Adversarial Machine Learning against keystroke dynamics

Parimarjan Negi, Ankita Sharma

Stanford University

Project Objective

- To generate adversarial keystroke samples that make an otherwise robust classifier accept the artificially generated samples as belonging to the valid user
- Compare different classifiers vs adversarial samples and explore ways to improve defence

Data

- **CMU data set**
  - All users type a unique password (.tie5Roanl)
  - 51 users, 400 instances over 8 sessions
  - 31 timing feature recorded; key-hold time and consecutive keyup-keydown, keyup-keyup time
  - Agnostic to modifier key preference
- **Suitability**
  - Construction of adversarial attacks is easier since all users type the same password
- **Preprocessing**
  - Use S.D to normalize features and scores
  - **Filter:** Exclude outliers (> 2 SD from mean) from user's samples.

Methodology

- *Threshold Score:* Equal Error Rate **(EER)** from Receiver Operating Characteristic **(ROC)** curve
- **Classifiers**

Data for baseline score: 200 genuine, 200 imposters.

- Manhattan Distance
- Autoencoder
- OneClassSVM
- Variational Autoencoder

- **Attackers**

Data: Samples from all other users (2000 samples)

- Average: Use average value for each feature. Generates 1 attack vector per user.
- K-means with 8,16,32 & 64 clusters on all the features. Each cluster serves as an artificially generated attack vector

Classifier Robustness

- Based on the EER scores, users divided into **Great (< 0.03), Ok (< 0.10), and Bad (> 0.10)**

Unseen Test Data: 200 genuine users; 500 impostor users

Figure : Average Error Rate per user group

Users	Manhattan	SVM	Autoencoder	Var AE
Great	0.070	0.090	0.091	**0.065**
OK	**0.084**	0.096	0.096	0.100
Bad	**0.134**	0.136	0.149	0.139

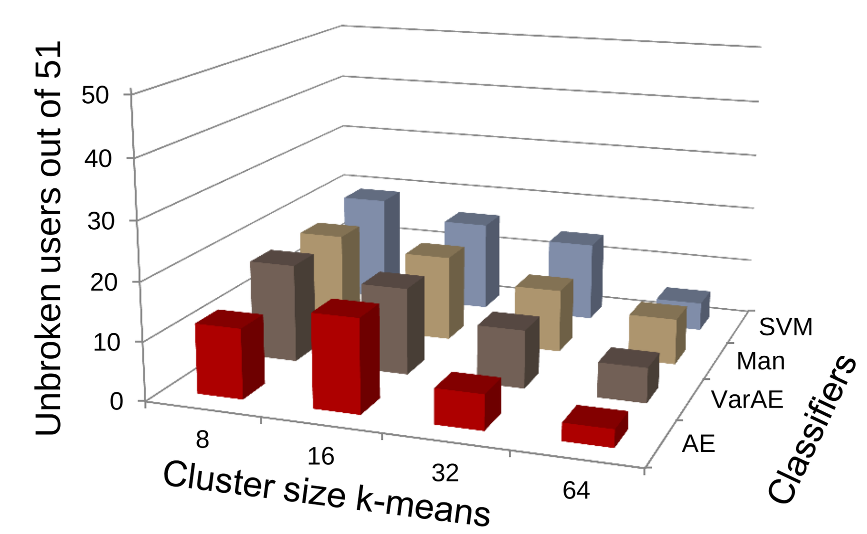
Figure: EER per user group

Users	Great	OK	Bad	All
EER	0.019	0.061	0.179	0.113

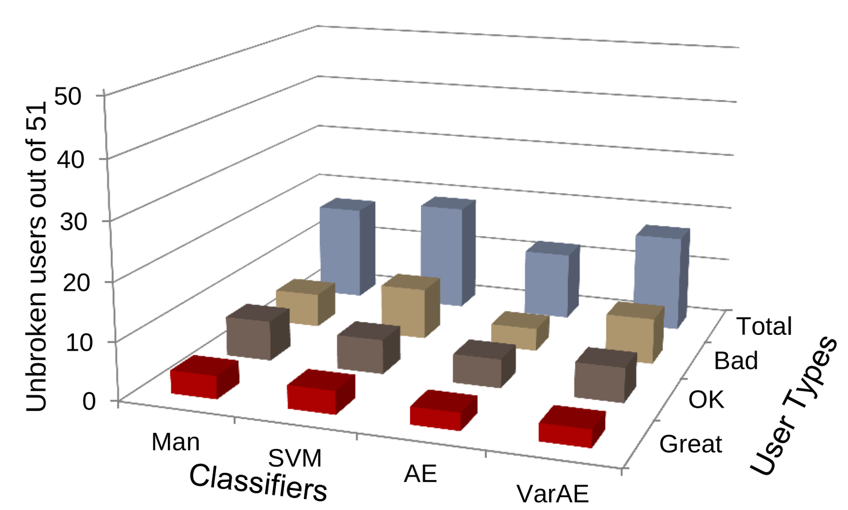
Attacker Performance

Total fails for classifier and attacker combination

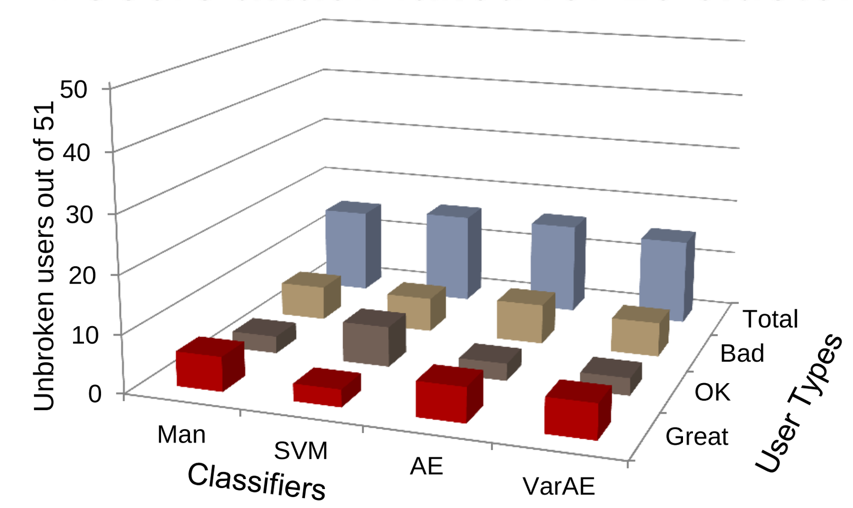
- 8 clusters already breaks 70% of users
- Total users that did not break decrease when cluster size is increased
- SVM, and Manhattan classifiers performed best



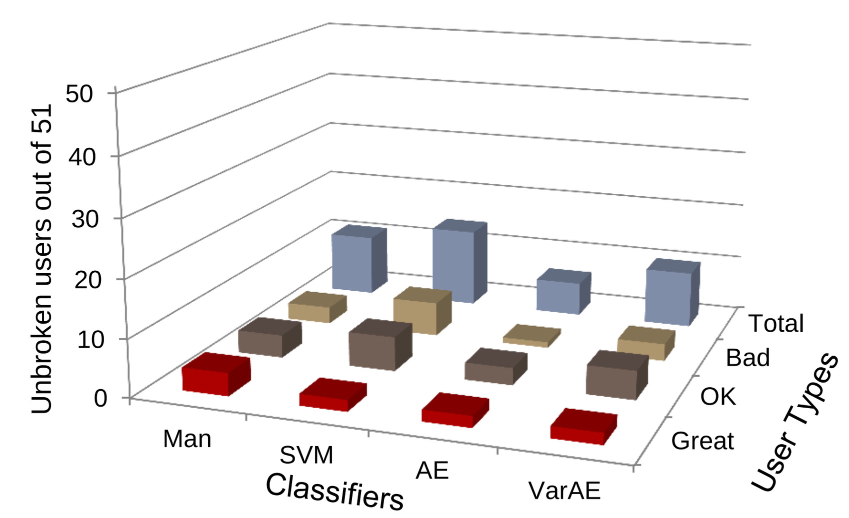
#Users attack failed for 8 clusters



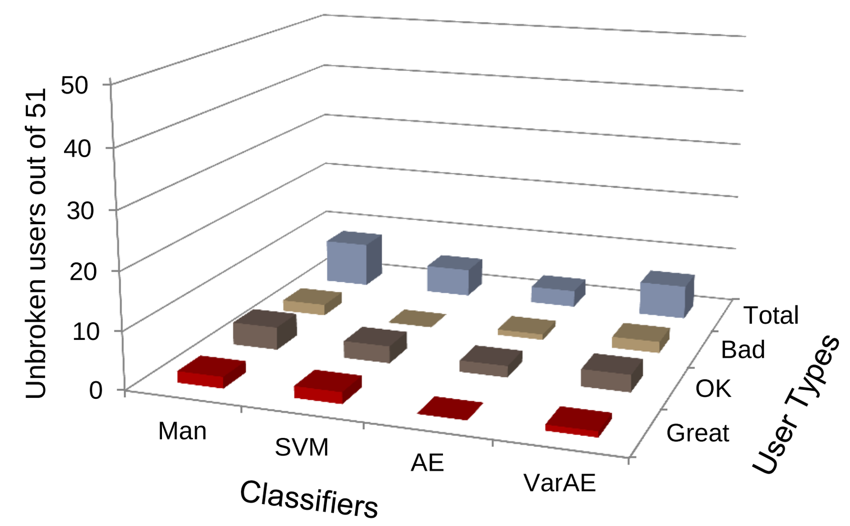
#Users attack failed for 16 clusters



#Users attack failed for 32 clusters



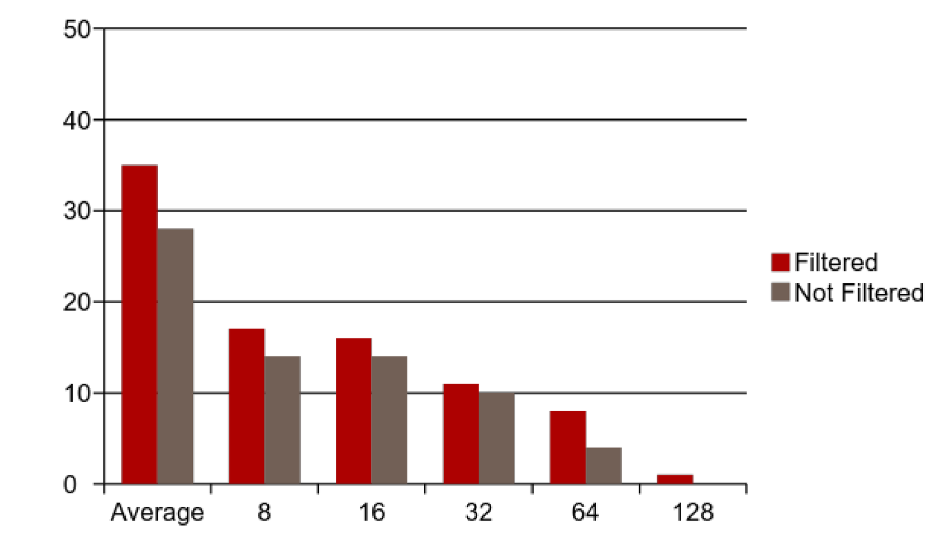
#Users attack failedfor 64 clusters



Enhancements

Following techniques were used to improve defenses:

- Skipping initial features: Improves resistance as skips the samples where user is getting used to password
- Filtering: Remove outlier samples from training data (Figure shows total broken users with Manhattan classifier against attackers)



- Using median/mean as threshold instead of EER: This technique can be used in practical scenario when the log-in attempt is from an unknown machine.

Figure: Median used as threshold with Manhattan Classifier And K-means attacker with 32 and 64 clusters

clusters	Great	OK	Bad	Total
32	8	15	12	35
64	8	14	8	30

Conclusion

- Most users' defense can be broken easily with just 8 cluster k-mean scheme
- Manhattan distance is simplest and most robust classifier probably due to certain degree of overfitting for others
- Score normalization, filtering improved average error rate

Future Work

- Get features like modifier key usage from other datasets
- Is it possible that the majority of the users just never got used to typing in this particular password? To make more general conclusions, it should be very useful to run these tests on some other datasets - especially - those that might have more 'natural' passwords, like user's name.
- Evaluate classifier and adversarial attack performances change as we feed it different amounts of data.