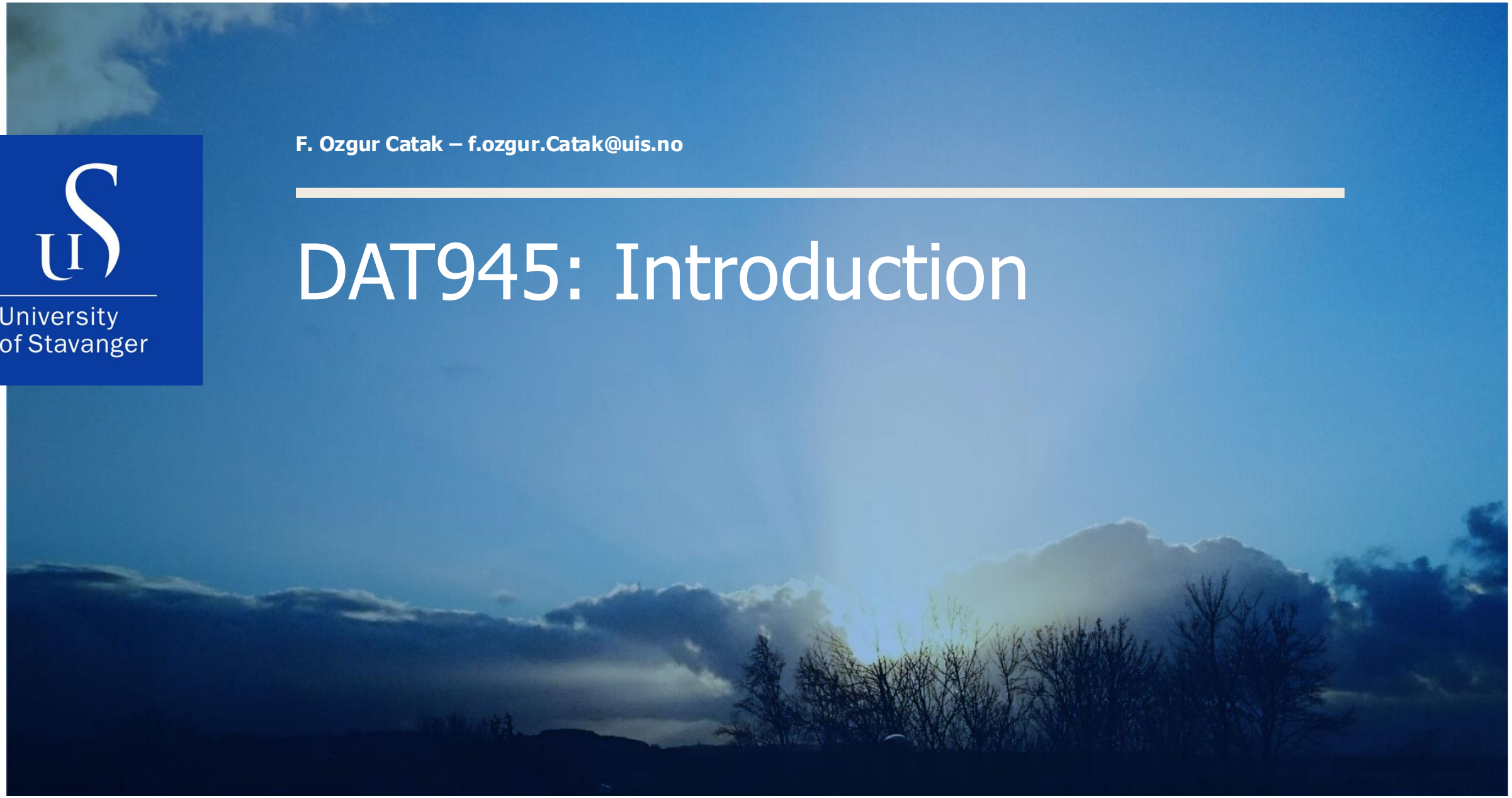
F. Ozgur Catak – f.ozgur.Catak@uis.no

University of Stavanger

DAT945: Introduction



Introduction

- Canvas Login
- Project based course
- Jupyter notebooks
- 2 Assignments
  - Assignment 1: Oct 19,
  - Assignment 2 (Final Project): (Nov 30th - 5th December online presentation - 10 mins presentation 5 mins questions)
    - **Expectation**: both a **comprehensive report (Format: IEEE/Springer etc)** and the **accompanying code** for the project.

Final Report Format

The report should include the following sections:

1. **Introduction**: Clearly state the research problem, objectives, and motivation for the project.
2. **Literature Review**: Provide an overview of the relevant literature and existing work in the field, highlighting key findings, methodologies, and gaps in knowledge.
3. **Methodology**: Describe the research methodology, algorithms, frameworks, and techniques used in the project. Explain the rationale behind the chosen approaches and any modifications or improvements made.
4. **Implementation Details**: Provide a detailed explanation of the implementation, including the dataset used, preprocessing steps, model architecture, hyperparameters, and training/validation procedures.
5. **Results and Analysis**: Present and discuss the results obtained from the experiments conducted. Include quantitative and qualitative analyses, visualizations, and comparisons with existing approaches, if applicable. Discuss any insights or observations derived from the results.
6. **Discussion**: Interpret the findings in light of the research objectives and address any limitations or challenges encountered during the project. Discuss the implications of the results and potential future directions for research.
7. **Conclusion**: Summarize the key findings, contributions, and implications of the project.

Final Report Format - Implementation

- Include their code in a separate folder or submit it as a separate file.
- The code should be well-documented, modular, and readable, with comments explaining the purpose and functionality of each section.
- It should also include instructions for running the code, specifying any dependencies or prerequisites.

Example Projects

- You can choose from the list or you can suggest another one similar to this list.
- Submisson deadline:

Healthcare

1."Explainable AI for Personalized Treatment Recommendations"
2."Privacy-Preserving Deep Learning for Medical Imaging Diagnostics"
3."Optimizing Clinical Trial Recruitment with Natural Language Processing"
4."Automated Detection of Rare Diseases from Genomic Data Using GANs”

Natural Language Processing (NLP)

1."Sentiment Analysis in Social Media Posts Using Transformer Models"
2. "Bias Detection and Mitigation in Large Language Models"
3."Multilingual Text Summarization with Reinforcement Learning”

Finance

1."Anomaly Detection in Financial Transactions Using Deep Learning"
2."Fraud Detection in Credit Card Transactions with Federated Learning”

Large Language Models (LLM)

1."Improving Language Model Robustness with Adversarial Training"
2."Exploring Ethical Considerations in Large Language Model Deployment"

Computer Vision

1."Adversarial Attacks and Defenses in Image Classification Models"
2."Real-Time Facial Recognition with Privacy Preservation"

Course Content

Module	Theme
1	Introduction to Trustworthy AI
2	Adversarial Machine Learning
3	Uncertainty Quantification in AI
4	Adversarial Machine Learning
5	Explainable and Interpretable AI
6	Cryptography and Federated Learning based Privacy Preserving AI

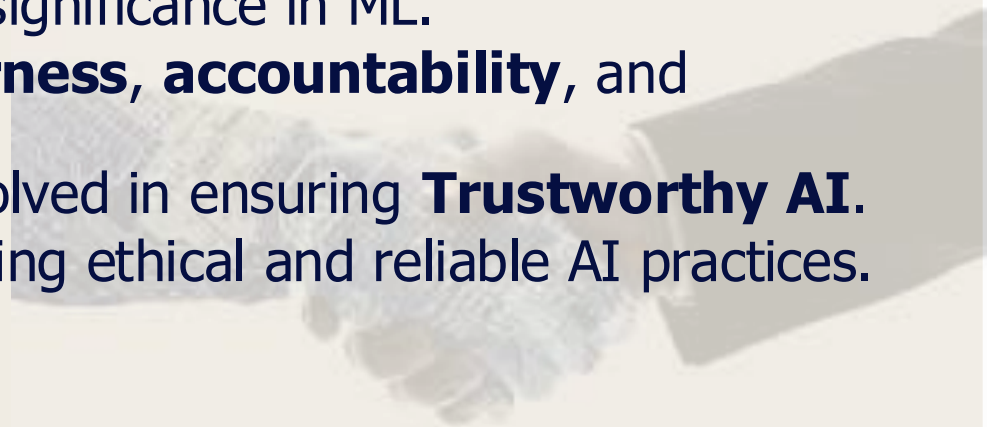
My GitHub Repo with several code examples
https://github.com/ocatak/trustworthyai

Required libraries

- **Cleverhans** (for adversarial attacks) - https://github.com/cleverhans-lab/cleverhans
- **Uncertainty Wizard** (for uncertainty quantification) - https://github.com/testingautomated-usi/uncertainty-wizard
- **XAI**:
  - https://github.com/marcotcr/lime
  - https://github.com/slundberg/shap
- **PyFhEL** (for cryptography) - https://pyfhel.readthedocs.io/en/latest/
  - PyFhEL installation could be painful on Mac computers. I am using Mac, at least we can use my solution to install it :)

Trustworthy AI

- **Introduction**:
  - AI has become an integral part of our lives, impacting various aspects of society.
  - **Trust** is crucial for AI systems to be accepted and adopted by users, organizations, and society at large.
  - In this course, we will explore the principles and techniques that contribute to building Trustworthy AI.
- **Learning Objectives**:
  - Understand the concept of Trustworthy AI and its significance in ML.
  - Explore key principles such as **transparency**, **fairness**, **accountability**, and **security** in AI systems.
  - Learn about the challenges and considerations involved in ensuring **Trustworthy AI**.
  - Discover **frameworks** and **standards** for promoting ethical and reliable AI practices.

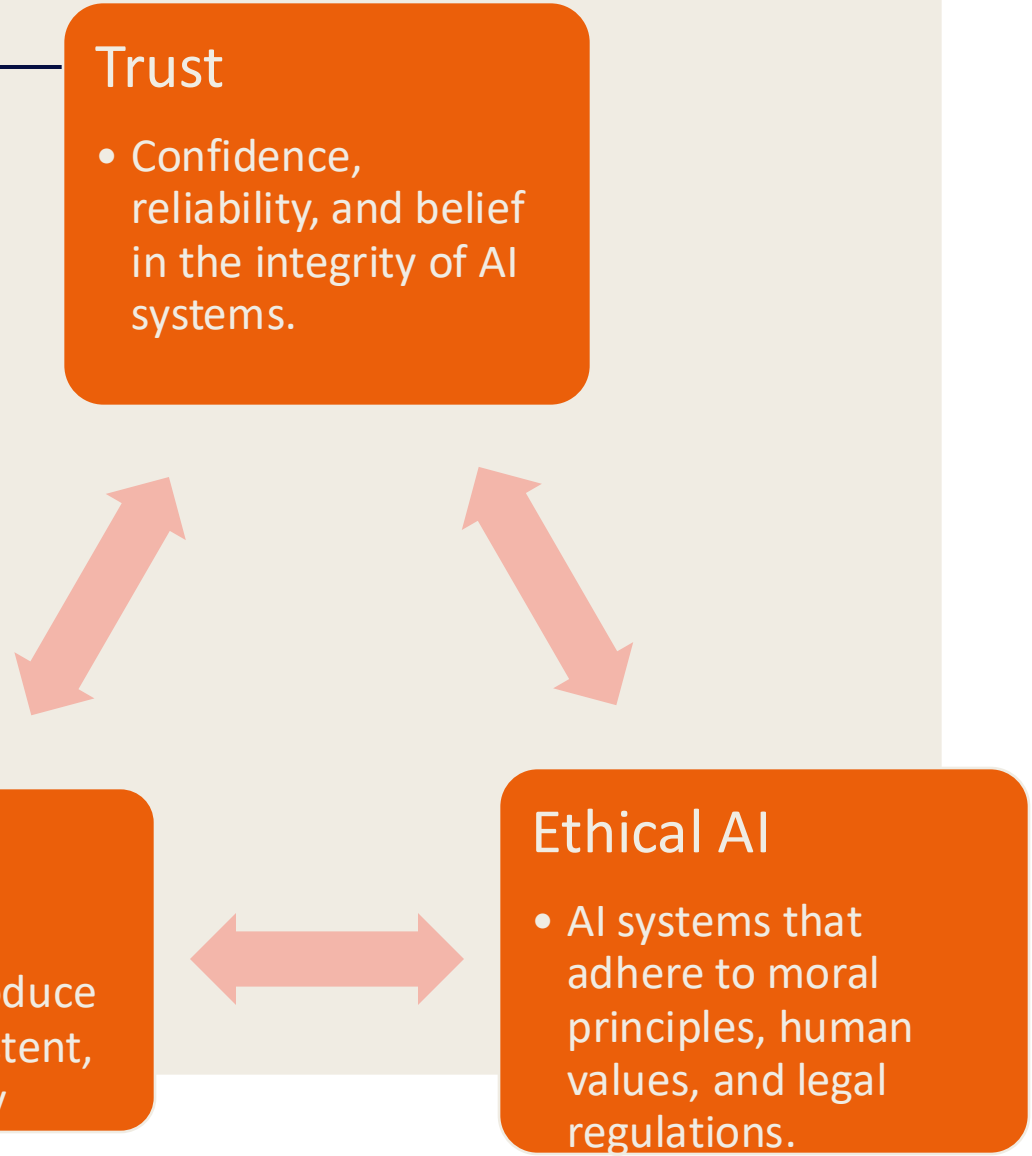


What is Trustworthy AI?

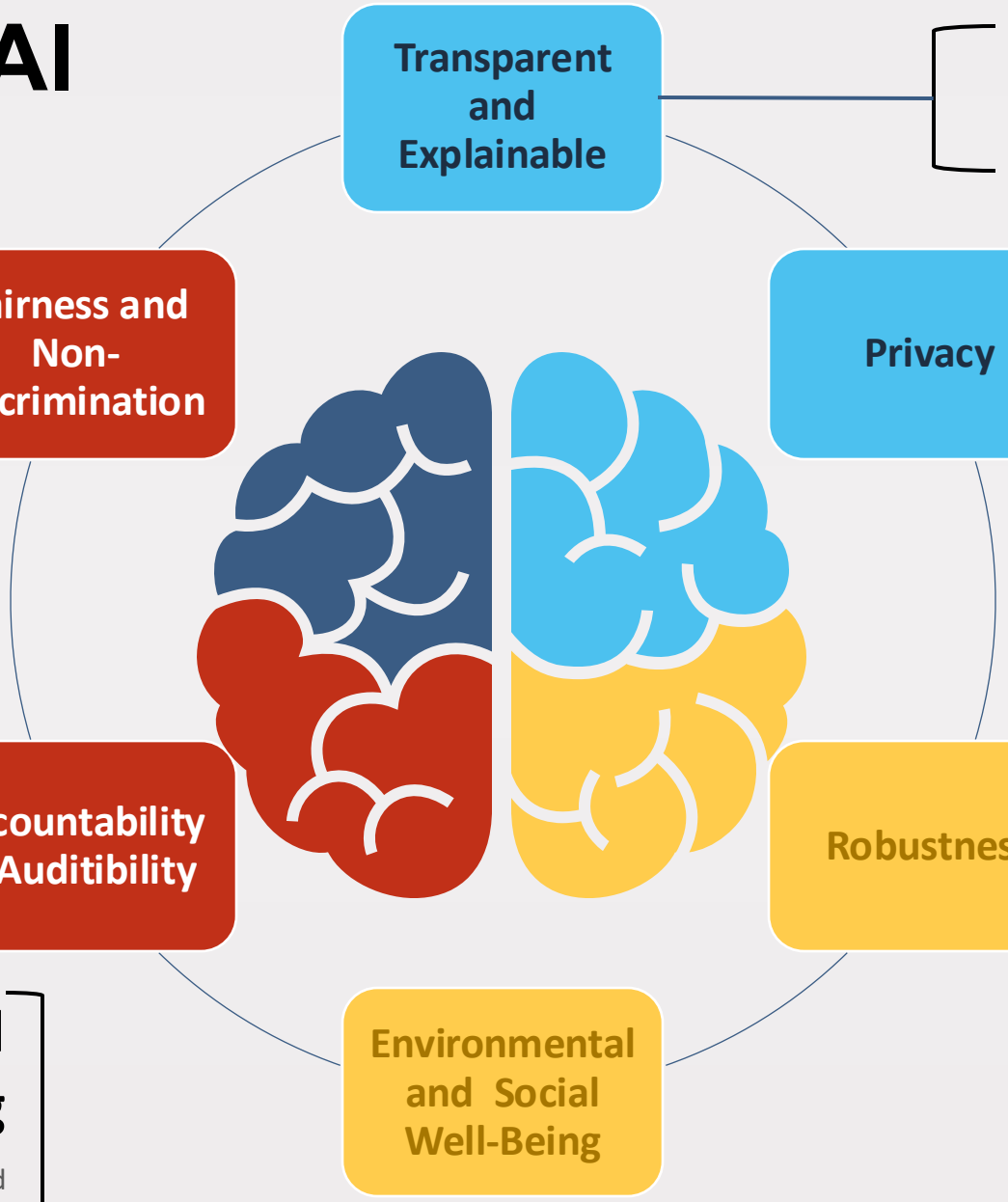
Key Points:

- Refers to the development and deployment of AI systems that are **reliable**, **ethical**, and **accountable**.
- Involves ensuring that AI systems are **transparent**, **fair**, **unbiased**, and **respect privacy** and **data protection**.
- Aims to **address the potential risks and challenges** associated with AI technologies and promote public trust.

- **Fairness**: to prevent discrimination based on race, gender, or other protected attributes.
- **Privacy-preserving techniques** to protect sensitive data while leveraging AI for insights.
- **Explainability and interpretability:**to understand the decision-making process.



Trustworthy AI



Transparent and Explainable

- The traceability of AI systems should be ensured
- Certain level of understanding about the decisions taken by an AI system

Privacy-preserving AI

AI systems should avoid leaking any private information

Robust AI

AI system should be robust to the adversarial perturbations of inputs and AI should make secure decisions

Environmental & Social Well-being

AI system should be sustainable and environmentally friendly.

Responsible AI

AI system should be assessed by a third party and, when necessary, assign responsibility for an AI failure,

Fairness and Non-discrimination

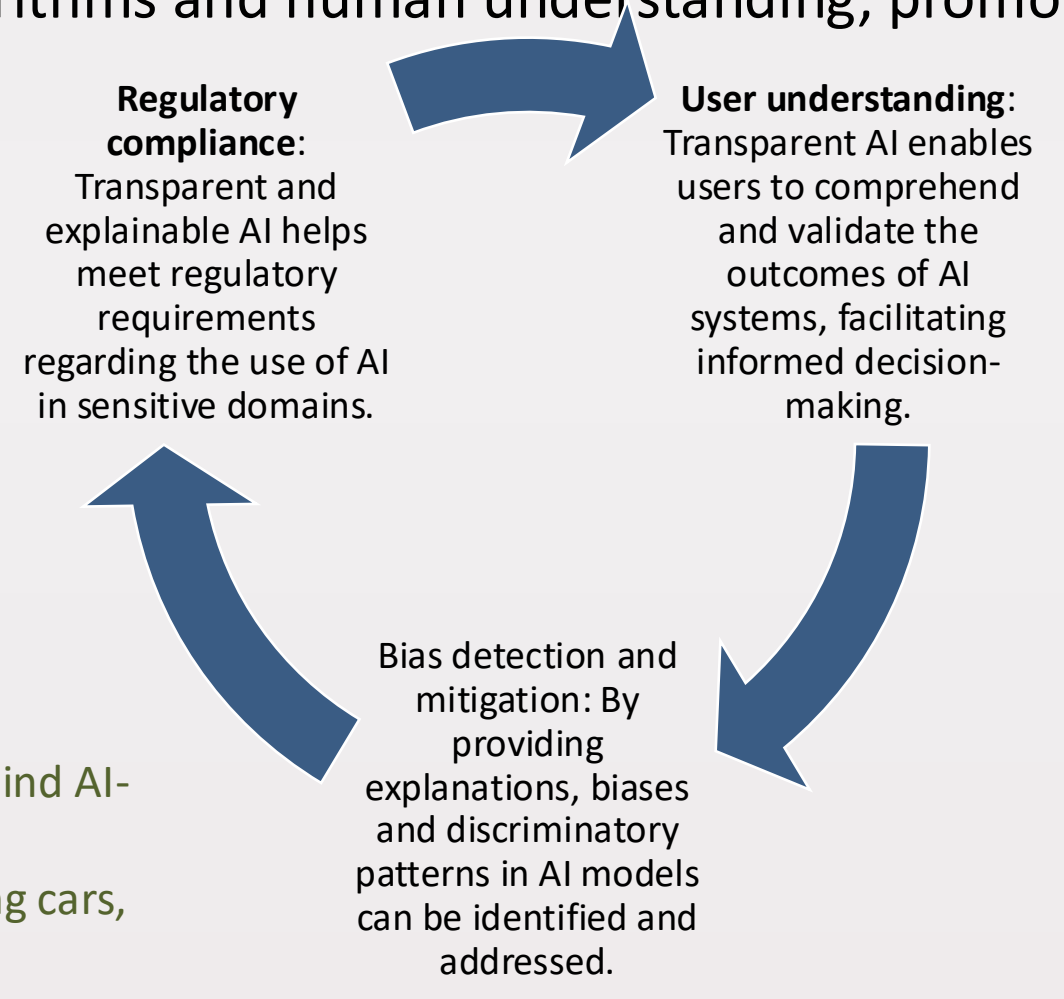
AI systems should not lead to any kind of discrimination in relation to race, religion, gender, sexual, ethnic, origin or any other personal condition

Transparent and Reliable

- Refers to the ability of AI systems to provide understandable and interpretable explanations for their decisions and behaviors.
- Aims to bridge the gap between complex AI algorithms and human understanding, promoting trust and accountability.

Importance:

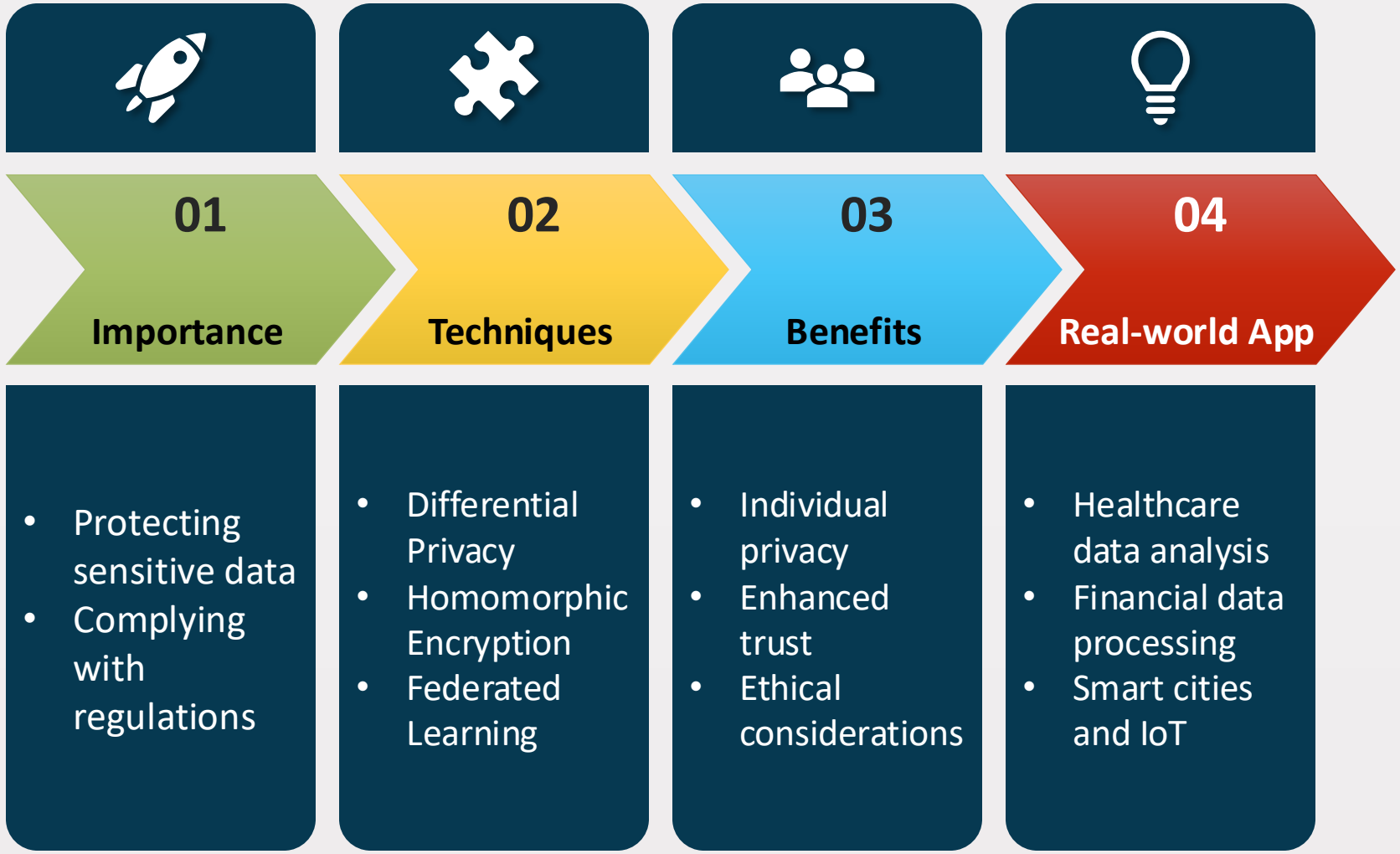
- **Building trust**: Transparent AI systems enable users and stakeholders to understand how AI arrives at its decisions, leading to increased trust and confidence.
- **Accountability**: Explainable AI allows for accountability and scrutiny of AI systems, ensuring they align with ethical standards, legal requirements, and societal expectations.



- **Loan approval systems**: Providing explanations for loan approval decisions to customers and regulators.
- **Healthcare diagnostics**: Explaining the factors and reasoning behind AI-based medical diagnoses.
- **Autonomous vehicles**: Transparent decision-making in self-driving cars, enabling users to understand why certain actions are taken.

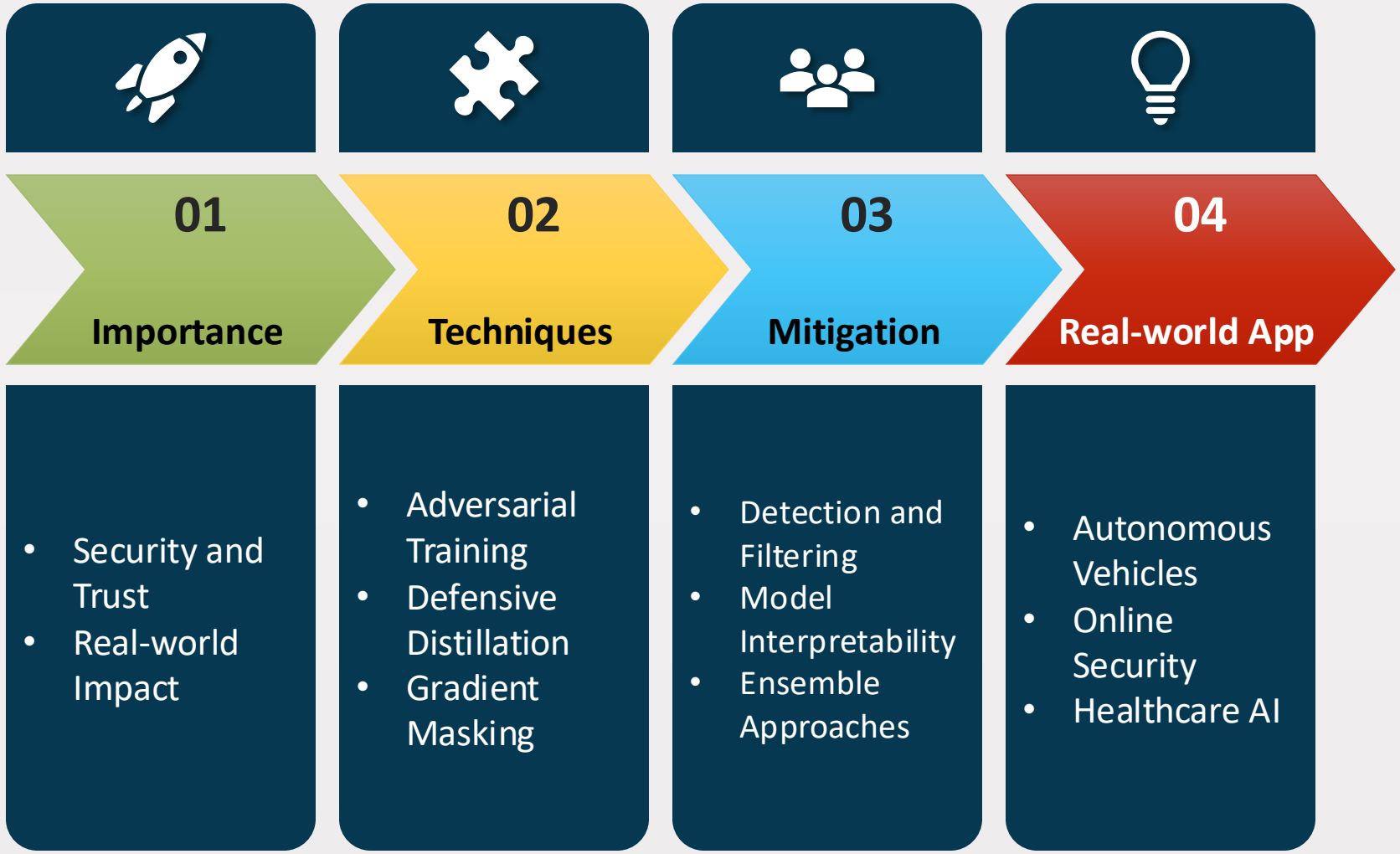
Privacy-preserving AI

Safeguarding Data Privacy in Artificial Intelligence



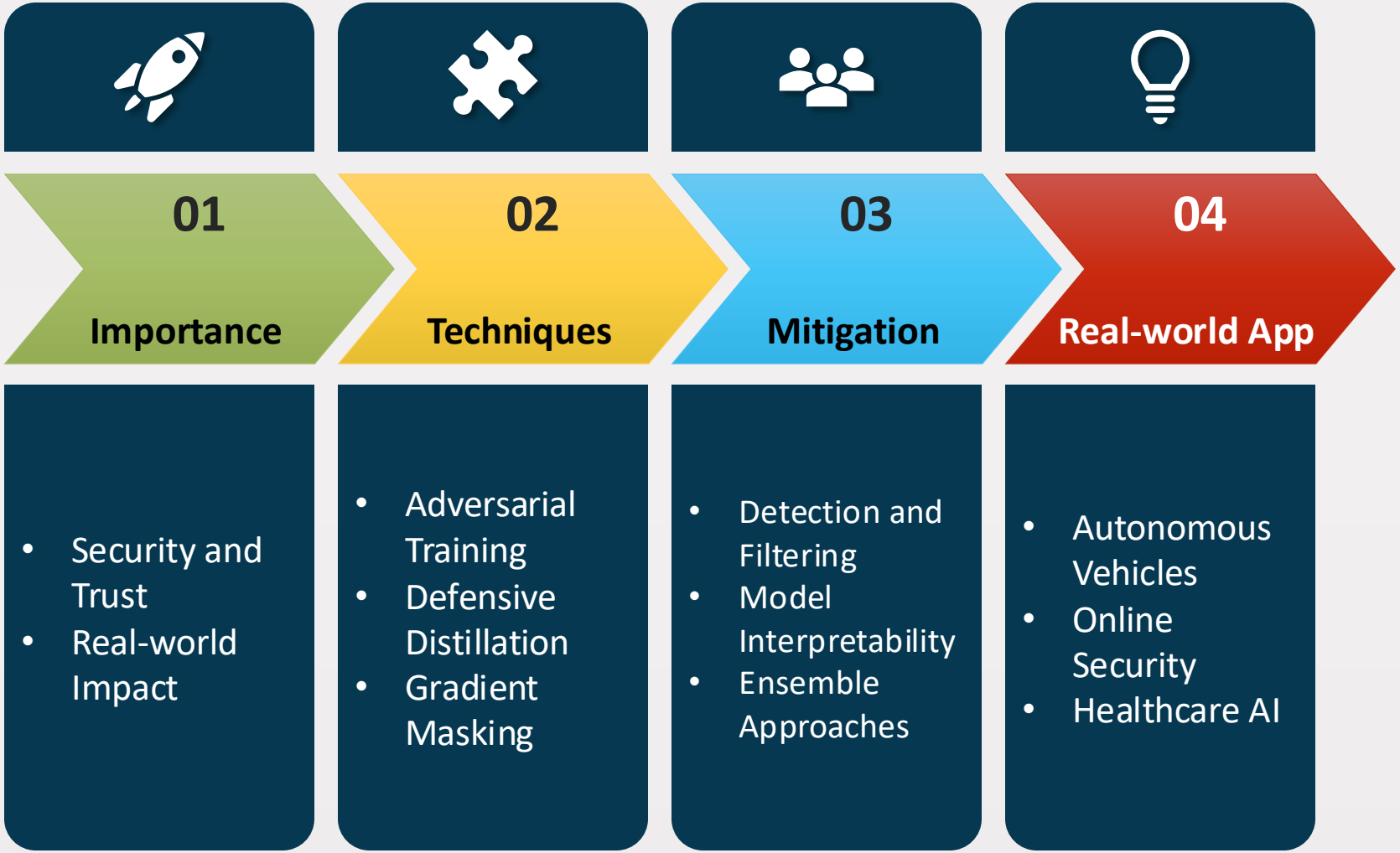
Robust AI

Defending Against Adversarial Attacks and Ensuring Resilience



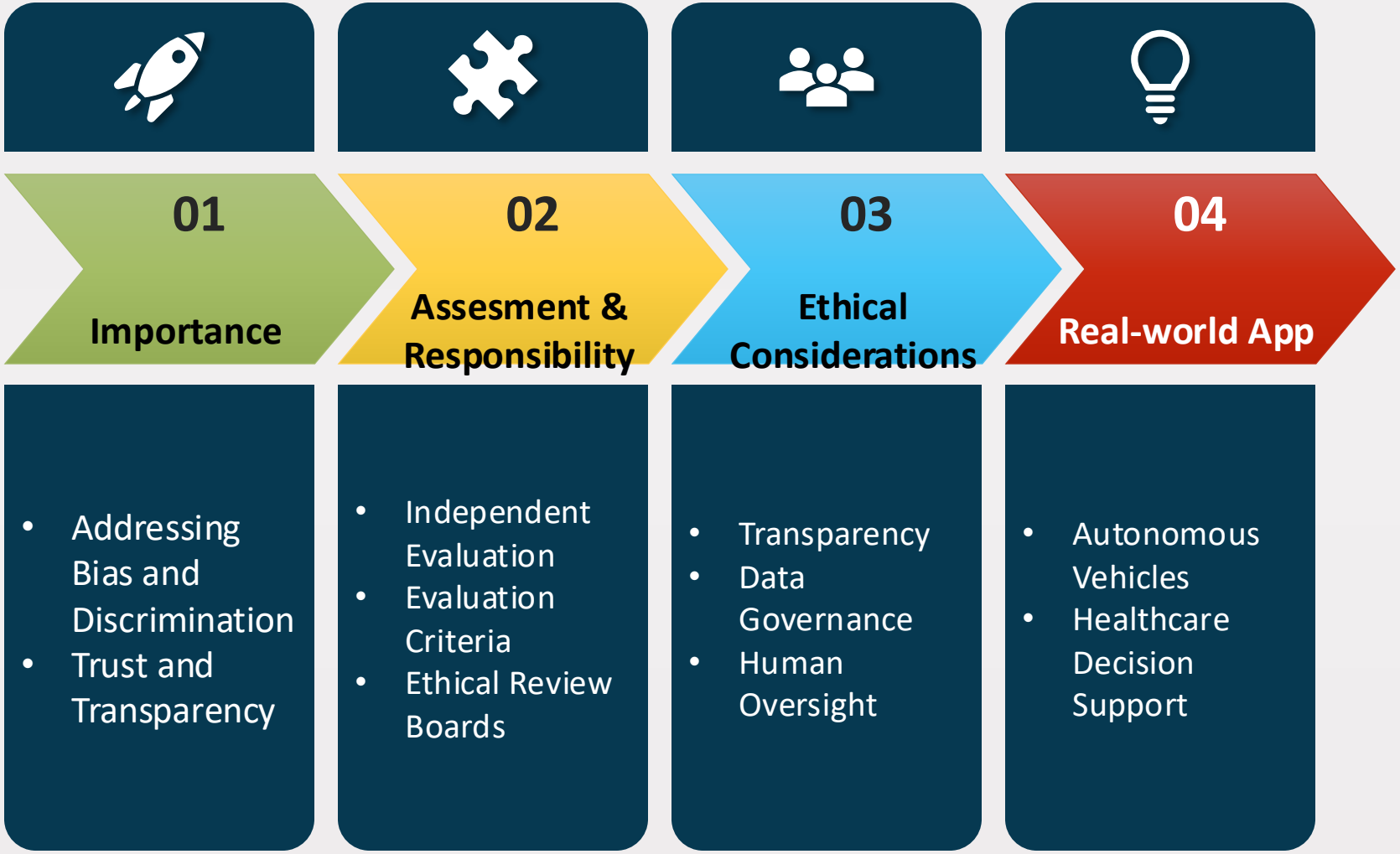
Environmental and Social Well-Being

Building Sustainable and Responsible AI Systems



Responsible AI

Assessing and Assigning Responsibility for AI Failures



Trustworthy AI Frameworks and Standards

Frameworks

- IEEE Ethically Aligned Design
- OECD AI Principles
- EU Ethics Guidelines for Trustworthy AI

Standards

- ISO/IEC 27001
- ISO/IEC 20547
- NIST AI Risk Management Framework
- AI4People Ethical Guidelines for AI

Benefits of Frameworks and Standards

- Enhanced transparency and explainability
- Increased accountability
- Improved public trust and acceptance of AI technologies
- Facilitation of international collaboration and harmonization

https://standards.ieee.org/wp-content/uploads/import/documents/other/ead_v2.pdf (Page 4)

Case Studies and Examples

Building Sustainable and Responsible AI Systems

Education

Trustworthy AI is reshaping education by enabling personalized learning experiences and adaptive assessments.

Case Study: Duolingo

Duolingo, a language-learning platform, employs AI algorithms to personalize language lessons and adapt assessments based on individual proficiency levels.

By utilizing trustworthy AI, educational platforms can deliver tailored instruction, improve student engagement, and facilitate effective learning outcomes

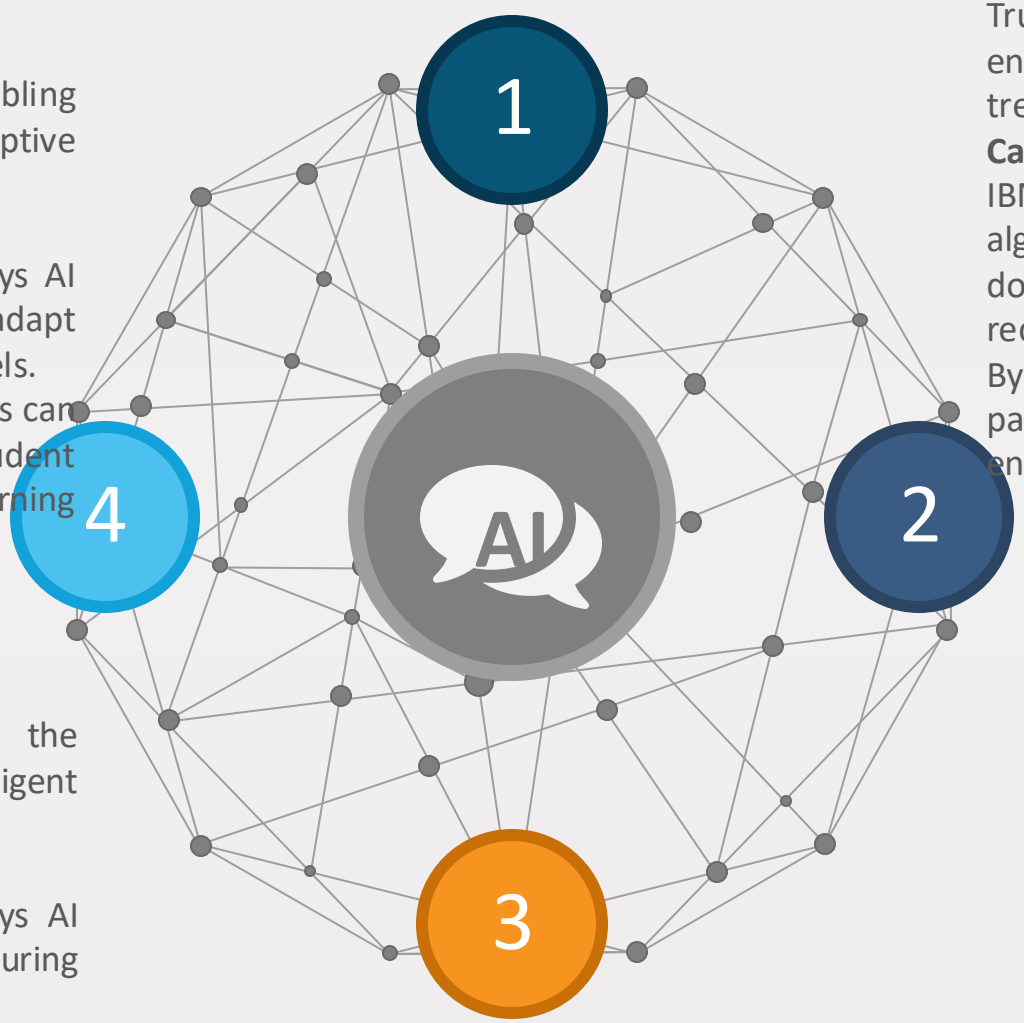
Transportation

Trustworthy AI plays a crucial role in the development of autonomous vehicles and intelligent traffic management systems.

Case Study: Waymo

Waymo, a subsidiary of Alphabet Inc., employs AI technologies to power its self-driving cars, ensuring safe navigation and efficient transportation.

By incorporating trustworthy AI, transportation systems can enhance road safety, reduce congestion, and optimize travel experiences.



Healthcare

Trustworthy AI is transforming healthcare by enabling accurate diagnosis and personalized treatment plans.

Case Study: IBM Watson Health

IBM Watson Health utilizes trustworthy AI algorithms to analyze medical data, aiding medical doctors in diagnosing complex diseases and recommending treatment options.

By leveraging AI, healthcare providers can improve patient outcomes, reduce medical errors, and enhance the overall quality of care.

Finance

Trustworthy AI is revolutionizing the finance industry by enhancing fraud detection and risk assessment capabilities.

Case Study: JPMorgan Chase

JPMorgan Chase utilizes AI-powered algorithms to identify patterns and anomalies in financial transactions, effectively detecting fraudulent activities.

By employing trustworthy AI, financial institutions can protect customer assets, strengthen security measures, and mitigate financial risks.

Techniques

03.

Cryptography & Federated Learning

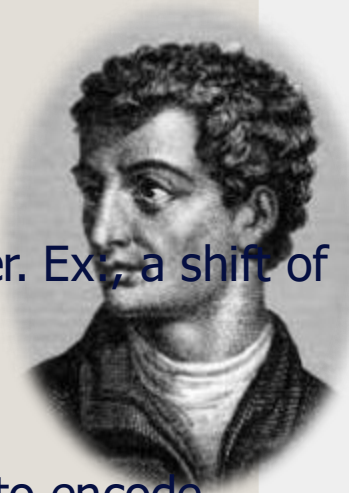
Cryptography – A Short History

- A powerful tool in safeguarding sensitive data. **May sound complex**. Fundamental to ensuring the **security and privacy of our medical records**.

- Start: ancient civilizations such as **Egypt** and **Mesopotamia**. **Need for secrecy in communication**
  - The use of **hieroglyphic symbols** or **substituting letters**.

Caesar Cipher

  - Attributed to Julius Caesar during the Roman Empire.
  - A simple substitution cipher, shifting each letter in the alphabet by a fixed number. Ex:, a shift of 3 would turn "A" into "D" and "B" into "E."



- **The Renaissance and the Polyalphabetic Ciphers**
  - Great advancements in **art**, **science**, and **cryptography**.
  - Leon Battista Alberti invented the polyalphabetic cipher, used multiple alphabets to encode messages. Increased the **complexity of deciphering** the encoded text.
- **German military during World War II.**

Developed in the 1920s and used during WWII. Its complex system of rotating wheels and electrical pathways made it extremely difficult to break the encrypted messages.
British Mathematician **Alan Turing** cracked the machine's principle

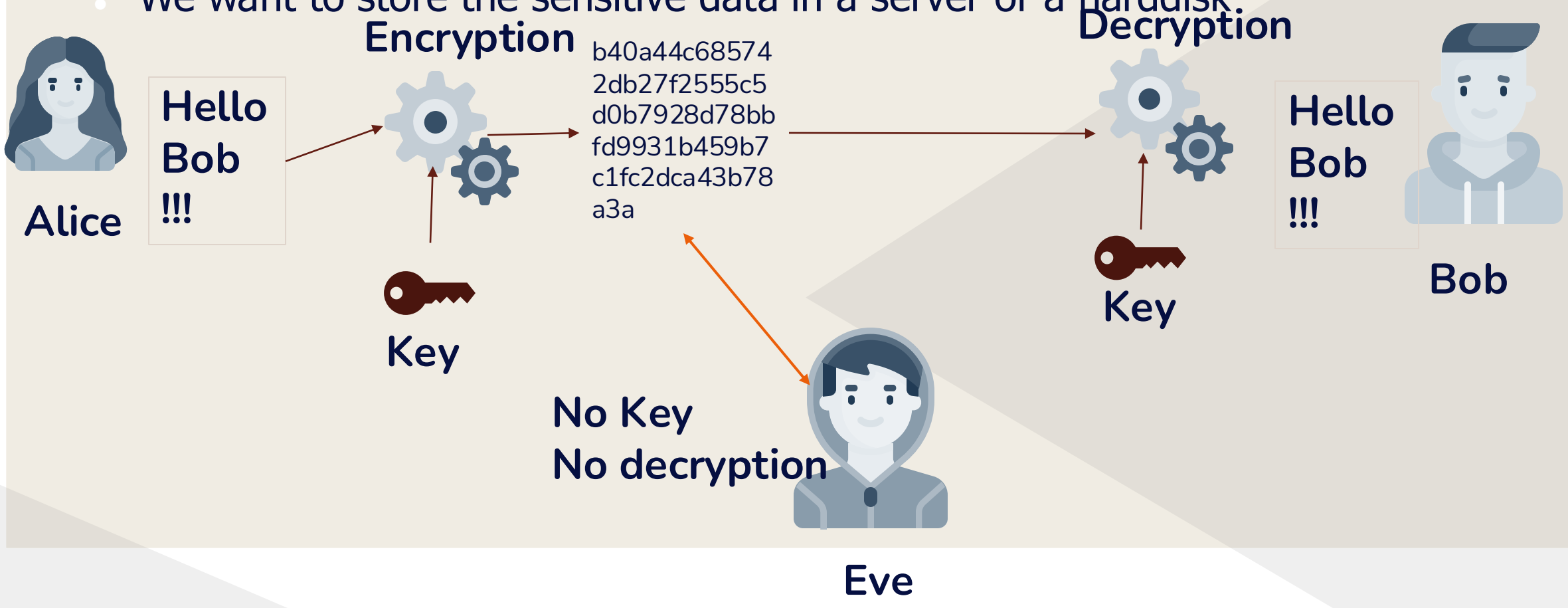
e Digital Age and Public Key Cryptography (Civilians started to use it !!!)

IBM 1970s. **Whitfield Diffie** and **Martin Hellman**.

  - Public key cryptography introduced a revolutionary idea

Encryption, Decryption, Crypto Key

- What are they?
- We want to send a senstive message over unsecure channel (like internet)
- We want to store the sensitive data in a server or a harddisk.



What is Homomorphic Encryption

PURPOSE

Data can remain **confidential** while it is processed in **untrusted environments.**

SAME STRUCTURE

Greek words for “same structure.”

HOMOMORPHISM

A structure-preserving map between two algebraic structures

PROTECTION

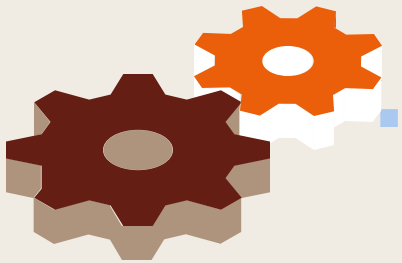
Describes the transformation of one data set into another while preserving

TRADITIONAL vs HOMOMORPHIC CRYPTO

TRADITIONAL

- **Functionality**:
  allows data to be securely **transmitted** or **stored**, **preventing unauthorized access**. To perform any computations on the encrypted data, it needs to be **decrypted first**, which exposes it to potential security risks.
- **Operations**:
  supports encryption and decryption.
- **Limitations**
  does not allow performing computations on encrypted data directly without decryption, limiting the ability to process sensitive information while preserving privacy.

Cloud Computing



AI Training
& Deployment

HOMOMORPHIC

- **Functionality**:
  ensures the **confidentiality and privacy** of data while allowing for **secure computations** on the encrypted data**, maintaining the privacy of sensitive information**.
- **Operations**:
  supports various operations, including **addition**, **multiplication**, and more, **directly on the encrypted data**. The results obtained after computation on the encrypted data remain encrypted.
- **Advantages**:
  allows for secure data processing in scenarios where privacy is critical, such as cloud computing, machine learning on sensitive data, or outsourced data analysis.

HOMOMORPHIC ENCRYPTION

PROPERTIES

No Trusted 3rd Parties

Data remains secure and private in untrusted environments
The data stays encrypted at all times, which minimizes the likelihood that sensitive information ever gets compromised.

QUANTUM SAFE

Fully homomorphic encryption schemes are resilient against quantum attacks

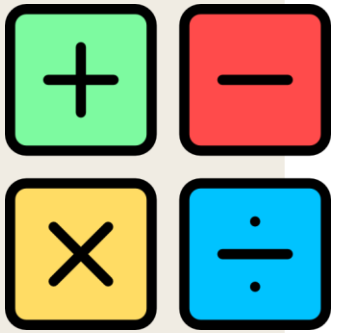
Tradeoff between data usability and privacy

There is no need to mask or drop any features in order to preserve the privacy of data.
All features may be used in an analysis, without compromising privacy.

A Very Simple Application

operation(plain)

Public key: (23,143)
Private key: (47,143)



Private What is the area? Width:7 height:3

Public

2 × 126 = 252

Enc(width) = width^{e} mod N
Enc(width) = 7^{23} mod 143
Enc(width) = 2

Enc(height) = height^{e} mod N
Enc(height) = 3^{23} mod 143
Enc(height) = 126

area = cipher^{d} mod N
area = 252^{47} mod 143
area = 21
7 × 3 = 21

Types of Homomorphic Encryption

Partially Homomorphic

- When you can only perform certain mathematical operations on the ciphertext but not others
  - **RSA cryptosystem**: partially homomorphic with respect to **multiplication**
    - **[a] x [b] OK**
    - **[a] + [b] Not Ok**
  - **Caesar Cipher**: partially homomorphic with respect to **addition**
    - **[a] + [b] OK**
    - **[a] x [b] Not OK**
  - **Paillier:** partially homomorphic with respect to **addition**
    - **[a] + [b] OK**
    - **[a] x b Ok (Encrypted x Plain)**
    - **[a] x [b] Not OK**

Fully Homomorphic

- When you can perform mathematical operations on the ciphertext

- **(Encrypted with Encrypted)**
  [a] + [b] OK
  [a] – [b] Ok
  [a] x [b] OK
- **(Encrypted with Plain)**
  [a] + b OK
  [a] – b Ok
  [a] x b OK

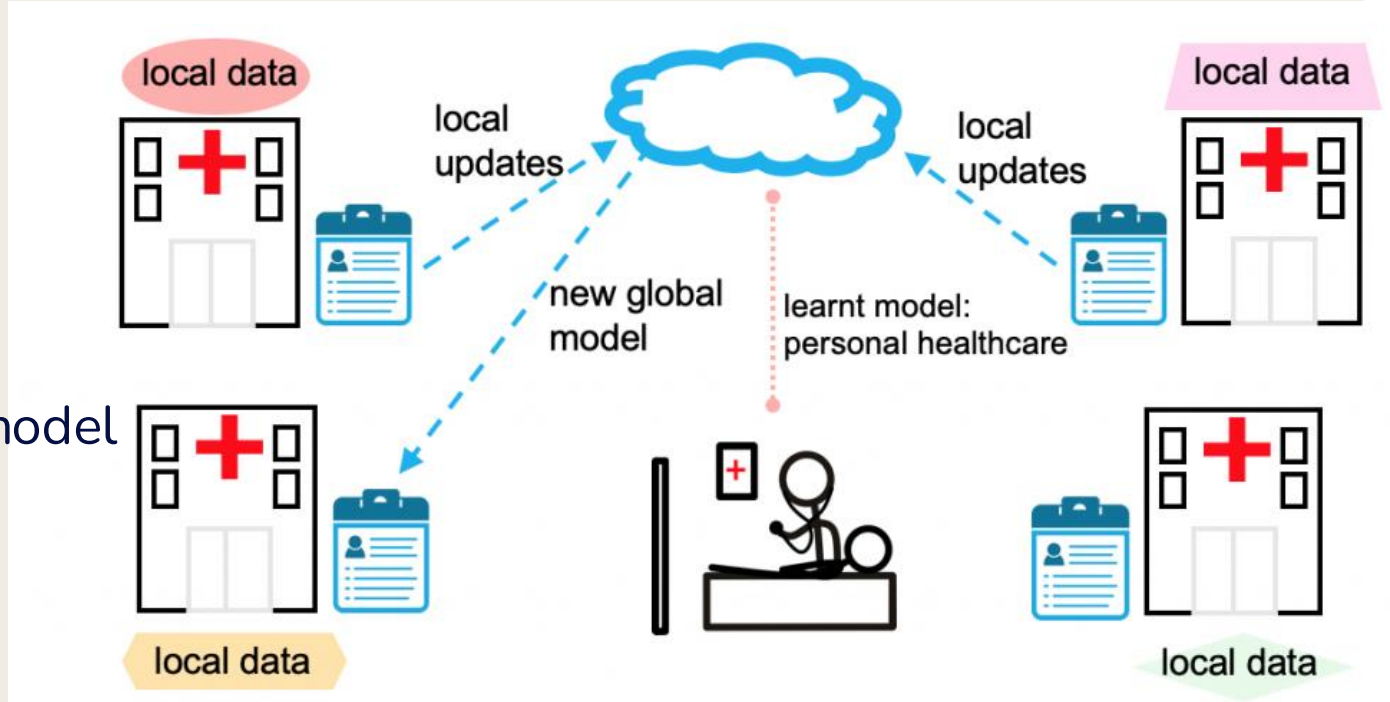
Somewhat Homomorphic

Federated Learning

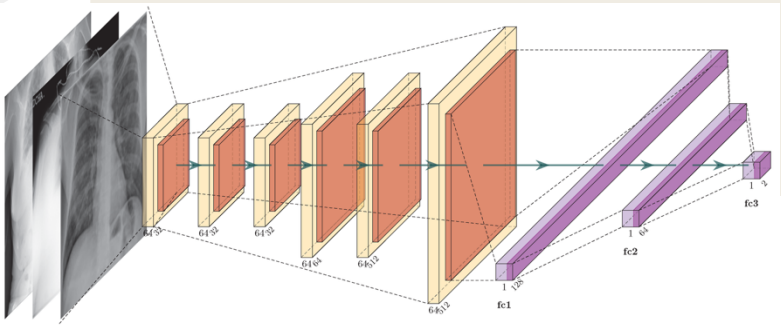
Federated learning is a distributed approach to ML that allows training models without centralizing data.
In traditional ML, data is collected from various sources and sent to a central server for model training.
In FL, the training **process takes place locally** on edge devices or servers where the data resides.
This **eliminates the need to transfer sensitive/private data** to a central server, addressing privacy.

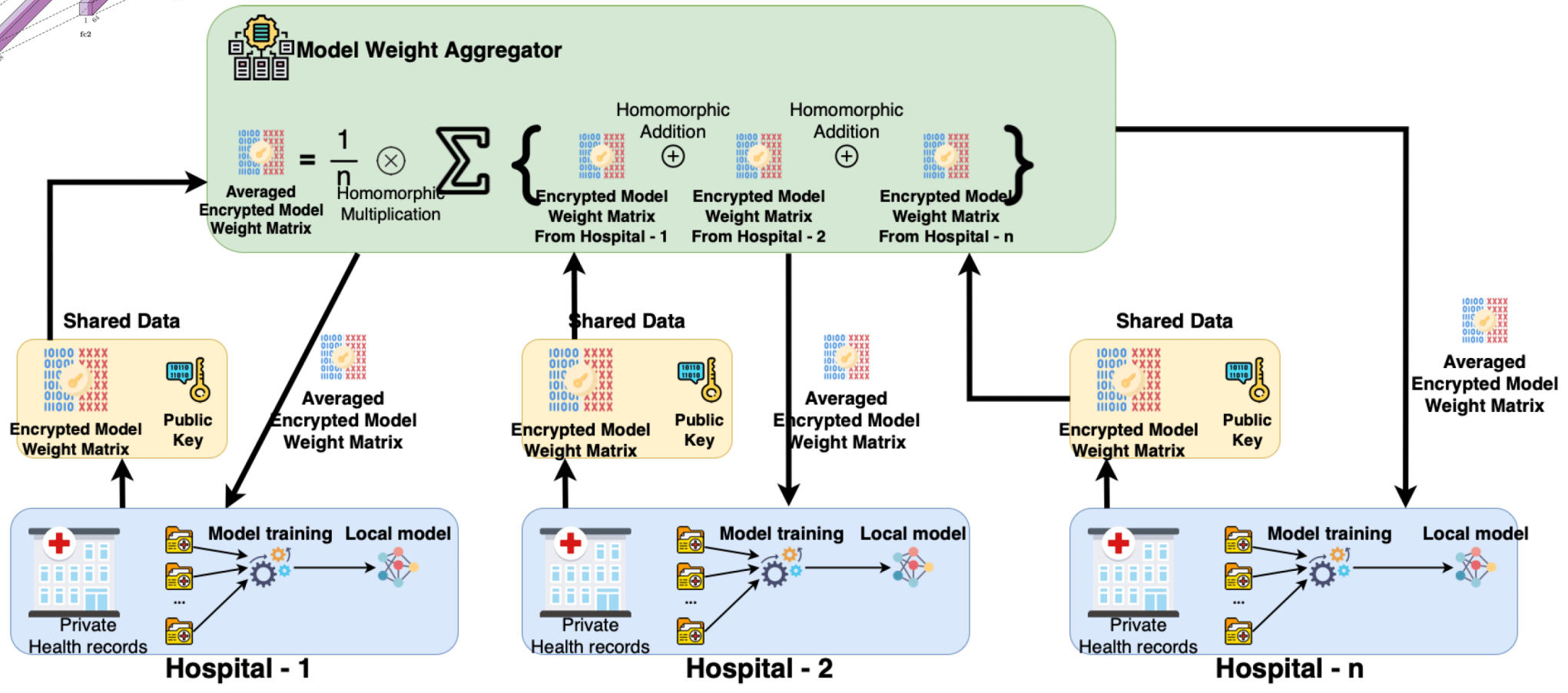
Benefits:

- **Decentralized Training**
- **Collaborative Model Updates**
- **Combine With Other Techniques**
- **Efficient Resource Utilization**: reduces bandwidth and computational requirements, as data processing and model training occur locally.



Homomorphic Encryption & Federated Larning





04.

Robustness

Uncertainty in AI

- **Uncertainty**: could potentially lead to unreliable (e.g., unsafe) behaviors of Apps,
  - if such uncertainty is not properly dealt with.

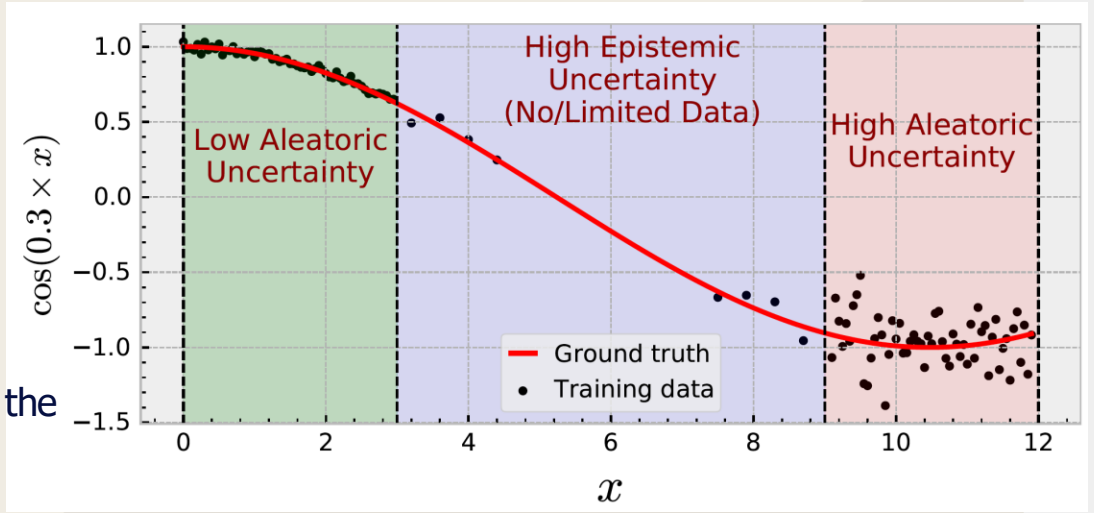
- **DNNs**: black box models (multilayered nonlinear structures)
  - non-transparent
  - Predictions not identifiable by humans

- **Apps**: black-box DL models have been used to make critical predictions

Uncertainty Sources

Epistemic Uncertainty

- Uncertainty in models (not in data)
- "Epistemic": Greek "episteme" = knowledge
- **Reducible: more data helps**
- There are two types Epistemic uncertainty
  - Model Uncertainty
    - Neural network model's neuron weights are not optimized well for the domain
  - Approximation uncertainty
    - Model structure (# of layers, activation functions (ReLU, Tanh, Sigmoid etc), optimizer functions (SGD, RMSProp, Adam, Adamax etc)



Aleatoric Uncertainty

- because of the noise input dataset.
- "**Aleatoric**": Latin "aleator" = dice players
- Noise in the training data
- **Stochastic**, **irreducible** in data (noise)
  - **More data doesn't help**
- For instance; noise sensor readings for a CPS application

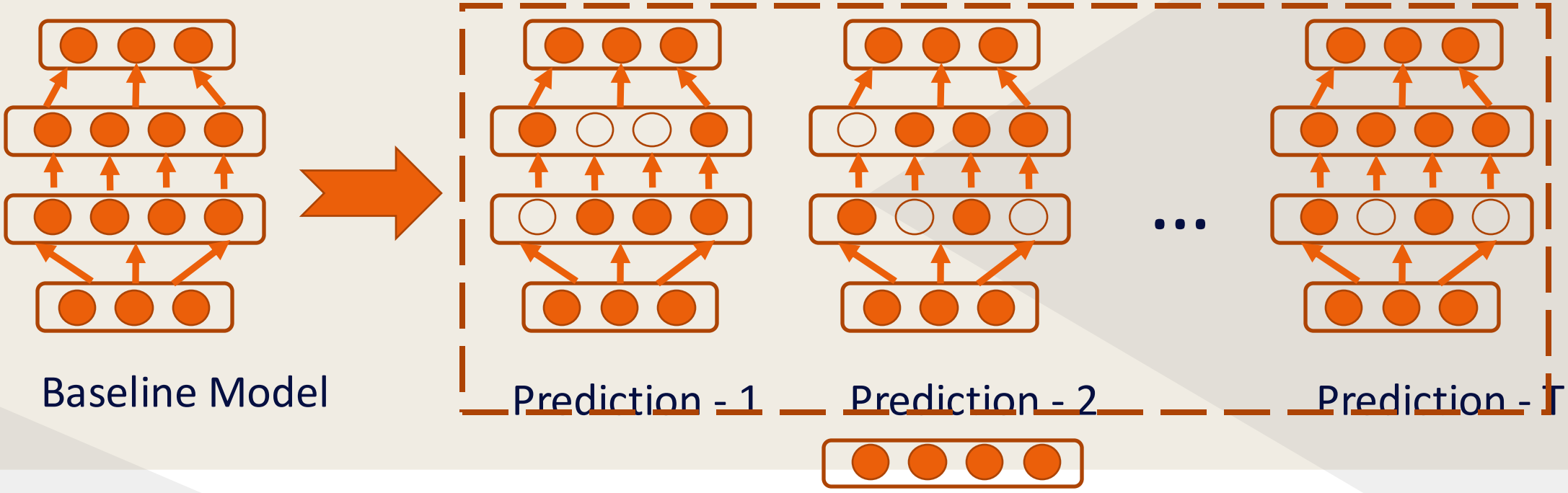
MC dropout Predictions

Important parameters:

- **T**: the number of predictions
- **p**: dropout ratio

$$\begin{bmatrix} y_{11} & y_{21} & y_{31} & y_{41} & \vdots & y_{T1} \\ y_{12} & y_{22} & y_{32} & y_{42} & \vdots & y_{T2} \\ y_{13} & y_{23} & y_{33} & y_{43} & \vdots & y_{T3} \end{bmatrix}$$

Softmax output vectors for each prediction (i.e. 3 classes, T predictions)



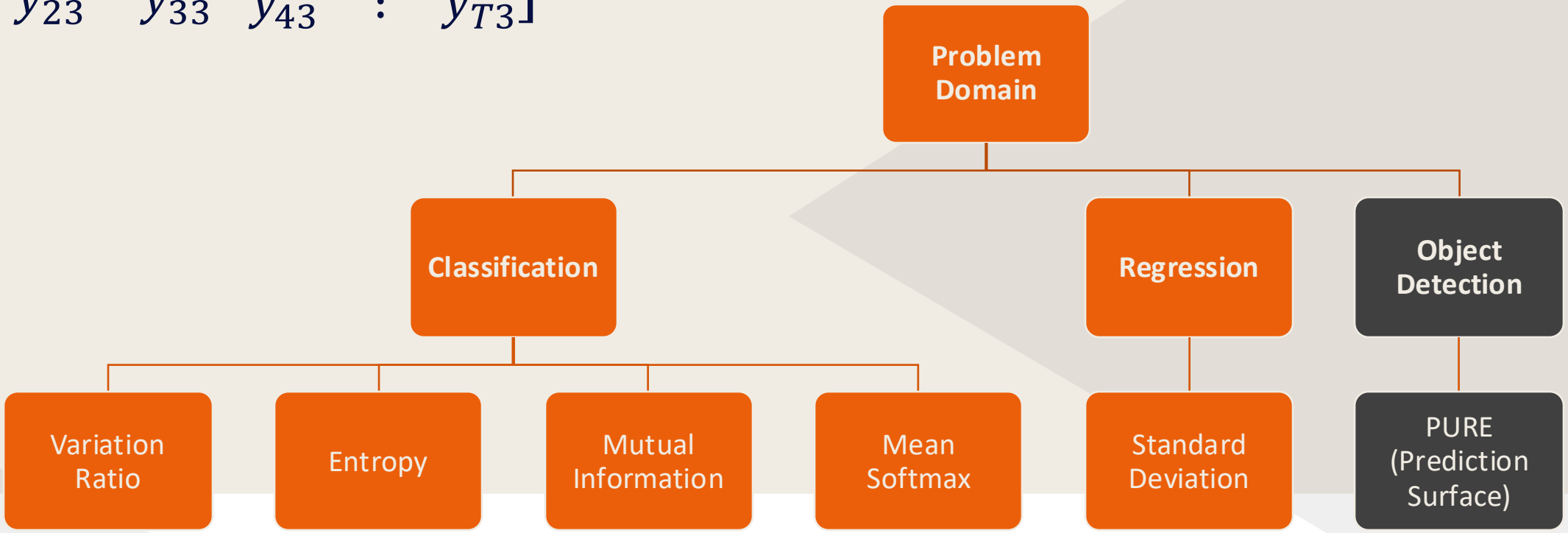
Uncertainty Quantification Metrics

Classification output

$$\begin{bmatrix} y_{11} & y_{21} & y_{31} & y_{41} & \vdots & y_{T1} \\ y_{12} & y_{22} & y_{32} & y_{42} & \vdots & y_{T2} \\ y_{13} & y_{23} & y_{33} & y_{43} & \vdots & y_{T3} \end{bmatrix}$$

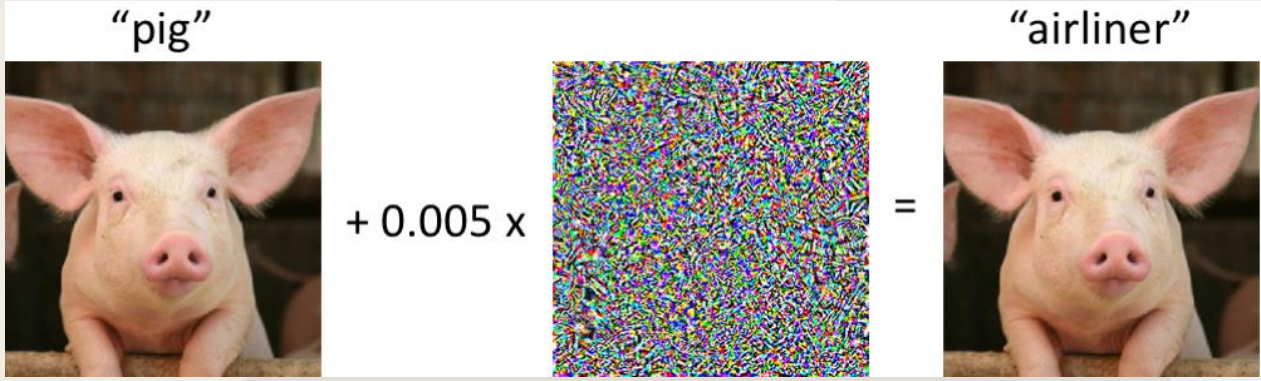
Regression output

$$[y_1 \quad y_2 \quad y_3 \quad \dots \quad y_T]$$



Adversarial Machine Learning

Adv. ML is not GAN. They are completely different.
DNN models are highly vulnerable for the craftly designed malicious inputs.
DNN models contain many vulnerabilities and weaknesses which make them difficult to defend in the context of adversarial machine learning.
For instance, they are often sensitive to small changes in the input data, resulting in unexpected results in the model's final output.
how an adversary would exploit such a vulnerability and manipulate the model through the use of carefully crafted perturbation applied to the input data.



Adv ML Attacks

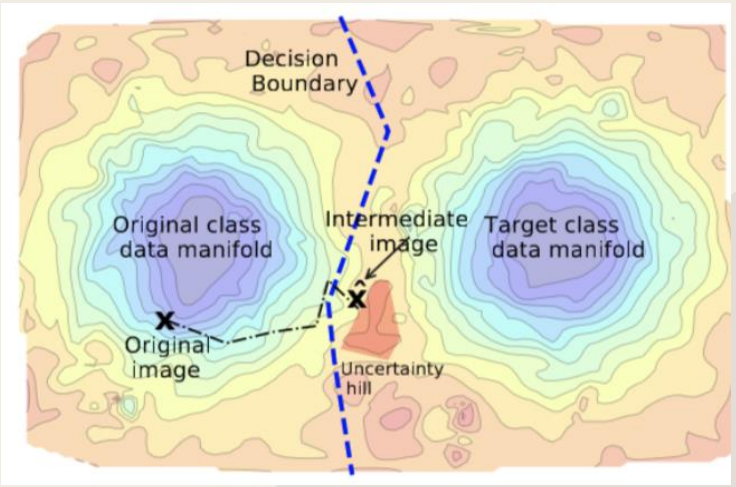
- Fast-Gradient Sign Method
- Iterative Gradient Sign Method
- Projected Gradient Descent
- Jacobian-Based Saliency Map
- Carlini & Wagner

Attack Strategy

The key concept of adversarial examples is to fool a DNN model's internal decision boundaries.

Training: DNN learns decision boundaries, which determine its behavior.

- represented by the network's parameters.
- **Loss Surface**
- If an attacker finds out where the decision boundaries are, they can change the input examples in such a way that "pushes" them over the boundary

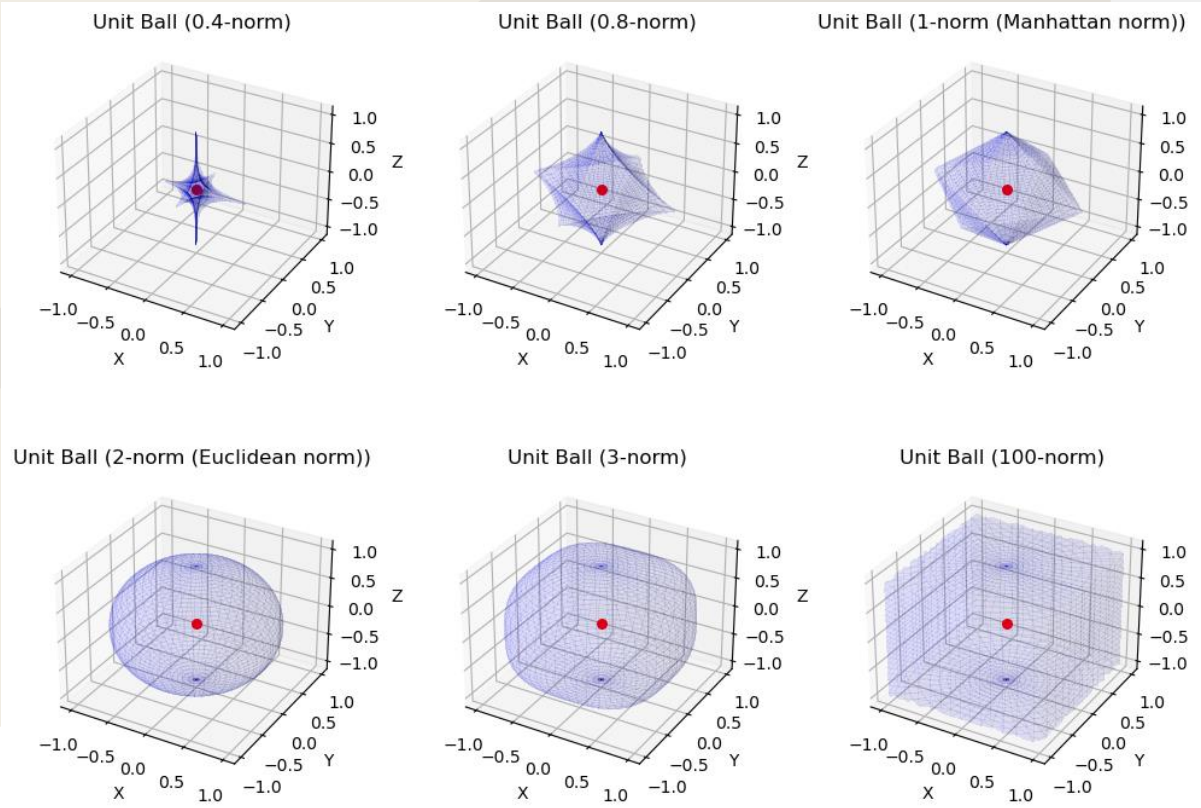


2.3 Lp Balls

In mathematics, the Lp ball, denoted as $B_p(r)$, is a set of points in a vector space that are Lp norm measures the magnitude of a vector and is defined as:

$$\|x\|_p = \left(\sum_{i=1}^{n} |x_i|^p \right)^{\frac{1}{p}}$$

where $x = (x_1, x_2, \ldots, x_n)$ is a vector of n elements, and p is a positive real number.



A Simple Attack

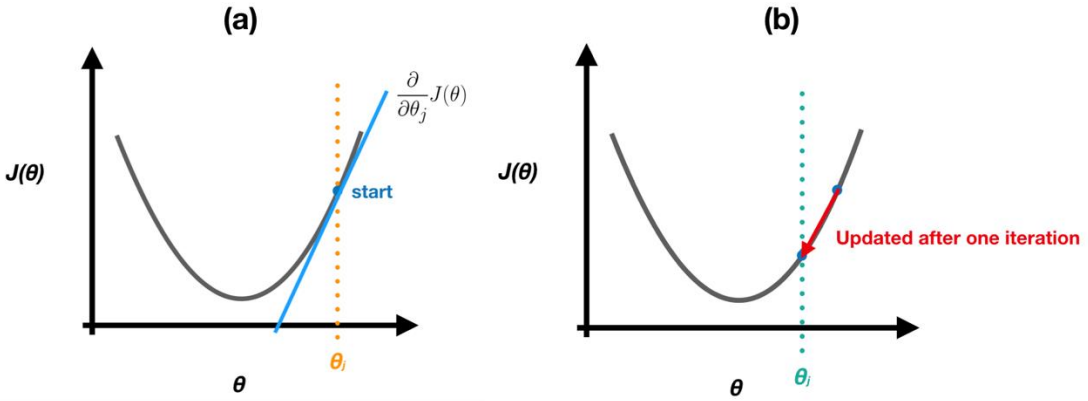
Generating adversarial examples with the Fast Gradient Sign Method [1]

Traditional DNN learning (Gradient Descent)

Repeat until converge {

$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta)$$

} where j represents the feature index number.



$$\theta = \theta - n * \nabla_\theta \mathcal{L}(\theta, x, y)$$

$$x' = x + \epsilon * sign(\nabla_x \mathcal{L}(\theta, x, y))$$

Attacker's aim to manipulate

- DNN parameters should stay the same
- Attack can change the input itself, **x**
- the goal is now to increase the model error.
- the difference to the true class should be large.

```
# Generate adversarial examples using FGSM
epsilon = 0.1
adv_x_test = fast_gradient_method(model, x_test, epsilon, np.inf)
```

Defensive Distillation

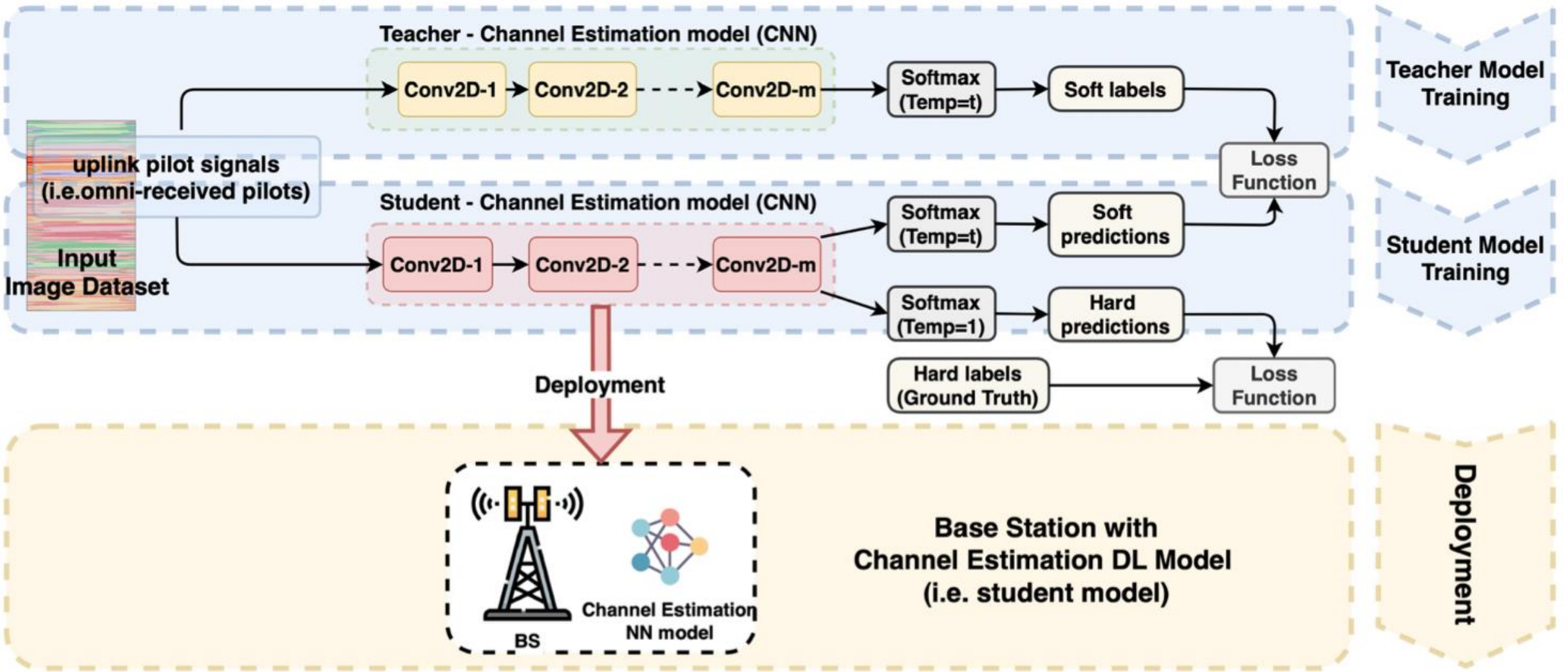


FIGURE 2. Overview of the system architecture with knowledge distillation.