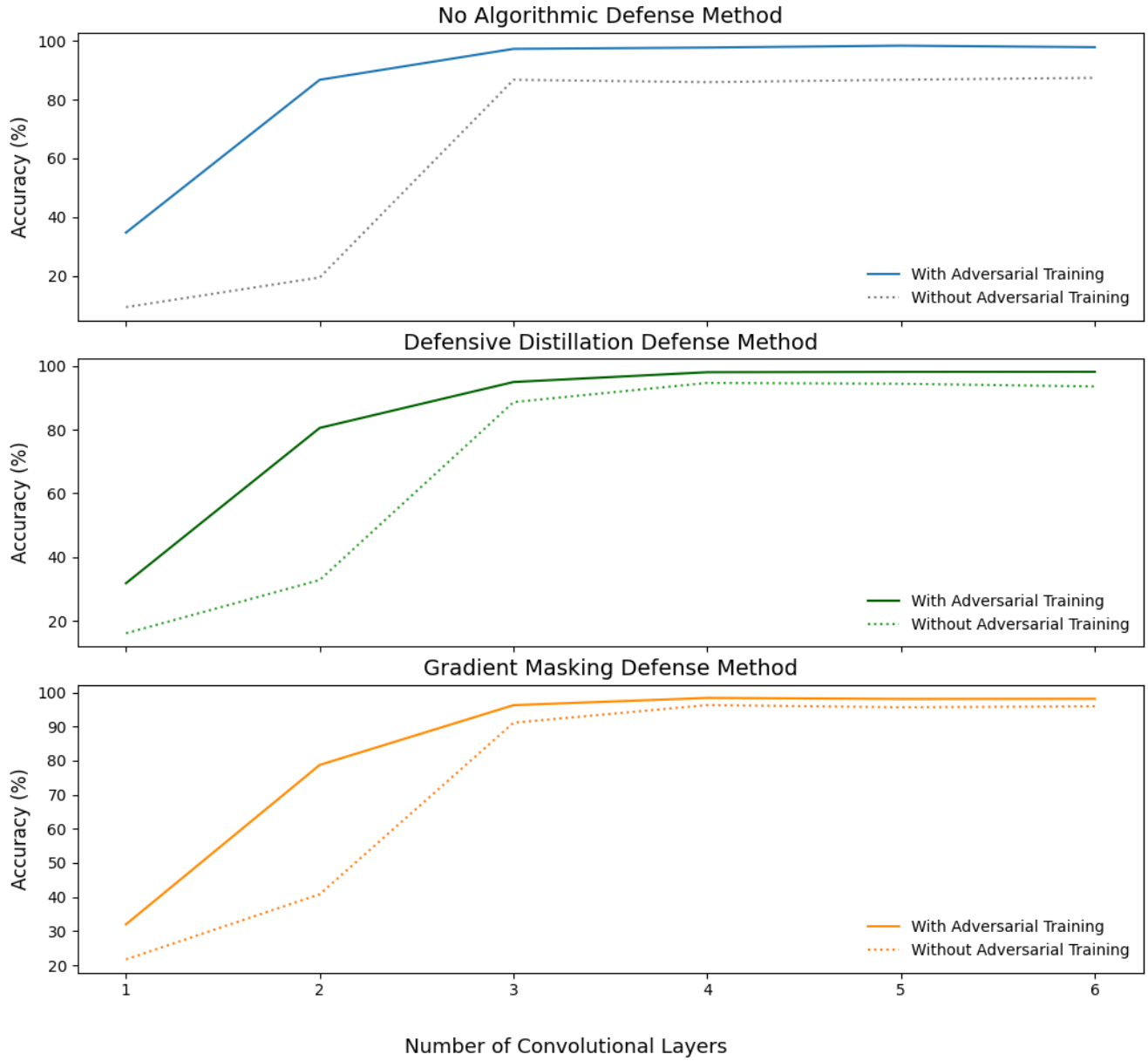

Figure 1. Accuracies of models with and without adversarial training versus number of convolutional layers. The top panel has no algorithmic defense, middle panel has defensive distillation, and the bottom panel has gradient masking. Line graph showing mean image classification accuracies of each model with 1-6 convolutional layers (N=10,000). Model accuracies were computed after 7 epochs of training against an FGSM attack for each convolutional layer amount.

https://doi.org/10.59720/25-181

	None + AT Correct	None + AT Incorrect
No Algorithmic Defense Correct	8655	86
No Algorithmic Defense Incorrect	1129	130

Table 2. No algorithmic defense contingency table. The amount of images correctly classified by the 6 convolutional layer, non-algorithmic defense models with and without adversarial training. Used for McNemar analysis. Models trained for training 7 epochs per convolutional layer.

testing conducted on the effectiveness of adversarial training which found a similar <90% accuracy when using adversarial training against adversarial attacks (3).

We acknowledge that using a pre-trained model might have reduced the gap between the adversarial training conditions and their no adversarial training counterparts, but we leave this investigation for future work. Both algorithmic defense methods require substantial hyperparameter tuning for optimal effectiveness, while adversarial training does not. The results of this study suggest that adversarial training may lessen the need for such hyperparameter tuning (**Figure 1**). This is because in every instance, adversarial training provided a statistically significant increase in accuracy when it was applied to a system with no defense, gradient masking, or defensive distillation. As such, based on the results from our experiments with this dataset and attack method, we recommend adversarial training for all defense systems for which adversarial training data is available.

We conjectured that the inclusion of adversarial training boosts algorithmic defense methods in all instances because it is more specific than algorithmic defense methods. Since the defender is able to alter the training dataset to include examples of the specific attack, it makes the model resilient to the kinds of manipulations that the attack performs. However, algorithmic defense methods are preventative, altering the machine learning process itself to lessen the impact of an adversarial attack. They do not adapt to the attacks. So, algorithmic defense methods are less specific and more indirect. Adversarial training requires the defender to know the attack formula being used. Indeed, one limitation of this work

	DD + AT Correct	DD + AT Incorrect
Defensive Distillation Correct	9287	66
Defensive Distillation Incorrect	526	121

Table 3. Defensive distillation contingency table. The amount of images correctly classified by the 6 convolutional layer, defensive distillation models with and without adversarial training. Used for McNemar analysis. Models trained for 7 training epochs per convolutional layer.

	GM + AT Correct	GM + AT Incorrect
Gradient Masking Correct	9536	56
Gradient Masking Incorrect	277	131

Table 4. Gradient masking contingency table. The amount of images correctly classified by the 6 convolutional layer, gradient masking models with and without adversarial training. Used for McNemar analysis. Models trained for 7 training epochs per convolutional layer.

is that the attack code was shared across training and testing. If someone uses a different attack or different implementation of the FGSM attack, the results could be different.

Another limitation of the work is our use of the MNIST dataset. The simplicity of the images in this dataset makes the classification task rather easy, as shown by the classification accuracy on unattacked data (data not shown), and raises the risk of overfitting.

In future work, we intend to investigate the extent to which adversarial training on known attack methods is successful against unknown attacks. We anticipate that diverse adversarial training examples and inclusion of algorithmic defense methods may be helpful. This work would bring to light the importance of knowing the specific attack(s) to be defended against when performing adversarial training. Nonetheless, the explicit training and reshaping of the model's decision boundaries from adversarial training makes a model fundamentally more robust compared to one with only algorithmic defense methods.

Overall, we found that adversarial training significantly increases accuracy in all test cases on the MNIST dataset using a custom base classifier. By exploring the impact of augmenting training data with adversarial examples in various machine learning settings, we aim to contribute substantially to the field of machine learning cybersecurity. This work offers insights into the most viable strategies for safeguarding machine learning systems against sophisticated adversarial threats.

MATERIALS AND METHODS

In this study, the MNIST dataset, a collection of 70,000 28x28 grayscale images of the numbers 0-9, with a standard train-test split, was utilized (9). The integer pixel values were normalized to be represented as floats. The single color channel was duplicated so the images would be represented in RGB format and the images were resized to 32x32. These alterations were necessary to make the MNIST dataset compatible with the image classification networks.

To assess the model's robustness, a custom architecture was employed (**Figure 2**). The evaluated networks had between one and six convolutional layers. They were built from scratch and did not use a pretrained network. In each convolutional layer, the convolution (kernels), then the ReLU activation function, then batch normalization, and then finally max pooling were applied. This order is standard for

https://doi.org/10.59720/25-181

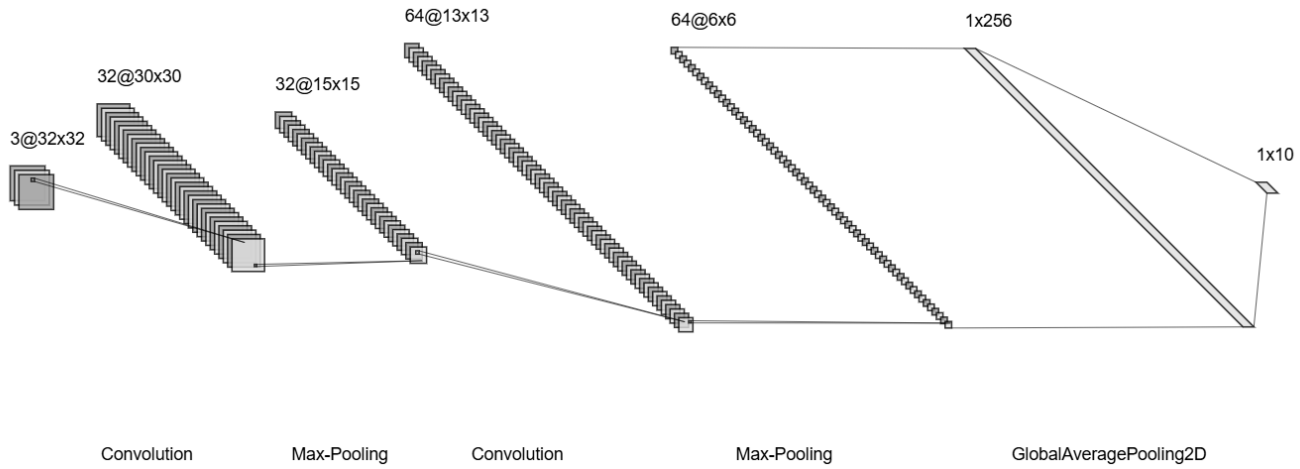

Figure 2. Architecture flowchart of our neural network. This figure shows the inputs, outputs, and layer types (convolutional layers, max-pooling layers, and the final dense layers) within our network architecture. Made using Le Net style (9).

convolutional neural networks. All networks then had a final pooling layer, a dense layer with 256 nodes, and a final dense layer with 10 nodes representing the 10 MNIST classes.

In this study, when a model was trained with adversarial training, an FGSM attack was implemented on each of the 60,000 images in the training data and the attacked images were appended to the training dataset. The FGSM attack was implemented using descriptions of the method found online. In particular, the gradient of the loss function (original classification task) was calculated to determine the best way to perturb each pixel while being constrained to a single-step (ε=0.03). Training proceeded for 7 epochs in every experimental condition. For example, in the defensive distillation with adversarial training condition, the model was trained on clean images with defensive distillation for 4 epochs and then on attacked and clean images (adversarial training) for 3 epochs. The test set contained only the attacked versions of the 10,000 test images.

Defense methods were implemented based on descriptions from the papers above with teacher and student model temperatures set to 20 for Defensive Distillation. The temperature acts as a "bias" term of sorts on the probability distribution for the classification of the image. A higher temperature makes the distribution more "ambiguous" (8) or less extreme, thus a temperature to 20 according to the experiments in (8) was set, striking a balance between effectiveness and destructive power of the defense.

The McNemar test used to verify the statistical significance of Adversarial Training was implemented by examining pairs of predictions that both models correctly classified, one correctly classified, or both incorrectly classified. Specifically, by comparing the cases in which one model correctly classified the image while the other one incorrectly classified it, this provided insight into the statistical significance of the introduction of the non-algorithmic defense.

Code and Data Availability Statement

The code from this project is available at https://github.com/BradenYian/Science_Fair_Project.

ACKNOWLEDGMENTS

We acknowledge Veritas AI and Soy Choi for suggestions and help throughout the early stages, development, and planning of the project.

Received: August 18, 2024
Accepted: December 11, 2025
Published: July 05, 2026

REFERENCES

1. Alhajjar, Elie. "Adversarial Machine Learning Poses a New Threat to National Security." The Cyber Edge. www.afcea.org/signal-media/cyber-edge/adversarial-machine-learning-poses-new-threat-national-security. Accessed 1 July 2024.
2. Athalye, Anish, *et al.* "Synthe g Robust Adversarial Examples." *International Conference on Machine Learning*, vol. 80, 2018, pp. 284–293. proceedings.mlr.press/v80/athalye18b/athalye18b.pdf
3. Goodfellow, Ian, *et al.* "Explaining and Harnessing Adversarial Examples." *arXiv*, 2015. https://doi.org/10.48550/arXiv.1412.6572
4. Eykholt, Kevin, *et al.* "Robust Physical-World Attacks on Deep Learning Visual Classification." *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 1625–1634. https://doi.org/10.1109/cvpr.2018.00175
5. Ilyas, Andrew, *et al.* "Black-box Adversarial Attacks with Limited Queries and Information." *Proceedings of the 35th International Conference on Machine Learning*, vol. 80, 2018, pp. 2137–2146. proceedings.mlr.press/v80/ilyas18a.html
6. Athayle, Anish, *et al.* "On Evaluating Adversarial Robustness", 2019, *arXiv*:1902.06705.
7. Papernot, Nicolas. "Gradient Masking in Machine Learning." ARO Workshop on Adversarial Machine Learning. seclab.stanford.edu/AdvML2017/slides/17-09-aro-aml.pdf. Accessed 1 July 2024.
8. Papernot, Nicolas, *et al.* "Distillation as a Defense to Adversarial Perturbations Against Deep Neural

https://doi.org/10.59720/25-181

Networks." *IEEE Symposium on Security and Privacy (SP)*, 2016, pp. 582–597. https://doi.org/10.1109/SP.2016.41
9. Deng, Li. "The MNIST Database of Handwritten Digit Images for Machine Learning Research." *IEEE Signal Processing Magazine*, vol. 29, no. 6, 2012, pp. 141–142.
10. LeNail, Alex. "NN-SVG Tool". Alex LeNail Academic Website. https://alexlenail.me/NN-SVG/LeNet.html Accessed 21 Nov. 2025.