Machine Learning for Security Analytics and Anomaly Detection

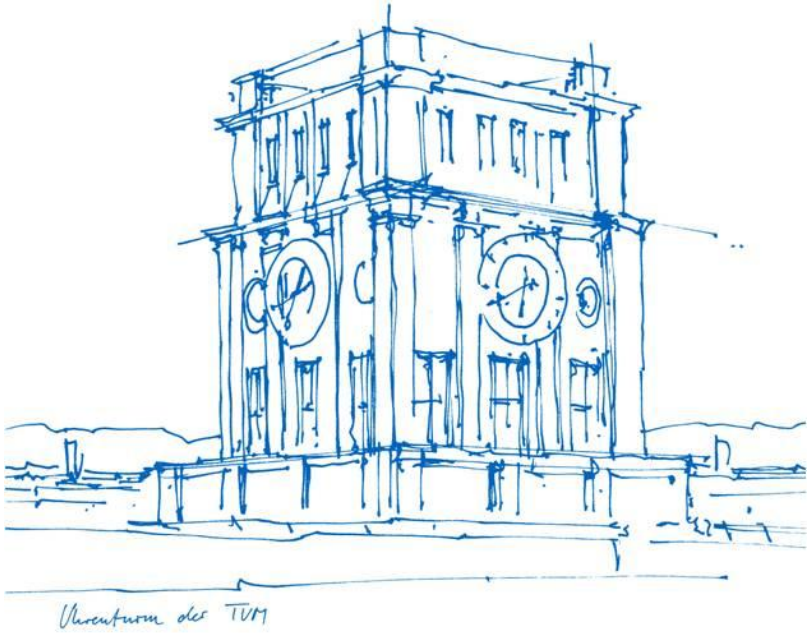
Bojan Kolosnjaji

Chair of IT Security

Faculty of Informatics

Technical University of Munich

Date: 12.07.2018.



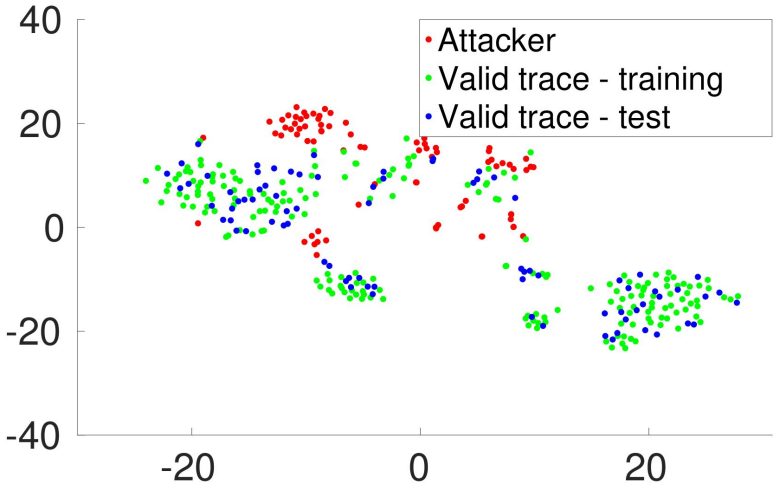
Security Analytics

- Detection (Preparedness)
  - Rule-based detection - easy to evade
  - Attacks have more variety - difficult to capture
- Analysis (Auditing, Forensics)
  - Benefits from Big Data - extract information
  - Use information retrieval methods to analyze threats

Pernul, G., Schryen, G., Schillinger. R.,2017, Security in Highly Connected IT Systems, Results of the Bavarian Research Alliance FORSEC, Uni Regensburg

Anomaly-based approaches

- Used in many areas to analyze outlier events: medical image analysis, video surveillance, fault detection; methods translate to IT security
- Unsupervised vs. Supervised Anomaly Detection
- Multiple conditions: unbalanced datasets, low number of labeled data, adversarial noise, resource constraints...



Anomaly-based approaches

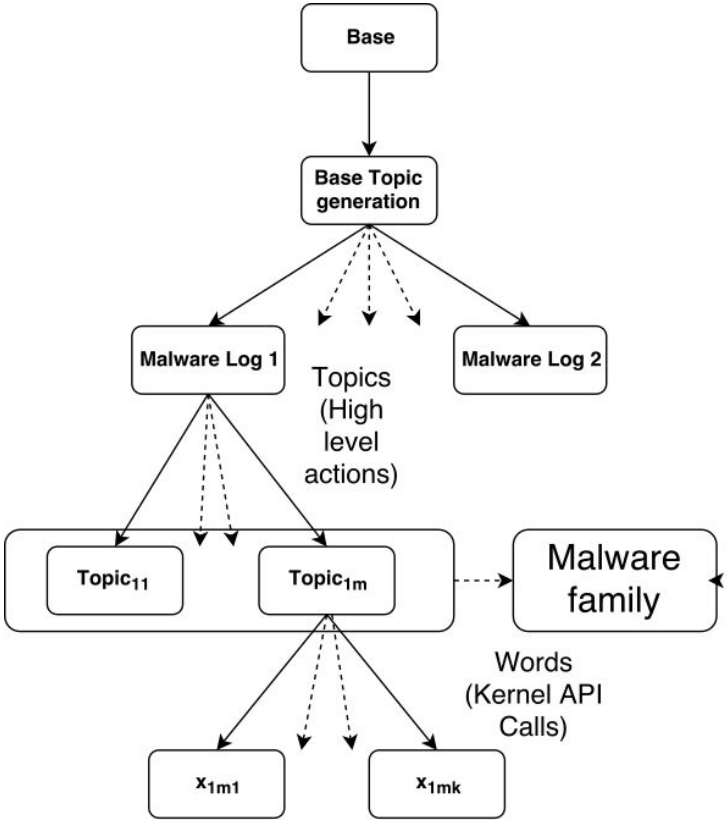
- We use anomaly-based approaches in security analytics
  a. Overcome the rigid signature-based approaches (e.g. Yara)
- We develop methods for anomaly-based approaches in constrained environments:
  a. Resource constraints: low memory, bandwidth
  b. Environment constraints: adversarial environment, online learning
- We test our methods using large-scale malware sample sets and other gathered data

Topic Models for Malware Analysis

- Dynamic Malware Analysis - generates behavioral data for malware samples
- We need to extract relevant information from this data
- Topic Modeling -> known method from Natural Language Processing
  - Information retrieval from document data
  - Detection of high level topics from low level events
  - Semantics-aware -> topics can have meaning
- Large-scale -> from thousands to millions of documents for large malware sample sets
  - VirusTotal: millions of new malware samples per day
  - Requires improvements in information retrieval

Topic Models for Malware Analysis

- Hierarchical Topic Model
- Documents -> System Call Logs
- Topics -> groups of system calls
- Topic probabilities determine the malware family



Topic Models for Malware Analysis

- We use topic modeling and semi-supervised learning to retrieve latent topics[1]

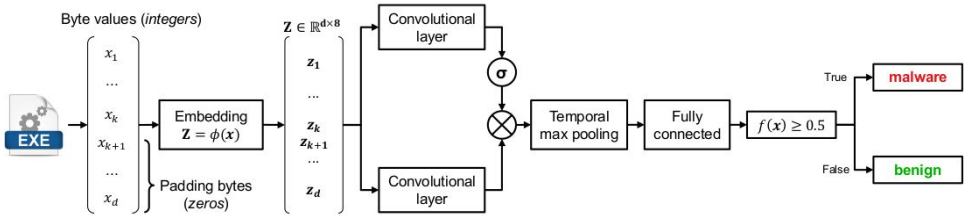
Registry manipulation	Memory management	File manipulation	Process Handling
NtWriteFile	VirtualAllocEx	NtReadFile	OpenProcess
RegOpenKeyExW	VirtualQueryEx	NtWriteFile	ReadProcessMemory
RegCloseKey	VirtualQuery	NtDelayExecution	WriteProcessMemory
RegEnumValueW	VirtualFreeEx	LdrGetProcedureAddress	CloseHandle
RegQueryValueExW	VirtualFree	NtSetInformationFile	LocalAlloc
LdrGetProcedureAddress	LdrGetProcedureAddress	NtCreateFile	LocalFree
RegOpenKeyExA		NtQueryDirectoryFile	

- Topic models can be used as feature extractors, improve malware classification (97.5% on F1 score compared to 88% using baseline methods)

[1] Kolosnjaji, B., et al, 2016., Adaptive semantics-aware malware classification. In *International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment* (pp. 419-439). Springer, Cham.

Adversarial Attacks on Malware Detectors

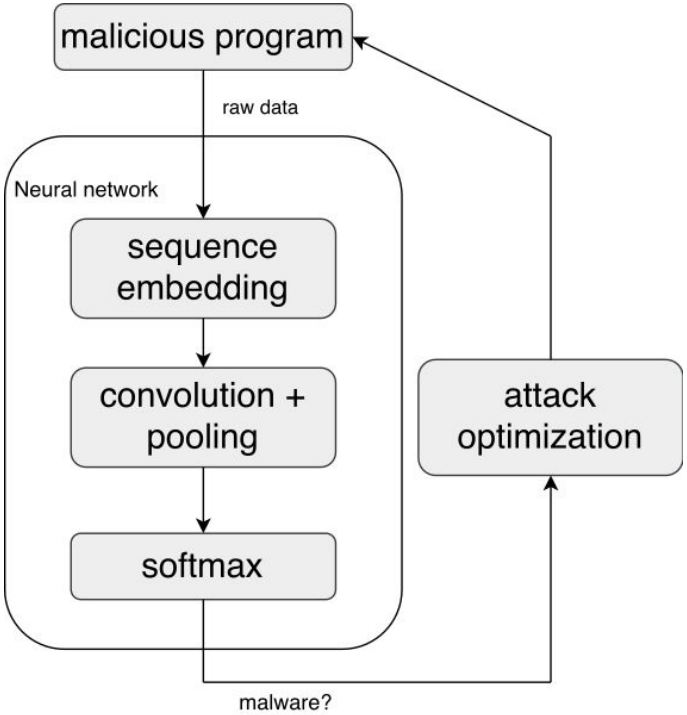
- Neural Networks recently proposed for malware detection based on static features:
  - Raw bytes
  - Instruction sequences
  - PE Import Table, other PE Metadata
- One example: Raff et al., 2017: *Malware Detection by Eating a Whole EXE*[1]
  - However, neural networks are more vulnerable to adversarial examples



[1] Raff, E., Barker, J., Sylvester, J., Brandon, R., Catanzaro, B., & Nicholas, C. (2018). Malware Detection by Eating a Whole EXE, AAAI

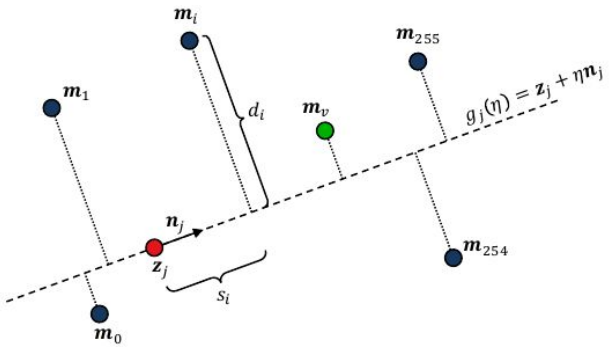
Adversarial Attacks on Malware Detectors

- Evasion attack on malware detectors
- Neural networks especially vulnerable
  - Small change in input →fast change in classification results



Adversarial Attacks on Malware Detectors

- Evasion attack on malware detectors[1]



Algorithm 1 Adversarial Malware Binaries

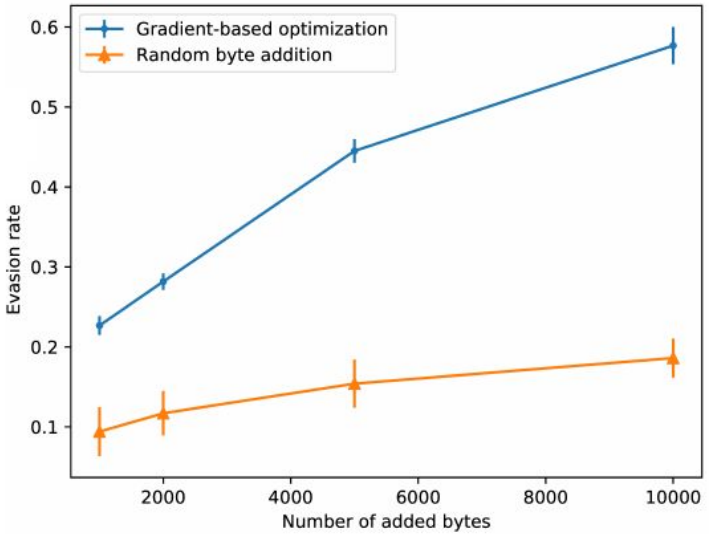
Input: $\boldsymbol{x}_0$, the input malware (with k informative bytes, and $d-k$ padding bytes); q, the maximum number of padding bytes that can be injected (such that $k+q \leq d$); T, the maximum number of attack iterations.

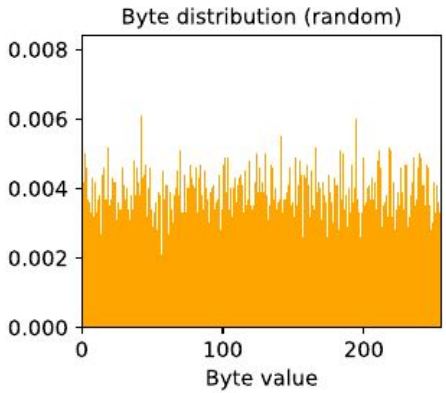
Output: $\boldsymbol{x}'$: the adversarial malware example.

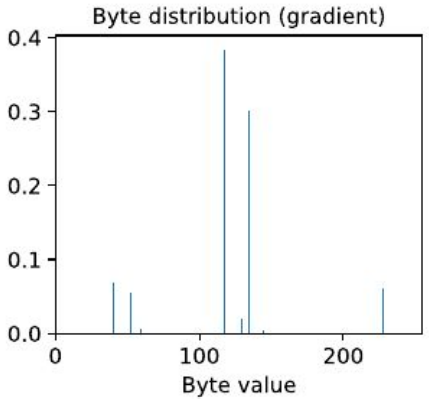
Set $\boldsymbol{x} = \boldsymbol{x}_0$.
Randomly set the first q padding bytes in $\boldsymbol{x}$.
Initialize the iteration counter $t = 0$.
repeat
    Increase the iteration counter $t \leftarrow t + 1$.
    for $p = 1, \ldots, q$ **do**
        Set $j = p + k$ to index the padding bytes.
        Compute the gradient $\boldsymbol{w}_j = -\nabla_\phi(x_j)$.
        Set $\boldsymbol{n}_j = \boldsymbol{w}_j / \|\boldsymbol{w}_j\|_2$.
        for $i = 0, \ldots, 255$ **do**
            Compute $s_i = \boldsymbol{n}_j^\top (\boldsymbol{m}_i - \boldsymbol{z}_j)$.
            Compute $d_i = \|\boldsymbol{m}_i - (\boldsymbol{z}_j + s_i \cdot \boldsymbol{n}_j)\|_2$.
        end for
        Set x_j to $\arg\min_{i:s_i>0} d_i$.
    end for
until $f(\boldsymbol{x}) < 0.5$ or $t \geq T$
return $\boldsymbol{x}'$

[1] Kolosnjaji et al., 2018, Adversarial Malware Binaries: Evading Deep Learning for Malware Detection in Executables, arXiv:1803.04173

Adversarial Attacks on Malware Detectors



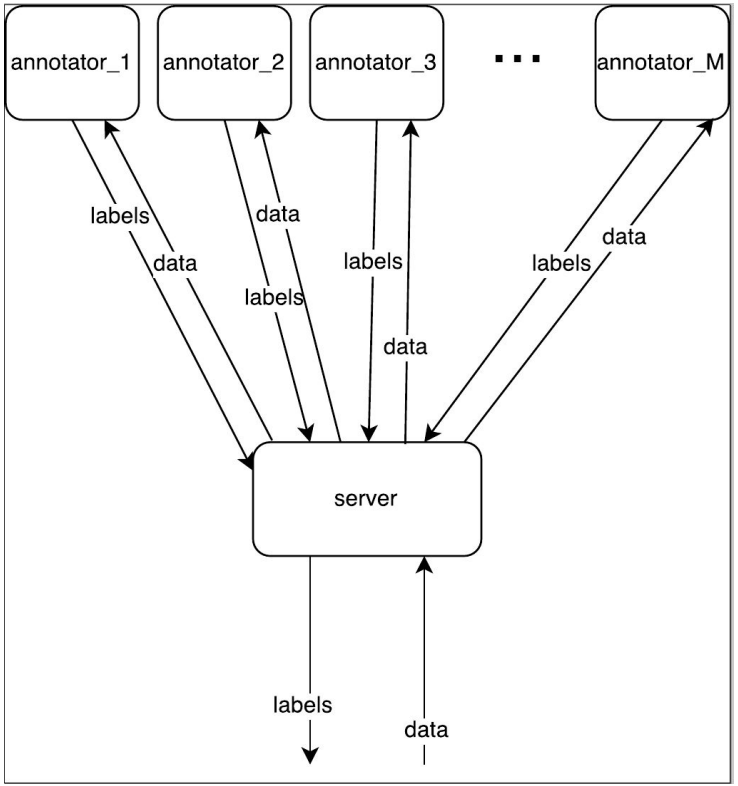




[1] Kolosnjaji, B., Demontis, A., Biggio, B., Maiorca, D., Giacinto, G., Eckert, C., Roli, F., 2018, Adversarial Malware Binaries: Evading Deep Learning for Malware Detection in Executables, arXiv:1803.04173

Communication-efficient Learning

- Multiple **annotators** label the data, some are more **reliable** than others, some could be **malicious**
- How to select a minimal set of annotators and **save budget** while optimizing the accuracy?

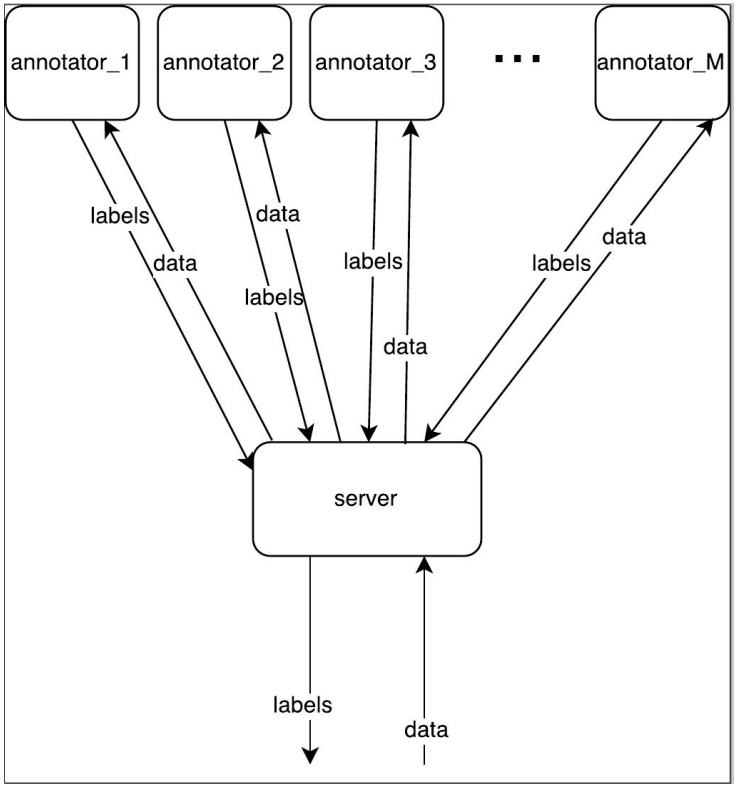


Communication-efficient Learning

- Select most reliable annotators? Difficult,

  every annotator has different **expertise**

- We want to have a **joint optimization model**



Communication-efficient Learning

- We have input data x_n, annotations z_{ni}, ground truth y_n
- Goal 1: Minimize the loss w.r.t. the ground truth training set
- Goal 2: Minimize the loss w.r.t. the client annotations
- Goal 3: Make the annotator weight vector sparse -> select only a few clients

$$L(w, v) = -\frac{1}{N}\sum_{n=1}^{N}(\hat{y}_n \log \sum_{i=1}^{M} v_i y_{ni} + (1-\hat{y}_n)\log(1-\sum_{i=1}^{M} v_i y_{ni})) +$$

$$\sum_{i=1}^{M} v_i = 1$$

client weight vector

$$\psi \sum_{i=1}^{M}(y_{ni} - z_{ni})^2 + \lambda \mid v \mid$$

Communication-efficient Learning

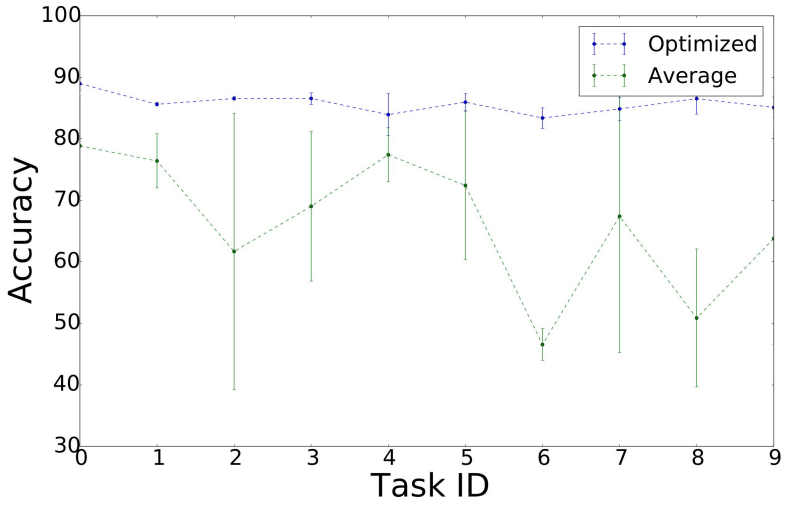
- Gradient-based optimization

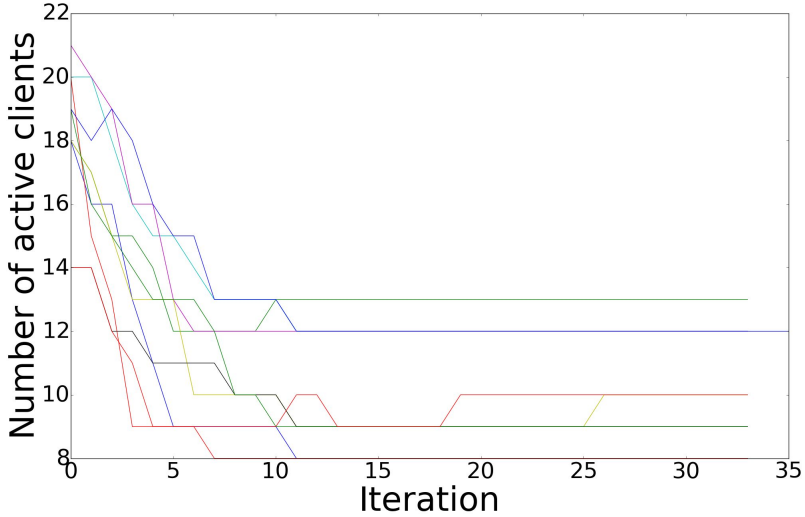
$$\frac{\partial L}{\partial w_j} = -\frac{1}{N}\sum_{n=1}^{N}(y_n \frac{1}{\sum_{i=1}^{M} v_i \hat{y}_{ni}} v_j \frac{\partial y_n}{\partial w_j} + (1-y_n)\frac{1}{1-\sum_{i=1}^{M} v_i \hat{y}_{ni}} v_j \frac{\partial y_n}{\partial w_j} + \psi \sum_{i=1}^{M} 2(y_{ni} - z_{ni}\frac{\partial y_n}{\partial w_j}))$$

$$\frac{\partial L}{\partial v_i} = -\frac{1}{N}\sum_{n=1}^{N}(y_n \frac{1}{\sum_{i=1}^{M} v_i y_{ni}} + (1-\hat{y_n})\frac{1}{1-\sum_{i=1}^{M} v_i y_{ni}}(-y_n)$$

Communication-efficient Learning

- Experiments on Amazon Mechanical Turk



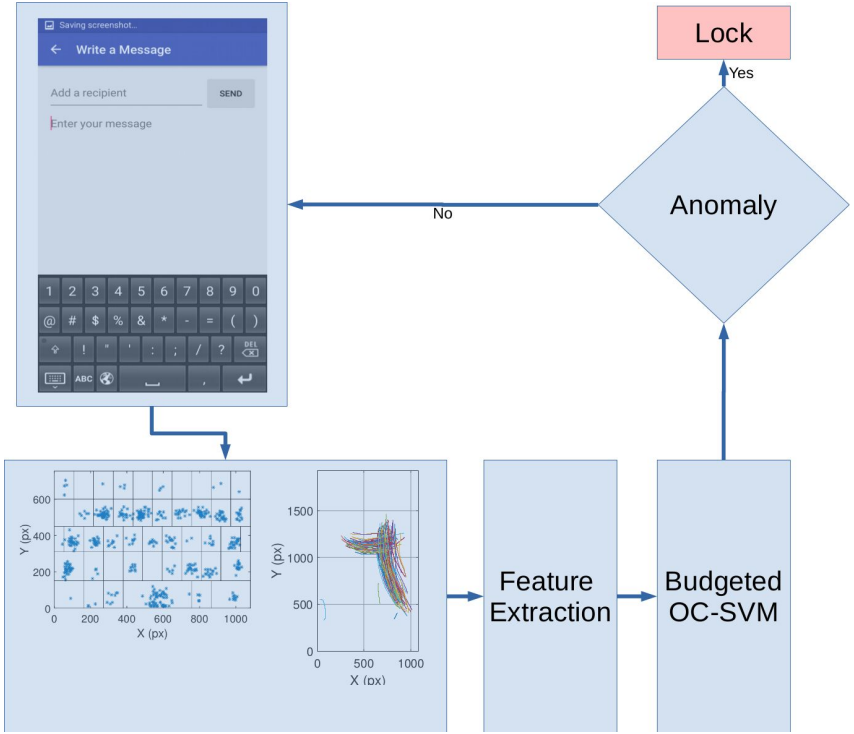


Behavior-based Authentication

- Entry authentication: password, PIN, fingerprint
  - Prone to attacks
  - Many times not used, inconvenient
- Continuous authentication: based on user sensor data
  - Attractive alternative
  - Needs to be trained online
  - Model grows with the data

Behavior-based Authentication

- We design a behavior-based authentication system:
  - Data Collection from sensors
  - Feature Engineering
  - Model: Budgeted One-Class SVM
  - Anomaly Detector



Behavior-based Authentication

- Model based on One-Class SVM [1]

$$min_{\alpha} \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j k(x_i, x_j)$$

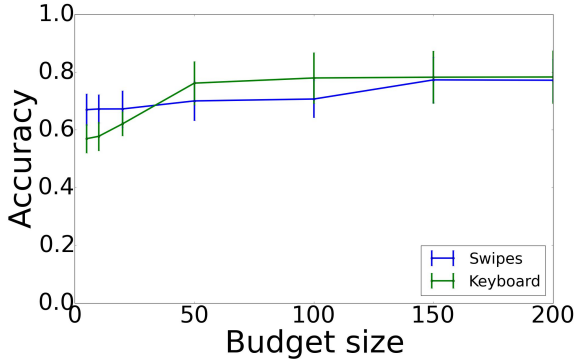
$$0 \leq \alpha_i \leq \frac{1}{vl}, \sum_i \alpha_i = 1$$

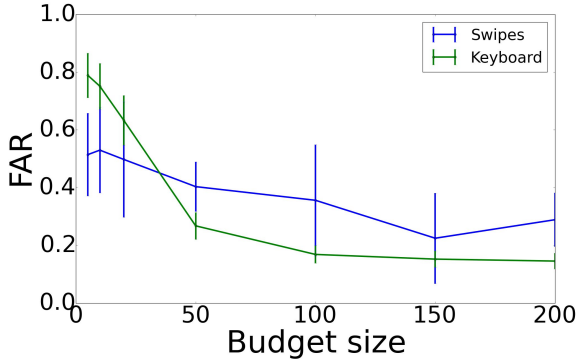
- Optimization while maintaining an upper bound on support vectors - Budgeted learning

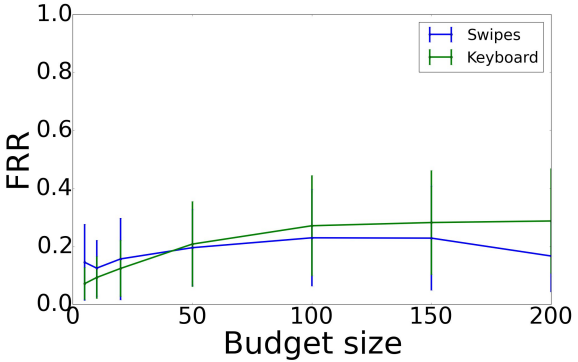
[1] Schölkopf, B., Williamson, R. C., Smola, A. J., Shawe-Taylor, J., & Platt, J. C. (2000). Support vector method for novelty detection. In *Advances in neural information processing systems*(pp. 582-588).

Behavior-based Authentication

- Evaluation on experiments with human subjects and typical smartphone use case scenarios







Conclusion

- High potential in machine learning methods for the purpose of IT security, as a consequence of variety/volume of malware, noisy data
- Additional constraints make the use of baseline approaches difficult
- Benefits from areas such as: adversarial learning, budgeted learning, approximate inference, ensemble learning