COMP 430/530: Data Privacy and Security - Fall 2020
Homework Assignment # 4

Question 1. [20 pts] Read the paper "Exploiting Machine Learning to Subvert Your Spam Filter" by Nelson et al., which can be accessed at this link: `https://people.eecs.berkeley.edu/~tygar/papers/SML/Spam_filter.pdf`. This is one of the early works on attacking statistical machine learning-based spam filters. Then, answer the following questions.

(a) Is this a training-time attack or test-time attack?

(b) The attack in this paper best falls under which of the following categories (as covered in the lecture): privacy of the training data, confidentiality of the model, integrity of the model, integrity of the test data? Pick the most relevant and explain why.

(c) As stated in Section 3.4, an enabling observation behind the attack is that: "the spam scores of distinct words do not interact; that is, adding a word w to the attack does not change the score $f(u)$ of some different word $u \neq w$". Based on the explanation of the learning method in Section 2.3 and the equations therein, explain how the authors arrive at this observation.

(d) Describe how the above observation is used when generating an attack payload/strategy.

(e) Explain the RONI strategy to defend against the attack.

(f) Explain how the *dynamic threshold* technique is used to defend against the attack.

(g) In the lectures, we discussed that outlier detection may also be a potential defense. How can outlier detection be applied to this paper's scenario? In particular, the authors admit what aspect/feature of the attack payload may make it possible for the attack to be detected?

Question 2. [15 pts] Watch Nicolas Papernot's talk on adversarial examples in Usenix Enigma 2017: `https://www.youtube.com/watch?v=hUukErt3-7w`. Then, answer the following questions.

(a) What is meant by "transferability" in the context of adversarial examples? Define in one or two sentences.

(b) What are "cross training data transferability" and "cross technique transferability"? How are they different?

(c) Explain how transferability is used to create black-box attacks on remotely hosted models.

(d) According to Papernot, are ensembles effective in defending against adversarial examples? What experimental support does he show regarding this?

(e) According to Papernot, how are adversarial examples related to AI safety? Give a concrete example of how robustness to adversarial examples is connected to AI safety.

Question 3. [20 pts] Read the paper "Audio Adversarial Examples: Targeted Attacks on Speech-to-Text" by Carlini and Wagner, which can be accessed at this link: `https://people.eecs.berkeley.edu/~daw/papers/audio-dls18.pdf`. Then, answer the following questions.

(a) Is this a training-time attack or test-time attack?

(b) The attack in this paper best falls under which of the following categories (as covered in the lecture): privacy of the training data, confidentiality of the model, integrity of the model, integrity of the test data? Pick the most relevant and explain why.

(c) What is the difference between targeted and untargeted audio adversarial examples? Give an example for each.

(d) In image adversarial examples, we measure the amount of perturbation using the number of pixels changed or how much the (colors of the) pixels were modified. How does this paper quantify

perturbation in audio adversarial examples?

(e) Give a technical description of how the targeted attack from Sections 3-B and 3-C is later used to HIDE speech, i.e., the user speaks but the speech recognition system does not recognize it.

(f) In the lectures, we discussed 3 defense strategies against adversarial examples. Describe how each defense strategy could be applied to defend against the attack presented in this paper.

Question 4. [45 pts] You are given the Iris dataset (*iris.csv*) with 4 features: sepal length, sepal width, petal length, petal width. The label is "variety". You are also given starter code in Python (*hw4_models.py*) to split the Iris dataset into training and test sets (60%/40% split) and build 3 supervised ML models: Decision Tree, Logistic Regression, Support Vector.

You can choose to build on top of *hw4_models.py*, or you can choose to work in your preferred programming language. If you choose to work in your preferred programming language, you must ensure that your feature set, train/test split ratio, and model types (Decision Tree, Logistic Regression, Support Vector) are the same as *hw4_models.py*.

(a) Implement a label flipping attack: Given n, your attack should perform random label flipping on $n\%$ of the training data. Execute your attack with $n = 5\%, 10\%, 20\%, 40\%, 60\%$. How do the accuracy values of the ML models change according to n? Does the attack impact all 3 ML models equally or are the impacts different? Provide the necessary implementation, experimental results, and discussion to answer these questions.

Note: You may need to repeat your experiments with each n multiple times to account for the impact of randomness in random label flipping.

(b) Perform a clean-label poisoning attack. Describe your attack strategy and measure its impact in terms of: % of instances poisoned vs drop in ML model accuracy. Which attack causes higher accuracy drop: the clean-label poisoning attack you just implemented or the label flipping attack from part (a)?

(c) Perform a backdoor attack by adding new instances into the training set. What is your trigger pattern? How many instances did your attack need to add in order to be successful? Experimentally demonstrate (using many trials) that your attack is working.

Note: In the instructor's experience, you can achieve a backdoor with close to 100% success rate with as few as 5 additions.

(d) Implement a function called *evade_model(model, test_instance)* to perform an evasion attack.

Inputs of the function: (i) *model* is the trained ML model, e.g., DT, LR, SVC. (ii) *test_instance* is the instance for which we want to evade correct classification.

Say that *model* is classifying *test_instance* as "x". *evade_model* should perturb *test_instance* a small amount such that *model* classifies the perturbed instance as anything other than "x". After finishing, *evade_model* should return the perturbed instance and its resulting class.

Note: There are multiple design decisions and no single correct answer. Example: Will you perturb multiple features simultaneously or perturb features one by one or perturb only one feature? Describe and justify your design decisions as you implement them, and report how much perturbation was necessary to achieve evasion (as long as they are close to our reference implementation, you will get full credit).

(e) Experimentally test whether ensemble learning (i.e., using an ensemble of the 3 ML models given to you) is an effective solution to the evasion attack you implemented in part (d). Your ensemble should use majority vote to assign labels. In case votes are tied, pick randomly.