Securing Traffic Sign Recognition Systems in Autonomous Vehicles

Thushari Hapuarachchi, Long Dang, Kaiqi Xiong
ICNS Lab and Cyber Florida,
University of South Florida, Tampa, FL, 33620 USA
Email: saumya2@usf.edu, longdang@usf.edu, xiongk@usf.edu

***Abstract*—Deep Neural Networks (DNNs) are widely used for traffic sign recognition because they can automatically extract high-level features from images. These DNNs are trained on large-scale datasets obtained from unknown sources. Therefore, it is important to ensure that the models remain secure and are not compromised or poisoned during training. In this paper, we investigate the robustness of DNNs trained for traffic sign recognition. First, we perform the error-minimizing attacks on DNNs used for traffic sign recognition by adding imperceptible perturbations on training data. Then, we propose a data augmentation-based training method to mitigate the error-minimizing attacks. The proposed training method utilizes nonlinear transformations to disrupt the perturbations and improve the model robustness. We experiment with two well-known traffic sign datasets to demonstrate the severity of the attack and the effectiveness of our mitigation scheme. The error-minimizing attacks reduce the prediction accuracy of the DNNs from 99.90% to 10.6%. However, our mitigation scheme successfully restores the prediction accuracy to 96.05%. Moreover, our approach outperforms adversarial training in mitigating the error-minimizing attacks. Furthermore, we propose a detection model capable of identifying poisoned data even when the perturbations are imperceptible to human inspection. Our detection model achieves a success rate of over 99% in identifying the attack. This research highlights the need to employ advanced training methods for DNNs in traffic sign recognition systems to mitigate the effects of data poisoning attacks.**

***Index Terms*—Autonomous Vehicles, Traffic sign recognition, Deep neural networks, Data poisoning attacks, Nonlinear transformations**

I. Introduction

Traffic sign recognition systems are essential for detecting road signs and aiding drivers or control modules in making informed driving decisions. Modern autonomous vehicle models, such as Tesla's Model 3 [1], integrate traffic sign recognition systems as essential components of their driving assistance technology [2]. According to [3], the classification algorithms employed in traffic sign recognition systems can be categorized into two groups: machine learning-based and deep learning-based approaches. Deep learning algorithms include deep neural networks (DNNs) which are widely used nowadays because they can automatically extract high-level features from images. However, researchers have expressed concerns about the security of traffic sign recognition systems, as their dependence on DNNs makes them susceptible to various attacks such as data poisoning [4], [5].

DNNs for traffic sign recognition systems need a lot of data for training to make sure that they perform without any errors that might cause fatal accidents. As a result, data may be collected from various sources, both trusted and untrusted, such as the Internet, for training purposes. However, these data may be poisoned. One such method of data poisoning is the error-minimizing attacks [6], which involve adding imperceptible perturbations to the data. These data are difficult to be detected as poisoned due to the low intensity of the perturbations. Once the system is trained on these data, the training accuracy is very high. However, when the system predicts images captured by the vehicle on the road, the prediction accuracy [4] drops, leading to incorrect predictions. Fig. 1-a) illustrates this scenario.

In this paper, we first implement the error-minimizing attacks [6] on DNNs trained for traffic sign recognition. We manipulate the strength of the attack by varying the perturbation intensity. The strength of the attack is measured using the prediction accuracy. Our experimental results show that the attack is stronger when the perturbation intensity is higher. However, the perturbations become visible at higher intensities and can be easily detected by data collectors. To detect this attack, we propose a detection model capable of identifying poisoned data even when the perturbations are imperceptible to human inspection. The detection model is built using a simple convolutional neural network (CNN) model and gave a success rate of more than 99%. Moreover, we propose a data augmentation-based training method (a mitigation scheme) to mitigate error-minimizing attacks, as illustrated in Fig. 1-b). The proposed training method utilizes nonlinear transformations such as grayscale to disrupt the perturbations and improve the model's resilience. This mitigation scheme is able to achieve almost the same performance as a system trained on clean data. Furthermore, we show that the error-minimizing attack has a limitation: when DNNs are exposed to clean (non-poisoned) data during training, the attack becomes ineffective.

The error-minimizing attacks, as introduced by [6], have not been implemented on traffic sign recognition datasets. When applying the error-minimizing attacks on traffic sign recognition datasets, we face several challenges compared to the baseline datasets including CIFAR-10, and ImageNet: 1) handling complex datasets that have larger number of classes, 2) limited variety of images within the same class, 3) high variations in image colors in the same class, and 4) algorithms require longer convergence times. We explain how we addressed these challenges in Sec. III-A. Additionally, we com-

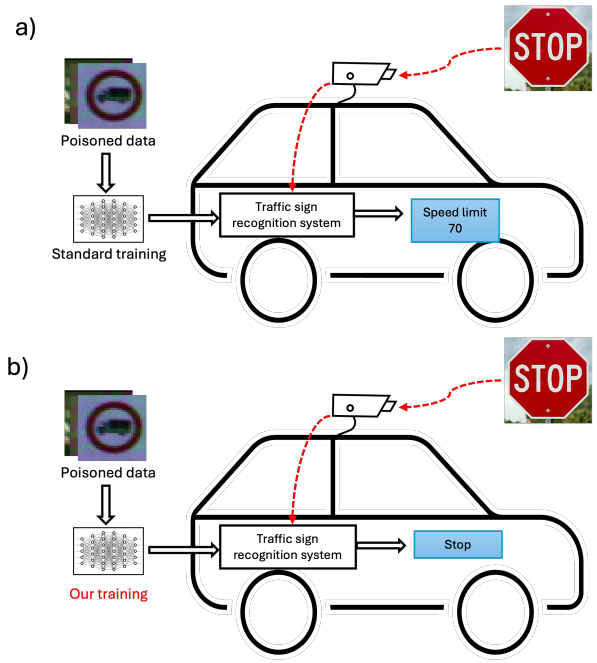


Fig. 1: Overview of this research. a) We train the traffic sign recognition system using data poisoned by error-minimizing attacks with *standard training*. When the trained system is used for predicting traffic signs on the road, the signs are misclassified. b) We train the traffic sign recognition system with *our mitigation scheme*, where we propose a data-augmentation-based training method to mitigate the effects of error-minimizing attacks. Then, the trained traffic sign recognition system is able to provide the correct predictions.

pare the proposed mitigation scheme with adversarial training, a widely used approach for mitigating evasion attacks [7]. The experimental results demonstrate that our mitigation scheme outperforms adversarial training. In summary, we make the following key contributions.

- We exploit the error-minimizing attacks to poison DNNs in traffic sign recognition systems. The attacks significantly drop the prediction accuracy of the traffic sign recognition systems.
- We propose a detection model to identify the data poisoned by the error-minimizing attacks. The detection model is built using a simple CNN model and resulted in a success rate of more than 99%.
- We propose a data augmentation-based training method (a mitigation scheme) to mitigate the effect of error-minimizing attacks. The proposed training method utilizes nonlinear transformations to disrupt the perturbations added by the error-minimizing attacks. Our experimental results show that the proposed training method outperforms adversarial training in mitigating the error-minimizing attacks.

II. Related Work

DNNs are susceptible to both evasion and data poisoning attacks, depending on the stage at which the attack is executed [4], [5], [8]. Data poisoning attacks involve injecting malicious or misleading samples into the training dataset, ultimately degrading the model's generalization and reducing its prediction accuracy. These attacks can be particularly stealthy, as poisoned samples may appear benign but cause systematic misclassifications. On the other hand, evasion attacks occur at prediction time by adding carefully crafted adversarial perturbations to input images, tricking trained DNNs into misclassifying them [9], [10]. Such attacks pose significant risks in real-world applications, especially for autonomous driving systems that rely heavily on accurate traffic sign recognition [8], [11].

In this paper, we explore the error-minimizing attacks [6] to attack DNNs trained for traffic sign recognition. This attack strategically perturbs training samples to maximize the model training accuracy while minimizing the model's prediction ability on legitimate (clean) data. Several studies [12]–[14] have proposed remedies for the error-minimizing attacks including data augmentation techniques and adversarial training [15]. However, these studies mostly focused on the baseline datasets. In contrast, our study highlights the effectiveness of this attack in traffic sign recognition systems and proposes a mitigation scheme incorporating nonlinear transformations to reduce the impact of the error-minimizing attacks.

III. Methodology

A. Error-minimizing Attacks

The error-minimizing attacks [6] make poisoned data by adding a type of imperceptible perturbation to the original data. This perturbation is generated in a way that minimizes the loss of the model trained on it while preventing the DNN from learning the actual features of the images. In this study, we adopted Huang et al. [6]'s approach for performing the error-minimizing attacks. They proposed solving the following bi-level optimization problem to generate the perturbations.

$$\arg\min_{\theta} \left[\min_{\|\delta\|_p \leq \epsilon} \mathcal{L}(f(X^D + \delta; \theta), Y^D) \right], \tag{1}$$

where X^D and Y^D denote a set of training images and a set of target labels, respectively. The goal of the inner optimization problem is to find the set of perturbations denoted by δ. Huang et al. [6] used the Projected Gradient Descent (PGD) method to solve this inner optimization problem. First, they initialized δ. In each iteration, a fixed model with parameters θ is evaluated on $X^D + \delta$ and the loss is calculated. δ is updated for T iterations while minimizing the loss. ϵ controls the intensity of the added perturbation. The L_p norm of δ is bounded by ϵ to ensure that the perturbations remain imperceptible. The outer optimization problem finds the model parameters that minimize the same loss. In this step, δ is fixed and the model parameters are optimized to achieve the minimum loss when trained on $X^D + \delta$. The optimization process is terminated when the model's training accuracy is higher than a given value λ.

We implemented this attack on the traffic sign recognition system, an aspect that was not explored by Huang et al. [6].

Applying this attack on traffic sign data presents greater challenges compared to the baseline datasets due to several factors. 1) The higher number of classes in the traffic sign datasets compared to the baseline datasets increases the complexity of the perturbation generation process. The datasets we considered, German Traffic Sign Recognition Benchmark (GTSRB) [16] and the Chinese Traffic Sign Recognition Database (CTSRD) [17], have 43 and 58 classes, respectively. 2) The limited variety of images within the same class makes it easier for models to learn them effectively, which makes the attacks more difficult. To address this, we generate class-wise perturbations instead of sample-wise perturbations. 3) Variations in image appearance, such as differences between night and day, add another layer of difficulty. Some images are particularly dark, making it more difficult to introduce imperceptible perturbations. 4) Due to the higher complexity of the data, the algorithms require longer convergence times. Hence, we set the number of PGD iterations to solve the inner minimization problem in Eq. 1 to 1 instead of 10 which is suggested by [6].

In this study, we used ResNet18 as the model architecture when generating the attack. We chose ResNet18 because is it a widely used high performance DNNs model for image classification [18]. We considered a white-box attack setting and attacked the same model. The stopping criterion for generating perturbations was set to 99% (λ) training accuracy. We considered three values for ϵ. We defined the strength of the attack based on ϵ, as the attack's performance highly depends on the intensity of the added perturbations.

B. The Detection Model

As shown in Fig 2, the error-minimizing attacks perturbations are barely visible to the human eyes, especially when $\epsilon = 4$. Hence, model trainers are unable to determine whether the data is poisoned by merely observing the images. Additionally, it is not feasible to inspect all images manually. Therefore, we proposed a detection model to identify whether each image is poisoned or clean. The output of the detection model is either 'poisoned' or 'clean'.

Fig 3 shows the model architecture of the detection model. The model consisted of two convolutional layers with a kernel size of 3×3 followed by a max pooling layer with a kernel size of 2×2, flattening, and two fully connected layers. The output size of each layer is shown in Fig 3. The final layer used a sigmoid activation function to classify images into two categories i.e, poisoned or clean image. All other layers have ReLU activation function. The detection model utilized binary cross-entropy loss as the loss function and Adam as the optimizer.

C. The Mitigation Scheme

The goal of the mitigation scheme is to recover the traffic sign recognition system that has been attacked by the error-minimizing attacks. To achieve this, we proposed a data augmentation-based training approach instead of the standard training method. Algorithm 1 shows the proposed mitigation

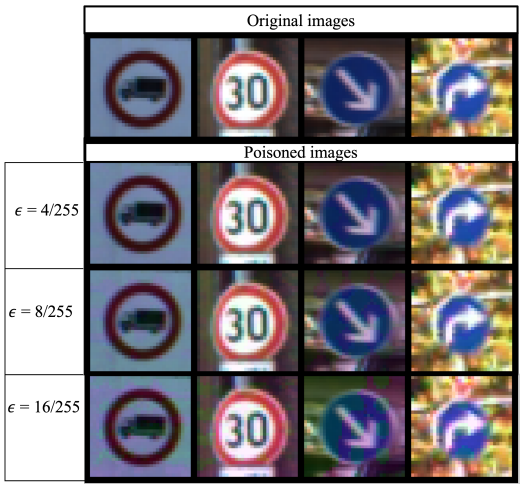


Fig. 2: Original images and poisoned images with different poison rates.

Fig. 3: The detection model.

scheme. First, we selected a set of nonlinear transformation techniques (T). When selecting T, we prioritized the nonlinear transformations explored in previous studies for mitigating the error-minimizing attacks [12]–[14]. Then, we randomly chose one transformation (t_i) from the set and applied it on the training dataset. Next, we trained the traffic sign recognition system using the transformed data. We chose ResNet18 as the model architecture and trained it for 20 epochs with the Stochastic Gradient Descent (SGD) optimizer and cross-entropy loss as the loss function. After training, we validated the model on a clean dataset, which reflected how the system performs on traffic sign images captured by the vehicle while driving. We defined the accuracy that is obtained on the clean validation dataset as *prediction accuracy*. If the prediction accuracy reached the desired value (α), we stopped the training process. Otherwise, we chose another transformation, applied it to the training dataset, and combined the newly transformed dataset with the dataset from the previous iteration. In this way, we expanded the training dataset size in each iteration until we achieve the desired prediction accuracy.

We evaluated our mitigation scheme on two traffic sign image datasets: GTSRB [16], and CTSRD [17]. In our experimental evaluation, we chose three nonlinear transformations for T: grayscale, Color Jitter, and Random Invert. To implement these transformations, we used PyTorch's built-in options [19]. For instance, we used the *transforms.Grayscale(3)* command to apply the grayscale transformation, where 3 represents the number of channels we want the transformed

Algorithm 1 Mitigating error-minimizing attacks

Input: Poisoned Traffic sign data (D), Traffic Sign recognition system (F), Clean prediction dataset (C), A set of nonlinear transformation techniques (T), Target prediction accuracy (α)
Return: Robust Traffic Sign recognition system (F_i)
while $v < \alpha$ **do**
1. Randomly choose t_i from T
2. Transform D using $t_i \rightarrow t_i(P)$
3. $t_i(P)$ + $D_{i-1} \rightarrow D_i$ $\triangleright$ $D_0 = \emptyset$
4. Train F on $D_i \rightarrow F_i$
5. Predict F_i on $C \rightarrow$ prediction accuracy, v

end while

image to have. We chose grayscale because, several studies [12]–[14] have already shown that grayscale transformation is effective against the error-minimizing attacks. Moreover, grayscale transformation controls the channel-wise perturbations by replacing all three channel values in a pixel with a single value. For the Color Jitter transformation, we used the command *transforms.ColorJitter(brightness=.5, hue=.3)*, which reduced the brightness and hue of the image. By modifying the brightness and hue, the color jitter transformation expands the feature space of the model and reduces the risk of overfitting to adversarial examples [20]. To apply the Random Invert transformation, we used the command *transforms.RandomInvert(p=1.0)*, where $p = 1.0$ indicates that the transformation is applied to all the images in the dataset. This transformation modifies the pixel values of an image by inverting them.

IV. Experimental Evaluation

For the experimental evaluation, we used two datasets: the German Traffic Sign Recognition Benchmark (GTSRB) [16], and the Chinese Traffic Sign Recognition Database (CTSRD) [17]. Table I includes the specifications of these datasets. GTSRB dataset contains 31,367 images for training and 7,842 images for prediction (aka validation), spanning 43 classes. The CTSRD dataset includes 3,336 training images and 834 prediction images, covering 58 classes. All the experiments are conducted in PyTorch 1.13.1 framework.

TABLE I: Dataset details.

Dataset	Training set size	Prediction set size	# of classes
GTSRB [16]	31367	7842	43
CTSRD [17]	3336	834	58

A. Error-minimizing Attacks on Traffic Sign Recognition Systems

To perform the error-minimizing attacks, we adopted the code in Huang et al. [6]. ResNet18 was used as the base model for generating the error-minimizing perturbations. In our experiments, we fixed the seed to 42 to ensure the reproducibility. We generated class-wise perturbations, as they are more effective than sample-wise perturbations [6]. Perturbations were generated with three maximum perturbation limits (ϵ) of 4/255, 8/255, and 16/255. When $\epsilon = 4/255$, the perturbations are barely visible, making it difficult for human eyes to distinguish poisoned images (see Fig. 2). When $\epsilon = 16/255$, the perturbations became slightly visible. Hence, we defined the strength of the attack based on the ϵ: a higher ϵ indicates a stronger attack due to the increased intensity of the perturbations. Following the white-box attack settings, we attacked the same model (ResNet18) using the poisoned data.

The second column in Table II shows the prediction accuracy of the model that is trained on clean data (no attack). The prediction dataset is also clean because it reflects the images captured by the vehicle while driving on the road. The model achieved prediction accuracies of 99.90% and 98.56% for the GTSRB and CTSRD datasets, respectively. The fourth column of Table II shows the accuracy of the prediction after the error-minimizing attacks. The attack is able to reduce the prediction accuracy from 99.90% to 10.6% for the GTSRB dataset and from 98.56% to 25.18% for the CTSRD. These results indicate that the poisoned models failed to accurately predict the signs in the traffic sign images. However, the strength of the attack can be changed by modifying ϵ. When ϵ is reduced to $4/255$, the prediction accuracy increased to 52.10% for GTSRB dataset and 70.26% for CTSRD dataset. These results demonstrate that the attack becomes weaker as ϵ decreases.

Next, we demonstrate a limitation of the proposed attack method. Our experiments showed that the error-minimizing attacks are not effective when only part of the training data is poisoned. To illustrate this, we first poisoned the entire training dataset. Then, we poisoned a proportion of the training dataset and observed the attack's performance. Fig. 4 shows the prediction accuracy curves when poison proportions (p) are changed from 1 to 0.5. When the full dataset is poisoned, i.e., $p = 1.0$, the prediction accuracy is around 10%. When only 95% of the dataset is poisoned, the prediction accuracy is over 90%. When the poison proportion is further reduced, the model gives even higher prediction accuracies. These results demonstrate that the error-minimizing attacks are not effective in fooling traffic sign recognition systems when a portion of the training dataset is clean.

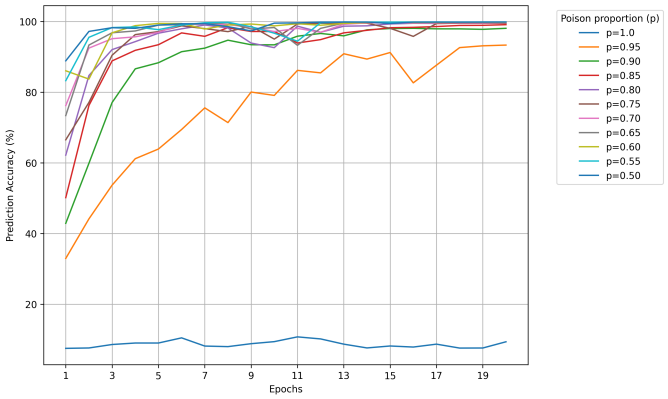


Fig. 4: The error-minimizing attacks with different poison proportions.

TABLE II: The prediction accuracy of the traffic sign recognition systems.

Dataset	No Attack (%)	ϵ	Attack (%)	Our Mitigation Scheme (%)	Adversarial Training (%)
GTSRB [16]	99.90	16/255	10.6	96.05 (+85.45)	91.52 (+80.92)
		8/255	22.38	99.59 (+77.21)	98.87 (+76.49)
		4/255	52.10	99.86 (+47.76)	98.93 (+46.83)
CTSRD [17]	98.56	16/255	25.18	98.08 (+72.90)	96.04 (+70.86)
		8/255	45.44	98.32 (+52.88)	95.20 (+49.76)
		4/255	70.26	98.32 (+28.06)	94.96 (+24.70)

B. The Detection Model

The detection model is evaluated on a version of GTSRB dataset available in Pytorch datasets. The dataset is divided into training and prediction sets, containing 39,209 images for training and 12,630 images for prediction. We converted 50% of both training and prediction datasets into poisoned data. The detection model's performance is presented in Table III. The model achieved a success (prediction accuracy) rate of over 99% regardless of the strength of the attack (ϵ). These results imply that the detection model can accurately distinguish poisoned data from the clean data. Moreover, the success rate of the detection model is slightly reduced when the ϵ is low, probably due to the lower intensity of the perturbations, making them harder to detect. Fig. 5 shows the loss curves for the detection models trained on poisoned data with different ϵ. The detection model trained on data poisoned with a small ϵ exhibited a higher initial binary cross-entropy loss compared to training with a large ϵ. However, by the end of training, all detection models reached similar loss values.

TABLE III: Accuracy of the detection model.

ϵ	Training Accuracy	Success Rate
16/255	100%	99.97%
8/255	99.99%	99.66%
4/255	99.80%	99.11%

C. The Mitigation Scheme

The fifth column of Table II shows the prediction accuracy after applying our mitigation scheme against the error-minimizing attacks. To mitigate the weaker attacks (ϵ = 8/255, 4/255) on the GTSRB, we only needed to use grayscale transformation, which increased the prediction accuracy up to 99%. To mitigate the attack with higher strength (ϵ = 16/255) on the GTSRB dataset, we used grayscale, Color Jitter and Random Invert as the nonlinear transformations, improving the prediction accuracy up to 96.05%. We applied the same transformations to the CTSRD dataset attacked with the error-minimizing attack at ϵ = 16/255. It improved the accuracy from 25.18% to 98.08%, achieving almost the same prediction accuracy as the model trained on clean data. For mitigating the weaker attacks (ϵ = 8/255, 4/255) on CTSRD, we used Color Jitter and Grayscale. Fig. 6 shows the evaluation of prediction accuracy during training with and without the error-minimizing attack and after the applying our mitigation scheme. The red lines denote the prediction accuracy when the model is trained with standard training on poisoned data, which is very low. The blue lines show the high prediction accuracy when the same training approach is used on clean data. The green lines indicate the prediction accuracy when the poisoned data is trained using our mitigation scheme, which is almost the same as training with clean data. These results show that the mitigation scheme can overcome the effects of the error-minimizing attacks and provide a prediction accuracy nearly equivalent to that of clean data.

We compared our mitigation scheme with adversarial training [7], a widely used approach for mitigating evasion and data poisoning attacks. We implemented adversarial training using PGD attacks [15]. Following the default settings in [15], we used a perturbation radius of 8/255, a step size of 0.8/255, and the number of PGD steps is set to 10. We trained the same ResNet18 model for 20 epochs to be comparable with our other experiments. The prediction accuracies after applying adversarial training are shown in the sixth column of Table II. Our mitigation scheme outperforms adversarial training regardless of the attack strength. When the attack strength is low (ϵ = 4/255, 8/255), adversarial training performs as well as our mitigation scheme. However, when the attack strength is high (ϵ = 16/255), our mitigation scheme performs significantly better than adversarial training.

V. Conclusions and Future Work

Nowadays, traffic sign recognition systems in autonomous vehicles are predominantly based on DNNs. These DNNs can be compromised through various attacks, including data poisoning. In this paper, we exploited the error-minimizing attacks to poison DNNs used for traffic sign recognition during training. However, our experiments demonstrated that the attack is effective only when the entire training dataset is poisoned. Furthermore, we showed that the error-minimizing attacks can be mitigated by employing a data-augmentation-based training method. The proposed mitigation scheme was more effective than the computationally expensive adversarial training approach for mitigating the error-minimizing attacks. Furthermore, our findings highlighted the necessity of utilizing advanced training techniques for traffic sign recognition systems to enhance their resilience against data poisoning attacks. In the future, we aim to evaluate the robustness of our mitigation scheme on diverse datasets and to consider advanced model architectures in the detection model. We also plan to improve the error-minimizing attacks by addressing the limitations identified in this study.

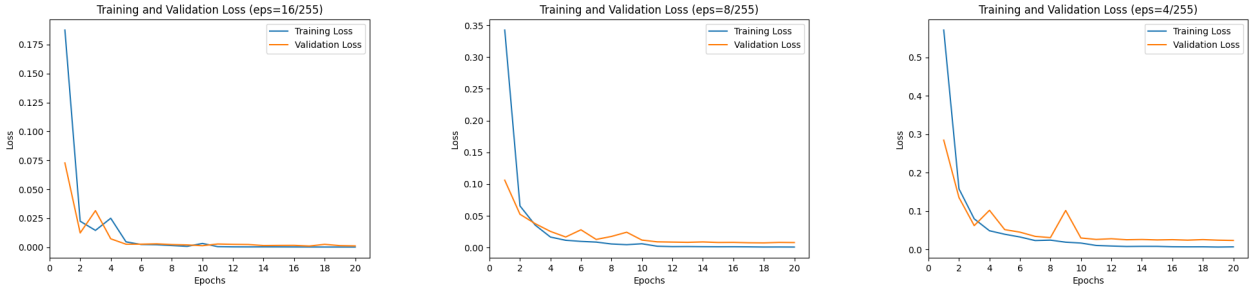
VI. Acknowledgment

We acknowledge NSF for partially sponsoring the work under grants #1620868 with its REU, #2228562, and #2236283. We also thank Cyber Florida for a seed grant.

References

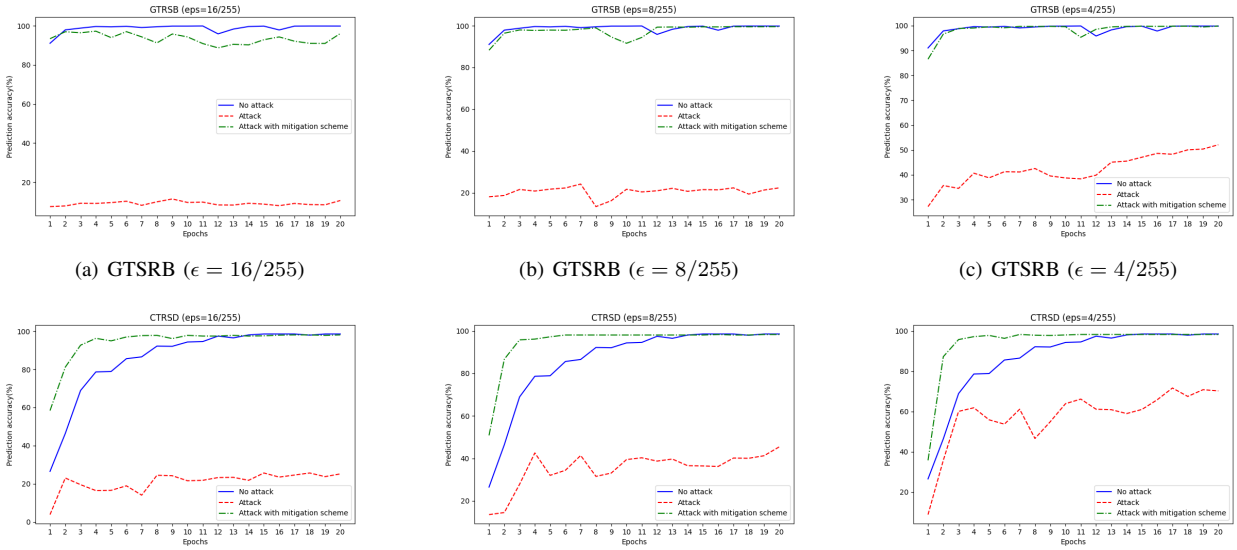
[1] "Tesla's model 3." [Online]. Available: https://www.tesla.com/en_hk/model3

[2] H. Cao, L. Yuan, G. Xu, Z. He, Z. Fang, and Y. Fang, "Secure traffic sign recognition: An attention-enabled universal image inpainting mechanism against light patch attacks," *CoRR*, 2024.



(a) $\epsilon = 16/255$ (b) $\epsilon = 8/255$ (c) $\epsilon = 4/255$

Fig. 5: Training and validation loss curves of the detection model.



(a) GTSRB ($\epsilon = 16/255$) (b) GTSRB ($\epsilon = 8/255$) (c) GTSRB ($\epsilon = 4/255$)

(d) CTSRD ($\epsilon = 16/255$) (e) CTSRD ($\epsilon = 8/255$) (f) CTSRD ($\epsilon = 4/255$)

Fig. 6: Prediction accuracy during training with and without error-minimization attacks, as well as after the application of our mitigation scheme.

[3] X. R. Lim, C. Lee, K. Lim, T. S. Ong, A. Alqahtani, and M. Ali, "Recent advances in traffic sign recognition: Approaches and datasets," *Sensors*, vol. 23, 2023.

[4] W. Jiang, H. Li, S. Liu, X. Luo, and R. Lu, "Poisoning and evasion attacks against deep learning algorithms in autonomous vehicles," *IEEE Trans. Veh. Technol.*, vol. 69, no. 4, pp. 4439–4449, 2020.

[5] J. Lin, L. Dang, M. Rahouti, and K. Xiong, "ML attack models: Adversarial attacks and data poisoning attacks," *arXiv:2112.02797*, 2021.

[6] H. Huang, X. Ma, S. M. Erfani, J. Bailey, and Y. Wang, "Unlearnable examples: Making personal data unexploitable," in *International Conference on Learning Representations*, 2021.

[7] A. Madry, "Towards deep learning models resistant to adversarial attacks," *arXiv:1706.06083*, 2017.

[8] V. S. Barletta, C. Catalano, M. Colucci, M. De Vincentiis, and A. Piccinno, "Measuring the risk of evasion and poisoning attacks on a traffic sign recognition system," in *IEEE International Workshop on Technologies for Defense and Security (TechDefense)*, 2024.

[9] N. Papernot, P. D. McDaniel, I. J. Goodfellow, S. Jha, Z. B. Celik, and A. Swami, "Practical black-box attacks against machine learning," in *Asia Conference on Computer and Communications Security*, 2017.

[10] W. Wang, Y. Yao, X. Liu, X. Li, P. Hao, and T. Zhu, "I can see the light: Attacks on autonomous vehicles using invisible lights," in *Conference on Computer and Communications Security*, 2021, pp. 1930–1944.

[11] T. Limbasiya, K. Z. Teng, S. Chattopadhyay, and J. Zhou, "A systematic survey of attack detection and prevention in connected and autonomous vehicles," *Veh. Commun.*, vol. 37, p. 100515, 2022.

[12] T. Hapuarachchi, J. Lin, K. Xiong, M. Rahouti, and G. Ost, "Nonlinear transformations against unlearnable datasets," *arXiv:2406.02883*, 2024.

[13] Z. Liu, Z. Zhao, A. Kolmus, T. Berns, T. van Laarhoven, T. Heskes, and M. Larson, "Going grayscale: The road to understanding and improving unlearnable examples," *arXiv:2111.13244*, 2021.

[14] Z. Liu, Z. Zhao, and M. A. Larson, "Image shortcut squeezing: Countering perturbative availability poisons with compression," in *International Conference on Machine Learning*, vol. 202, 2023, pp. 22 473–22 487.

[15] S. Fu, F. He, Y. Liu, L. Shen, and D. Tao, "Robust unlearnable examples: Protecting data privacy against adversarial learning," in *The Tenth International Conference on Learning Representations*, 2022.

[16] S. Houben, J. Stallkamp, J. Salmen, M. Schlipsing, and C. Igel, "Detection of traffic signs in real-world images: The German Traffic Sign Detection Benchmark," in *International Joint Conference on Neural Networks*, no. 1288, 2013.

[17] "Chinese traffic sign dataset." [Online]. Available: https://nlpr.ia.ac.cn/pal/trafficdata/recognition.html

[18] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.

[19] PyTorch, "Illustration of transforms," accessed on Dec 26, 2024. [Online]. Available: https://pytorch.org/vision/0.20/auto_examples/transforms/plot_transforms_illustrations.html

[20] J. Xiao, W. Guo, and J. Liu, "Exploring data augmentation effects on a singular illumination distribution dataset with colorjitter," in *International Conference on Image Processing and Media Computing*, 2024.