

A Software Tool for Securing Deep Learning against Adversarial Attacks for CAVs

Final Report

by

Pierluigi Pisu, Ph.D., Clemson University
Gurcan Comert, Ph.D., Benedict College
Negash Begashaw, Ph.D., Benedict College
Chunheng Zhao, Ph.D. candidate, Clemson University
Kalpiti Vadnerkar, Ph.D. candidate, Clemson University

Contact information

Pierluigi Pisu, Ph.D.
4 Research Drive, Greenville, SC 29607
Clemson University
Phone: (864) 283-7227; E-mail: pisup@clemson.edu

November 2024



Center for Connected Multimodal Mobility (C²M²)



Benedict College



THE CITADEL
THE MILITARY COLLEGE OF SOUTH CAROLINA

SCState
UNIVERSITY



UNIVERSITY OF
SOUTH CAROLINA

200 Lowry Hall, Clemson University
Clemson, SC 29634

DISCLAIMER

The contents of this report reflect the views of the authors, who are responsible for the facts and the accuracy of the information presented herein. This document is disseminated in the interest of information exchange. The report is funded, partially or entirely, by the Center for Connected Multimodal Mobility (C²M²) (Tier 1 University Transportation Center) Grant, which is headquartered at Clemson University, Clemson, South Carolina, USA, from the U.S. Department of Transportation's University Transportation Centers Program. However, the U.S. Government assumes no liability for the contents or use thereof.

Non-exclusive rights are retained by the U.S. DOT.

ACKNOWLEDGMENT

The authors would like to acknowledge the Center for Connected Multimodal Mobility (C²M²), which is a Tier 1 University Transportation Center, for supporting this research.

A Software Tool for Securing Deep Learning against Adversarial Attacks for CAVs, 2024

Technical Report Documentation Page

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.	
4. Title and Subtitle A Software Tool for Securing Deep Learning against Adversarial Attacks for CAVs		5. Report Date November, 2024	
		6. Performing Organization Code	
7. Author(s) Pierluigi Pisu, Ph.D.; ORCID: 0000-0003-4266-1336 Gurcan Comert, Ph.D.; ORCID: 0000-0002-2373-5013 Negash Begashaw, Ph.D.; ORCID: 0000-0002-4192-3069 Chunheng Zhao, Ph.D. candidate; ORCID: 0000-0002-3121-4779 Kalpit Vadnerkar, Ph.D. candidate; ORCID: 0009-0002-4230-6633		8. Performing Organization Report No.	
9. Performing Organization Name and Address Department of Automotive Engineering Clemson University 4 Research Drive, Greenville, SC 29607		10. Work Unit No.	
		11. Contract or Grant No.	
12. Sponsoring Agency Name and Address Center for Connected Multimodal Mobility (C ² M ²) USDOT Tier 1 University Transportation Center Clemson University 200 Lowry Hall, Clemson Clemson, SC 29634		13. Type of Report and Period Covered Final Report (October 2023 – October 2024)	
		14. Sponsoring Agency Code	
15. Supplementary Notes			
16. Abstract <p>This project focuses on the technological transfer of a robust perception algorithm previously developed to mitigate adversarial attacks, transforming it into a practical software tool with an intuitive interface. The initiative builds upon the prior project, <i>Securing Deep Learning against Adversarial Attacks for Connected and Automated Vehicles</i>, which successfully introduced a deep ensemble network combining discriminative and generative models to counter adversarial examples. This innovative approach utilized a causal latent graph embedded in a Bayesian model to estimate adversarial perturbations, demonstrating superior accuracy and robustness when trained solely on clean data. The current project advances this work by prioritizing usability and accessibility, emphasizing the development of a graphical user interface (GUI) to facilitate the generation, training, and testing of adversary-resilient neural networks. The anticipated outcome is a tool that democratizes access to robust AI systems, enabling diverse users to enhance the security of perception systems in various applications, regardless of their expertise in deep learning.</p>			
17. Keywords Connected and Autonomous Vehicles; Security; Deep Learning; Adversarial Examples; Classification.		18. Distribution Statement No restrictions.	
19. Security Classif. (of this report) Unclassified	20. Security Classif. (of this page) Unclassified	21. No. of Pages 10	22. Price NA

Table of Contents

DISCLAIMER	ii
ACKNOWLEDGMENT	iii
LIST OF FIGURES.....	vi
EXECUTIVE SUMMARY.....	1
CHAPTER 1	2
Introduction.....	2
CHAPTER 2	3
Literature Review.....	3
2.1 Web-Based Interface	3
2.1 Docker-Based Deployment	3
CHAPTER 3	5
Method.....	5
3.1 Data Management Module	5
3.2 Model Configuration Module	5
3.3 Model Training Module.....	6
3.4 Model Evaluation Module	7
3.5 Help and Support Module.....	8
CHAPTER 4	9
Conclusions	9
REFERENCES.....	10

LIST OF FIGURES

Figure 1: Overall GUI setup..... 5

Figure 2: Model configuration section..... 6

Figure 3: Model training section. 7

Figure 4: Model evaluation section..... 7

Figure 5: Help and support section. 8

EXECUTIVE SUMMARY

The overarching vision of this project is the technological transfer of the previously developed robust perception algorithm against adversarial attacks by developing a software tool that would allow for easy-to-use interface features, seamlessly adapting to various perception neural networks and automatically capable of generating the adversary resilient outputs within the desired perception system.

This proposal builds upon the PI's previously completed project "Securing Deep Learning against Adversarial Attacks for Connected and Automated Vehicles [1]." In this prior project, the goal was to develop a Deep Learning (DL) system with a tradeoff between accuracy and adversarial robustness. Recent studies find that DL is vulnerable against well-designed input samples. These misclassified samples were named adversarial examples. In the prior project, we developed a deep ensemble network for image classification based on the fusion of discriminative features and generative models. Specifically, a causal latent graph was built into a Bayesian model to model the distribution of adversarial perturbations. Experimental results showed that the proposed ensemble model achieves reduced accuracy loss against adversarial examples even when trained with only clean data. As a continuing work, the primary focus of this proposal is on the practical deployment and widespread adoption of the proposed robust deep neural networks.

The specific objectives of the proposed technology transfer consist of developing an intuitive graphical user interface (GUI) to generate, train, and test intrinsically robust neural networks to make adversarial attacks less effective, which means obtaining correct recognition results even in the presence of adversarial attacks. By incorporating a GUI, we aim to ensure that even those who might not be well-versed in the intricacies of deep learning can comfortably navigate the platform. This focus on usability is crucial as it broadens the potential user base, allowing researchers from various backgrounds to efficiently generate neural networks that are resilient against adversarial attacks.

CHAPTER 1

Introduction

Recent developments on connected and automated vehicles (CAVs) show that many companies, such as UBER and Waymo, are substantially investing in the development of perception modules based on deep learning algorithms [2]. Deep learning algorithms are susceptible to adversarial attacks aimed at modifying the input of the deep neural network (DNN) to induce misclassification, which may compromise vehicle decision-making and therefore functional safety. The project focuses on autonomous vehicle architectures with perception-planning-action pipeline [3]. The results of the perception will be used in the planning module to execute motion planning tasks. In this architecture, CAV faces the challenge of obtaining correct sensing information about the surrounding environment, including recognizing pedestrians and traffic signs.

The scope of this project is to realize a technology transfer of previously developed adversarial attack resilient perception algorithm for autonomous navigation in connected and automated vehicles. This is achieved through the following set of activities:

- Develop a graphical user interface (GUI) for automatically generating the robust perception algorithm given a baseline perception neural network including option for training and evaluation under different publicly available image data sets.
- Training students on the use of the developed GUI to foster next-generation personnel that will benefit USDOT and society at large.

As CAV technology is developing fast and going to enter the market soon, the proposed technology transfer addresses the problem of improving the resilience of CAVs to the possibility of adversarial attacks aimed at affecting the performance of the perception module of CAVs, therefore improving vehicle reliability and functional safety beyond currently adopted practices. We envision that our software tool will play an important role in securing automated vehicles and, thus, accelerating the spread of CAVs. The expected outcomes of the project fall well within the C2M2 research priority focus on artificial intelligence in multi-modal transportation cyber-physical systems. The results of this project will have a broad applicability not only to the transportation sector but also to many other applications utilizing machine learning for classification, including robotics, biometric identification, and speech recognition. Resiliency of machine learning algorithms to adversarial attacks is currently a very hot topic, and we expect to be able to leverage the project outcomes to pursue other research projects in the listed application areas, and to attract and engage industrial partners such as UBER and NVIDIA.

CHAPTER 2

Literature Review

A Graphical User Interface (GUI) is a visual method for users to interact with computer software or applications. It employs graphical components like windows, icons, buttons, and menus to facilitate user interaction with the software, enabling them to carry out a range of tasks [4]. GUIs are user-friendly tools that make it easy for the users with less technical programming skills to deploy the models and use them for their research purposes without going through technical programming processes. GUIs are employed for a variety of coding and software applications in Engineering and Science. Web Browsers and Operating systems are good examples of GUI. A variety of programming languages, such as Python, JavaScript, and C, offer several programming tools and packages to develop professional GUIs. Tkinter and Flexx are sets of tools that can render GUIs using Python. They can run Python scripts in a GUI format [5]. Implementing object detection code can be a challenging and complicated task. Using code written by other programmers and integrating it into your own scripts for deployment can be particularly difficult. On the other hand, GUI software can facilitate the use of object detectors and eliminates the need of coding and technical complexities for users. In this project, we will implement GUI software for our proposed robust object detector that allows for visualizing the impact of adversarial attack on traditional objection detectors in comparison with our robust object detector.

To develop the interface tool, we evaluated several implementation strategies to determine the most effective approach for delivering a user-friendly, accessible, and efficient platform. The two primary options we considered were a web-based interface and a Docker-based deployment.

2.1 Web-Based Interface

The web-based interface operates directly within a user's browser without requiring any additional installations or configurations, making it highly accessible.

- **Pros:**
 - **Accessibility:** No installation required; accessible from any device with a web browser.
 - **Ease of Use:** Intuitive for users; easy to update and maintain.
 - **Scalability:** Easily scalable; can handle increasing user numbers and data sizes with server adjustments.
 - **Centralized Updates:** Updates and patches can be rolled out seamlessly without user intervention.
- **Cons:**
 - **Dependence on Internet Connection:** Requires a continuous internet connection, which could limit access in areas with poor connectivity.
 - **Browser Limitations:** Performance and feature support might vary across different browsers.

2.1 Docker-Based Deployment

Docker allows applications to be packaged along with their dependencies into containers, ensuring consistency across multiple development and release cycles.

- **Pros:**
 - **Consistency:** Provides a consistent environment for all users, regardless of their underlying operating system.
 - **Isolation:** Each application runs in its own container, isolated from others,

- enhancing security.
- **Portability:** Containers can be run on any system that supports Docker, making it highly portable.
- **Cons:**
 - **Setup Complexity:** Requires users to have Docker installed and understand basic Docker operations.
 - **Resource Overhead:** Can be resource-intensive, especially for users with limited system capabilities.

After considering the different factors, we decided to implement our Machine Learning Interface Tool as a web-based application. The main reasons for this choice were accessibility and ease of use, which ensured that users could access the tool without needing to install additional software or manage complex configurations. The web-based approach also allows for rapid deployment of updates and new features, ensuring all users simultaneously benefit from enhancements, without needing to manually update their installations.

CHAPTER 3

Method

To validate and demonstrate our approach to securing deep learning models against adversarial attacks, we developed a comprehensive web-based platform using modern software development technologies. The implementation utilizes React with TypeScript for robust type safety and better development experience, along with a modular component architecture that ensures scalability and maintainability.

The application architecture is structured into five primary functional domains, each serving a specific purpose in the workflow of developing and testing robust deep learning models:

3.1 Data Management Module

The data management module serves as the foundation of the experimental pipeline, providing researchers with sophisticated tools for data handling and preparation:

- A dataset library with integrated access to standard benchmarks (e.g., CIFAR-10, CIFAR-100 [6]), detailed metadata, quick-load functionality, and sample visualization.
- A data upload system supporting flexible file formats (CSV, XLSX, JSON), real-time upload status feedback, automated validation, and preview generation.
- An interactive data preview interface with tabular views and visualization options.

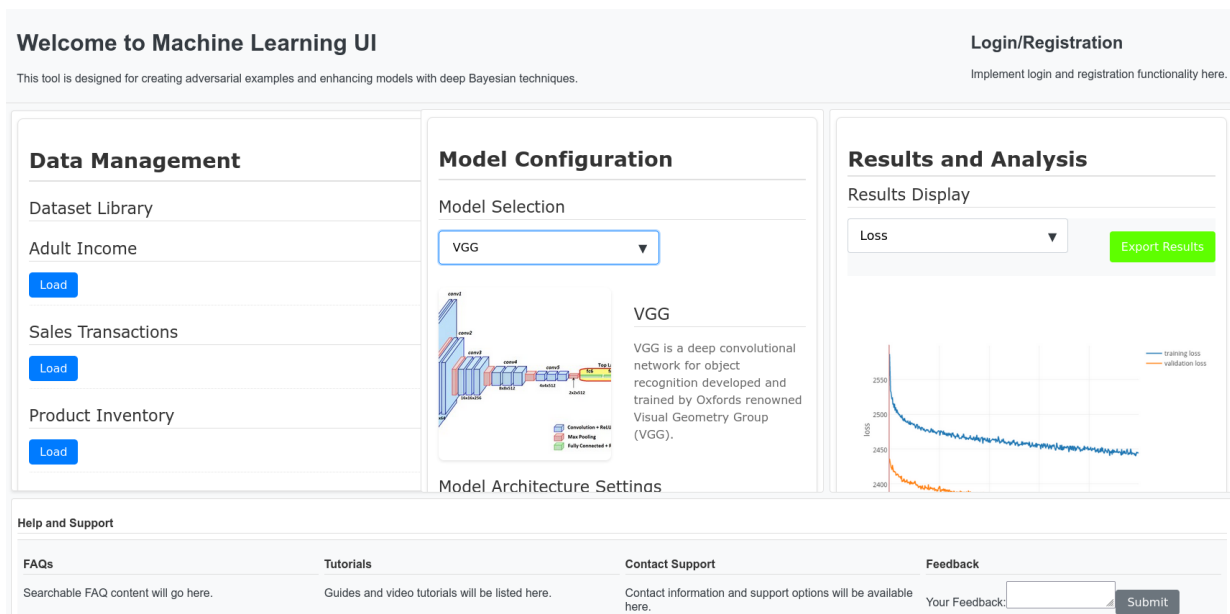


Figure 1: Overall GUI front-end.

3.2 Model Configuration Module

The Model Configuration section provides precise control over your neural network architecture and training parameters, enabling you to:

- Easily Select Models: Visualize and compare available architectures, including popular options like VGG [7], with detailed diagrams, performance characteristics, and

recommended use cases. Quickly switch between different models to find the best fit for your task.

- Fine-Tune Architectures: Adjust model parameters with granular control over feature layers, hidden layers, and encoder/decoder layers. Advanced options are available for experienced users. The system offers parameter validation and optimization suggestions.
- Optimize Training: Fine-tune the training process by adjusting batch size, learning rate, and the number of training iterations. Control dataset partitioning and cross-validation settings for optimal model performance.

Model Configuration

Model Selection

VGG

VGG

VGG is a deep convolutional network for object recognition developed and trained by Oxford's renowned Visual Geometry Group (VGG).

Model Architecture Settings

Feature Layer: Mid

Dimension of Hidden Layers: 500

Number of Encoder Layers: 2

Number of Decoder Layers: 2

Model Training Settings

Training Dataset: CIFAR-10

Number of Training Iterations: 300

Training Batch Size: 100

Figure 2: Model configuration section.

3.3 Model Training Module

The tool is specifically designed to automate the process of generating robust deep neural networks using the provided datasets and pre-trained models. The model architecture and parameters can be saved as model files and used for evaluation. The model training uses TensorFlow framework with CUDA and can be compatible with ROS for interfacing with robots and sensors.

Number of Encoder Layers:

2

Number of Decoder Layers:

2

Model Training Settings

Training Dataset:

CIFAR-10

Number of Training Iterations:

300

Training Batch Size:

100

Learning Rate:

0.00005

Apply Settings and Train

Status: Stop

Figure 3: Model training section.

3.4 Model Evaluation Module

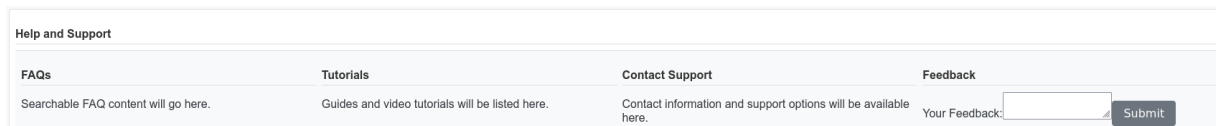
Different types of adversarial perturbation (e.g., FGSM [8], PGD [9]) are included in this tool for evaluation. After the robust neural network is built, the adversarial attacks will be performed on the neural network to test the performance. We evaluate the loss, attack success rate and clean data accuracy using Adversarial Robustness Toolbox [10].



Figure 4: Model evaluation section.

3.5 Help and Support Module

Users can view FAQs, read tutorials, and submit feedback using the interface.



The screenshot shows a web interface titled "Help and Support". It features a horizontal navigation bar with four tabs: "FAQs", "Tutorials", "Contact Support", and "Feedback". Below the tabs, there are four corresponding content areas. The "FAQs" area contains the text "Searchable FAQ content will go here." The "Tutorials" area contains "Guides and video tutorials will be listed here." The "Contact Support" area contains "Contact information and support options will be available here." The "Feedback" area contains a form with the label "Your Feedback:" followed by a text input field and a "Submit" button.

Figure 5: Help and support section.

Our prior research presented a robust image classification model. In this project, we also tested the integration of a robust regression model specifically for delineating object bounding boxes and the previously proposed classification model. We used an existing deep Bayesian regression model [11] as an enhancement to our prior work to further introduce robustness. However, due to the performance limitation of generative classifiers, we found that the generated robust image classifier didn't scale well with larger datasets like full ImageNet and COCO. The extension to these more sophisticated datasets for object detection requires more tuning and research outside the scope of this project.

CHAPTER 4

Conclusions

This project represents a significant step forward in bridging the gap between advanced adversarial robustness research and its practical application. By integrating the robust perception algorithm into an easy-to-use GUI, we aim to lower the barrier to adoption and empower a broader range of users to defend against adversarial threats. Currently, our software tool can train robust image classification models with the selection of various datasets, model architectures, and training parameters. The GUI can train and test intrinsically robust neural networks to make a variety of adversarial attacks less effective, which means obtaining correct recognition results even in the presence of adversarial attacks. This effort underscores the importance of translating cutting-edge research into deployable tools, paving the way for robust, secure perception systems in diverse fields, from autonomous vehicles to other mission-critical AI applications.

REFERENCES

- [1] <https://cecas.clemson.edu/C2M2/project-reports/>
- [2] A. Qayyum, M. Usama, J. Qadir, and A. Al-Fuqaha, "Securing connected & autonomous vehicles: Challenges posed by adversarial machine learning and the way forward," arXiv preprint arXiv:1905.12762, 2019.
- [3] Autonomous vehicles market developments, <https://www.analyticsinsight.net/report-autonomous-vehicles-market-developments-in-2019/>
- [4] Levy, S. (2023, June 6). graphical user interface. Encyclopedia Britannica. <https://www.britannica.com/technology/graphical-user-interface>
- [5] Shepherd, A. (2023). What is GUI Python and Its Alternatives!. Wondershare. <https://mockitt.wondershare.com/ui-ux-design/gui-python.html#Part3.3>
- [6] Krizhevsky, A. and Hinton, G., 2009. Learning multiple layers of features from tiny images.
- [7] Simonyan, K. and Zisserman, A., 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.
- [8] Goodfellow, I.J., Shlens, J. and Szegedy, C., 2014. Explaining and harnessing adversarial examples. arXiv:1412.6572.
- [9] Madry, A., Makelov, A., Schmidt, L., Tsipras, D. and Vladu, A., 2017. Towards deep learning models resistant to adversarial attacks. arXiv preprint arXiv:1706.06083.
- [10] <https://adversarial-robustness-toolbox.readthedocs.io/en/latest/>
- [11] Choi, J., Chun, D., Kim, H., & Lee, H. J. (2019). Gaussian yolov3: An accurate and fast object detector using localization uncertainty for autonomous driving. In Proceedings of the IEEE/CVF International conference on computer vision (pp. 502-511).