Adversarial Machine Learning and Its Impact on Automation Systems

Rech Leong Tian Poh

2427218L

Proposal

Motivation

With the increasing trend of integrating Machine Learning models in Automated Systems, the need for security becomes a major concern. This is because Machine Learning systems has further made unsupervised learning and automation become possible, and yet they may contain sensitive personal data that should not be leaked. Safety also becomes a concern, especially when our lives are in the hands of these Machine Learning systems, such as Driverless Vehicles. With that being said, data scientists and engineers are desperate in the search for defensive measures that can be taken in order to minimize the chances and the potential damage of Adversarial Attacks against Machine Learning systems. In order to do that, one must first understand the vulnerabilities of Machine Learning systems, and how can they be exploited and turned into mass attacks on potentially life threatening or security compromising applications.

Aims

The aim is to develop a targeted adversarial attack on a facial recognition model such that for any given untampered input image, the model returns the true class of the image, and for any given tampered input image, the model returns the same false class of the image. In doing so, the attack would achieve success in both stealth and ease of use, thus indicating the extent to which vulnerabilities of Machine Learning systems can be exploited.

Progress

- Performed Research of Adversarial Attacks against object detection and facial recognition Machine Learning models
- Attended AI Security Workshop for greater insight on attacks against Machine Learning Systems
- Chosen scope of Machine Learning system to be Facial Recognition
- Chosen type of adversarial attack to perform (Targeted)
- Chosen vulnerabilities of the target facial recognition model to exploit
- Successfully generated adversarial examples to be input into the trojaned facial recognition model
- Successfully written code to perform model retraining

- Successfully written code to produce personalized model
- Successfully retrained a convolutional neural network – facial recognition model to suit own requirements

Problems and risks

Problems

In order to implement the attack on a facial recognition model, plenty of execution time had to be allowed for both of the following programs/scripts:

- Generation of adversarial examples
- Retraining of the facial recognition model

Execution time of any of the above could take anywhere between 18 hours to 29 hours, depending on input size and complexity.

Retraining of the model also requires very high computational resources for which the school had limited access to. As such, I had to employ the use of Google Cloud Compute Engine Virtual Machines to develop the simulated attack. However, a new account had to be created after full consumption or expiry of the free credits.

In order to fully understand the scope and application of the attack, one first had to have prior knowledge of machine learning systems, which I did not to begin with. The initial research stages were not only to find out more about the types of attacks, but also to familiarize myself with Machine Learning in general as well.

Risks

Classification of input images could be inaccurate. In order to mitigate this, further retraining needs to be done to further increase the accuracy of misclassification.

Effectiveness of the adversarial images could be insufficient to produce the desired output. As such, in order to mitigate this issue, the algorithm to produce the adversarial images must be more robust and complex.

Plan

Week	Action Items
16 Dec 19 – 20 Dec 19	- To finish retraining of the facial recognition model - To be able to select target class to misclassify

23 Dec 19 – 27 Dec 19	- To finish writing Literature Review - To finish generating adversarial examples for personalized input images
30 Dec 19 – 3 Jan 20	- To continue implementation to generate more robust adversarial images - To further increase accuracy of misclassification via retraining
6 Jan 20 – 10 Jan 20	- To complete the implementation of trojan attack - To export all source code and models used into a portable folder for ease of subsequent execution
13 Jan 20 – 17 Jan 20	- To perform evaluation of the attack with different transparency of trojan watermarks - To assess the robustness of adversarial images on the trojaned model - To draw an architecture diagram of the overall attack process (deliverable)
20 Jan 20 – 4 Feb 2020	- To write the first draft of the final report - To submit the report draft to external lecturer (Effective writing class) as well as to Prof. Sye Loong for vetting
6 Feb 2020 – 13 Feb 2020	- To compile feedback from lecturers and write up the second draft of the final report, to be submitted for vetting again
15 Feb 2020 – End of Project period	- To finalize report - To Export all minimal software artifacts to perform the trojan attack and upload to Github - To finish up all other documentation and prepare for submission