B. Tech (ECE)

Project-I (BECE497J) **Review-2**

FEDERATED LEARNING AND XAI FOR SEMICONDUCTOR FAULT DETECTION

Team Members Details

AFFAN AHMAD(22BEC0644)

DEVASHISH NARAYAN(22BEC0178)

ATUL KUMAR(22BEC0610)

Project Guide:

Dr. Sumathi G,

Professor Sr. Grade 1,

School of Electronics Engineering

School of Electronics Engineering, Vellore Institute of Technology.

OUTLINE:

- Design Objectives
- Software Details
- Code Block 1
- Code Block 2
- Code Block 3
- Code Block 4
- Code Block 5
- Result
- References

Design objectives:

1. Privacy-Preserving Model Training
Implement **Federated Learning** so wafer defect data remains at each client location and is never shared directly.

2. Lightweight yet Accurate Model
Use **MobileNetV2** to ensure low computational cost while maintaining high classification accuracy.

3. Integration of Explainability
Apply **Grad-CAM** to visualize important image regions influencing the model's predictions.

4. XAI-Guided Model Improvement
Use explainability feedback to **reweight training** and **improve focus on defect regions**.

Software Details:

We are using **Python** as the primary programming language along with **Google Colab** for free GPU-based model training and federated learning simulation. **TensorFlow Federated (TFF)** is employed to implement a simple 3–4 client FL setup, while **TensorFlow 2.x/Keras** is used to build and train a lightweight CNN model such as MobileNetV2. Data handling and preprocessing are done using **NumPy** and **Pandas**, with **Grad-CAM** integrated for generating explainability heatmaps as part of the training feedback loop.

Block 1 - Enhanced Data Preprocessing for Federated Learning

Preprocessing Wafer Data for Federated Learning

- **Function:** preprocess_wafer_npz_federated
- **Goal:** Load wafer .npz dataset, preprocess images, split into Train/Val/Test, and create **federated client datasets** (IID & Non-IID).

Steps:

- **Load & Map Data** – Load images/labels, convert one-hot → class IDs, analyze class distribution.
- **Split Dataset** – Stratified split into Train/Val/Test with balanced classes.
- **Image Preprocessing** – Normalize [0–2] → [0–1], resize (224×224), convert grayscale → RGB, adjust contrast.
- **Federated Client Creation** –
  - **IID:** Equal random distribution.
  - **Non-IID:** Each client gets dominant “primary classes” + few secondary classes (simulating fab specialization).

Outputs:

- clients_data → datasets for each client
- val_ds, test_ds → centralized validation & testing
- num_classes, train_ids, client_class_distributions

Block 2 - Enhanced MobileNetV2 Model for Federated Learning

Federated MobileNetV2 Model (with XAI-ready design)

- **Purpose:** Build a **MobileNetV2-based CNN** optimized for **federated learning** with strong augmentation & explainability support.

Pipeline:

- **Input & Augmentation:** Random rotation, flips, contrast, brightness for better generalization
- **Base Model**: MobileNetV2 (no pretrained weights, include_top=False), trainable backbone
- **Feature Processing**: Global Average Pooling → compact features; BatchNorm & Dropout to reduce overfitting
- **Dense Layers**: Dense(512) → BN → Dropout; Dense(256) → Dropout for non-linear decision-making
- **Output**: Dense(num_classes, softmax) for multiclass wafer classification
- **Compilation**: Adam (LR=0.001); loss: categorical_crossentropy; metrics: accuracy, top-3 accuracy, precision, recall

Benefit: Lightweight, robust, and federated/XAI-compatible CNN for wafer defect detection.

Block 3 - XAI Analyzer with Grad-CAM

Purpose: Explain wafer defect classification with Grad-CAM.

Features:

- **Setup**: Loads model, applies ReplaceToLinear(), configures Grad-CAM (tf-keras-vis).
- **Heatmaps**: Uses last conv layer (out_relu in MobileNetV2); fixes class index handling; visualizes model focus.
- **Analysis**: Samples client data, runs predictions, applies Grad-CAM; computes Focus Ratio (max/mean activation) and Defect Coverage (defect vs edges).
- **Outputs**: Dictionary with focus_score (attention concentration) and defect_coverage (overlap with defect zones).

Benefit: Verifies the model predicts accurately and attends to wafer defects, improving trust.

Block 4 - Federated Learning Framework

Goal: Train wafer defect models using FedAvg with Grad-CAM feedback.

Core Workflow:

- **Initialization**
  - Global MobileNetV2 model + WaferXAIAnalyzer.
- **Federated Round**
  - Each client trains locally → uploads weights.
  - Metrics: loss, accuracy, focus score, defect coverage (via XAI).
  - Aggregation: **weighted averaging** by client data size.
- **Global Evaluation**
  - Update global model with aggregated weights & Validate on centralized data.
- **Full Training Loop**
  - Repeat for multiple rounds.
  - Early stop if accuracy > 90%.
  - Periodic XAI feedback to assess attention focus.

Outputs: Final global model, training history (client + global), XAI insights on model focus.

Block 5 - Main Execution Pipeline

Purpose: Run the **complete federated learning workflow** with XAI feedback for wafer defect detection.

Steps:

- **Data Prep**: Load .npz dataset, preprocess images, split into non-IID clients, validation, and test sets
- **FL Init**: Create FederatedWaferLearning, set number of clients & classes
- **Training Loop**: Run 15 federated rounds; clients train locally → FedAvg aggregation; Grad-CAM evaluates focus & defect coverage
- **Final Model**: Save global model (federated_wafer_defect_model.h5); return model, training history, and federated learner

Outcome:

- Federated global model optimized for wafer defect detection.
- XAI insights guide model attention across clients.

Result:

Round	Train	Validation
1	16.37	13.21
2	43.22	39.14
3	51.39	50.47
4	58.66	56.93
5	68.21	65.28
6	73.49	72.34
7	75.8	73.26
8	79.18	78.11
9	82.76	80.01
10	83.31	81.98

Summary & Benefits

- **Objective:** Train a robust wafer defect classification model across multiple clients while providing **explainable AI (XAI) insights**.

Pipeline Overview:

- **Data Preparation**
  - Load .npz wafer maps.
  - Preprocess: resize, normalize, augment, convert to RGB.
  - Split into **non-IID federated clients**, centralized validation, and test sets.
- **Model Architecture**
  - **MobileNetV2** backbone (trainable).
  - Dense layers + BatchNorm + Dropout.
  - Softmax output for multiclass classification.
- **Federated Learning**
  - Each client trains locally → **FedAvg aggregation**.
  - Multiple rounds with optional **XAI feedback** for attention/focus.
  - Early stopping if global accuracy > 90%.

- **XAI Integration**
  - **Grad-CAM heatmaps** to analyze model focus.
  - Metrics: focus_score, defect_coverage.
  - Guides attention and improves trust in predictions.
- **Outputs**
  - Global federated model (.h5)
  - Training history per round & per client
  - XAI-driven insights on model attention

Key Benefit:

Combines **federated learning** and **explainability**, ensuring accurate and interpretable wafer defect detection across distributed fab data.

References:

[1]Md Meftahul Ferdaus et al., “Significance of activation functions in developing an online classifier for semiconductor defect detection,” Knowledge-Based Systems, vol. 248, pp. 108818–108818, Apr. 2022, doi: https://doi.org/10.1016/j.knosys.2022.108818.

[2]F. López de la Rosa, J. L. Gómez-Sirvent, R. Morales, R. Sánchez-Reolid, and A. Fernández-Caballero, “A deep residual neural network for semiconductor defect classification in imbalanced scanning electron microscope datasets,” Applied Soft Computing, vol. 131, no. C, p. 109743, Oct. 2022, doi: https://doi.org/10.1016/j.asoc.2022.109743.

[3]Y. Yang and M. Sun, “Semiconductor Defect Pattern Classification by Self-Proliferation-and-Attention Neural Network,” IEEE Transactions on Semiconductor Manufacturing, vol. 35, no. 1, pp. 16–23, Feb. 2022, doi: https://doi.org/10.1109/tsm.2021.3131597.

[4]T. Patel, Ramalingam Murugan, Gokul Yenduri, R. H. Jhaveri, Hichem Snoussi, and T. Gaber, “Demystifying Defects: Federated Learning and Explainable AI for Semiconductor Fault Detection,” IEEE Access, vol. 12, pp. 116987–117007, Jan. 2024, doi: https://doi.org/10.1109/access.2024.3425226.

[5]Pedram Tabatabaeemoshiri, N. Kumar, Anis, D. Ting, and Vivek Regeev, “A Novel Approach to Test-induced Defect Detection in Semi-conductor Wafers, Using Graph-Based Semi-Supervised Learning (GSSL),” IEEE Access, vol. 13, pp. 1–1, Jan. 2025, doi: https://doi.org/10.1109/access.2025.3535103.

[6]S. L. Yuen et al., “GENSS: Defect Classification Method on Extremely Small Datasets for Semiconductor Manufacturing,” International Computer Science and Engineering Conference (ICSEC), pp. 419–424, Sep. 2023, doi: https://doi.org/10.1109/icsec59635.2023.10329633.

[7]F.-L. Chen and S.-F. Liu, “A neural-network approach to recognize defect spatial pattern in semiconductor fabrication,” IEEE Transactions on Semiconductor Manufacturing, vol. 13, no. 3, pp. 366–373, Aug. 2000, doi: https://doi.org/10.1109/66.857947.

[8]R. K. Vankayalapati, Z. Yasmeen, A. Bansal, V. Dileep, and N. Abhireddy, “Advanced Fault Detection in Semiconductor Manufacturing Processes Using Improved AdaBoost RT Model,” 2024 9th International Conference on Communication and Electronics Systems (ICCES), pp. 467–472, Dec. 2024, doi: https://doi.org/10.1109/icces63552.2024.10859691.

[9]B. Haddad, S. Yang, L. J. Karam, J. Ye, N. Patel, and M. Braun, “Multifeature, Sparse-Based Approach for Defects Detection and Classification in Semiconductor Units,” IEEE Transactions on Automation Science and Engineering, vol. 15, no. 1, pp. 145–159, Jan. 2018, doi: https://doi.org/10.1109/tase.2016.2594288.

Thank You