AICS 2019 Workshop Challenge Problem: Malware Classification under Adversarial Conditions

Machine learning capabilities have recently been shown to offer astounding ability to automatically analyze and classify large amounts of data in complex scenarios, in many cases matching or surpassing human capabilities. However, it has also been widely shown that these same algorithms are vulnerable to attacks, known as adversarial learning attacks, which can cause the algorithms to misbehave or reveal information about their inner workings. In general, attacks take three forms: a) data poisoning attacks inject incorrectly or maliciously labels data points into the training set so that the algorithm learns the wrong mapping, 2) evasion attacks perturb correctly classified input samples just enough to cause errors in classification and 3) inference attacks which repeatedly test the trained algorithm with edge-case inputs in order to reveal the previously hidden decision boundaries. Protection against adversarial learning attacks include techniques which cleanse training sets of outliers in order to thwart data poisoning attempts [1], and methods which sacrifice up-front algorithm performance in order to be robust to evasion attacks [2]. As machine learning based AI capabilities become incorporated into facets of everyday life, including protecting cyber assets, the need to understand adversarial learning and address it becomes clear.

In support of AICS, we have developed a cyber security-related challenge problem and associated data set for use by researchers. We believe that providing a challenge problem through AICS will help focus the community toward a common technical goal and will contribute to the lively exchange of ideas at the workshop.

Task and Dataset

The proposed task for this challenge is to construct AI-based malware classifiers that are robust to adversarial evasion attacks. We define the **malware classification** task as determining the class of a sample that has previously been identified as malware. For this challenge problem, the data consists of 15,670 instances split into five classes according to a subset of malware types: Virus, Worm, Trojan, Packed Malware, and AdWare. For each sample, a dynamic analysis on a Windows virtual machine was performed and the sequence of Windows API calls was extracted. Additional processing was performed to collect unigrams, bigrams, and trigrams of API calls. We make the following assumptions about the dataset:

- All instances are malware
- The classes cover all instances under consideration
- Each instance has a single unambiguous label

DISTRIBUTION STATEMENT A. Approved for public release. Distribution is unlimited. This material is based upon work supported by the United States Air Force under Air Force Contract No. FA8702-15-D-0001. Any opinions, findings, conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the United States Air Force.

In the released datasets, the class labels and API calls are obfuscated as integers, but the correspondence between any particular API call as a unigram and as part of a bigram or trigram is preserved. See below for an example.

Data Schema

The feature matrix for the 2019 AICS challenge problem is represented as an NxM sparse matrix and serialized in JSON format with the following schema:

```
{
        “data”: JSON_ARRAY(K),
        “row_index”: JSON_ARRAY(K),
        “col_index”: JSON_ARRAY(K),
        “col_schema”: JSON_ARRAY(M),
        “shape”: [N, M],
        “labels”: JSON_ARRAY(N)
}
```

The “`data`” field gives the values of the nonzero entries of the feature matrix, while “row_index” and “column_index” give the row and column indices corresponding to these values. So that the nonzero entries of the feature matrix are specified by

feature_matrix[row_index[i], col_index[i]] = data[i]

for i=1,…,K. The “col_schema” array provides names for the columns of the feature matrix. For unigrams, a column name will be a distinct integer corresponding to that specific API call, for bigrams (and trigrams), the column name will be a pair (triple) of integers separated by a semi-colon. For example, the trigram “27;9;150” represents the sequence of API calls 27, 9, and 150. The “shape” array indicates the dimensions of the feature matrix (note that it would be possible to infer these dimensions from the other data, but we include this field for convenience). Finally, the “labels” array indicates the class of each instance (this field will be available in the training data, but will be absent in the test data).

Baseline Solutions

We used Keras on top of TensorFlow to train a feedforward neural network, with two fully-connected layers, each with 160 neurons using ReLu activation. We also trained additional baseline classifiers: a random forest and an SVM using radial basis function kernels. The performance achieved on the test set is presented in the table below:

	Accuracy	Weighted F1	Macro F1
Neural Network	0.89	0.872	0.691
Random Forest	0.919	0.913	0.808
SVM	0.908	0.895	0.757

Submission

Submissions to the challenge problem will consist of two parts: 1) a CSV file containing results from applying the developed classifier to the test set, and 2) a paper describing the developed classifier. The results file will be evaluated based on performance on the test set while the Challenge paper submission will be selected for inclusion in the AICS workshop as full published paper. Please note that challenge paper submissions have the same submission deadlines as regular papers.

AICS 2019 Workshop Adversarial Learning Challenge Problem important dates:

- Training data released: 24 August 2018
- Test data released: 29 October 2018
- Papers due: 5 November 2018
- Challenge problem results due: 5 November 2018
- Accepted papers announced: 26 November 2018

Challenge problem winners will be announced at the workshop. Participants' solutions will be evaluated based on a harmonic mean of macro-averaged F1 scores on un-attacked pristine test data and attacked test data. The baseline solution described above is intended to serve as a guide that participants can use to gauge the performance of their techniques. If you have any questions about the challenge, please send an email to aics@ll.mit.edu

Download the Dataset

Download the dataset from:
ftp://ftp.ll.mit.edu/outgoing/AICS_2019/training_data.tgz

References:

1. "Manipulating Machine Learning: Poisoning Attacks and Countermeasures for Regression Learning", M. Jagielski, A. Oprea, B. Biggio, C. Liu, C. Nita-Rotaru, B. Li, IEEE Symposium on Security and Privacy (SP), 2018, San Francisco, CA.

2. "Distillation as a Defense to Adversarial Perturbations Against Deep Neural Networks', N. Papernot, P. McDaniel, X. Wu, S. Jha, A. Swami, IEEE Symposium on Security and Privacy (SP), 2016, Oakland, CA