Payatu

{Client Name}

{Month} {Year}

{Client Name} AI/ML Security Assessment

Client Details

Company Name: {Client Name}

Contact Person: {Person Name}

Address: {Address Name}

Email: {Email Address}

Telephone: {Telephone Number}

Document History

Version	Date	Author	Remark
1.0	{Date}	{Author}	Document Creation

Table of Contents

1. About Payatu 4

2. Project Details 5

2.1 Executive Summary 5

2.2 Scope and Objective 5

2.3 Project Timeline 5

2.4 Technological Impact Summary 6

2.5 Business Impact Summary 6

2.7 Vulnerability Chart .... **Error! Bookmark not defined.**

2.8 Vulnerabilities by Category .... **Error! Bookmark not defined.**

3. Technical Findings .... **Error! Bookmark not defined.**

3.1 PY-CL-001: Whitebox Adversarial Attack: When attacker has access to single face image sample for target .... **Error! Bookmark not defined.**

3.2 PY-CL-002: Whitebox Adversarial Attack: When attacker has access to multiple image samples for target .... **Error! Bookmark not defined.**

3.3 PY-CL-003: Generic adversarial noise generation for targeted attack . **Error! Bookmark not defined.**

3.4 PY-CL-004: Generic adversarial noise generation for non-targeted attack .... **Error! Bookmark not defined.**

3.5 PY-CL-005: Model Extraction attack .... **Error! Bookmark not defined.**

4. We Prescribe .... **Error! Bookmark not defined.**

5. Appendix .... **Error! Bookmark not defined.**

1. About Payatu

Payatu is a research-powered security testing service organization specialized in IoT and embedded products, web, mobile, cloud & infrastructure security assessments, and in-depth technical security training. Our state-of-the-art research, methodologies, and tools ensure the safety of our client's assets.

At Payatu, we believe in following one's passion, and with that thought, we have created a world-class team of researchers and executors who are bending the rules to provide the best security services. We are a passionate bunch of folks working on the latest and leading-edge security technology.

We are proud to be part of a vibrant security community and don't miss any opportunity to give back. Some of the contributions in the following fields reflect our dedication and passion.

- nullcon - nullcon security conference is an annual security event held in Goa, India. After years of effort put in the event, it has become a world-renowned platform to showcase the latest and undisclosed research.
- hardwear.io - Hardware security conference is an annual hardware security event held in The Hague, Netherlands. It is being organized to answer emerging threats and attacks on hardware. We aim to make it the largest platform, where hardware security innovation happens.
- Dedicated fuzzing infrastructure - We are proud to be one of the few security research-based companies to own an in-house infrastructure and hardware for distributed fuzzing of software such as browsers, client, and server applications.
- null - It all started with null - The open security community. It’s a registered non-profit society and one of the most active security community. null is driven totally by passionate volunteers.
- Open source - Our team regularly authors open source tools to aid in security learning and research.
- Talks and Training: Our team delivers talks/highly technical training in various international security and hacking conference, i.e., DEFCON Las Vegas, BlackHat Las Vegas, HITB Amsterdam, Consecwest Vancouver, nullcon Goa, HackinParis Paris, Brucon Belgium, zer0con Seoul, PoC Seoul to name few.

We are catering to a diverse portfolio of clients across the world, who are leaders in banking, finance, technology, healthcare, manufacturing, media houses, information security, and education, including government agencies. Having various empanelment and accreditations, along with a strong word of mouth, has helped us win new customers. Our thorough professionalism and quality of work have brought repeat business from our existing clients. We thank you for considering our security services and requesting a proposal. We look forward to extending the expertise of our passionate, world-class professionals to achieve your security objectives.

2. Project Details

2.1 Executive Summary

AI/ML Security Assessment of {Client Name} has been performed, considering below common security issues:

- ✓ If the model can be extracted.
- ✓ Several Whitebox and blackbox attacks.

2.2 Scope and Objective

This application consists of a state of the art machine learning model called FaceNet.
Used to generate face embeddings and thus helping face recognition
We have simulated a demo-application that largely resembles customer's use cases. And tested the model against Adversarial learning attacks and Model extraction attacks.

2.3 Project Timeline

The security assessment was performed for {Number of Days} days from {Start Date} to {End Date}.

2.4 Technological Impact Summary

We have performed security assessments on the ML application. The application consists of a FaceNet model used to create face embeddings and use the embeddings to recognize input faces.
An attacker can create adversarial noise when added to an image, which can lead to the case of targeted impersonation or mis-recognition.
It took less than 5 mins to perform the targeted adversarial learning attack. Which increases the likelihood of attack in future.
Generate adversarial noise can be used in different images to perform the same targeted adversarial learning attack. Which means that it is not required to run the Adversarial Learning attack to create every adversarial sample.

2.5 Business Impact Summary

We identified the following business impacts:

- Targeted attack can be performed by attacker to impersonate as anyone from dataset
- Non targeted attacks makes it possible for attackers to go undetected/unrecognized in the system
- When the model is deployed on user accessible Hardwares like CCTV cameras, mobile application, etc. it is possible to extract the model and can be mis-used by attacker
- Extracted model may also lead to business competitors

Vul ID	Finding	Severity	Status
1	Whitebox Adversarial Attack: When attacker has access to single face image sample for target	HIGH	
2	Whitebox Adversarial Attack: When attacker has access to multiple image samples for target	HIGH	
3	Generic adversarial noise generation for targeted attack	HIGH	
4	Generic adversarial noise generation for non-targeted attack	MEDIUM	
5	Model Extraction attack	LOW	

1 Whitebox Adversarial Attack: When single face of victim is available

Potential Impact: HIGH

Description: From a single image of face of the target person, An attacker can generate adversarial samples to impersonate as target

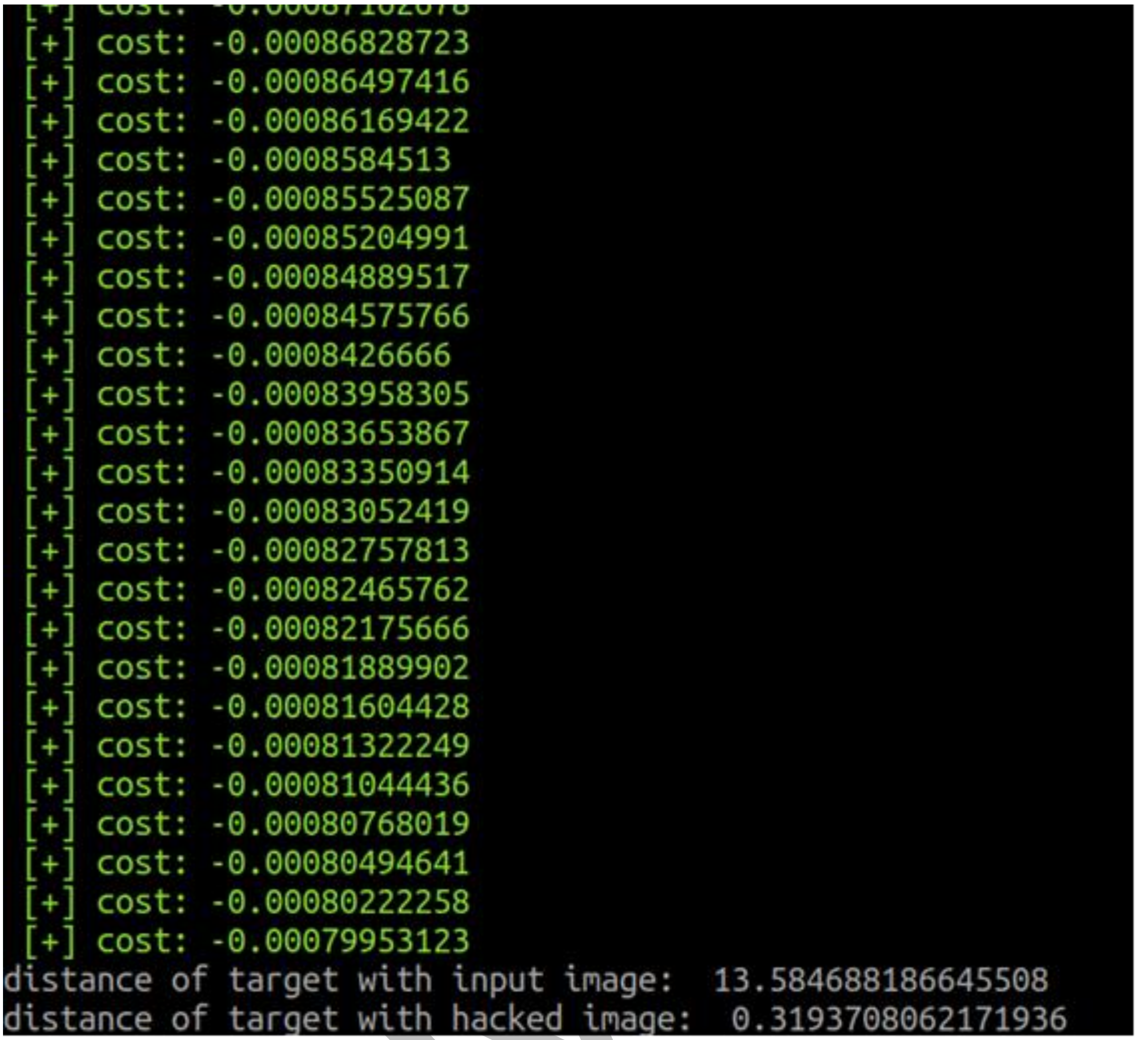
Affected Resources: Model

Business Impact: Attacker successfully impersonate a target to perform unauthorized actions

Technical Impact: Attacker can perform targeted adversarial learning attack on the model

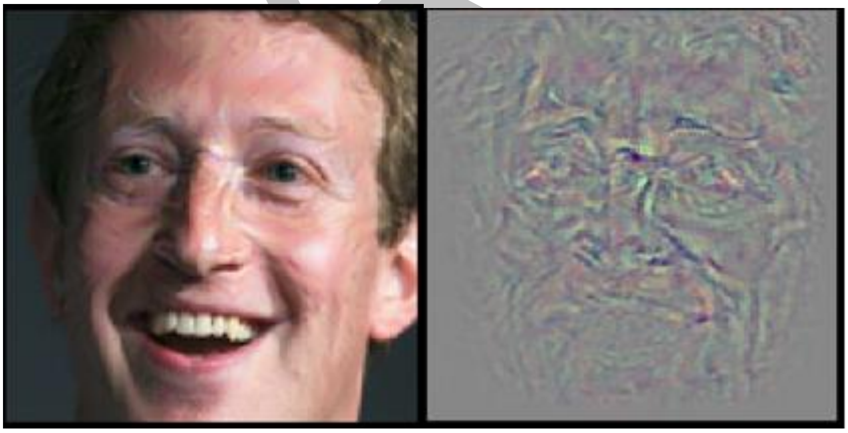
Steps to Reproduce: Following are the example faces used for attack. Image on left is input image and the one on right is target face.

Run the Adversarial Learning Attack script, to reduce distance between embeddings of above two faces.

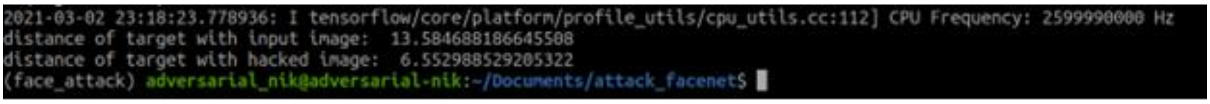


```
 [+] cost: -0.00087162078
 [+] cost: -0.00086828723
 [+] cost: -0.00086497416
 [+] cost: -0.00086169422
 [+] cost: -0.0008584513
 [+] cost: -0.00085525087
 [+] cost: -0.00085204991
 [+] cost: -0.00084889517
 [+] cost: -0.00084575766
 [+] cost: -0.0008426666
 [+] cost: -0.00083958305
 [+] cost: -0.00083653867
 [+] cost: -0.00083350914
 [+] cost: -0.00083052419
 [+] cost: -0.00082757813
 [+] cost: -0.00082465762
 [+] cost: -0.00082175666
 [+] cost: -0.00081889902
 [+] cost: -0.00081604428
 [+] cost: -0.00081322249
 [+] cost: -0.00081044436
 [+] cost: -0.00080768019
 [+] cost: -0.00080494641
 [+] cost: -0.00080222258
 [+] cost: -0.00079953123
distance of target with input image:  13.584688186645508
distance of target with hacked image:  0.3193708062171936
```

Following adversarial sample will be generated in the target directory. Image on left is the adversarial sample and the image on right is added adversarial noise.



Validate attack by running adversarial samples through FaceNet application and by finding its distance from input image.



```
2021-03-02 23:18:23.778936: I tensorflow/core/platform/profile_utils/cpu_utils.cc:112] CPU Frequency: 2599990000 Hz
distance of target with input image:  13.584688186645508
distance of target with hacked image:  6.552988529205322
(face_attack) adversarial_nik@adversarial-nik:~/Documents/attack_facenet$ 
```

Remediation:

- More preprocessing of input should help to change the adversarial noise, ultimately disabling the attack
- Adversarial training of the face classifier for all faces in the dataset

2 Whitebox Adversarial Attack: When attacker has access to multiple image samples for target

•

Potential Impact: HIGH
Description: Using multiples faces of the target person, attacker can average the face embeddings for target and can create robust adversarial samples
Affected Resources: Model
Business Impact: Attacker can successfully impersonate a target to perform unauthorized actions
Technical Impact: Averaged embeddings from multiple faces of target can act as a good directional target while generating adversarial samples
Steps to Reproduce:

- Use the same script shown above perform adversarial learning attack with target embedding as the average of all face embeddings for victim's face
- Validate the attack using validation script

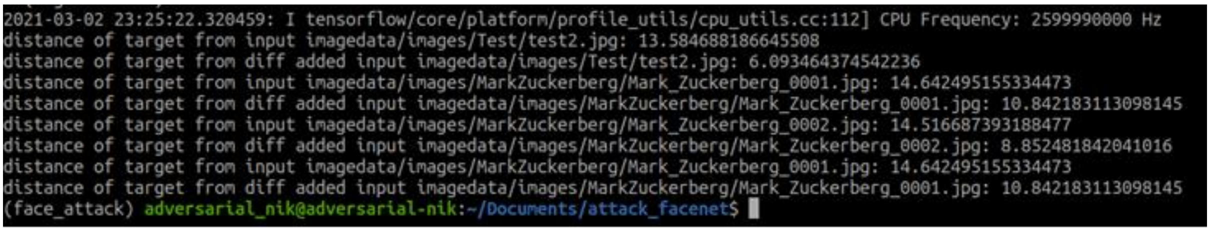
Remediation:

- More preprocessing of input should help to change the adversarial noise, ultimately disabling the attack
- Adversarial training of the face classifier for all faces in the dataset

1 Generic adversarial noise generation for targeted attack

Potential Impact: HIGH
Description: The adversarial noise crafted for targeted adversarial attack can be used to generate more adversarial samples without running the adversarial training. Which helps the attacker more adversarial samples in less time.
Affected Resources: Model
Business Impact: This allows attacker to generate adversarial sample faster
Technical Impact: Reduces the complexity of adversarial learning attacks. Hence making it quicker for attackers to impersonate.
Steps to Reproduce:

1. Generate targeted adversarial noise for target using script in section 1.1
1. Add the generated noise to different attacker's faces and check if the distance between embeddings of target face and attacker's face is reduced



```
2021-03-02 23:25:22.320459: I tensorflow/core/platform/profile_utils/cpu_utils.cc:112] CPU Frequency: 2599990000 Hz
distance of target from input imagedata/images/Test/test2.jpg: 13.584688186645508
distance of target from diff added input imagedata/images/Test/test2.jpg: 6.093464374542236
distance of target from input imagedata/images/MarkZuckerberg/Mark_Zuckerberg_0001.jpg: 14.642495155334473
distance of target from diff added input imagedata/images/MarkZuckerberg/Mark_Zuckerberg_0001.jpg: 10.842183113098145
distance of target from input imagedata/images/MarkZuckerberg/Mark_Zuckerberg_0002.jpg: 14.516687393188477
distance of target from diff added input imagedata/images/MarkZuckerberg/Mark_Zuckerberg_0002.jpg: 8.852481842041016
distance of target from input imagedata/images/MarkZuckerberg/Mark_Zuckerberg_0001.jpg: 14.642495155334473
distance of target from diff added input imagedata/images/MarkZuckerberg/Mark_Zuckerberg_0001.jpg: 10.842183113098145
(face_attack) adversarial_nik@adversarial-nik:~/Documents/attack_facenet$
```

Remediation:

- More preprocessing of input should help to change the adversarial noise, ultimately disabling the attack
- Adversarial training of the face classifier for all faces in the dataset

2 Generic adversarial noise generation for non targeted attack

Potential Impact: **MEDIUM**

Description: The adversarial noise crafted for non targeted adversarial attack can be used to generate more adversarial samples that can be misclassified or unrecognized by the system

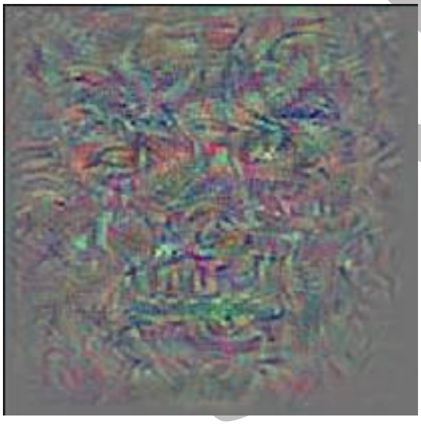
Affected Resources: Model

Business Impact: Misclassified faces can lead to authorisation system failure and unpredictable unauthorized actions.

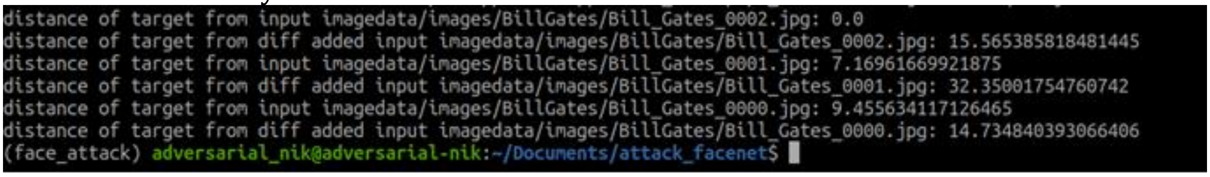
Technical Impact: An attacker can use non targeted adversarial noise to generate adversarial samples quickly. The adversarial samples may go unrecognized or misclassified by the system

Steps to Reproduce:

1. Generate non targeted adversarial noise for target using script in section 1.1 This should generate noise similar to following



1. Add the generated noise to different attacker’s faces and check if the distance between embeddings of adversarial and attacker’s faces is increased. This implies that the attacker’s face will be misclassified by the FaceNet model



```
distance of target from input imagedata/images/BillGates/Bill_Gates_0002.jpg: 0.0
distance of target from diff added input imagedata/images/BillGates/Bill_Gates_0002.jpg: 15.565385818481445
distance of target from input imagedata/images/BillGates/Bill_Gates_0001.jpg: 7.16961669921875
distance of target from diff added input imagedata/images/BillGates/Bill_Gates_0001.jpg: 32.35001754760742
distance of target from input imagedata/images/BillGates/Bill_Gates_0000.jpg: 9.455634117126465
distance of target from diff added input imagedata/images/BillGates/Bill_Gates_0000.jpg: 14.734840393066406
(face_attack) adversarial_nik@adversarial-nik:~/Documents/attack_facenet$
```

Remediation:

- More preprocessing of input should help to change the adversarial noise, ultimately disabling the attack
- Adversarial training of the face classifier for all faces in the dataset

3 Model Extraction attack

Potential Impact: LOW

Description: Models deployed on consumer hardwares like CCTV cameras, mobile applications, etc. can be extracted and used by attackers.

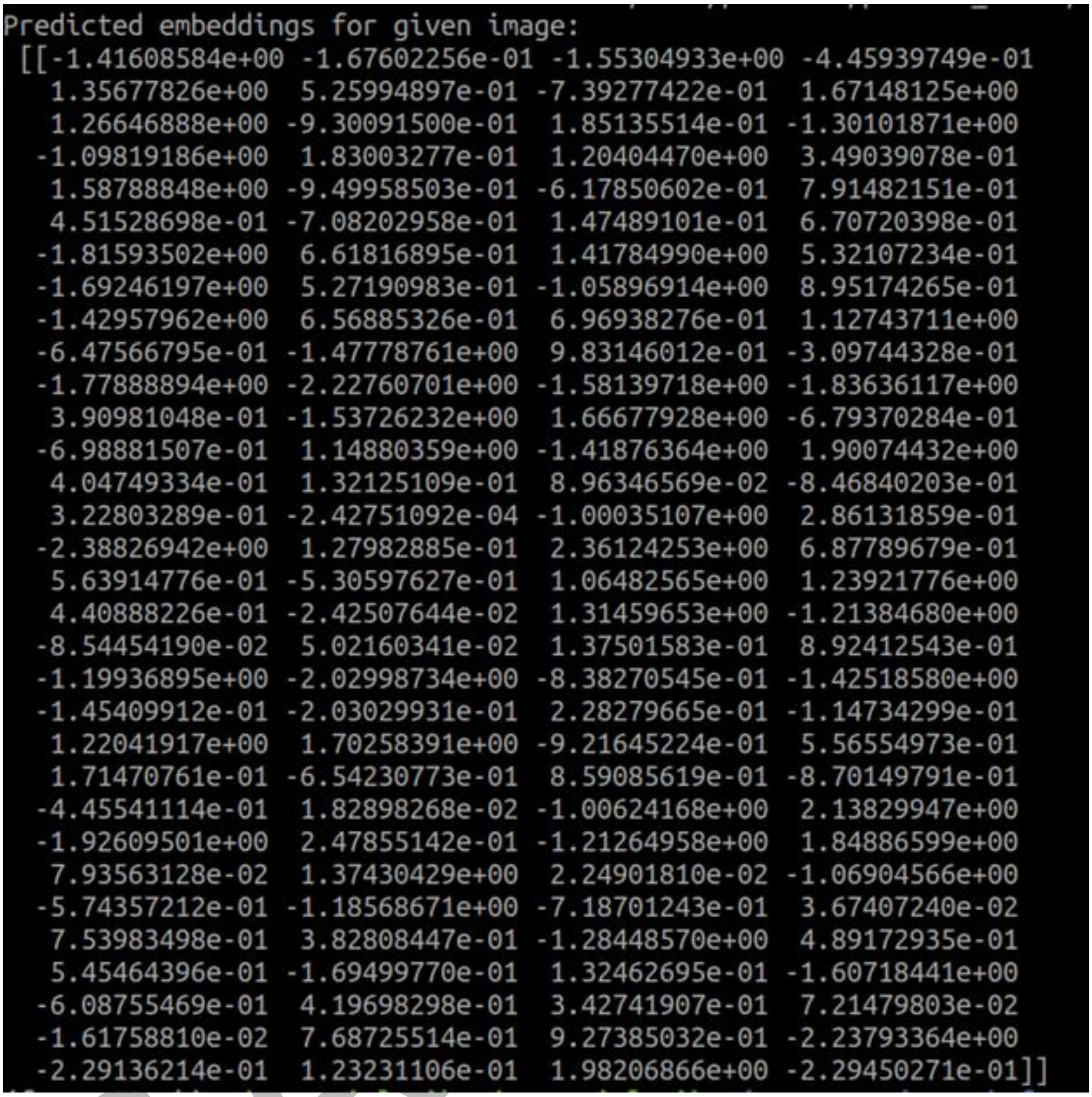
Affected Resources: Prediction pipeline

Business Impact: IP loss and revenue loss

Technical Impact: Extracted models can be used by attacker efficiently craft the adversarial learning attack and also potentially duplicate the the entire application

Steps to Reproduce:

1. Locate where the model is stored
1. Since we already know that the model is FaceNet, we do not have to perform any reverse engineering on model to figure out the input and output shape of model
1. Use stored model to make generate face embeddings



```
Predicted embeddings for given image:
 [[-1.41608584e+00 -1.67602256e-01 -1.55304933e+00 -4.45939749e-01
   1.35677826e+00  5.25994897e-01 -7.39277422e-01  1.67148125e+00
   1.26646888e+00 -9.30091500e-01  1.85135514e-01 -1.30101871e+00
  -1.09819186e+00  1.83003277e-01  1.20404470e+00  3.49039078e-01
   1.58788848e+00 -9.49958503e-01 -6.17850602e-01  7.91482151e-01
   4.51528698e-01 -7.08202958e-01  1.47489101e-01  6.70720398e-01
  -1.81593502e+00  6.61816895e-01  1.41784990e+00  5.32107234e-01
  -1.69246197e+00  5.27190983e-01 -1.05896914e+00  8.95174265e-01
  -1.42957962e+00  6.56885326e-01  6.96938276e-01  1.12743711e+00
  -6.47566795e-01 -1.47778761e+00  9.83146012e-01 -3.09744328e-01
  -1.77888894e+00 -2.22760701e+00 -1.58139718e+00 -1.83636117e+00
   3.90981048e-01 -1.53726232e+00  1.66677928e+00 -6.79370284e-01
  -6.98881507e-01  1.14880359e+00 -1.41876364e+00  1.90074432e+00
   4.04749334e-01  1.32125109e-01  8.96346569e-02 -8.46840203e-01
   3.22803289e-01 -2.42751092e-04 -1.00035107e+00  2.86131859e-01
  -2.38826942e+00  1.27982885e-01  2.36124253e+00  6.87789679e-01
   5.63914776e-01 -5.30597627e-01  1.06482565e+00  1.23921776e+00
   4.40888226e-01 -2.42507644e-02  1.31459653e+00 -1.21384680e+00
  -8.54454190e-02  5.02160341e-02  1.37501583e-01  8.92412543e-01
  -1.19936895e+00 -2.02998734e+00 -8.38270545e-01 -1.42518580e+00
  -1.45409912e-01 -2.03029931e-01  2.28279665e-01 -1.14734299e-01
   1.22041917e+00  1.70258391e+00 -9.21645224e-01  5.56554973e-01
   1.71470761e-01 -6.54230773e-01  8.59085619e-01 -8.70149791e-01
  -4.45541114e-01  1.82898268e-02 -1.00624168e+00  2.13829947e+00
  -1.92609501e+00  2.47855142e-01 -1.21264958e+00  1.84886599e+00
   7.93563128e-02  1.37430429e+00  2.24901810e-02 -1.06904566e+00
  -5.74357212e-01 -1.18568671e+00 -7.18701243e-01  3.67407240e-02
   7.53983498e-01  3.82808447e-01 -1.28448570e+00  4.89172935e-01
   5.45464396e-01 -1.69499770e-01  1.32462695e-01 -1.60718441e+00
  -6.08755469e-01  4.19698298e-01  3.42741907e-01  7.21479803e-02
  -1.61758810e-02  7.68725514e-01  9.27385032e-01 -2.23793364e+00
  -2.29136214e-01  1.23231106e-01  1.98206866e+00 -2.29450271e-01]]
```

Remediation:

- Models stored on device should be encrypted and the prediction pipeline should be highly obfuscated
- Store model on remote location if possible