Fishy Faces: Crafting Adversarial Images to Poison Face Authentication

Giuseppe Garofalo, Vera Rimmer, Tim Van hamme,
Davy Preuveneers and Wouter Joosen

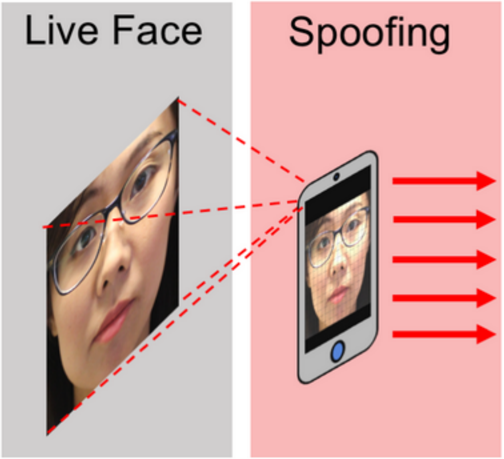
WOOT 2018, August 13-14 (Baltimore, MD, USA)

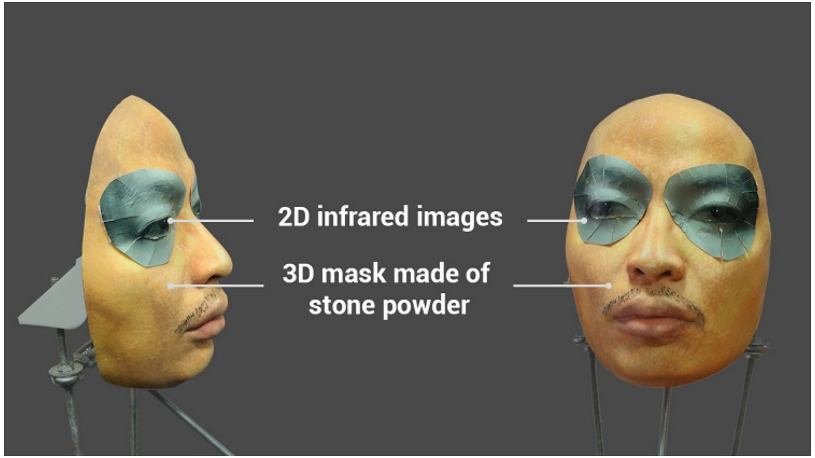
KU LEUVEN DistriNet

Face authentication

Face authentication

- Wide adoption of face recognition in mobile devices
- Face authentication is a highly security-sensitive application
- Several attacks have been proposed (e.g replay attacks[1], Bkav’s mask[2] etc.)





[1] Face Anti-spoofing, Face Presentation Attack Detection
[2] Bkav’s new mask beats Face ID in "twin way": Severity level raised, do not use Face ID in business transactions.

Face authentication - Machine Learning

- Authentication relies on Machine Learning (ML) algorithms
  - they learn how to recognise the user through time and changes
- ML algorithms are not security-oriented per se
  - Adversarial ML arms-race investigates the **existing vulnerabilities**, models **active attacks** and seeks for **proactive countermeasures**

Why poison face authentication?

- Adversarial ML has been applied to face recognition[1], **but not face authentication**
- Face authentication systems are **adaptive**
  - ML model is periodically re-trained
  - Gives an attacker **access prior to training**
- Feasibility and efficacy of poisoning attacks against face authentication is yet unknown

[1] Biggio, B., Didaci, L., Fumera, G., and Roli, F. Poisoning attacks to compromise face templates. In 2013 International Conference on Biometrics (ICB) (June 2013), pp. 1–7.

Background

Background - Machine learning

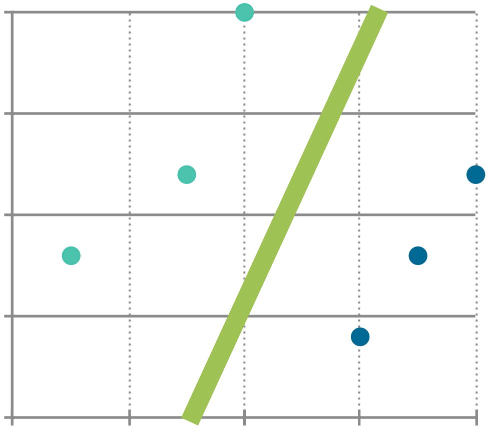
- Machine learning algorithms as a tool for learning patterns
  - Patterns comprise *biometric traits* used for authenticating a person
- The classification task is divided into two phases:
  - Training on a set of labelled points, i.e. the training set
  - Testing the model by predicting the label of new points, i.e. the test set
- Each point is a feature vector
- Training minimizes a *loss function*

Background - Adversarial Machine Learning

- Adversarial ML investigates the ML algorithms in the adversarial environment
- The two main scenarios are:
  - the evasion of the classification rule (post-training)
  - the **poisoning** of the training set

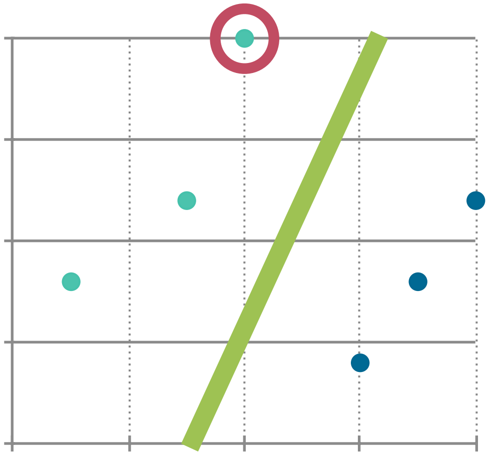
Background - Poisoning SVM

- Poisoning requires the attackers to inject / control a malicious sample into the training set



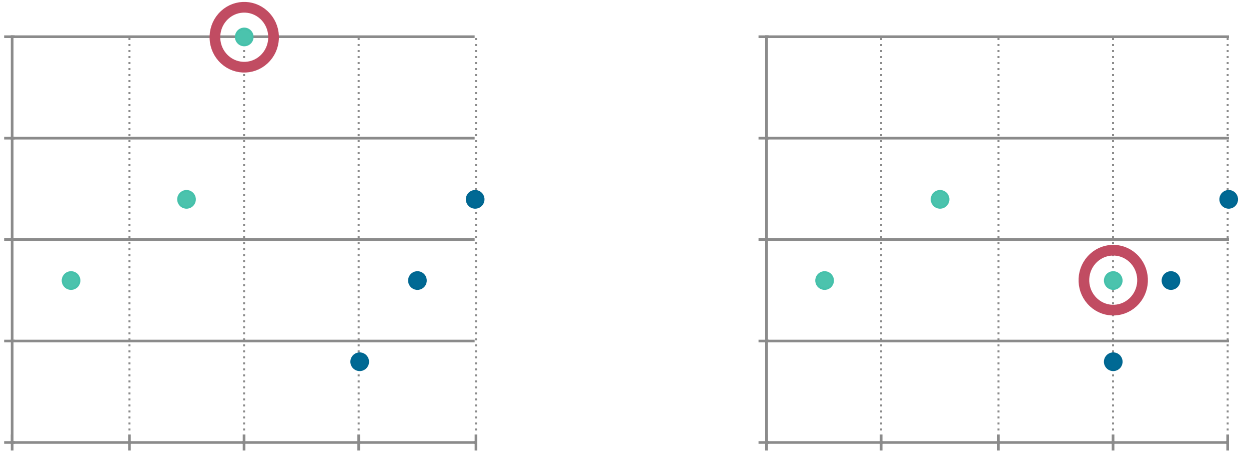
Background - Poisoning SVM

› Poisoning requires the attackers to inject / control a malicious sample into the training set



Background - Poisoning SVM

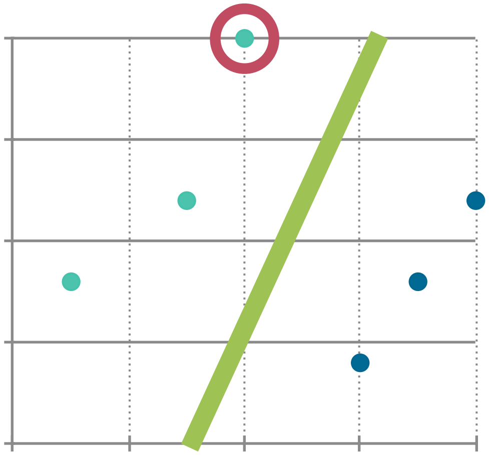
› The attack point is moved towards a desired direction to *maximize* a loss function (instead of minimizing it)



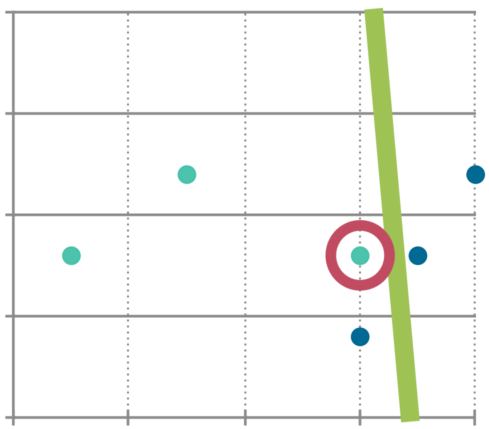
KU LEUVEN DistriNet

Background - Poisoning SVM

› The re-training phase triggers the poisoning effects



1 misclassification



Background - Attack point search

- The best attack point is the one that maximizes the loss function the most
- In this work, we apply an existing theoretical algorithm[1]
  - Poisoning attack against SVM
  - Focus on the hinge loss as a classification error estimate
  - Gradient Ascent strategy to search the attack point

[1] Biggio, B., Nelson, B., and Laskov, P., Poisoning attacks against SVM. (2012).

System under attack

System design

- Our target authenticator is composed of two parts:
  - Feature extractor
  - Classification model

Input image

System design

- Feature Extractor
  - OpenFace Library
  - Based on Google's FaceNet[1] (Convolutional Neural Network)

[1] Schroff, F., Kalenichenko, D., and Philbin, J. Facenet: A unified embedding for face recognition and clustering. In Proceedings of the IEEE conference on computer vision and pattern recognition (2015), pp. 815–823.

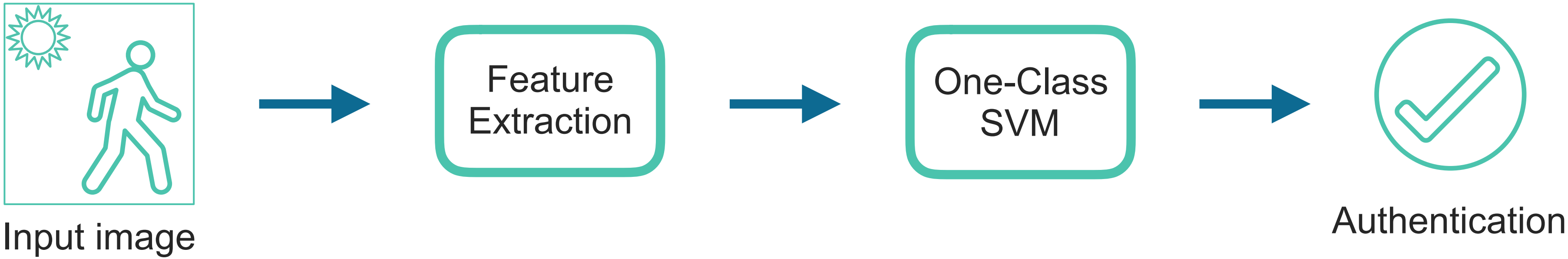
System design

- One-Class SVM for classification[1]
  - Trained only on images of the user
  - Takes a hyper-parameter which defines the upper-bound to the percentage of training errors

[1] Inspired by: Gadaleta, M., and Rossi, M. Idnet: Smartphone-based gait recognition with convolutional neural networks. Pattern Recognition 74 (2018), 25 – 37.

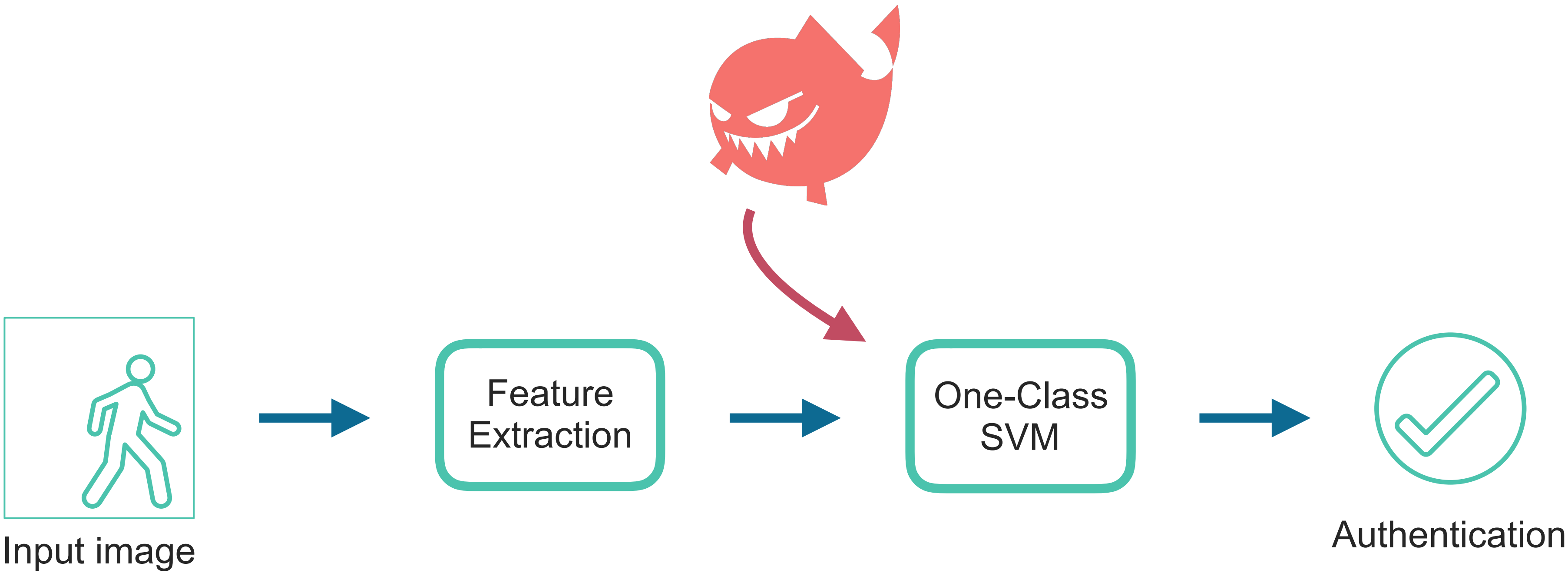
System design

- Once trained, the model is used to authenticate the user

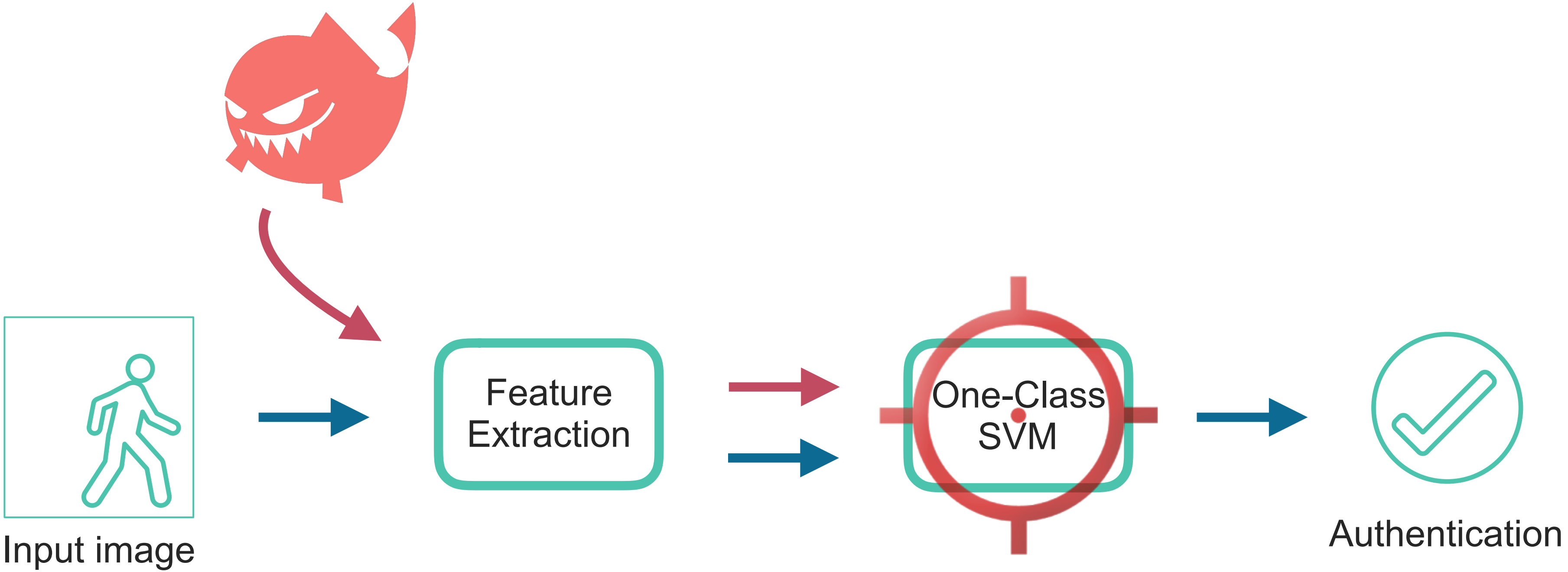


Attack methodology

Methodology - Threat Model



Methodology - Threat Model



Methodology - Threat Model

- Attacker's goals:
  - Denial-of-Service: to increase the false negative rate of the target authenticator
  - Impersonation: to allow other identities to be authenticated as the rightful user

Methodology - Threat Model

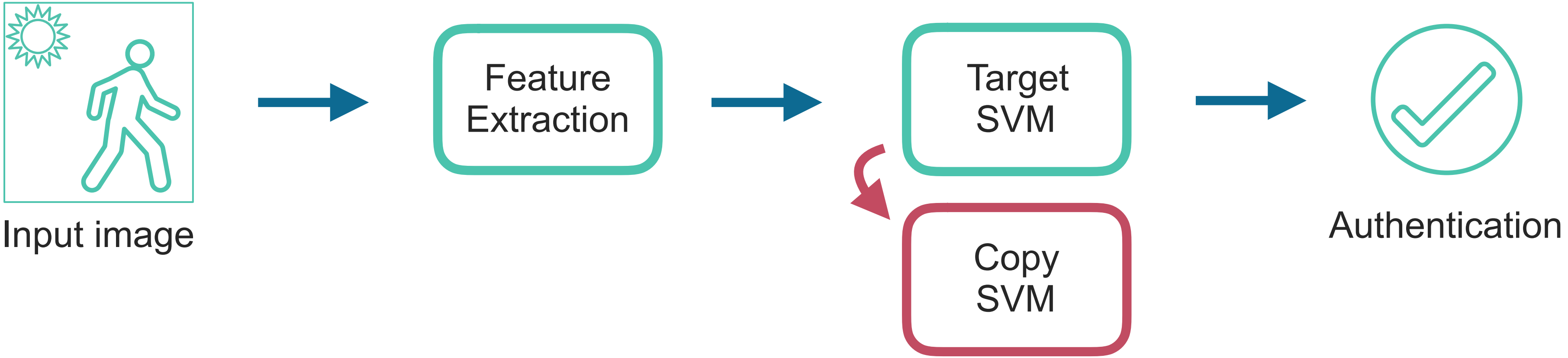
- Attacker's goals:
  - Denial-of-Service: to increase the false negative rate of the target authenticator
  - Impersonation: to allow other identities to be authenticated as the rightful user
- Attacker's resources:
  - Able to poison the training set by injecting malicious images
  - Has the knowledge of the model's detail (including training images and model parameters)

Methodology

- The attack methodology is divided into two parts:
  - Obtain the attack point by using the gradient ascent strategy
  - Reverse the feature extraction process to inject a real-world image

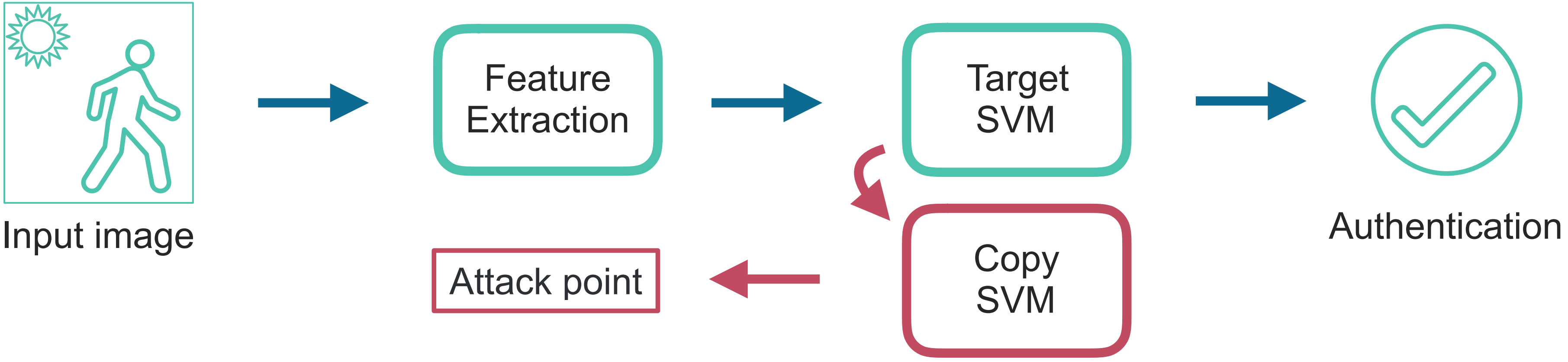
Methodology - Step-by-step

- Obtain the images used for training the model to train an exact copy of our target



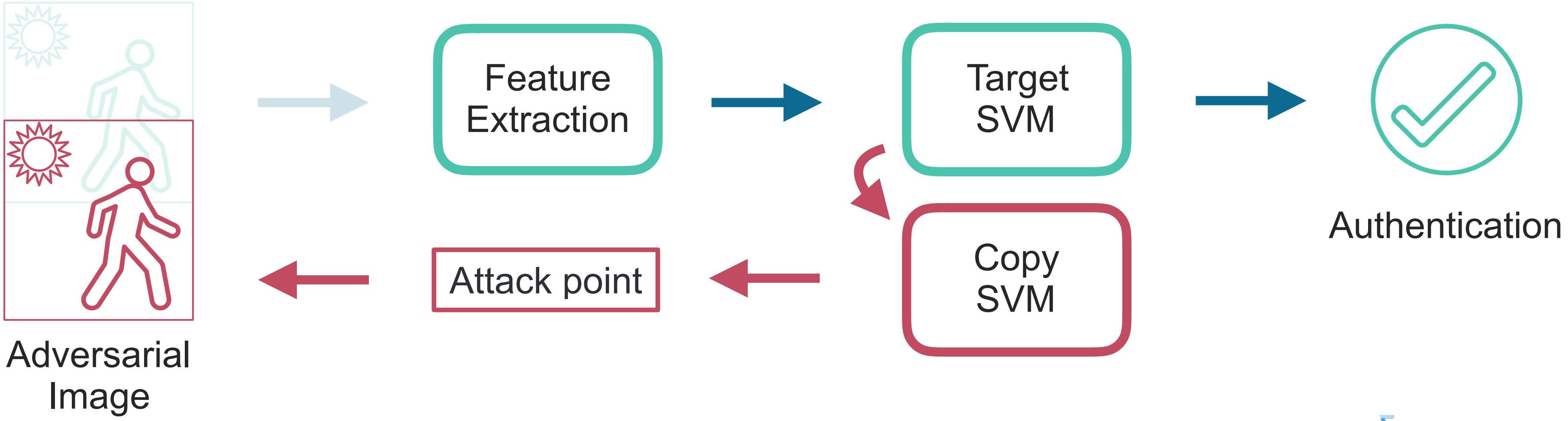
Methodology - Step-by-step

- Find the best attack point using the gradient ascent strategy
  - the “best” attack point is the one which maximizes the classification error
  - It is found by modifying the feature vector of a validation set image



Methodology - Step-by-step

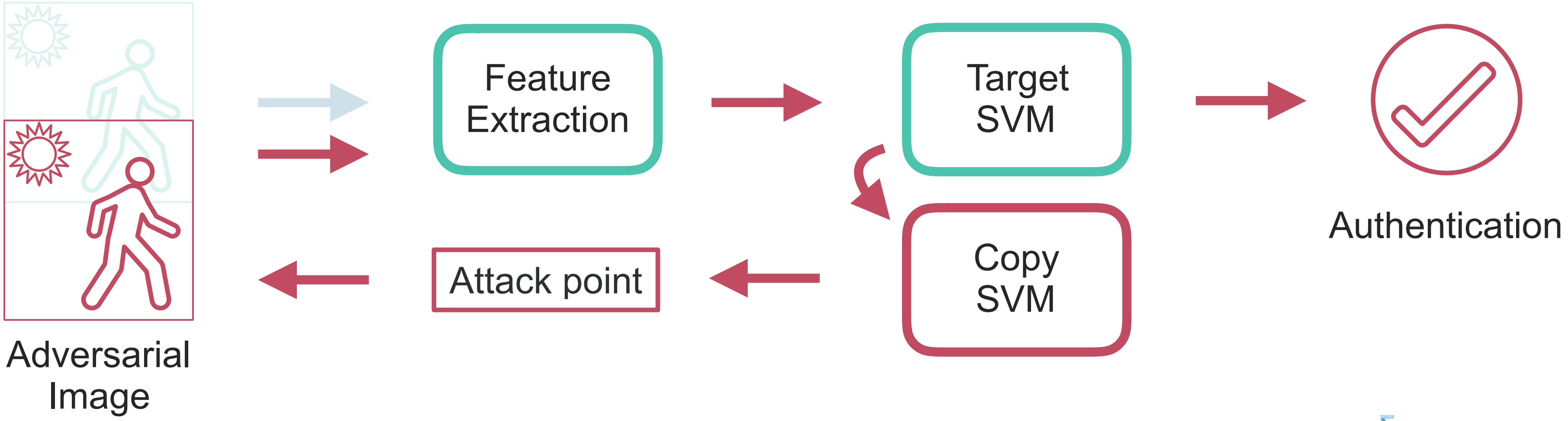
- Find a face image corresponding to the best attack point
  - A best-first search strategy to reverse the CNN function is exploited



KU LEUVEN DistriNet

Methodology - Step-by-step

- Present the image to the system which will be re-trained over the new sample, affecting the authentication procedure



Methodology - Attack example

- The target One-Class SVM is trained to recognize one identity
  - Data is collected from the FaceScrub celebrity dataset
  - Training set is composed by 30 images

Authenticated user

Methodology - Attack example

Attack point

Raw attack point

The attack point is computed by using the gradient ascent technique, starting from the feature vector of a randomly-chosen validation image

KU LEUVEN DistriNet

Methodology - Attack example

Attack point

A **sliding window** is used to apply modifications to the image so that its feature vector becomes **very similar** to the attack point

Raw attack point

Pre-processing

Reverse-CNN

Methodology - Attack example

Attack point ≈ **Feature Vector** ← Feature Extraction

Raw attack point

Pre-processing

Reverse-CNN

Methodology - Attack example

- After the injection, the classification accuracy drops from 4% to 44% (by 40%!)

False positive
Unauthorised User

False negative
Authorised User

Injected image

Methodology - Attack example

› Using just a **random image**, the classification accuracy drops by 2%

True negative
Unauthorised User

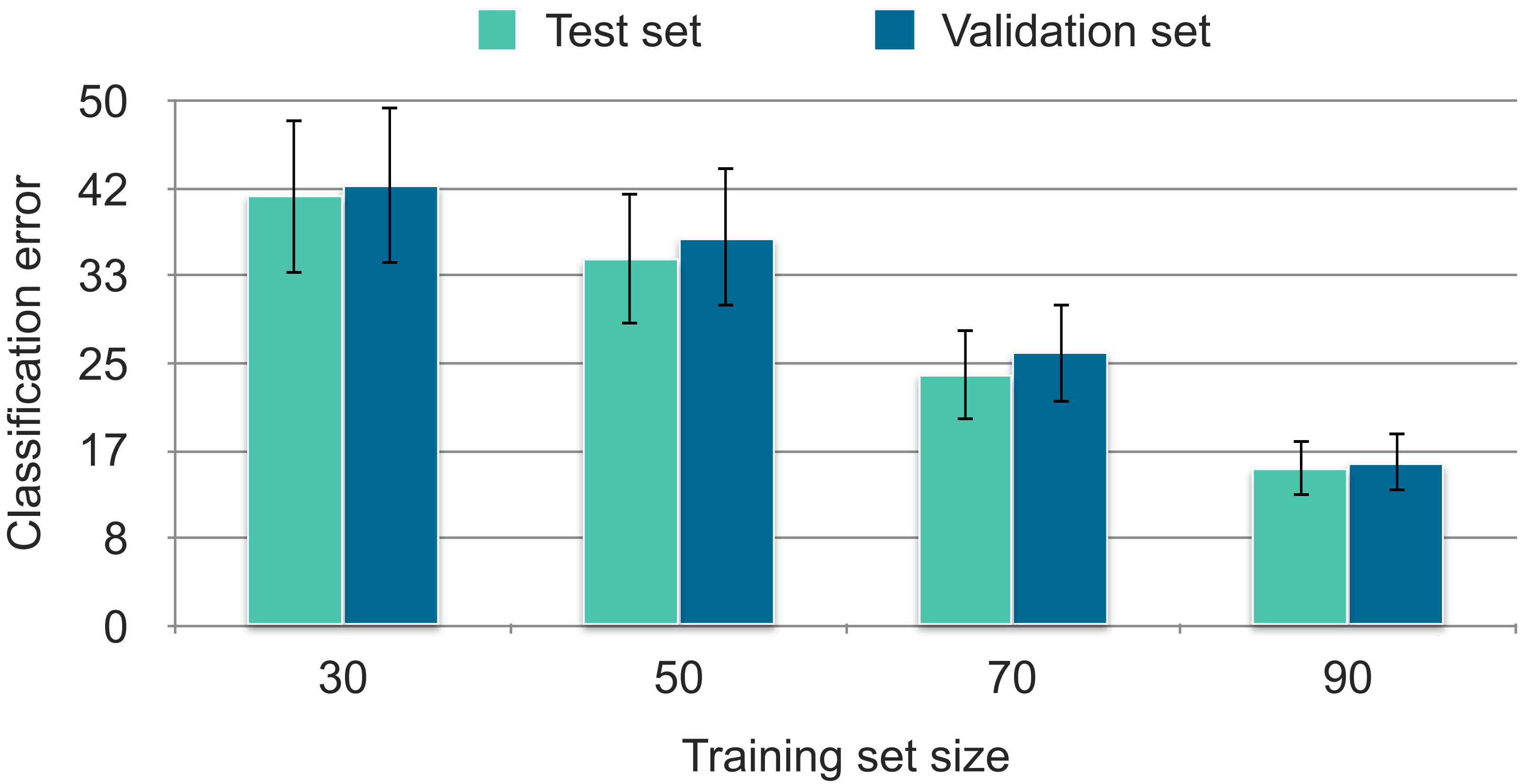
True positive
Authorised User

Injected image

KU LEUVEN DistriNet

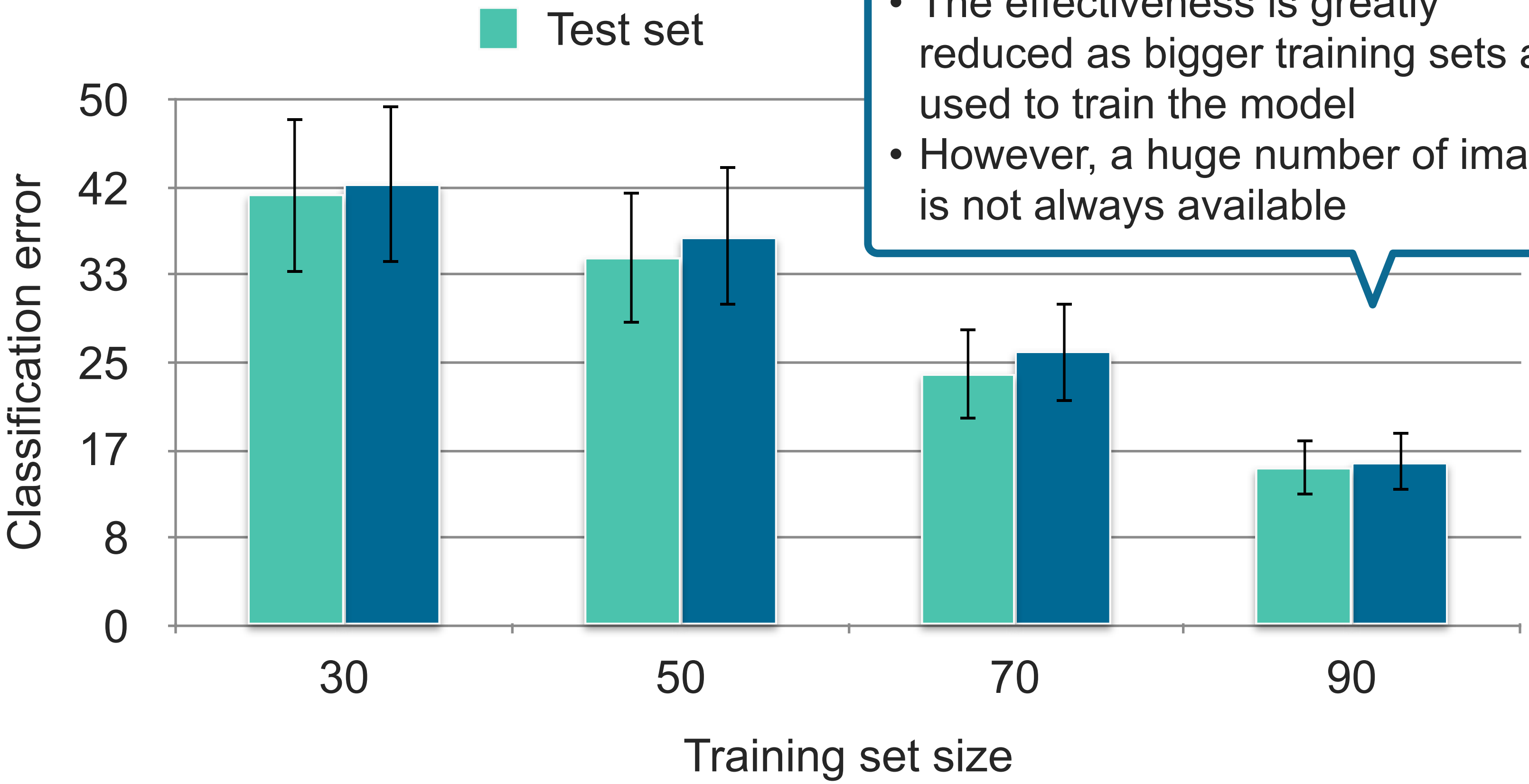
Evaluation

Evaluation



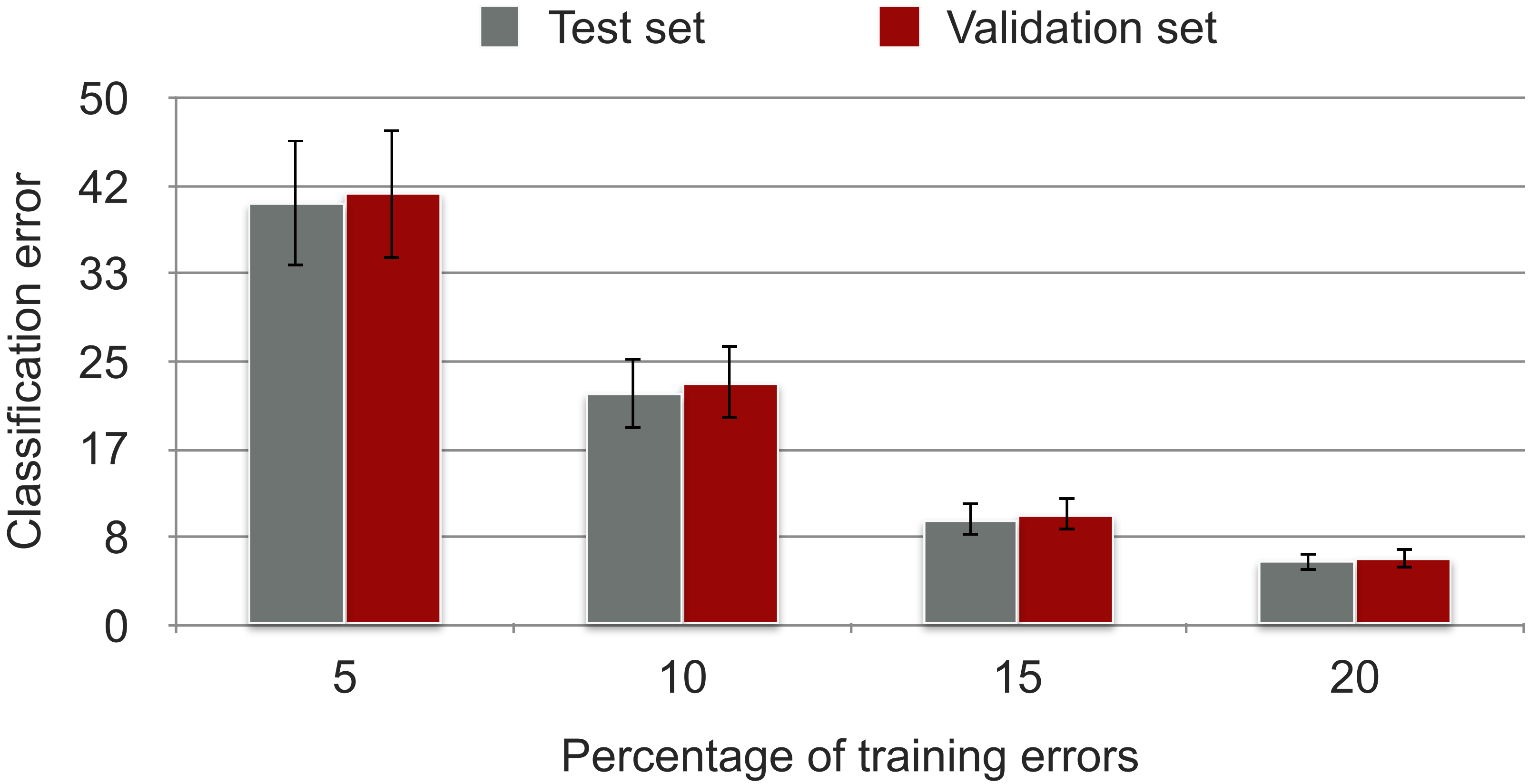
Test set
Validation set
Classification error
50
42
33
25
17
8
0
30
50
70
90
Training set size

Evaluation



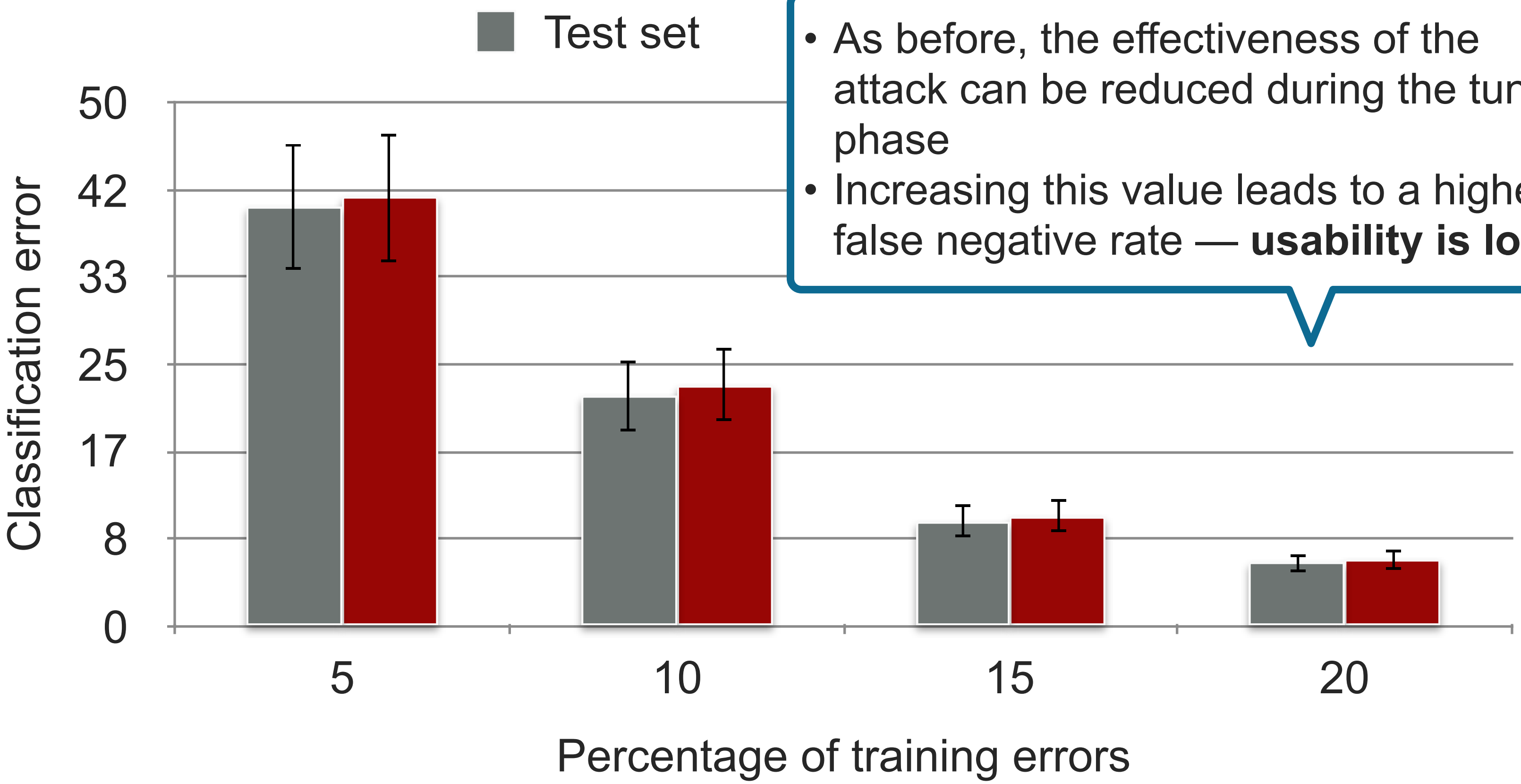
- The effectiveness is greatly reduced as bigger training sets are used to train the model
- However, a huge number of images is not always available

Evaluation



KU LEUVEN DistriNet

Evaluation



- As before, the effectiveness of the attack can be reduced during the tuning phase
- Increasing this value leads to a higher false negative rate — **usability is lower**

KU LEUVEN DistriNet

Limitations

Limitations

- The poisoning attack relies on two assumptions on the attacker's capabilities
  - Knowledge of the training images of the target user

Limitations

- The poisoning attack relies on two assumptions on the attacker's capabilities
  - Knowledge of the training images of the target user

• **Transferability property** can be exploited to train a model without knowing training images

Limitations

- The poisoning attack relies on two assumptions on the attacker’s capabilities
  - Knowledge of the training images of the target user
  - Ability to inject an image into the training set

Limitations

- The poisoning attack relies on two assumptions on the attacker's capabilities
  - Knowledge of the training images of the target user
  - Ability to inject an image into the training set

- **Continuously-adapted injection** strategies may be useful to break the authentication step

Conclusion

Conclusion

- In this work we:
  - Apply a poisoning attack against a state-of-the-art face authentication model — **obtain classification error of over 50% with one injected image**
  - Demonstrate how to defend against such attacks through careful design choices
  - Show the feasibility to attack a multi-stage authentication process involving face recognition with a reverse-mapping strategy

Conclusion

- In this work we:
  - Apply a poisoning attack against a state-of-the-art face authentication model — **obtain classification error of over 50% with one injected image**
  - Demonstrate how to defend against such attacks through careful design choices
  - Show the feasibility to attack a multi-stage authentication process involving face recognition with a reverse-mapping strategy
- This work urges to integrate awareness of adversarial ML attacks into all stages of the authentication system design

DistriNet

Thank you!

https://distrinet.cs.kuleuven.be/