Title: Improving Robustness of ML Classiers against Realizable Evasion Attacks Using Conserved Features.

This paper investigates the effectiveness of conventional approach of evaluating the robustness of ML models against *evasion attacks;* to design robust ML model against the attacks that can be realized in actual malware. There are 3 main contributions in this paper: (1) The paper presents an evaluation of feature-space evasion attack model specifically using 4 ML-based approaches for PDF malware detection, For structure-based: **SL2013** and **Hidost**, for content-based: **PDFRate-R** and **PDFRate-B**. (2) It proposes a method for boosting robustness of feature-space models by identifying *conserved features.* (3) It explores the extent to which said ML robustness can be generalized to multiple distinct realizable attacks.In this paper we focus on PDF malware detection,where the **label space is binary** i.e, either a PDF file is benign (which we can code as -1), or malicious (which we can code as +1).

Evasion Attacks: Realizable Evasion attack : y = h(x) ; y: label (-1/+1), h(X):learned model, x : feature vector extracted from PDF(x ϵ X). e : malicious entity, ϕ(e) :extract feature vector,attacker transforms e into e' (preserving malicious functionality) where,ϕ(e') = x' ; x' : associated feature vector and so h(x')= will yield erroneous label. Feature-space model of Evasion attack: attacker starts with a malicious feature vector x instead of an entity and directly modify x to x' ; x' ϵ X so that y' = h(x'). C(x,x') = cost of converting x to x'. attacker is penalized for greater modifications to given feature vector x : measured using l_p norm (difference b/w original and malicious instance). **Evasion Defense:** An arbitrary attack : O(h:D) where h: classifier , D: Dataset which returns 'an evasion'. let u(h ; O(h:D)) measure the defense that we want to optimize, then defense against the attack O(h:D) is : $\max_h$ u(h ; O(h:D)).

Iterative Retraining : In particular, we use a variant of iterative retraining:
1. Start with the initial classier.
2. Execute the evasion attack for each malicious instance in training data to generate a new feature vector.
3. Add all new data points to training data (removing any duplicates), and retrain the classier.
4. Terminate after either a fixed number of iterations, or when no new evasions can be added.

Results of Robustness:(**RAR**: Realizable Attack Retraining, **FSR**: Feature-Space Ret.,**CFR**: Conserved Feature Ret.,Performance against non-adversarial data using **AUC**(area under curve))

SL2013 against EvadeML: Original Classifier: 16%, RAR:96%, FSR:62%, On non adversarial data:Original-AUC=0.9999, RAR-AUC=0.9999, FSR-AUC=0.9947. **Hidost** against EvadeML: Original Classifier: 2%, RAR: 98%, FSR: 70%, On non adversarial data:Original-AUC=0.9997, RAR-AUC=0.9997, FSR-AUC=0.9930. **PDFRate-R** against EvadeML: Original Classifier: 2%, RAR: 96%, FSR: 100%, On non adversarial data: Original-AUC=0.9998, RAR-AUC=1.0000, FSR-AUC=0.9895.**PDFRate-B** against EvadeML:Original Classifier: 100%, FSR: 100%, On non adversarial data: Original-AUC=1.0000, FSR-AUC=0.9988. **SL2013** against EvadeML: Original Classifier: 16%, RAR: 96%, CFR: 87%, CFR-JS: 100%, On non adversarial data: Original-AUC=0.9999, RAR-AUC=0.9999, CFR-AUC=0.9992, CFR-JS-AUC=0.9997. **Hidost** against EvadeML: Original Classifier:2% ,RAR:98%,CFR:100%, CFR-JS: 53%, On non adversarial data:Original-AUC=0.9997,RAR-AUC=0.9997,CFR-AUC=0.9982,CFR-JS-AUC=0.9965.
PDFRate-B against EvadeML: Org.Classifier:100%, CFR:100%, CFR-JS: 100%, non adver.data: Original-AUC=1.000, CFR-AUC=0.9999, CFR-JS_AUC =0.9997.

Conclusion: This study was specific to PDF malware detection. However,the framework is quite general, and could be used in the future to consider other similar questions, such as the effectiveness of robust deep learning against physical attacks.

Name: Vishal Ramane , UID:2019430010
Name: Harsh Barucha, UID:20194300005

Title: Improving Robustness of ML Classiers against Realizable Evasion Attacks Using Conserved Features.

Proceedings: 28th USENIX Security Symposium, August 14 to 16 2019, Santa Clara,CA,USA.