Educational Machine Learning Modules for Undergraduates in Cybersecurity

Daniel Simpson, Dr. Maanak Gupta, Department of Computer Science, Tennessee Technological University
Dr. Mahmoud Abdelsalam, Department of Computer Science, Manhattan College

Objective

Create educational modules that present machine learning concepts in a way that is accessible to undergraduate Cybersecurity students.

Introduction

Machine learning resources have advanced to the point of being useful in nearly every field in recent years, including the field of Cybersecurity. However, much of the machine learning content is generally reserved for late undergraduate or graduate level courses.

The goal of this research is to provide undergraduate Cybersecurity and machine learning course content in the form of independent but complementary teaching modules. Each module includes lectures about a given topic relating to both Cybersecurity and machine learning as well as at least one lab assignment. These labs focus on some implementation of the materials covered in the lecture and seek to present machine learning content to Cybersecurity students in an accessible and applicable way. Alternatively, these modules can also be used in introductory undergraduate data science courses to demonstrate to data science students ways in which their field can intersect with Cybersecurity. These modules do not have to be implemented into existing courses to be useful, as their stand-alone nature makes them potentially useful for workshops for undergraduates or independent study.

Dataset

The samples used in labs 5, 6a, and 6b come from the Microsoft BIG 2015 dataset. These are sterilized bytecode samples of malware belonging to eight families of malware: Ramnit, Lollipop, Kelihos version 1, Kelihos version 3, Vundo, Tracur, Gatak, and Obfuscator_ACY. Further processing of these samples leaves the students with a midsized dataset of the samples converted to .PNG images of the bytecode, allowing them to be worked with based on texture.

Methods

Of the ten labs in development corresponding to six modules covering areas of machine learning, the three labs covered here relate to malware classification. Malware classification is a particularly useful intersect between the machine learning and Cybersecurity fields as it seeks to solve a complex problem with cumbersome feature selection. Successful machine learning solutions to the problems of detecting and classifying malware are applicable to work happening in the real world and will prepare students for industry. The labs provided with these educational modules make use of virtualization with Oracle's VirtualBox virtual machine software. The students are provided with a Ubuntu 20.04 environment in which to perform the labs in Python, making use of the TensorFlow, Keras, Scikit-learn, Pandas, and Cleverhans libraries to complete various tasks. In sequence the modules complement one another, but the environments are provided such that students are not at a disadvantage for only having been assigned one of the labs independently.

Lab 5: Malware Classification

In the lectures associated with this module, students will learn about the general problem of malware classification. They will be introduced to the general types of malware such as viruses, worms, trojans, ransomware, and more. This includes some basic information about the distinguishing features of each and their methods of spreading. Following this, the students will be introduced to the concept of malware families, sets of malware variants that are believed to have been derived from the same source samples.

Classification of these is discussed and the depth of this classification problem is expanded with the introduction of polymorphic and metamorphic malware which are capable of obfuscating themselves in different ways. Having been introduced to some of the traditional features used to classify malware into various families, the students will learn about Convolutional Neural Networks, a type of deep learning model used to classify images.

Given this information, the students will build a CNN model in the associated lab that will try to abstract away from the complex feature set that is traditionally used to identify malware and to instead classify them based on the bytecode samples having been converted to images. A classified sample of this can be seen in figure 1. The intent of this lab is to classify the malware samples based on the visual texture of their bytecode rather than traditional methods. This serves two purposes. The first is that it allows the students to interact with a very complex problem using a simpler technique in order to make this content much more accessible to undergraduate level students. The second is that it demonstrates an unconventional method of classifying malware that the students may not have encountered in previous studies. The mechanics of the lab itself focuses on adding layers to the convolutional model and tuning it as they train it.

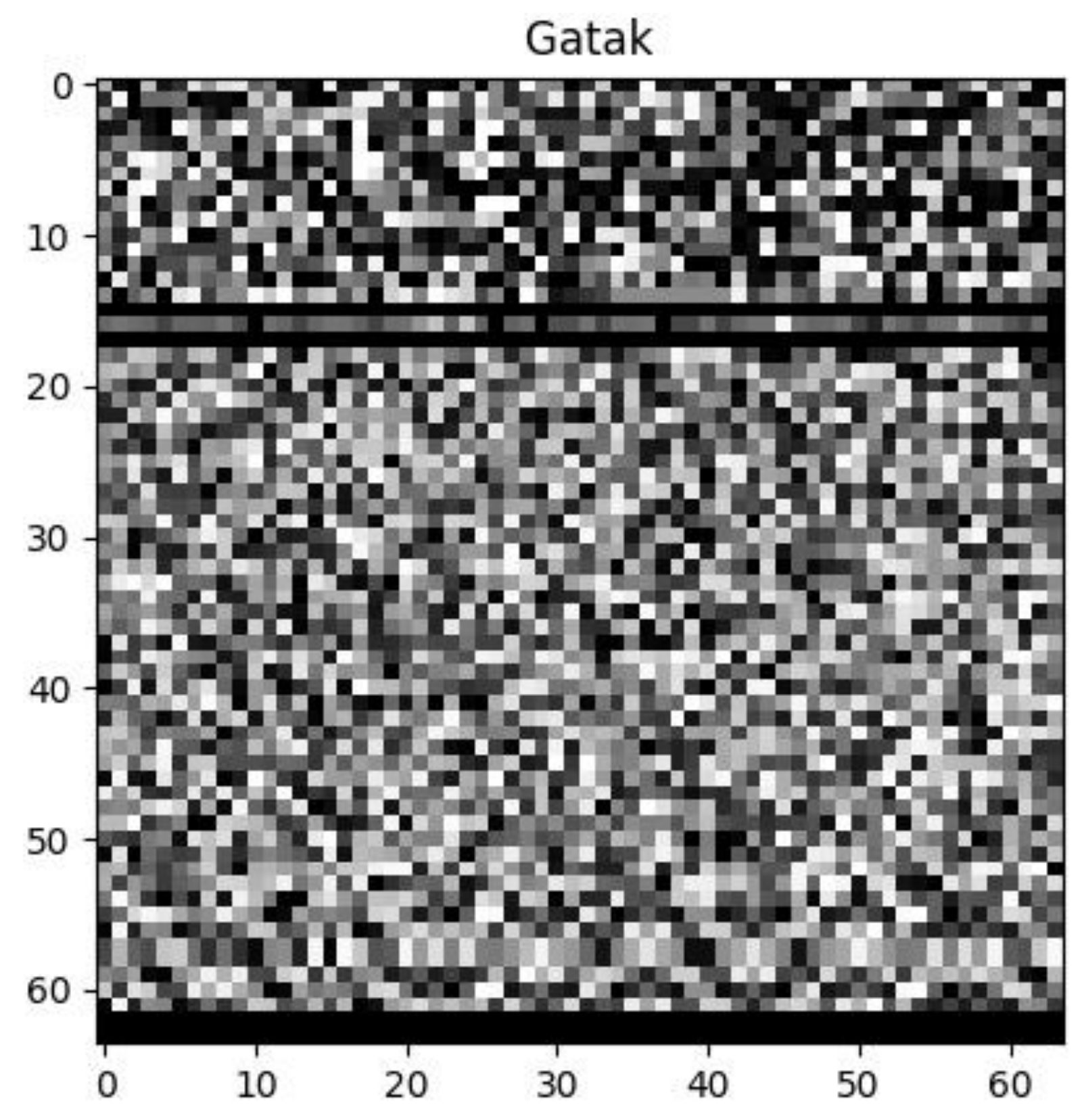

Figure 1. A 64x64 sample of Gatak malware correctly classified by the CNN classifier in lab 5

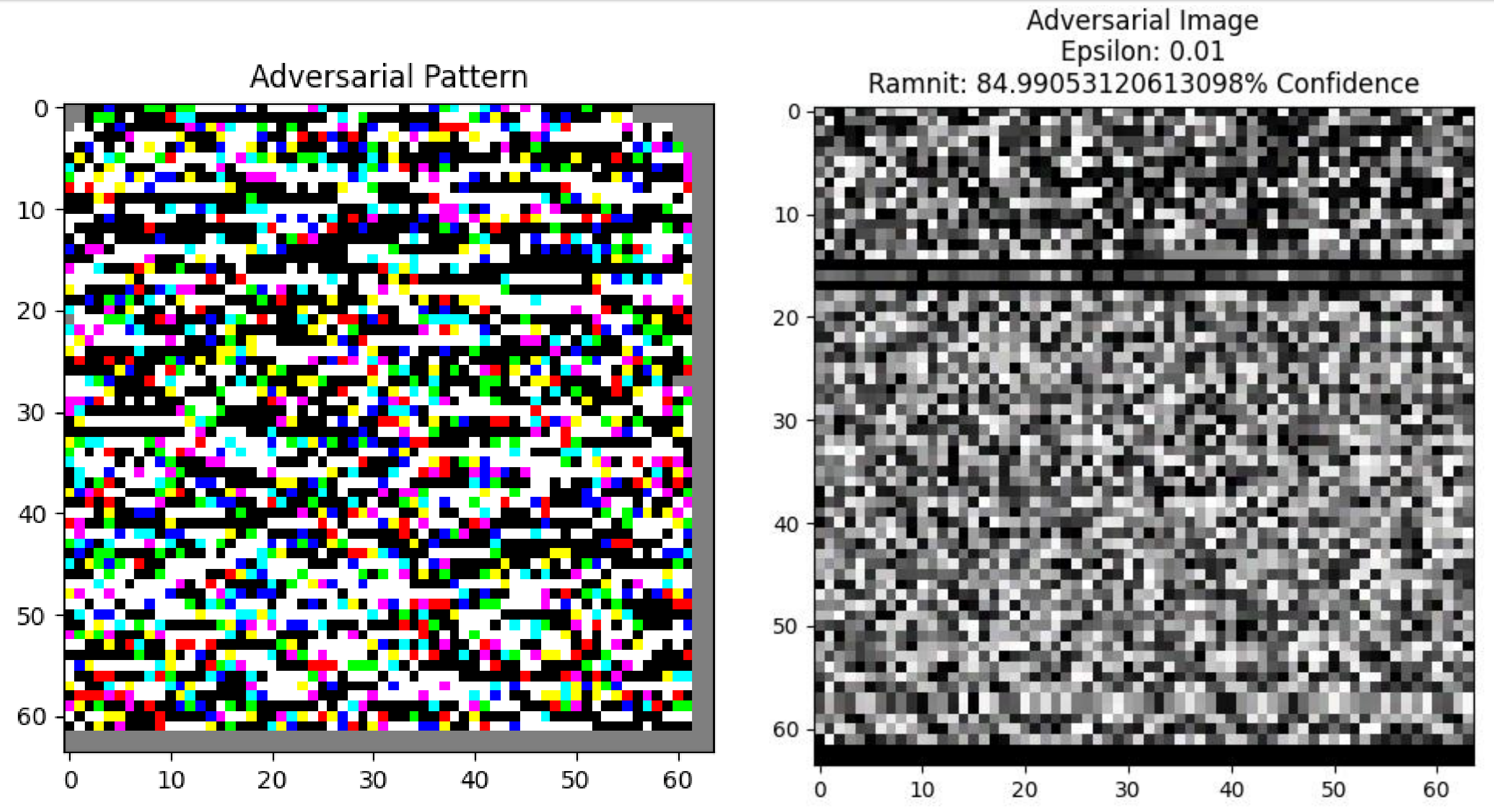

Figure 2 (left), and Figure 3 (right). Figure 2 depicts a perturbation calculated with respect to the model's loss when classifying the Gatak sample in Figure 1. Figure 3 depicts an adversarial sample generated by applying this perturbation to the original Gatak sample.

Lab 6a: Adversarial Attack with FGSM

In sequence, module 6 on adversarial machine learning would follow module 5 on malware classification. Students working on module 6 independently will not be at a disadvantage, however, as a functional model as would have been produced at the end of module 5 is provided. In the lectures for this module, students will learn that machine learning models can be attacking using adversarial inputs and various attack methods. Background information regarding the CNN model, the dataset provided, and the basic introduction to malware covered in module 5 is also available in module 6 in case this is being used independently.

Students will study in particular the Fast Gradient Sign Method of adversarial attack, which perturbs the sample images in minimal ways with respect to the model's loss. A sample perturbation is visible in figure 2. This allows for adversarial images to be misclassified by the model but to not have noticeable visual differences from a non-adversarial sample. In the lab 6a, students will create such an adversarial image in an attempt to make their model misclassify it. They will experiment with different values for epsilon, the weight for the perturbation to be applied to the original image, as they fine-tune the attack to a minimal value that still successfully attacks the model. An adversarial sample generated with the perturbation from figure 2 applied to the Gatak sample from figure 1 is visible in figure 3 as having been misclassified by the model with a high confidence value. This will demonstrate vulnerabilities of the model used in module 5 and similar such models as well as prepare students for the hardening of that model to be performed in lab 6b, the adversarial training lab under development.

Lab 6b: Adversarial Training

Still in development, lab 6b builds on 6a as students will employ the Fast Gradient Sign Method as implemented in the Cleverhans adversarial library to generate an adversarial dataset with which to further train the model. Students will work with dataset generating mechanisms as well as training mechanisms in order to harden the classifier against the FGSM attack demonstrated in lab 6a. To maintain the independence of the modules, all components necessary for lab 6b will be provided.

Future Work

Functional examples of labs 5 and 6a are complete as of the time of this writing, with 6b still under development. Classroom testing is projected to begin in the Fall semester of 2021, during which time feedback on the modules in their current form will be gathered and their effectiveness will be assessed. Adjustments and modifications will be made as needed, but these will require testing before they can be undertaken. Future work includes the completion of lab 6b, classroom testing in the fall, the gathering of feedback from students and professors using these modules, and modifications made in response to this feedback.

References

Abadi, Mart et al. "Tensorflow: A system for large-scale machine learning." *CoRR*, 2016, 1605.08695. https://arxiv.org/abs/1605.08695. Accessed 28 Oct 2020.

Goodfellow, Ian et al. "Explaining and Harnessing Adversarial Examples." arXiv, 2015, 1412.6572. https://arxiv.org/abs/1412.6572v1. Accessed 22 Jan 2021.

Google Brain Team. "Adversarial example using FGSM." Tensorflow.org, 19 March 2021. https://www.tensorflow.org/tutorials/generative/adversarial_fgsm. Accessed 22 Jan 2021.

Mallet, Hugo. "Malware Classification using Convolutional Neural Networks – Step by Step Tutorial." Towards Data Science, 27 May 2020. https://towardsdatascience.com/malware-classification-using-convolutional-neural-networks-step-by-step-tutorial-a3e8d97122f. Accessed 30 Oct 2020.

Papernot, Nicolas et al. "Technical Report on the CleverHans v2.1.0 Adversarial Examples Library." CoRR, 2016, abs/1610.00768. http://arxiv.org/abs/1610.00768. Accessed 29 Jan 2021.

Ronen, Royi and Corina Feuerstein. "Microsoft Malware Classification Challenge (BIG 2015)." Kaggle, 3 February 2015, https://www.kaggle.com/c/malware-classification/overview. Accessed 13 Oct 2020.

Acknowledgments

This work was funded by the National Science Foundation under grants 2025682 and 2025686 in collaboration between Tennessee Technological University and Manhattan College.

Tennessee Technological University 2021