Dear Ambra, Marco, Battista, Davide, Igino, Giorgio and Fabio:

This is Gabby, from Prof. Julia Rubin's group at the University of British Columbia, Canada. I'm sorry to bother you again, but after multiple attempts, I was still unable to reproduce results similar to those reported in the Sec-SVM paper. I have re-implemented the Sec-SVM solution myself and also used the implementation proposed by Pierazzi et al. [1].

I performed all the experiments with the original feature vectors from the Drebin dataset (https://www.sec.cs.tu-bs.de/~danarp/drebin/download.html). I found there exist some differences between their feature vectors and those you used in your experiments, which lead to different feature set distribution among the top features, so I also experimented with selecting features so that the distribution becomes similar to that included in Table 2 of the paper. Nevertheless, none of my results showed the same performance reported in the paper. I hope you can help me look into that.

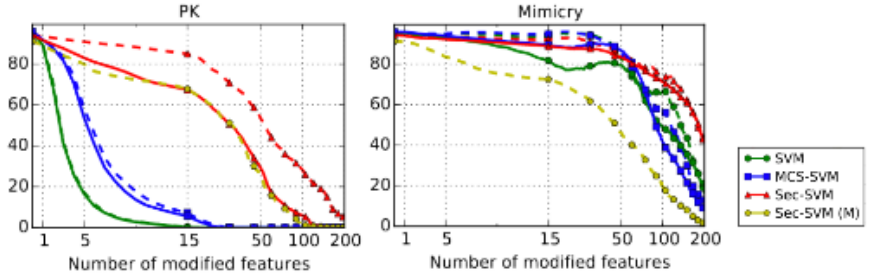
TABLE 2
Number of features in each set for SVM, Sec-SVM, and MCS-SVM. Feature set sizes for the Sec-SVM (M) using only `manifest` features are reported in brackets. For all classifiers, the total number of selected features is $d' = 10,000$.

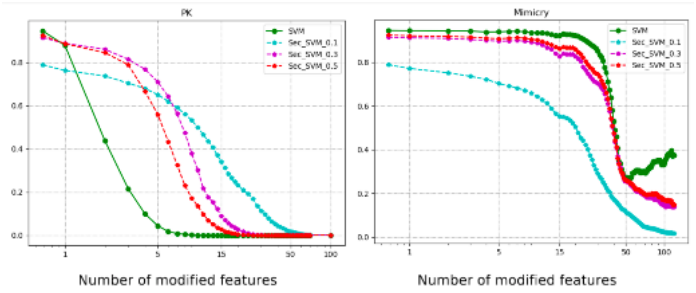
		Feature set sizes			
manifest	S_1	13 (21)	dexcode	S_5	147 (0)
	S_2	152 (243)		S_6	37 (0)
	S_3	2,542 (8,904)		S_7	3,029 (0)
	S_4	303 (832)		S_8	3,777 (0)

(Table 2 included in the paper)

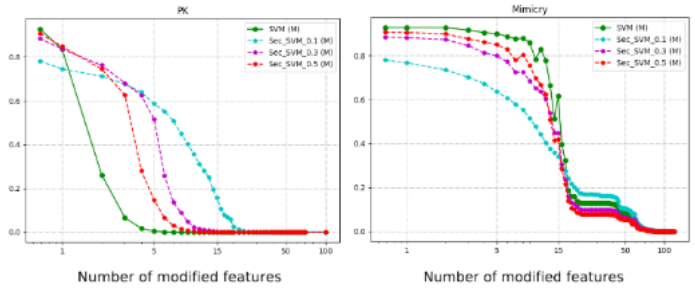
Details
--
Here are more details about what I did using the Sec-SVM implementation from Pierazzi et al. [1]. First, I randomly selected samples from the publicly-available Drebin dataset following the split reported in the paper: 30,000 samples for training, 30,000 as surrogate dataset for attacks, and the remaining for testing. I performed perfect knowledge attacks (PK) and mimicry attacks (Mimicry) with feature addition (solid lines in your experiment)**.** I measured the detection rate at 1% of false positive rate and averaged the results from 10 different runs. For each experiment setting, I tried 3 different weight constraints {[-0.5, 0.5], [-0.3, 0.3], [-0.1, 0.1]}

In the paper, the best results are achieved when model weights are bounded within [-0.5, 0.5]. Below are the figures from the paper and from my experiment with different weights. While the original SVM (the green line) performs similar in both experiments, Sec-SVM (the red line) does not and gets to almost 0 accuracy with just 15 added features under the PK attack and around 50 in the Mimicry attack. In fact, Sec-SVM seems to perform worse than "classic" SVM here.



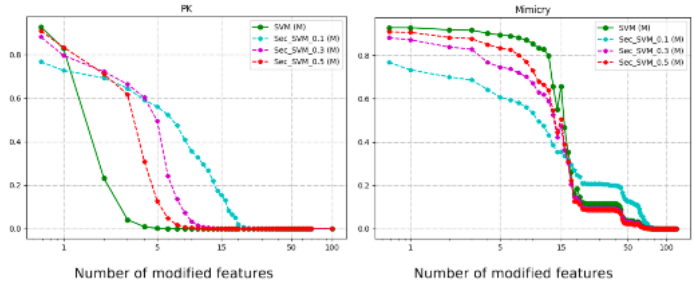


I've also tried to restrict the training to manifest features only (Sec-SVM (M)), but got similar results:



Finally, I noticed, for both experiments above I could not achieve the same feature set distributions as included in Table 2 of the paper (e.g. the biggest difference being number of features in Category S7: Sensitive API Calls: there were only around 300 in my training data in total, whereas more than 3000 features from S7 were included in the top 10,000 features in your original experiment.) I was not sure how much feature set distribution would influence the results, so I ran another experiment with manifest features only (Sec-SVM (M)), and I forced the feature set distribution to be similar to what included in Table 2. Specifically, to train a classifier with the same feature set distribution as in Table 2, I sorted the features from S1 - S4 by absolute usage difference (i.e. abs(p(x_k = 1|y = +1) – p(x_k = 1| y = -1))) and selected top K features from each feature category where K is the number of features as in Table 2.

The results from this experiment are as follows:



While Sec-SVM improved the robustness of models under PK attacks to certain degrees, I was not able to produce the model as robust as presented in the paper and I do not see any substantial improvement in the Mimicry attack case.

I was wondering if you can kindly share your insights, based on your experience, how to fix this problem.

Thank you for your time and hope to hear back from you soon!

Have a great day!

Best Regards,
Gabby