EE P 595A: TinyML

Lecture 9: Responsible AI in TinyML and Summary of Topics Covered

Dept. of Electrical and Computer Engineering

University of Washington

Instructor: Dinuka Sahabandu (sdinuka@uw.edu)

ELECTRICAL & COMPUTER ENGINEERING

UNIVERSITY of WASHINGTON

Topics Covered

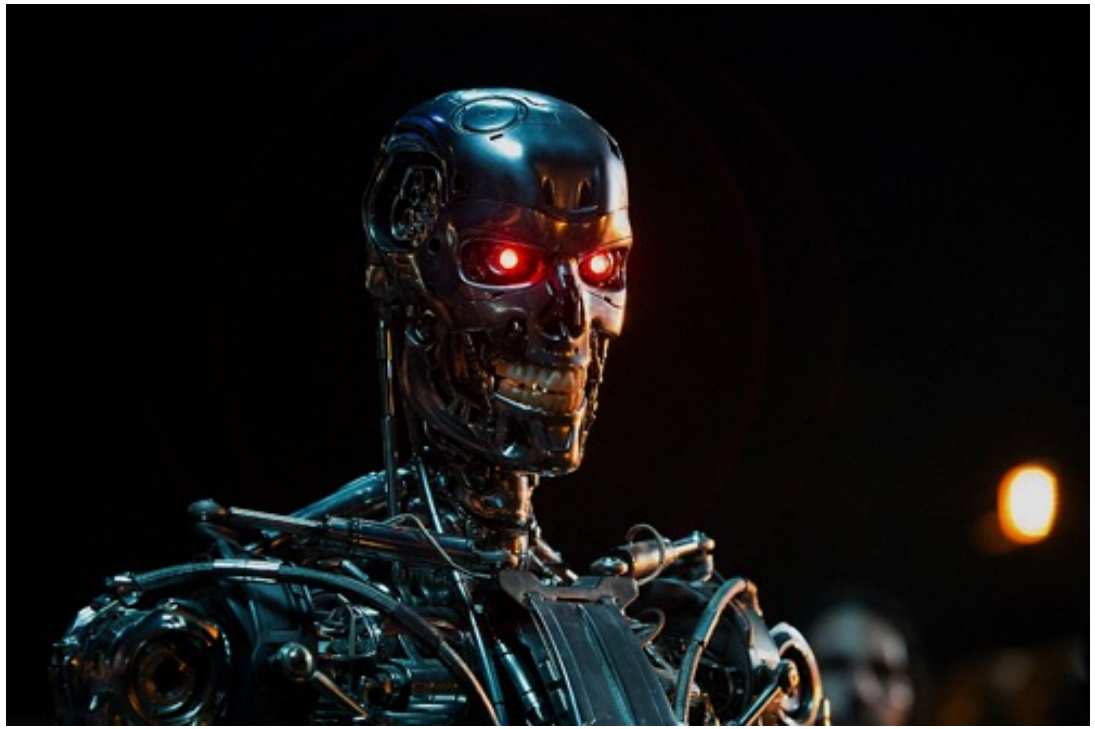
- Introduction to Responsible AI
- Responsible AI in TinyML
- Privacy and Responsible AI
- Summary of Topics Covered
- The Future of TinyML

Problems of AI

AI offers great benefits, but at the same time it introduces new risks and ethical challenges

Artificial intelligence can...

- Change current practices
- Influence human decisions
- Regulate human behavior



Problems of AI - Examples

Microsoft's disastrous Tay experiment shows the hidden dangers of AI

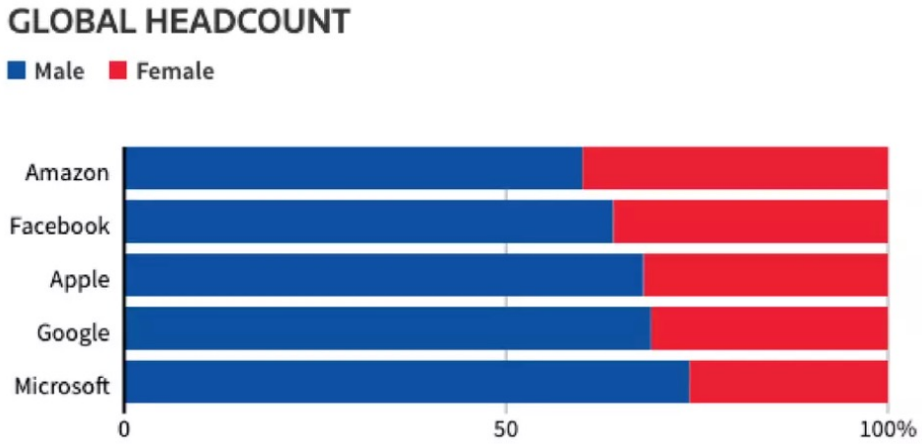
Google Calls Hidden Microphone in Its Nest Home Security Devices an 'Error'

Predictive policing algorithms are racist. They need to be dismantled.

Problems of AI - Examples

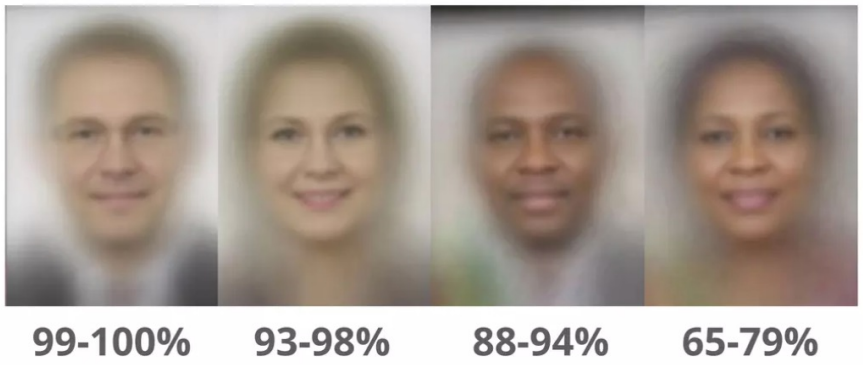
Recruiting Tools

Amazon recruiting tool shut down for bias against women after it codified discriminatory practices due to narrow data sets.



Recognition AI

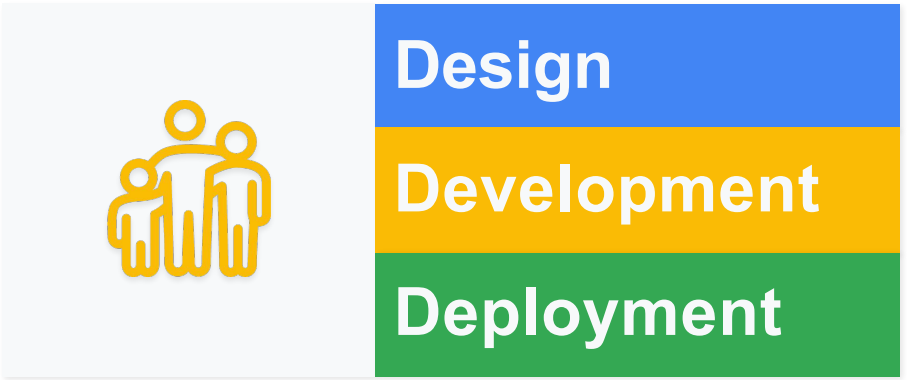
Calls for regulation on use of facial recognition after consistently higher error rates for darker-skinned and female faces.



What can we do?

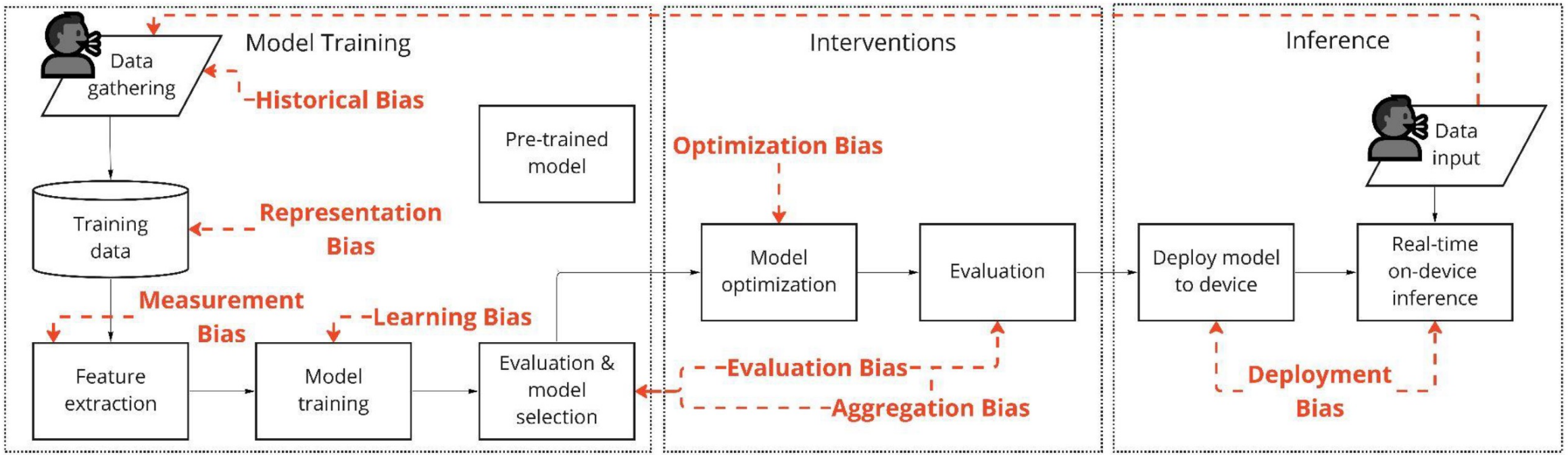
Responsible AI

- As creators of artificial intelligence systems, we have a duty to guide the development and application of AI in ways that fit our social values
- There is a growing trend towards creating **"human-centered" AI**
- Ethics is no longer just **reactive** but **proactive**

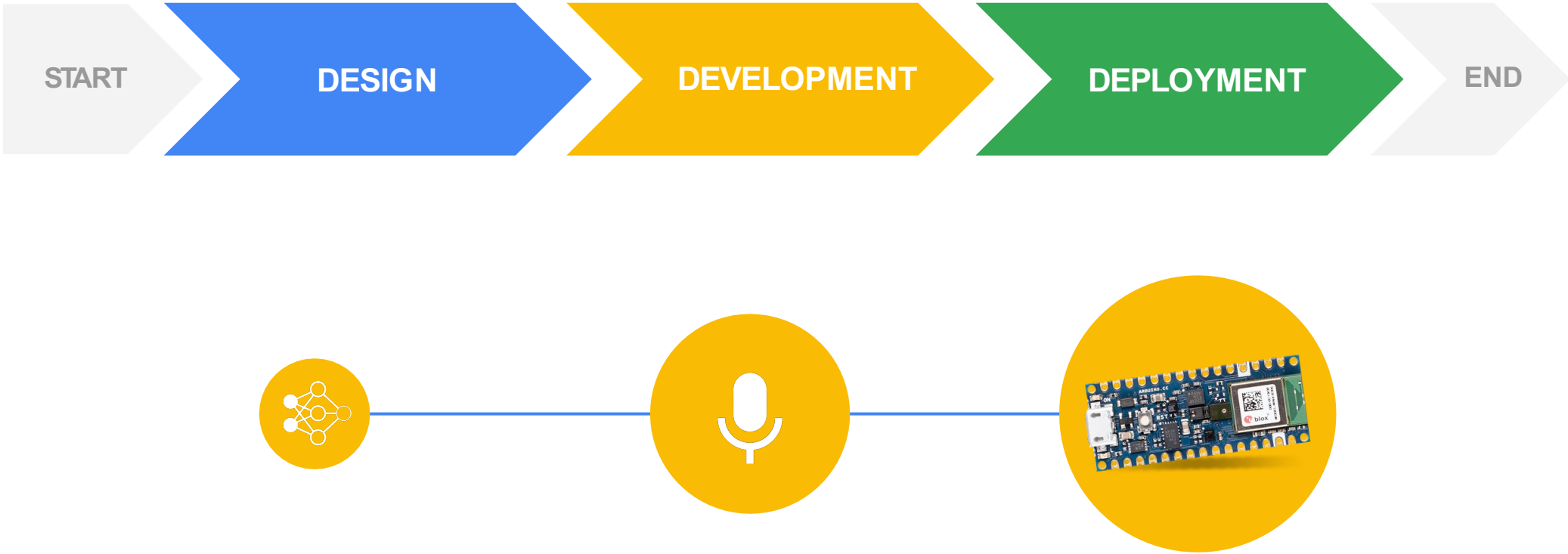


- This means keeping human values in the loop throughout all stages of a product's lifecycle

Bias in TinyML



Responsible AI – Human Centering Design



Responsible AI – Human Centering Design

- **What** am I building?
- **Who** am I building this for?
- What are the **consequences** for the user if it ***fails***?

- **What data** will be collected to train the model?
- Is the dataset **biased**?
- How can we **ensure** the model is **fair**?

- How will **model drift** be monitored?
- How should **security breaches** be addressed?
- How should the user's **privacy** be protected?

Biased Dataset

Garbage in, Garbage out!

Example of Biased Dataset

1. Define the target variable

Using biometric sensors for a health wearable device, how should you define “healthy”?

- Heart Rate
- Blood Pressure
- Number of Steps

Example of Biased Dataset

2. Label the data

Horse

Human

Human

Labels applied to the training data must serve as ground truth !

Example of Biased Dataset

3. Prejudice Reflected in Data



Dataset: 65% of people cooking are women

Algorithm predicts: 85% of people cooking are women

Solutions from Industry

- Google Research Initiative to collect data and refine speech recognition algorithms to work better for individuals with speech impairments

Solutions from Industry

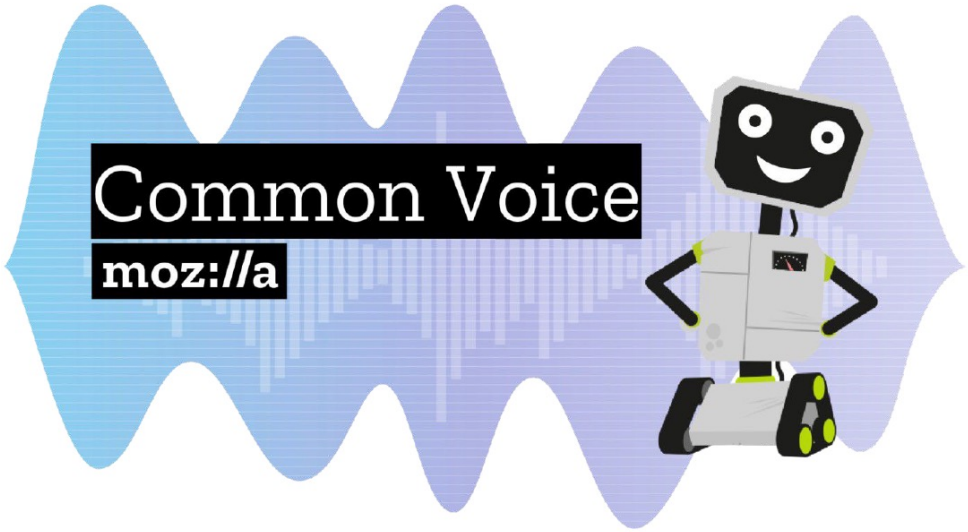
- Common Voice Dataset

- **Accent:**

23% United States English, 8% England English, 5% India and South Asia, 4% Australian English, 3% Canadian English, 2% Scottish English, 1% Irish English, 1% Southern African, 1% New Zealand English

- **Age**

23% 19–29, 14% 30–39, 10% 40–49, 6% < 19, 4% 50–59, 4% 60–69, 1% 70–79



Privacy and Responsible AI

Fit Leaking

Citizen Lab Research on Fitness Tracker Privacy

Image showing pattern-of-life activity in a suspected Russian military installation on an airbase in Syria. Image: Strava.

Why is privacy valuable?

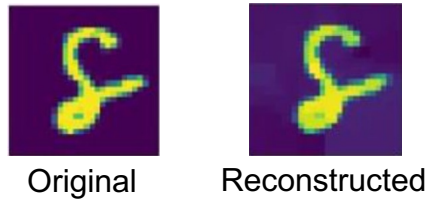
- **Prevent information-based harms**
  - Minimize opportunities for hackers to gain inappropriate access to data
- **Prevent informational injustice and discrimination**
  - Consider the context, the type of information, and who has access
- **Preserve autonomy and human dignity**
  - Obtain informed consent

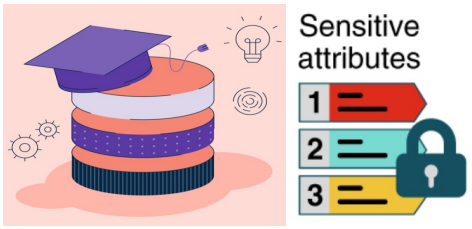
How can privacy be preserved?

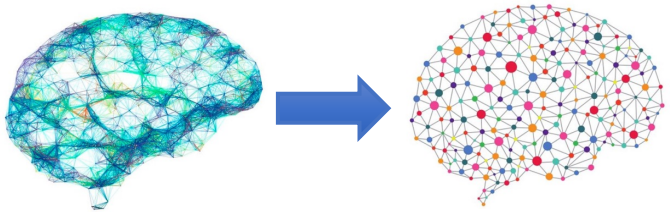
- **Minimize**
  - Avoid collecting unnecessary data, and dispose or delete data periodically
- **Protect**
  - Use encryption techniques to protect data
- **Map the flow of Information**
  - Context, the type of information, and who has access
- **Information Consent**
  - Be transparent with users about how their data is being collected and used

Privacy Attacks on Deep Learning

- Recreate one or more training samples
- Extract dataset properties which were not explicitly encoded as features
- Create a substitute model that learns the same task
- Check whether an input sample was used as part of the training set

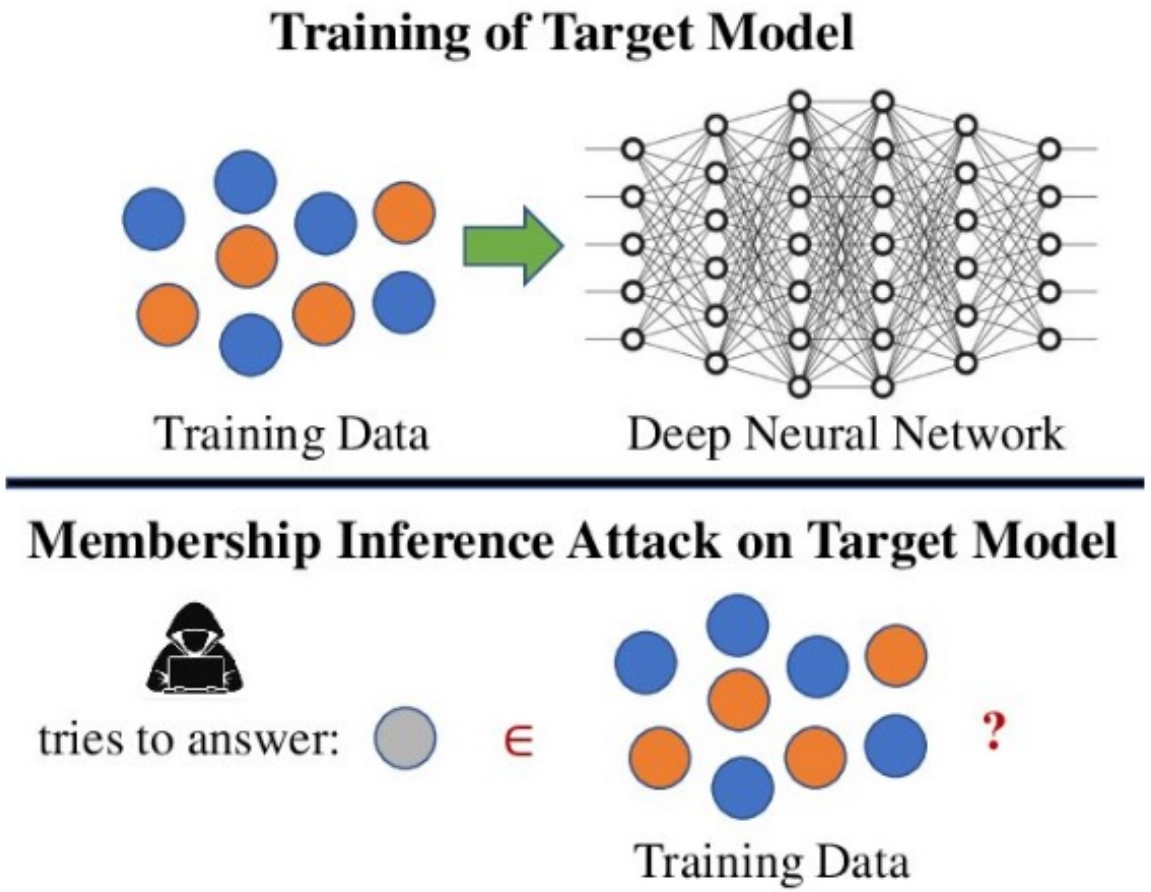






Membership Inference Attack (MIA)

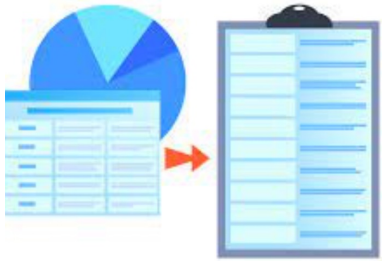
- A widely studied privacy attack in machine learning
- Allow an adversary to determine whether a specific data point was part of the training set

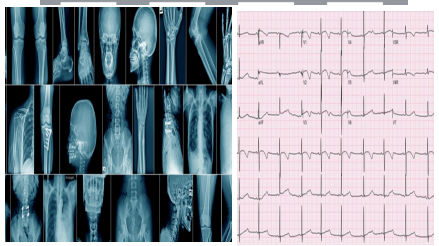


- **Member:** Input sample belongs to training set
- **Nonmember:** Input sample does not belong to training set

Examples of MIAs

- Identifying the hospital where medical inputs (e.g., X-rays, ECGs) were taken
- Recognizing the citizenship of a person in an image
- Finding out the name of the company that input financial data belongs to

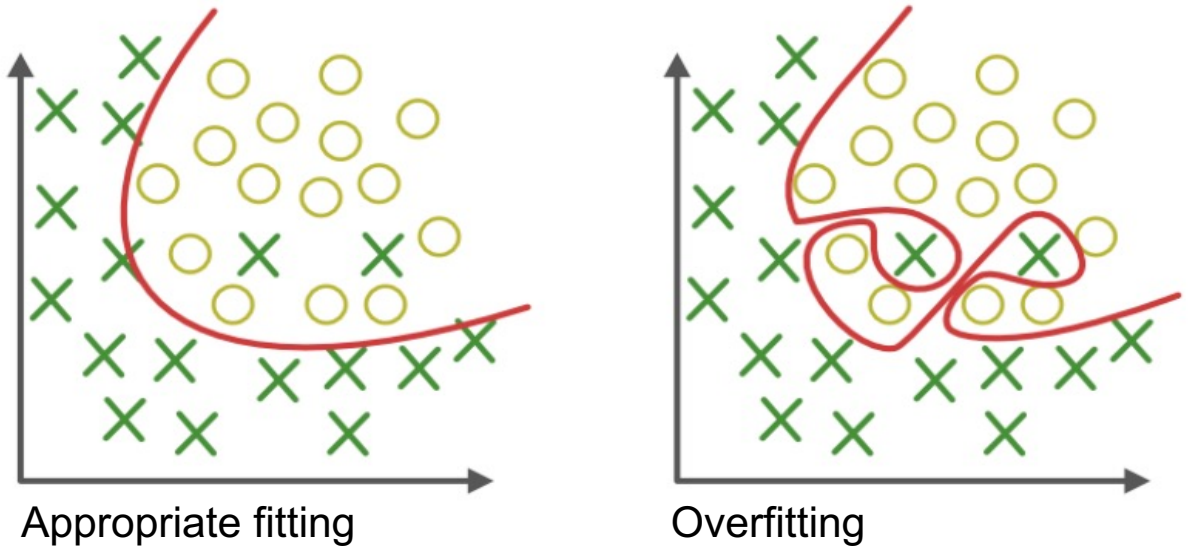




Insight of MIAs

Overfitting is the reason for the feasibility of mounting MIAs

- DNN models perform extremely well on member samples by memorizing patterns in the data → Higher confidence scores
- DNN models do not demonstrate the same performance on nonmember samples → Lower confidence scores



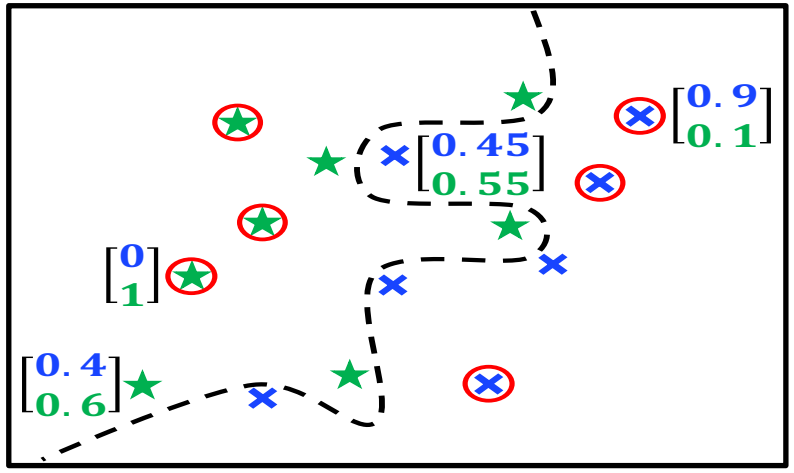
Confidence-based MIAs

- Confidence scores are used by adversary to distinguish between members and nonmembers (e.g., GAP attack)
- Attack Success Rate (ASR) of GAP attack:

$$ASR_{GAP} = 0.5 + \frac{Acc_{mem} - Acc_{nonmem}}{2}$$

- Acc_{mem}: Classification accuracy of members
- Acc_{nonmem}: Classification accuracy of nonmembers

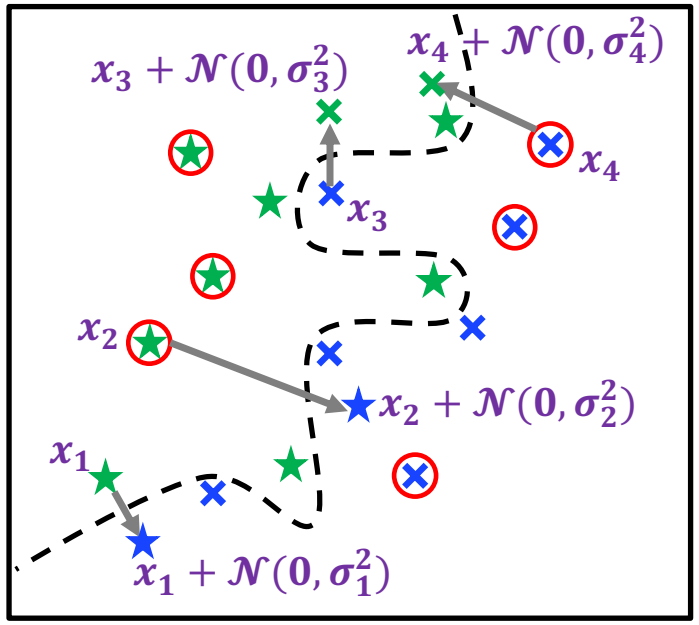
- Confidence score masking defenses (e.g., MemGuard) hide true confidence values to thwart confidence-based MIAs, but they are vulnerable to label-based MIAs



- **Member** samples are circled in **red**
- $\begin{bmatrix}\#\\\#\end{bmatrix}$: Confidence vector

Label-based MIAs

- Adversary perturbs a data sample with additive noise until the newly generated sample is misclassified
- It then estimates the distance of the sample to the decision boundary based on the magnitude of the added noise
- Data samples farther away from the boundary (high confidence samples) are identified as the members



- **Member** samples are circled in **red**
- $\mathcal{N}(0, \sigma^2)$: Zero-mean Gaussian noise samples with variance σ^2
- $\sigma_1^2, \sigma_3^2 < \sigma_2^2, \sigma_4^2$

Defenses Against LAB MIAs

- **Differential Privacy (DP)** [Chen et al., 2020], [Truex et al., 2019]**:** perturb model weights during training

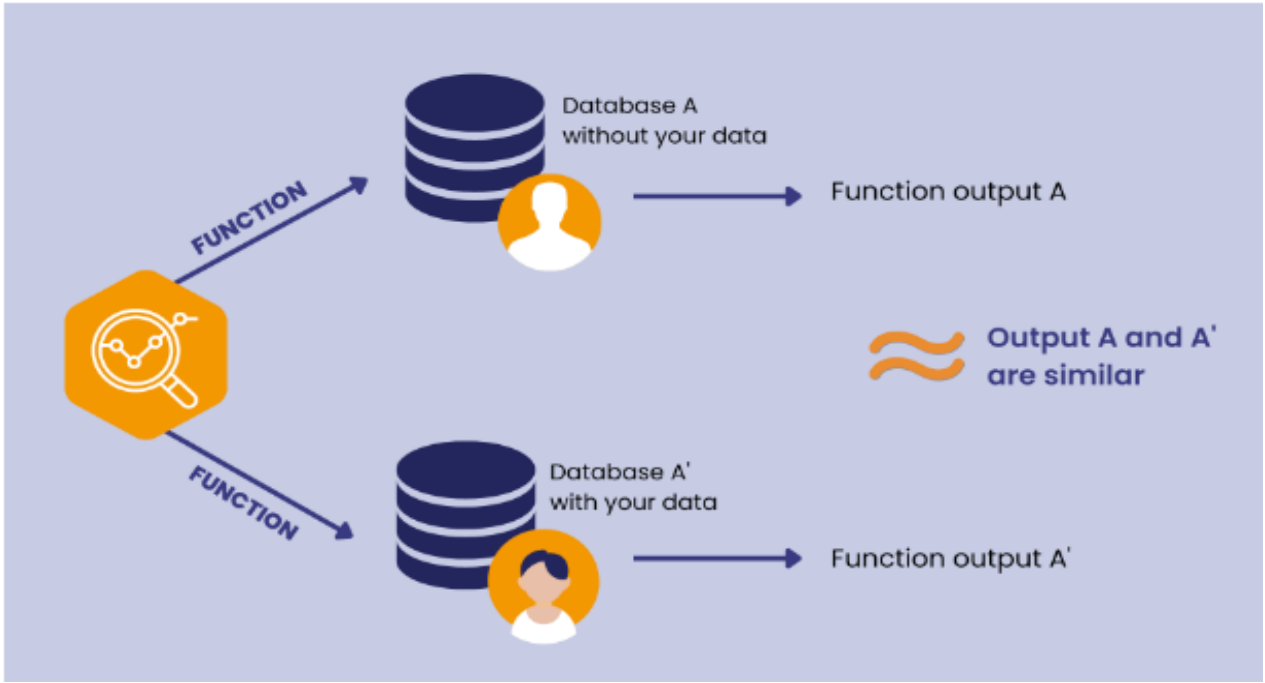
- **L1 and L2 Regularization** [Ying et al., 2020]**:** limits the values of the model weights during the training

- **Dropout** [Salem et al., 2019]**:** disregarding certain nodes in a layer at random during training

Introduction to Differential Privacy

- Aims to provide insights about the dataset without revealing information about any individual data point

- ML algorithm is considered differentially private if the probability of any outcome occurring is nearly the same for any two datasets that differ in only one record



Formal Definition of Differential Privacy

- An algorithm A is (ϵ, δ)-differentially private if for all datasets D_1 and D_2 differing by one element, and for all possible outputs S of the algorithm:

$$\Pr[A(D_1) = S] \leq e^{\epsilon} \cdot \Pr[A(D_2) = S] + \delta$$

- Parameters:
  - ϵ: Privacy loss parameter or privacy budget. Lower ϵ means stronger privacy
  - δ: Probability of the algorithm not being ϵ-differentially private

Introduction to DP-SGD Algorithm

- Stochastic Gradient Descent (SGD):
  - A popular optimization method for training machine learning models
  - Iteratively updates model parameters to minimize the loss function

- DP-SGD: A variant of SGD that ensures differential privacy by incorporating noise and clipping gradients

Mechanism of DP-SGD Algorithm

- **Noise Addition:** Adds Gaussian noise to gradients during each update to obscure individual data contributions
- **Gradient Clipping:** Limits the influence of any single data point by clipping gradients to a maximum norm C
- **Algorithm Steps:**
    - Initialize model parameters (e.g., model weights)
    - For each batch
        1. Compute per-sample gradients
        2. Clip the gradient by norm C
        3. Add Gaussian noise
        4. Update model parameters with the noisy gradients

Benefits of of DP-SGD Algorithm

- **Privacy Protection:** Safeguards against membership inference attacks by adding noise to the training process.
- **Scalability:** Applicable to large datasets and deep learning models.
- **Compatibility:** Integrates with existing ML frameworks (e.g., TensorFlow Privacy).

Challenges and Trade-Offs of DP-SGD Algorithm

- **Privacy vs. Utility:** Higer privacy (lower ϵ) often leads to reduced model accuracy
- **Hyperparameter Tuning:** Selecting appropriate values for noise scale and clipping norm is complex and critical for balancing privacy and utility
- **Computational Overhead:** Adding noise and clipping gradients can increase computational requirements

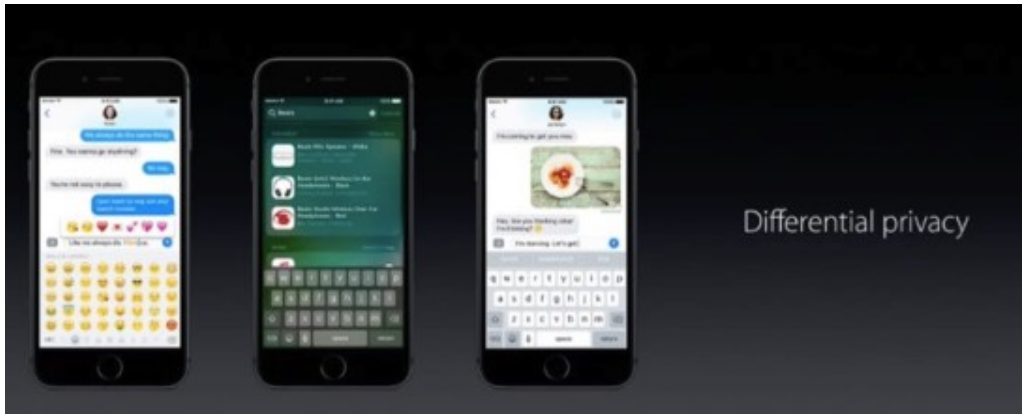
Real-World Applications of DP-SGD Algorithm

- **Google's RAPPOR:** Uses differential privacy to collect aggregate data from users while preserving individual privacy

google/**rappor**

RAPPOR: Privacy-Preserving Reporting Algorithms

- **Apple's Differential Privacy Implementation:** Utilizes differential privacy to collect data for improving user experience without compromising individual privacy



Defenses Against LAB MIAs Recap

- **Differential Privacy (DP)** [Chen et al., 2020], [Truex et al., 2019]**:**
  perturb model weights during training

- **L1 and L2 Regularization** [Ying et al., 2020]**:**
  limits the values of the model weights during the training

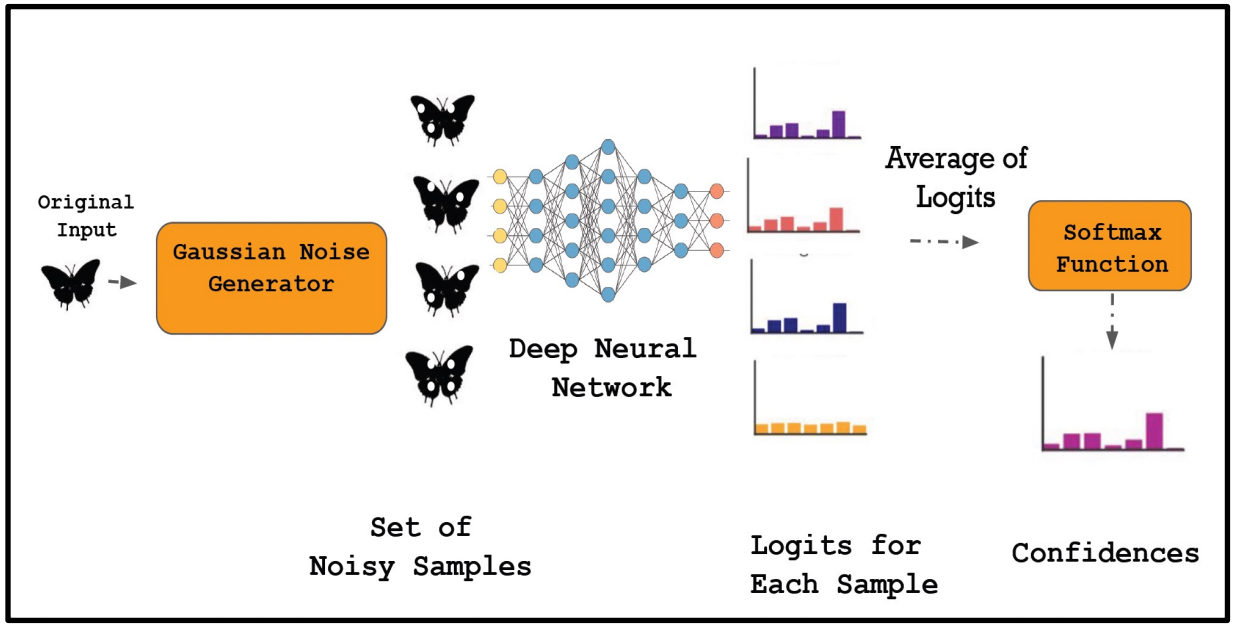
- **Dropout** [Salem et al., 2019]**:**
  disregarding certain nodes in a layer at random during training

❖ However, these defenses require re-training the DNNs

❓ *Can we defend against LAB MIAs without having to re-train the DNNs?*

LDL: A Defense Against MIAs at Inference Time

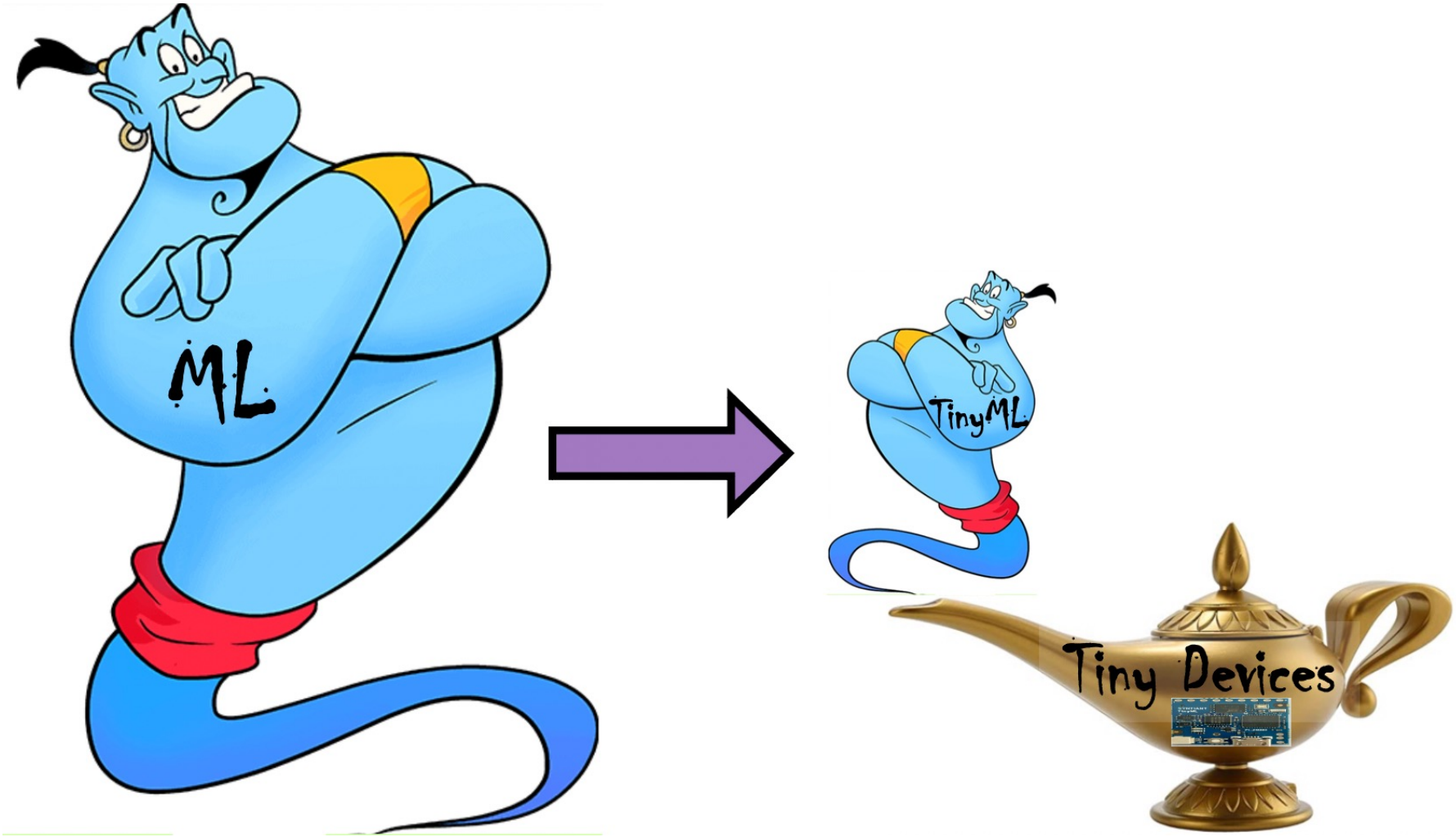
- **Insight**: ensure that magnitudes of noise required for misclassification of members and nonmembers are comparable
- Uses randomization at inference time and constructs a high dimensional sphere of label invariant around the samples



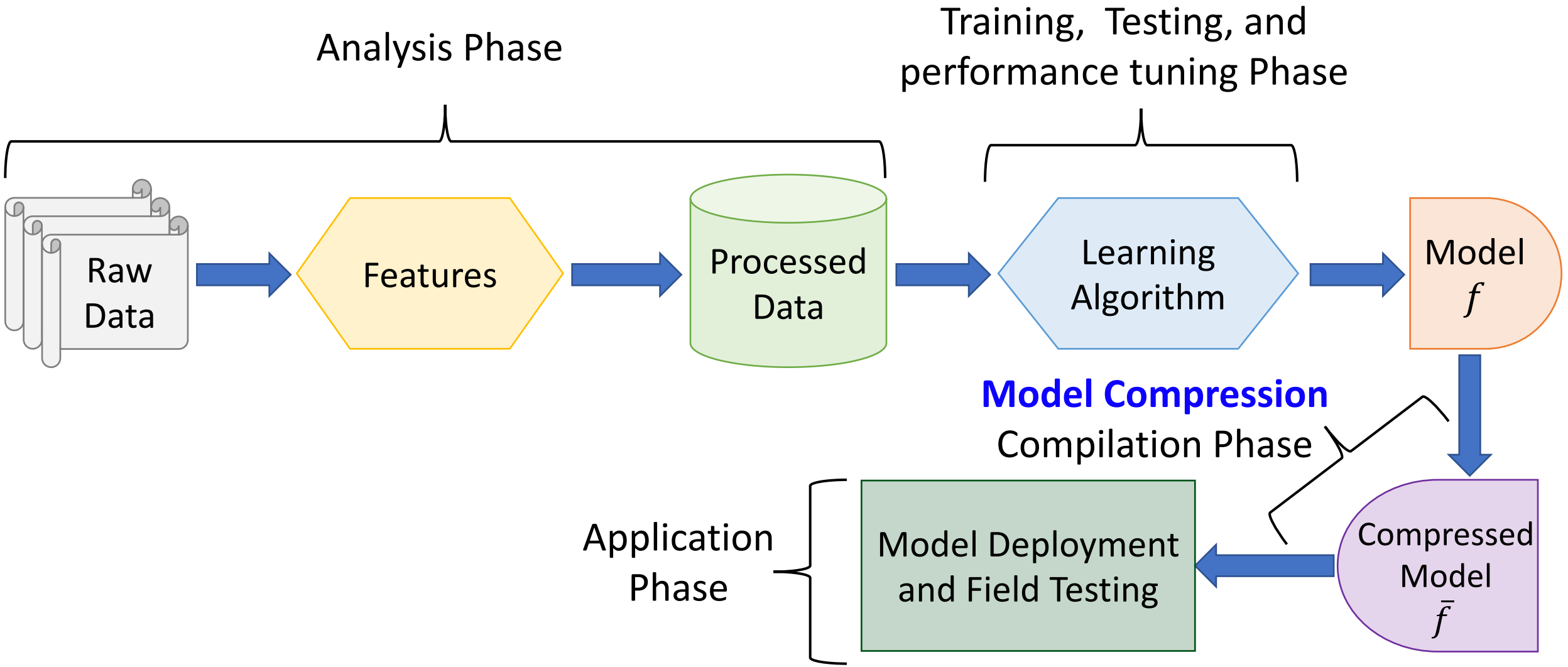
Rajabi, Arezoo, Dinuka Sahabandu, Luyao Niu, Bhaskar Ramasubramanian, and Radha Poovendran. "LDL: A Defense for Label-Based Membership Inference Attacks." In *Proceedings of the 2023 ACM Asia Conference on Computer and Communications Security*, pp. 95-108. 2023.

5 min Break

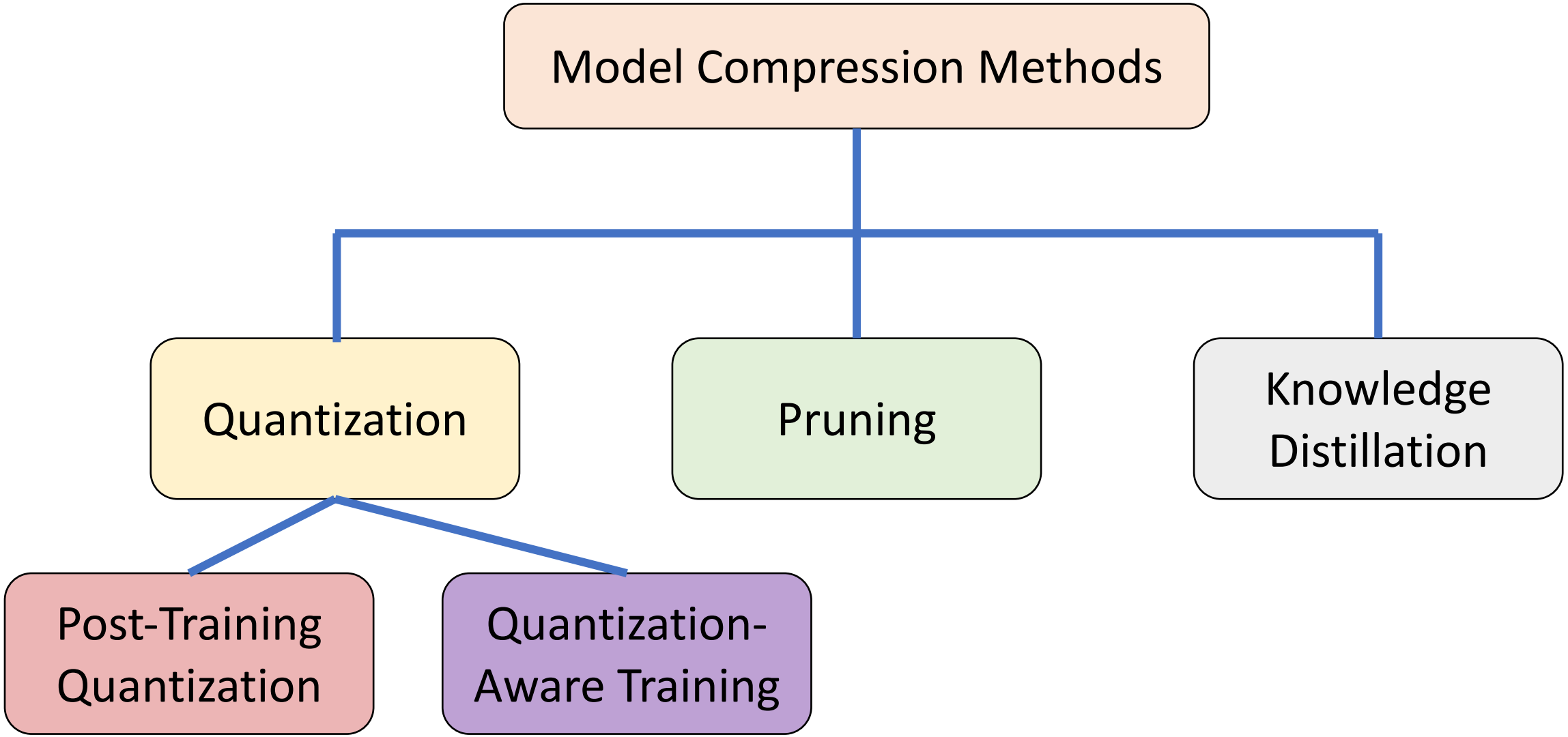
Review of TinyML



Review of TinyML



Review of TinyML – Model Compression Methods



Review of TinyML – Theory in Quantization

Theory in Quantization

K-means clustering-based Quantization

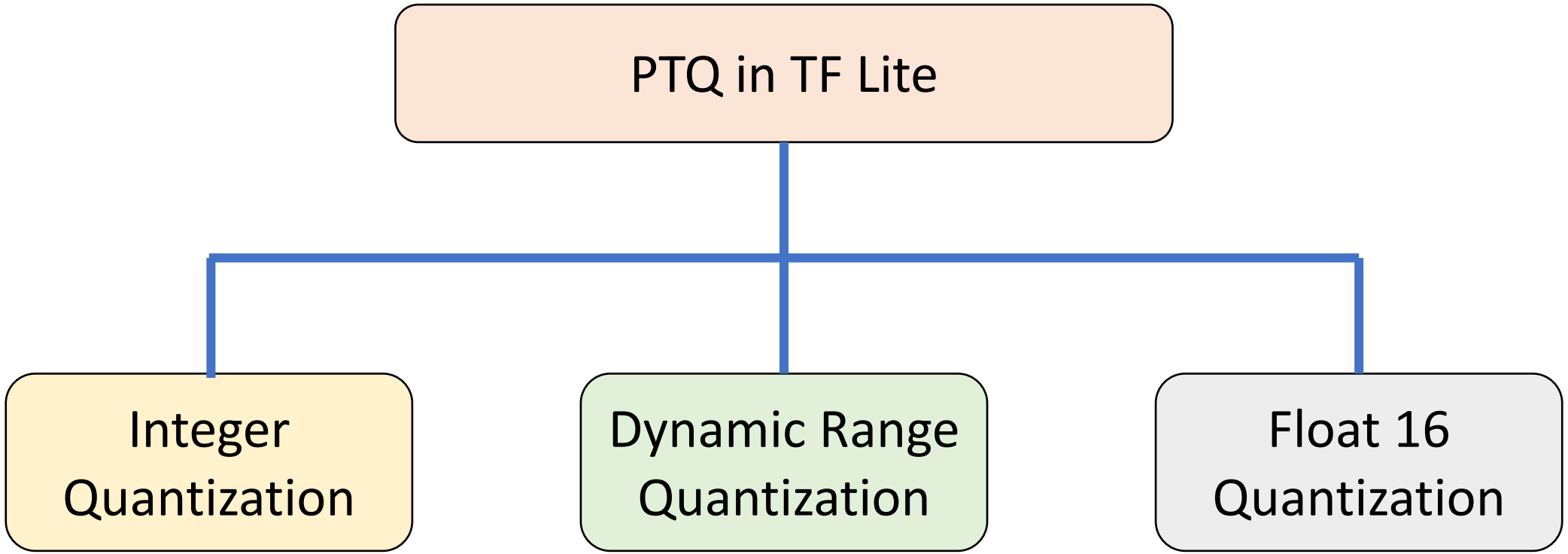
- Relatively Low Quantization Error
- Slow Implementation

Linear Quantization

- Relatively High Quantization Error
- Fast Implementation

TF Lite implements this

Review of TinyML – Post Training Quantization (PTQ)

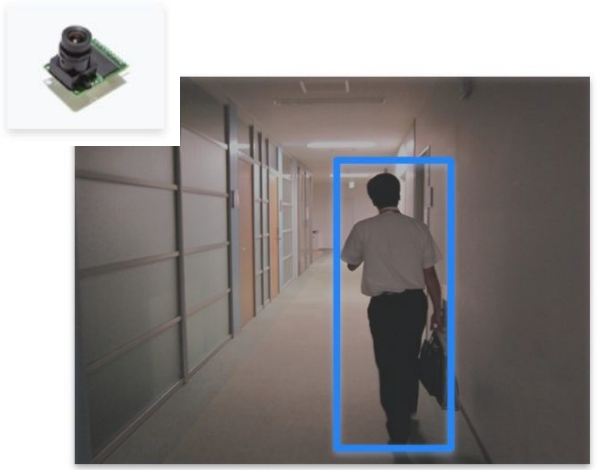


Review of TinyML - Applications

Sound

Keyword Spotting

Vision



Visual Wake Words
Mask Detection

Sensor

Anomaly Detection
Magic Wand,
Motion Classification
American Sign Language

EEP 595 Course Summary

Main learning objectives

- ✓Develop TinyML models to solve real-word problems
- ✓Implement TinyML applications & explanation of needed ML algorithms
- ✓Use TensorFlow for deep learning and TensorFlow Lite (TFLite) for TinyML
- ✓Using C language for deploying TinyML on Embedded Systems
- ✓Measure the performance of the deployed TinyML models

Each week we

1. Focused on a specific TinyML applications
2. Showed how to use TinyML to address the problem
3. Provided hands on experience in the labs

Course Summary: Topics Covered

- **Week 1: Introduction to TinyML**
  - TinyML Landscape, Applications and Challenges
  - TinyML Lifecycle and Workflow
  - Model Compression Techniques
  - Recap on Necessary ML Background: ML Algorithms, Neural Networks
  - Introduction to Hardware and Software Used in the Course

- **Week 2: Fundamentals of ML and TinyML**
  - Pruning ML models
  - Quantization Aware Training (QAT) and Post Training Quantization (PTQ)
  - Knowledge Distillation
  - Tiny Deep Learning
  - TensorFlow Lite (TFLite) for TinyML

Course Summary: Topics Covered

- **Week 3: TinyML for Keyword Spotting**
  - Background on Keyword Spotting and Streaming Audio
  - Challenges and Constraints in Keyword Spotting
  - Keyword Spotting Architecture and Data Collection
  - Model Training, Evaluation Metrics, and Deployment

- **Week 4: TinyML for Visual Wake Words**
  - Introduction to Visual Wake Words and Its Challenges
  - Visual Wake Words Dataset
  - MobileNets
  - Transfer Learning for Visual Wake Words
  - Model Training, Evaluation Metrics, and Deployment

Course Summary: Topics Covered

- **Week 5: TinyML for Anomaly Detection**
  - Background on Anomaly Detection and Signal Processing
  - Real and Synthetic Datasets
  - Unsupervised Learning (K-Means Clustering and Autoencorders)
  - Threshold Choice
  - Model Training, Evaluation Metrics, and Deployment

- **Week 6: Wizard Magic Wand**
  - Gesture Tracking through Bluetooth
  - CNN for Magic Wand Sketch
  - Data Collection and Labeling
  - Model Training, Evaluation Metrics, and Deployment

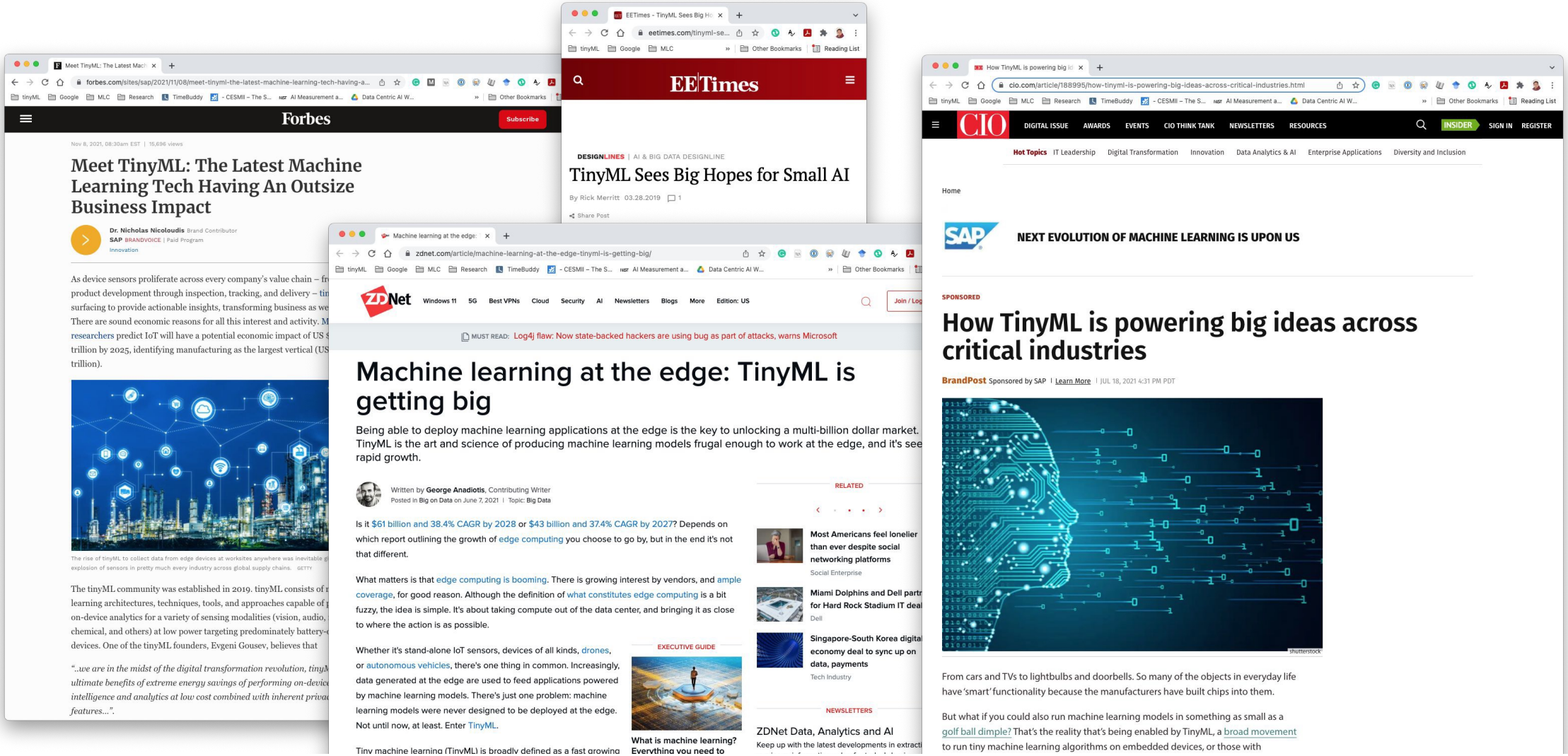
Course Summary: Topics Covered

- **Week 7: TinyML for Motion Classification**
  - Background on Motion Classification
  - Accelerometer, Gyroscope, Barometer, and Magnetometer
  - Supervised learning for motion classification
  - Sensor interface for Mechanical Stress in Transport
  - Model Training, Evaluation Metrics, and Deployment
- **Week 8: TinyML for American Sign Language (ASL) Interpretation**
  - Background on ASL and ASL Interpretation
  - Gesture Motion Datasets and Features
  - Analyzing Gesture Motion Data using Neural Networks
  - TinyML Framework for ASL Interpretation
  - Model Training, Evaluation Metrics, and Deployment

Course Summary: Topics Covered

- **Week 9: Responsible AI in TinyML and**
  - Introduction to Responsible AI
  - Responsible AI in TinyML
  - Privacy and Responsible AI
  - Summary of Topics Covered
- **Week 10: Final Project Presentations on Wednesday, May 29th**
  - **Each Group has 15 minutes (Suggested presentation – 12 minutes; Q&A— 3 mins)**
  - **Signup for the presentation order (Will provide in Slack #project channel)**
- **Week 11: Additional time for the final project Report**
  - **Final report due on June 7th 11:59pm, 2024**

The future of the TinyML

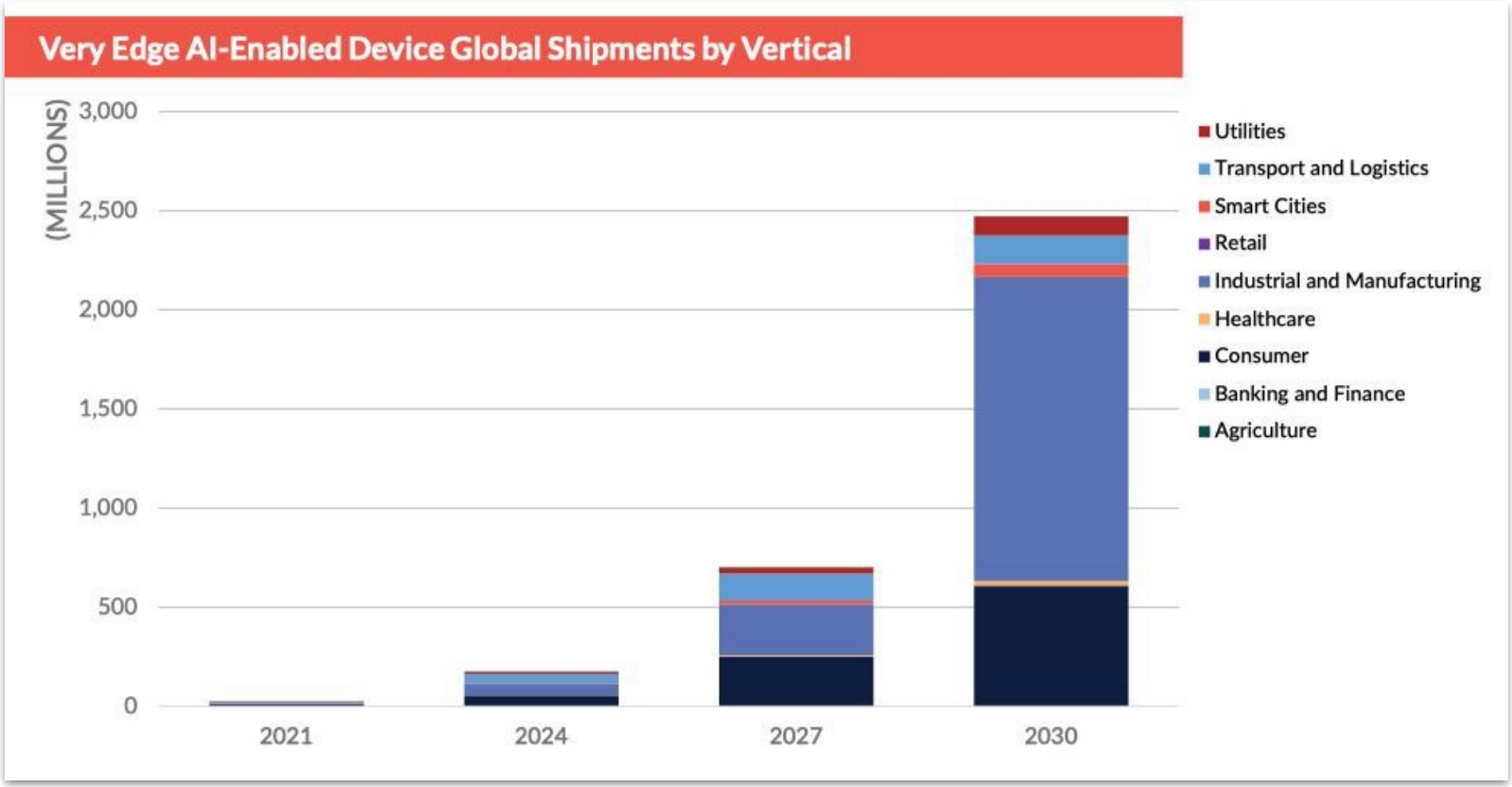


The future of the TinyML

As device sensors proliferate across every company's value chain – from new product development through inspection, tracking, and delivery – tinyML is surfacing to provide actionable insights, transforming business as we know it. There are sound economic reasons for all this interest and activity. McKinsey researchers predict IoT will have a potential economic impact of US $4-11 trillion by 2025, identifying manufacturing as the largest vertical (US $1.2-3.7 trillion).

Source: Forbes

The future of the TinyML



Source: ABI Research

5 min Break

Today's Lab

- Both Software and Hardware Lab!
- Training models using TensorFlow-Privacy
- Quantize the model for TinyML
- Deploy the model to the Arduino board

Lab 9 – Software - Road Map

1. Data Processing
2. Model training using TensorFlow-Privacy
3. Model Evaluation
4. Covert the TensorFlow model to TensorFlow Lite Micro model
5. Compress the model

Lab 9 - Hardware - Road Map

1. Connect Arduino Boards to your computer

2. Implement compressed ML model

3. Deploy the model on to board

4. Open Serial Monitor and test the model

Training the Model with Differential Privacy

Open EEP595-TinyML-Lab9.ipynb in Google Colab

Let’s train our model!

Source: https://towardsdatascience.com/k-means-a-complete-introduction