# Deceptify

# Final Project Definition Template

By:

Aviv Nataf

Omer Bartfeld

Oded Warmshtien

Hadar Bar Asher

Gur Lurye

Supervisor:

Menahem Leibovitz

Git:

https://github.com/OmerBart/Deceptify

**Table of Contents**

# 1. Project description

Organizational security risks largely result from employee behavior, exacerbated by the gap between technological advancements and societal preparedness. This gap is widening with AI's ability to create convincing impersonations, leading to a surge in social engineering attacks. Our project uses AI, including deepfakes, to simulate such attacks, enhancing organizational readiness. We aim to construct AI-driven campaigns tailored to organizational settings, enabling clients to customize attack types and parameters. This approach helps employees understand emerging threats, particularly in industries lacking computer technology proficiency.Our solution addresses the need for awareness and readiness regarding AI-driven cyber threats. By providing comprehensive analytics, we facilitate data-driven decision-making, awareness, and continuous improvement of defense strategies.

In summary, our project empowers organizations to navigate the complexities of the digital landscape and fortify defenses against AI-enabled social engineering attacks. We equip employees with knowledge and skills for long-term resilience.

# 2. Related Work

Our project involves extensive research on security breaches and social engineering theory, examining pre-trained models that can generate attacks. We also identify and utilize existing models, while developing strategies to maintain maximum confidentiality due to the potential misuse of our tools. A graphical interface with a local server, secured by robust security measures, offers users an intuitive experience while prioritizing safety. By combining user-friendly design and stringent security protocols, we provide a comprehensive solution that prioritizes both usability and security.

# 3. Functional Description / Requirements

We will optimize existing tools to create deepfake-based attacks, using pytiket for a local GUI and flask for a web server. We will also utilize PyTorch, TensorFlow, and open-source models to develop and train deep learning models for generating videos, audio, and other deepfakes.

The graphical user interface (GUI) will be a local interface, similar to a Jupyter notebook server, allowing clients to initiate attacks and receive analytical reports and recommendations for effective strategies. Clients will have the option to download all files locally, ensuring maximum security without relying on cloud or internet services. Regular software updates will be provided, incorporating new insights from attacks, recommendations, and security updates.

We will source datasets relevant to this project, and if needed will utilize a paid hosting service to allow for a more robust training environment.

# 4. Architecture

Our project adopts a Microservices architecture to provide a robust and flexible framework for simulating social engineering attacks using AI technology. This architectural pattern offers several benefits tailored to our project's goals

**-insert how we will finetune model-**

Each service offered will be equipped with its own dedicated functionality. For instance, within each generation service, a specialized mini-model will be deployed. These mini-models will be finely tuned to the specific requirements of their corresponding service, ensuring that the output they generate precisely matches the needs of the generation process. As an example, consider the voice generation service where customers will be prompted to submit recordings of up to 10 minutes in length, capturing the speech pattern of the intended subject.

Once trained, we will use the model to produce synthetic videos or images by inputting source media. Evaluate the quality of the generated deepfakes using metrics like FID or SSIM and refine the model as needed. Approach the use of deepfakes for social engineering cautiously, ensuring ethical compliance and obtaining necessary permissions.

Then, we will integrate a GUI with a local server to streamline the attacks' generation process and enhance user interaction. The GUI will allow users to input parameters and customize attacks conveniently, while the local server will ensure enhanced security and data privacy by conducting operations locally. Clients will receive regular software updates to stay equipped with the latest tools and models. This combined approach aims to empower organizations to proactively safeguard against AI-enabled social engineering attacks.

To construct a deepfake detector, we will leverage Long Short-Term Memory (LSTM) and Convolutional Neural Network (CNN) architectures. These models will be developed in-house to effectively identify manipulated media content. LSTM networks excel in processing sequential data, making them ideal for detecting temporal patterns in videos. Meanwhile, CNNs are proficient in extracting spatial features from images, enabling the detection of visual inconsistencies indicative of deepfake manipulation.

# 5. Work plan

Our work plan is presented in trello in the following link:

https://trello.com/b/ZkLf7GIW

# 6. Client side

The client UI will be an app which the client will need to download, and opening the app will open a local server on the user's computer;

Then at the server, a foundational element will be a beginner chatbot, leveraging Language Model (LLM) technology. This chatbot will serve as the initial point of interaction for customers, interpreting their requests and directing them to the appropriate function or service within our system. The services offered will span a range of functionalities including addressing general inquiries, generating video content, creating voice recordings, producing images, analyzing attacks, providing feedback on attacks, and potentially even facilitating real-time communication.

## 6.1. Usage Illustration

Some scenarios about the usage of the app are shown below.

In one scenario, the user will ask for an analysis of the most successful attack that the program has made, which in turn the AI will give the user the most successful attack and additionally will add statistics about the percentage of success and the number of times it was created.

In another scenario, the user will ask for a video generation attack, in which the AI will ask for some information about the person whom the video will be generated based of, in addition to photos, videos, and voice recording of the person to maximize how real the video

will look. After the user will send the information and data, the AI will generate the attack and send the user the completed attack.

In each attack generation scenario, following the execution of an attack, whether successful or unsuccessful, a comprehensive analytical report will be generated. Clients will be prompted to fill out a detailed form providing insights into various aspects of the attack, such as its success rate, the number of individuals affected, and the accuracy of the execution. This feedback mechanism will enable continuous improvement of our models and software. Once the form is completed, it will be transmitted to us via our cloud service or server, facilitating ongoing refinement of our capabilities.

## 6.2. Mockup

Graphic illustration of the different screens in your app.

# 7. Server Side

Due to the sensitive nature of this project, most of the operations will be ran locally on the client's computer to allow for added security. The main function of the server side will be to gather anonymous data from users, and to use that information to train and update the model, which when sufficiently updated will be pushed as an update to the clients.

In addition to data from clients' campaigns we will continue to build a more robust dataset of known and brand new attacks/technologies which will also be added to the tools/knowledge base of the model.

Locally the server will be responsible for loading all relevant files onto the target computer. This includes deploying models locally, as well as implementing front and back-end functionalities. To accommodate this process, clients may be required to allocate specific storage resources. This ensures seamless integration of our system with the client's infrastructure, optimizing performance and accessibility.

# 8. References

**DeepFakes attacks:**

[1] DeepFakes: a New Threat to Face Recognition?
Assessment and Detection Pavel Korshunov and Sebastien Marcel ´

[2] Detect Video Forgery by Performing Transfer Learning on Deep Neural Network

[3] Research Article: Face Spoof Attack Recognition Using Discriminative Image Patches
Zahid Akhtar and Gian Luca Foresti

[4] SWAPPED! Digital face presentation attack detection via weighted local
magnitude pattern

**DL & DNN & GAN Models:**

[5] Deep Learning for Deepfakes Creation and Detection: A Survey
Thanh Thi Nguyena, Quoc Viet Hung Nguyenb, Dung Tien Nguyena, Duc Thanh Nguyena,
Thien Huynh-Thec,Saeid Nahavandid, Thanh Tam Nguyene, Quoc-Viet Phamf, Cuong
M.Nguyeng.

[6] DeepSecure: A Real-Time Deep Learning-Based System for Enhancing
Cybersecurity in Social Media through DeepFake Detection using LSTM and
ResNext CNN

[7] Face Forensics in the Wild

[8] Fast Face-swap Using Convolutional Neural Networks

[9] FSNet: An Identity-Aware Generative Model for Image-based Face Swapping

[10] On the Detection of Digital Face Manipulation

[11] SwapItUp: A Face Swap Application for Privacy Protection

[12] Towards Generalizable Deepfake Detection with Locality-Aware AutoEncoder

[13] U-Net: Convolutional Networks for Biomedical Image Segmentation

[14] Watch your Up-Convolution: CNN Based Generative Deep Neural Networks are Failing to Reproduce Spectral Distributions

**DeepFake's models identifiable patterns:**

[15] Image inpainting

[16] Face X-ray for More General Face Forgery Detection

[17] Speaker Inconsistency Detection in Tampered Video

[18] A Comparative Evaluation of Local Feature Descriptors for DeepFakes Detection

[19] Learning Face Representation from Scratch

[20] BINARY GABOR PATTERN: AN EFFICIENT AND ROBUST DESCRIPTOR FOR TEXTURE CLASSIFICATION

[21] Face Recognition with Local Binary Patterns

[22] SURF: Speeded-Up Robust Features

[23] Automated Face Swapping and Its Detection

[24] Indexing chromatic and achromatic patterns for content-based color image retrieval

**DeepFake's videos temporal inconsistencies:**

[25] Detecting Deep-Fake Videos from Phoneme-Viseme Mismatches

[26] EXPOSING DEEP FAKES USING INCONSISTENT HEAD POSES

[27] Speaker Inconsistency Detection in Tampered Video

[28] Are ChatGPT and Deepfake Algorithms Endangering the Cybersecurity Industry? A Review

[29] Cyber Vaccine for Deepfake Immunity

**Variants of GAN (Generative adversarial networks):**

[30] The Cramer Distance as a Solution to Biased Wasserstein Gradients

[31] DEMYSTIFYING MMD GANS

[32] PROGRESSIVE GROWING OF GANS FOR IMPROVED
QUALITY, STABILITY, AND VARIATION

[33] SPECTRAL NORMALIZATION FOR GENERATIVE ADVERSARIAL NETWORKS