FOUNDATION MODEL FOR TIME SERIES ANALYSIS IN ALD PROCESS AND AUTOMATED SEMICONDUCTOR CHARACTERIZATION

A Dissertation
Presented to
The Academic Faculty

By

Congrui Li

In Partial Fulfillment
of the Requirements for the Degree
Master of Science in the
School of Electrical and Computer Engineering
College of Engineering

Georgia Institute of Technology

May 2025

FOUNDATION MODEL FOR TIME SERIES ANALYSIS IN ALD PROCESS AND AUTOMATED SEMICONDUCTOR CHARACTERIZATION

Thesis committee:

Dr. Asif Islam Khan
School of Electrical and Computer Engineering
Georgia Institute of Technology

Dr. Eric Vogel
School of Materials Science and Engineering & School of Electrical and Computer Engineering
Georgia Institute of Technology

Dr. Shimeng Yu
School of Electrical and Computer Engineering
Georgia Institute of Technology

Date approved: April 25, 2025

ACKNOWLEDGMENTS

I would like to sincerely thank my supervisor Professor Asif Islam Khan for his thoughtful guidance and support during my research process. He not only pointed me in meaningful directions for ALD process optimization and agent-based modeling, but also generously shared many fascinating videos and innovative papers with me. These moments truly showed me the possibility of applying artificial intelligence to FeFET manufacturing. Professor Khan not only helped me complete this paper, but also played a key role in inspiring me to pursue a doctoral degree and continue exploring research in connecting semiconductor devices and intelligent systems.

I am also deeply grateful to the fellows in our lab. As someone who didn't have much practical experience in the fabrication process at the beginning, I had many questions, and everyone was very patient and generously invested time and knowledge. Whether explaining process details or collaborating on projects, they make every interaction positive and enjoyable. I have learned a lot from them, both technically and personally, and I am truly grateful for the friendly and supportive environment they have created.

This paper is not only the result of my own work, but also a reflection of the people who shared my ideas, time, and encouragement with me. Thank you all for participating in this journey.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

This thesis focuses on the application of AI in semiconductor fabrication, including two different stages of research projects, one is the ALD process and the other is the device characterization process.

The first project developed a foundation model framework for time series analysis in atomic layer deposition (ALD) processes, specifically targeting high-K dielectrics such as HfO2 and ZrO2. This framework utilizes advanced machine learning techniques to analyze high-dimensional, heterogeneous data streams. By integrating time-series sensor data with multimodal datasets, including engineering logs and recipes, this project aims to predict material properties, improve process efficiency. The current progress is in the stage of data collection and processing.

The second project demonstrated a practical implementation through an automated semiconductor characterization system for ferroelectric devices. This work focuses on leakage detection and the validation for HZO thin flims from their electrical measurements, which transforms the traditional, manual, and time-consuming process of analyzing polarization voltage (P-V) loops into an automated machine learning pipeline. This study proposes an improved convolutional neural network architecture used for analyzing charge response data over time for devices and performs well on the metrics from the resulted confusion matrix, achieving more efficient device validation.

These projects both lay the foundation for AI driven semiconductor process optimization and provide practical experience for the effectiveness of deep learning in semiconductor fabrication.

CHAPTER 1

FOUNDATION MODEL FOR TIME SERIES ANALYSIS IN ALD PROCESS

1.1 The Problem

The purpose of this proposed study is to develop an artificial intelligence framework for predicting and optimizing material properties during atomic layer deposition (ALD) processes, with a focus on high-K dielectrics such as HfO_2 and ZrO_2.

The challenges currently faced by ALD technology include inconsistent material deposition, high energy consumption, and large greenhouse gas emissions.

This study attempts to address the key challenge by utilizing machine learning techniques, particularly foundation models, to analyze the high-dimensional, heterogeneous data streams generated during ALD processes. The ultimate goal is to find the correlations from high-dimensional process data and post-deposition film quality, which could help tasks including predicting material properties, improving process efficiency, and achieving sustainable manufacturing.

This study focuses on integrating time-series data (such as pressure sensor logs, spectral data) and multimodal datasets (such as engineering logs, recipes) into a unified framework. The proposed solution will adopt advanced pre-trained foundation models to enhance their ability to perform ALD domain-specific tasks, such as film quality prediction. The general goal of this project is consistent with the development of sustainable ALD processes.

1.2 Origin and History of the Problem

This research is part of Khan Lab's Life Cycle Sustainability Analysis (LCSA) of Atomic Layer Deposition (ALD) project, led by Principal Investigator Asif Khan.

Atomic layer deposition (ALD) is crucial in semiconductor manufacturing, providing atomic level precision for devices such as nanosheet FET and ferroelectric RAM. However, changes in process parameters, such as chamber pressure and precursor pulses, can affect the quality of the deposition and device performance. The recent analysis of pressure sensor data for $HfZrO_2$ recipe (November 2023 to May 2024) shows that the pressure peak continues to rise during later operation, indicating possible machine problems. These deviations may lead to suboptimal deposition, resulting in higher defect rates and reduced reliability. Also, the impact of ALD process on the environment is significant, reducing energy consumption and waste in ALD processes helps achieve industry goals of sustainability and net zero emissions.

One goal of the LCSA project is to address these challenges by creating an AI driven automated experimentation (AI/AE) platform that integrates multimodal physical processes, metrology, and simulation data, as shown in Figure Figure 1.1. The platform aims to co-optimize power, performance, area, and cost (PPAC) indicators while reducing the environmental footprint of ALD processes.

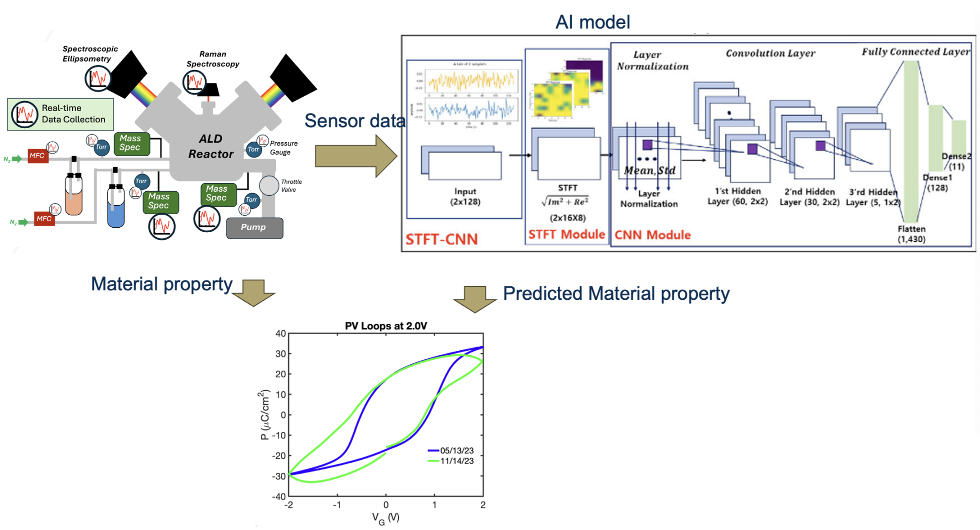

Figure 1.1: Flow Diagram

The advancement of artificial intelligence, especially the advancement of multimodal foundation models (FMs), provides an opportunity to address this issue. These models integrate various data streams for analysis and can be fine-tuned for specific tasks by pre-training on large and diverse datasets. These models use multimodal data such as text, images, and time series to provide comprehensive insights. For example, LLaVA-Gemma [1] demonstrated that the small FMs such as Gemma-2B and Gemma-7B can be efficiently adapted to multimodal situations by balancing computational efficiency and model accuracy.

For our project, we thought we could also use Foundation Models (FMs) to process data streams such as time series sensor data, optical measurements, and logs to perform predictive analysis and process optimization for ALD. Inspired by Samsung's multi-modal fundation model(MMFM) framework for advanced equipment data analytics in Semiconductor Manufacturing [2], this study focuses on developing tailored AI driven solutions for ALD process. By incorporating AI and data-driven models, we can predict material properties and optimize the ALD process, which directly contributes to the development of lifecycle sustainability frameworks and next-generation semiconductor technologies.

1.3 Related Work

The latest advances in the intersection of artificial intelligence (AI) and atomic layer deposition (ALD) provide promising methods for improving prediction accuracy, optimizing processes, and enhancing the sustainability of semiconductor manufacturing.

Adeleke et al. [3]conducted a comprehensive review on the application of machine learning (ML) techniques to ALD, emphasizing the integration of ML with traditional simulation methods to improve quality control and material discovery. Their work provided insights into how ML can facilitate better understanding and control of ALD processes.

A specific study by Yoon et al. [4] demonstrated the effectiveness of extreme gradient boosting (XGBoost) algorithms in predicting results for platinum nano-film coatings produced by ALD, demonstrating high prediction accuracy and rapid inference capabilities.

Similarly, Arunachalam et al. [5] applied ML methods to predict film thickness from in-situ spectroscopic ellipsometry data during ZnO ALD processes. Their findings emphasize the ability of ML, particularly k-nearest neighbor algorithms, to accurately predict deposition outcomes using real-time monitoring data.

Samsung's development of multi-modal large language models (LLMs) for autonomous semiconductor fabrication further demonstrates the utility of integrating multimodal data streams. Their approach combines sensor data, knowledge graphs, and LLMs to enhance equipment control, defect detection, and process optimization [2].

In semiconductor defect analysis, Jiang et al. [6] introduced FabGPT, a customized multimodal model customized for querying wafer defect knowledge. This system significantly enhances the efficiency and accuracy of defect diagnostics in integrated circuit fabrication.

Moreover, Dogan et al. [7] explores Bayesian machine learning methods to minimize defects in ALD coatings, specifically in Al_2O_3 passivation layers for metallic copper corrosion protection. This method optimizes ALD parameters to achieve defect-free, high-quality deposition results.

These related studies collectively emphasize the significant potential of using artificial intelligence and machine learning to improve ALD processes, with the goal of enhancing predictive capabilities, improving operational efficiency, and promoting the sustainability of advanced semiconductor manufacturing.

1.4 The Proposed Research

1.4.1 Research Objective

The primary objective is to develop a Multi-Modal Time Series Foundation Model for ALD processes. This model will:

1. Process Multimodal Data Streams
2. Predict Material Properties
3. Enable Sustainability

1.4.2 Proposed Methodology

Data Collection and Preprocessing

- Collect multimodal datasets from ALD processes, including sensor outputs, chamber logs, and optical spectra and apply naming techniques.
- Apply domain-adapted preprocessing techniques such as noise filtering, reversible instance normalization, and temporal alignment.

Model Development

- Use models like SOFTS[8] and MOMENT[9] for pre-training on diverse time-series datasets.
- Use dense embeddings, masked representation learning, and knowledge graphs to integrate datasets.
- Incorporate physics-informed embeddings to capture domain-specific features.

Evaluation Metrics

- Use Mean Absolute Error (MAE) and Root Mean Square Error (RMSE) for material property predictions.
- Calculate F1-scores for anomaly detection.
- Benchmark the model against some state-of-the-art techniques, including transformer-based and non-transformer models[10].

Sustainability Integration

Collaborate with the larger project's lifecycle sustainability framework to ensure that process optimizations is aligned with environmental goals.

1.4.3 Challenges

The main challenges are:

1. High-dimensional data: The ALD process generates complex high-dimensional data streams from sensors, logs, which may need dense embeddings.

2. Date alignment: Integrating different data types requires advanced alignment techniques.

3. Tasks generalization: Developing a single foundational model that can adapt to multiple tasks is challenging.

1.5 Current Progress and Project Development

1.5.1 Data Collection and Infrastructure Setup

As a first step toward building a foundation model for ALD process optimization, we have focused on data curation and infrastructure development. The primary equipment used in this study is the Veeco Fiji G2 Atomic Layer Deposition (ALD) System, which supports plasma-enhanced ALD processes critical for fabricating high-K dielectric films such as HfO_2 and ZrO_2.

The Veeco system continuously generates high-resolution time-series logs from its built-in sensors. These include data on precursor pulse timing, chamber pressure evolution, RF plasma activation, and substrate heating cycles, among others. However, these logs are initially stored in proprietary raw formats, which are difficult to access and interpret directly for modeling purposes.

To enable data-driven analysis, our cleanroom engineer has manually parsed these raw logs, extracting relevant sensor channels and metadata. The structured data is then uploaded to a dedicated cloud-based database bucket, where it is stored along with standardized metadata tags such as: toolname, recipename, category, subcategory, manufacturer, model, serial, and hash.

1.5.2 Challenges with Manual Parsing and Motivation for Automation

A key bottleneck in the early stages is manually parsing raw ALD logs, which is time-consuming and prone to errors, especially considering the large amount of historical data collected in many experimental runs. The process of converting raw logs into structured data suitable for consumption by artificial intelligence models took several months, during which inconsistencies in file formats, missing metadata, and inconsistent timestamps had to be manually resolved. And we have to consider the tags in advance for future use.

Recognizing the inefficiency, we are now actively working towards automating the data ingestion and parsing pipeline. Our goal is to develop a robust system that continuously extracts, parses, tags, and uploads new Veeco process logs to the database with minimal human intervention. This automation will not only accelerate the speed of data collection, but also improve the consistency and repeatability of data, which is a key requirement for basic models training and downstream tasks.

1.5.3 Data Access and Streaming Interface

In order to simplify the process of storing and analyzing structured data, we have started implementing a Node RED based interface that allows us to pull relevant time series data from cloud database buckets and stream it to our AI backend.

This setting allows automatic extraction of data slices by recipe name, process type, or time window, and supports batch training and real-time inference workflows. We pull the necessary data from the database for our model use.

1.5.4 Exploration of Foundation Models

As the project progresses towards building a prediction framework for film property using time-series sensor data from ALD processes, one of the key challenges is to choose a suitable model architecture that can be well generalized to multivariate, high-dimensional temporal data. To address this issue, we conducted extensive research on recent time series baseline models and identified MOMENT (Multi Task Open Model for Time Series Embedding and Normalization) as a highly promising candidate.

MOMENT is a family of open-source foundation models developed by Auton Lab at Carnegie Mellon University [9], the MOMENT architecture is shown in . It is specifically designed to support various time-series tasks, including forecasting, classification, anomaly detection, and representation learning. This general-purpose architecture is closely related to our goals, which involve predicting the characteristics of deposited thin films (such as dielectric quality, growth rate stability) based on the process sensor features recorded during ALD runs.

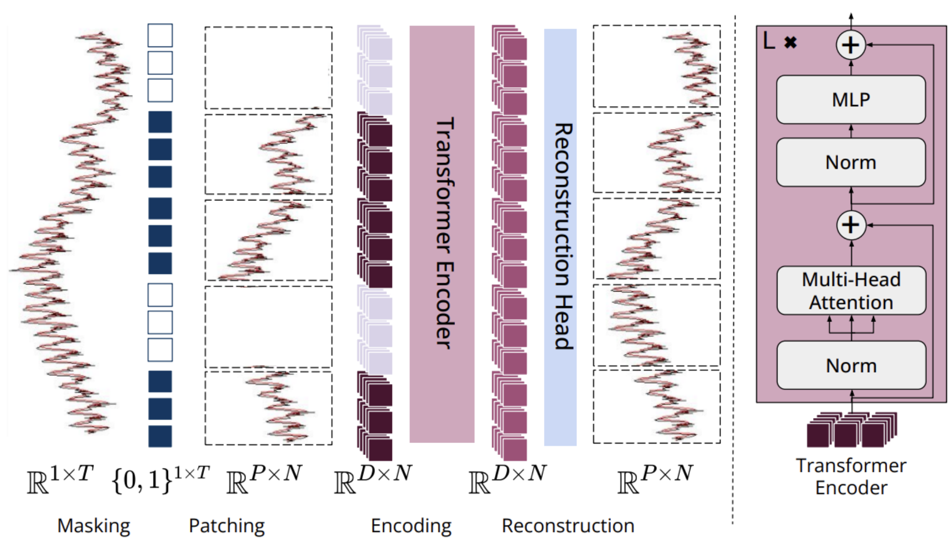

Figure 1.2: MOMENT MODEL Architecture

The MOMENT architecture is shown in Figure 1.2. The MOMENT-1 architecture provides pre-trained pipelines for different tasks, with three variants: small, basic, and large. The main functions of MOMENT include:

- The zero-shot and few-shot capabilities reduce the need for large labeled training datasets.
- Fine tuning support for specific tasks using distributed data.
- The encoder backbone network based on transformers has achieved attention-based modeling of long-distance dependencies across time.
- Supports multiple time-series inputs, which is suitable for processing sensor streams such as pressure, temperature, and more.

MOMENT can be accessed programmatically using the momentfm Python package. For our example, if we want a classification pipeline for binary film quality prediction, our initialization requires minimal configuration of that in Figure 1.3 and Figure 1.4. MOMENT can be accessed via the pre-trained model available on Hugging Face [11].

```
from momentfm import MOMENTPipeline

model = MOMENTPipeline.from_pretrained(
    "AutonLab/MOMENT-1-base",
    model_kwargs={
        'task_name': 'classification',
        'n_channels': 1,
        'num_class': 2
    },
)
model.init()
```

Figure 1.3: MOMENT Classification [11]

Given the structured sensor data provided by the Veeco Fiji G2 ALD system, including pressure and precursor pulse profiles, MOMENT provides a good starting point for

```python
from momentfm import MOMENTPipeline

model = MOMENTPipeline.from_pretrained(
    "AutonLab/MOMENT-1-base",
    model_kwargs={
        'task_name': 'forecasting',
        'forecast_horizon': 96
    },
)
model.init()
```

Figure 1.4: MOMENT forecasting [11]

analyzing trends related to thin film performance results. Especially, reconstructing and predicting pipelines is expected to learn representations of chamber pressure cycling and precursor dynamics, which are essential for the nucleation and growth efficiency of thin films in each cycle.

In addition, MOMENT's support for embedding extraction enables unsupervised representation learning across multiple ALD runs, laying the foundation for downstream tasks such as anomaly detection or virtual metric prediction.

1.6 Current progress and future work

Although the modeling phase is still in its early stages, but right now we have successfully:

- Set up the MOMENT framework locally, install the required packages, and prepare a sample pipeline using publicly available datasets to validate its core functionality.
- Explored the embedding extraction function of MOMENT to analyze whether the clustering of sensor traces from different recipes is consistent with known material outcomes.

We are currently preparing to map standardized Veeco logs (now included in our database) to MOMENT input format. This includes data formatting, channel alignment, and zero padding necessary for multi-channel input support.

In the next phase, we plan to:

1. Extend the MOMENT pipelines to support multi-channel, multi-resolution ALD data, including high-frequency chamber pressure logs and low-frequency metrology results (e.g., probe station data measurements).

2. Fine-tune MOMENT using paired Veeco process data and quality labels obtained from post deposition measurements.

3. Compare the performance of MOMENT against baseline models such as SOFTS and traditional CNN/RNN architectures on metrics including RMSE, F1-score, and zero-shot generalization.

CHAPTER 2
AUTOMATED SEMICONDUCTOR CHARACTERIZATION

2.1 Motivation

With the increasing complexity and scale of modern semiconductor devices, especially in emerging ferroelectric-based technologies such as $HfZrO_x$-based capacitors and FeFETs, the characterization process has become a critical bottleneck for technology development and process-device co-optimization. The traditional E-test workflow suffers from several fundamental limitations:

- **Manual Intervention:** Traditional characterization relies on expert-driven planning, execution, and validation, which introduces subjective variability and error.
- **Time Constraints:** The complete device characterization requires thousands of measurements under multiple test conditions, which makes comprehensive testing very time-consuming.
- **Data Quality Assurance:** Due to equipment limitations, probe misalignment, or device defects, measurement artifacts often contaminate datasets, it is necessary to conduct strict validation before extracting parameters.
- **Scaling Challenges:** Manual approaches cannot efficiently scale to characterize thousands of devices across multiple wafers.

These limitations significantly impede the rapid development cycle required for next-generation semiconductor technologies, particularly for emerging ferroelectric devices, the characterization is crucial for understanding reliability mechanisms and optimizing the fabrication process.

2.2 System Overview and Agentic Framework

To address these challenges, we propose an end-to-end automated characterization framework based on a modular, agent-based architecture. This framework combines advanced machine learning techniques with domain-specific electrical characterization expertise to create a self optimizing system for semiconductor device analysis.

Our framework integrates a hierarchy of autonomous agents designed to manage the entire semiconductor characterization pipeline (Figure 2.1). Each agent utilizes domain specific tools, statistical analysis, and AI-driven reasoning. Agents share memory and interfaces with backend simulators (e.g. Ginestra^{TM}) and front-end probe stations through code generation and data parsing.

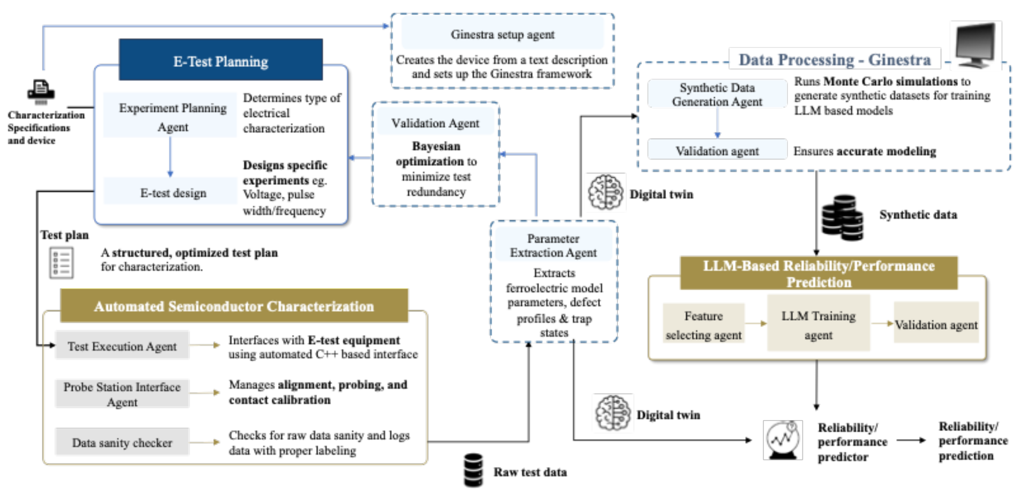

Figure 2.1: Overview of the proposed automated agentic framework for semiconductor characterization.

The workflow include:

- **Experiment Planning Agent** – Designs test conditions (voltages, frequencies) based on device specs.

- **Execution Agent** – Interfaces with measurement equipment through automated C++ code generation, manages probe station alignment, and executes the planned experiments.

- **Validation Agent** – Classifies measured device data as *Good*, *Leaky*, *Short*, or *No Contact*.
- **Parameter Extraction Agent** – Fits physical models to validated data and extracts device parameters to construct a digital twin.

My research focuses primarily on the Validation part, which can check the leaky films from their e-test mesurements. From the results from the validation, we keep the non-leaky ones and later we will do preisach and NLS, build the digital twin and also do the characterization and modeling of Ferroelectric polarization switching. So it's a critical quality gate to ensure that only valid measurement data proceeds to downstream modeling and analysis. To achieve the goal, we need to build a machine learning model to implement the logic.

2.3 Background and Problem Definition

Traditionally, leakage identification in ferroelectric devices is carried out using polarization-voltage(P-V) loop [12]. A typical P-V loop displays the charge response under sweeping gate voltage, and experts analyze hysteresis, saturation from that. Abnormal shapes, such as severe distorted or open loops, may indicate excessive leakage or shorts. However, this method is usually manual, which is time-consuming and prone to errors when expanding to a large number of devices. Thus, this work aims to develop a machine learning based testing validation agent that classifies devices as leaky, good, or other issues from raw measurement data to reduces human labor and ensures scalable validation for downstream tasks.

The validation task can be formally defined as follows: Given a set of time-series electrical measurements from semiconductor device characterization (typically time-series measurements of current, or charge, we can also get $I - V$, $Q - V$, or $I_t - V$ curves), automatically determine whether the signal represents valid device behavior.

As shown in Figure 2.2, the validation pipeline for semiconductor ferroelectric device characterization is illustrated.

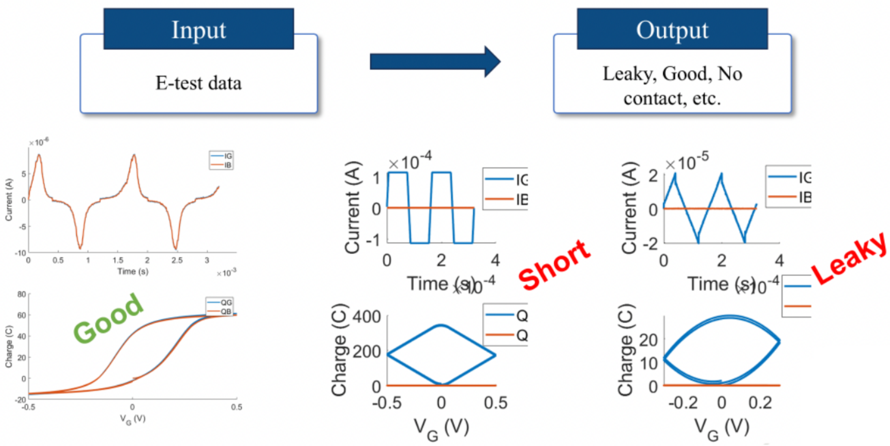

Figure 2.2: Validation Pipeline

In the full problem scope, this is a multi-class classification task with categories:

- `Good`: Valid measurement showing expected device characteristics
- `Leaky`: Measurement exhibiting gate leakage or partial dielectric breakdown
- `Shorted`: Complete device failure with direct electrical path
- `No Contact`: Measurement artifact due to probe misalignment or contact failure

For initial development and proof-of-concept validation, we constrain the problem to binary classification:

- `Non-Leaky`: Valid measurement suitable for parameter extraction
- `Leaky`: Invalid measurement showing gate leakage requiring rejection

This simplification allows us to design and validate core models and pipelines before scaling to full multi-class classification.

The input data consists of raw electrical measurement time-series, where each experiment produces traces of current, voltage, and charge values sampled over time. As evident in Figure 2.3, leaky devices typically exhibit distinct morphological characteristics in their measurement traces, including reduced saturation, increased asymmetry, and current

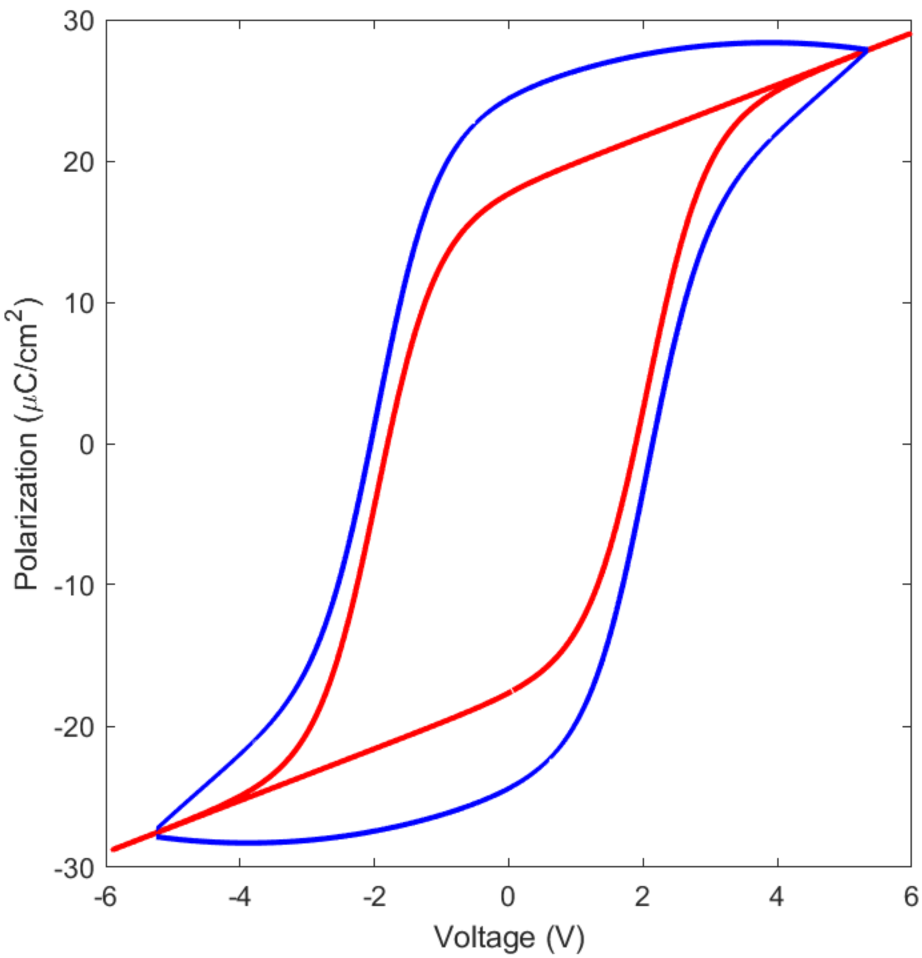

Figure 2.3: Synthetic data: red curves represent non-leaky devices, blue curves represent leaky behavior.

flow during supposed non-conduction phases. These patterns, while visually identifiable to domain experts through polarization-voltage hysteresis loop, present an opportunity for automated detection using pattern recognition techniques that can extract relevant features from time-series data sequence.

2.4 Related Work

The trend of using machine learning (ML) technology in semiconductor device fabrication and characterization is increasing, especially in automating data analysis, process optimization, and reducing the manpower required to identify anomalies. Previous studies have demonstrated both unsupervised and supervised methods, as well as deep learning techniques for data analysis, quality control, and process optimization in fabrication systems.

2.4.1 Deep Learning for Time Series Classification

In deep learning literature, time series classification has already been solved through various methods. Convolutional neural networks (CNNs) are a remarkable approach in time series analysis. Wang et al. [13] demonstrated that FCN can achieve performance comparable to or better than recursive architectures, while also having higher computational efficiency. Recurrent neural networks (RNNs) and their variants, such as LSTM and GRU, have also been used for sequential data because they can capture temporal dependencies [14]. However, these architectures are often plagued by computational complexity and training difficulties.

2.4.2 Neural Networks for Semiconductor Testing

In the field of ferroelectric device analysis, unsupervised clustering techniques have been successfully applied to distinguish characteristic behaviors in electrical measurements without the need for labeled data. For example, Hiranaga et al. [15] applied k-means and Gaussian Mixture Model (GMM) clustering to the nanoscale C-V hysteresis loop of doped HfO_2 thin films. Their method revealed distinguishable "butterfly" loops corresponding to normal ferroelectric switches, rather than asymmetric loops implying domain pinning or built-in field effects. Neumayer et al. [16] further extended unsupervised ML to the classification of polarization switching behavior by converting hysteresis loops into 2D representations and applying linear classifiers.

If we have labeled examples, we can employ supervised machine learning method, which has also been applied in validation and fault diagnosis tasks. For instance, Liu et al.[17] combined high-speed piezoresponse force microscopy (HSPFM) with a Bayesian-optimized support vector machine (SVM) to classify dynamic domain growth in ferroelectric materials, effectively distinguishing switching and stable regions. Similarly, Xu[18] proposed a graph neural network-based approach for fault detection in digital integrated circuits, which can identify defects such as leakage and misalignment by analyzing electrical signal variations. Compared with traditional test methods, these methods have higher accuracy and robustness.

2.4.3 Time-Series Characterization

Given that E-test data typically takes the form of sequential time voltage or time current measurements, deep learning models designed for time series analysis, such as recurrent neural networks (RNNs) and long short-term memory (LSTM) networks, can be applied here. Agar et al. [19] developed an autoencoder based on LSTM for processing piezoelectric response spectral data and revealing subtle switching features such as elastic hardening or charged domain wall nucleation that traditional statistical methods cannot distinguish.

These networks are able to capture the temporal dependencies and subtle deviations in waveform morphology, making them pretty suitable for classifying leaky and non-leaky device behavior from raw electronic test outputs, which is our case.

CHAPTER 3
METHODOLOGY FOR E-TEST DATA VALIDATION

3.1 Synthetic Dataset Generation

Due to the limited amount of E-test data measured in the early development stage, we generated a synthetic dataset to help model development and benchmark baseline classification performance. The synthetic dataset is generated using a Preisach model for ferroelectricity. Specifically, we use the Preisach model of ferroelectricity to simulate device behavior under controlled parameter perturbations, such as trap density, residual polarization, and electric field confinement, through Monte Carlo simulations using `Ginestra`™. The synthetic data is used to train the model.

3.2 Dataset Preprocessing and feature selection

The synthetic dataset used in this project comprises approximately 30,000 CSV files, each corresponding to a unique semiconductor device under electrical test. Each file follows the naming convention `dev#_label.csv`, where `#` denotes the device index and `label` indicates the binary classification target: `0` for non-leaky devices and `1` for leaky devices. Each file contains 1,002 rows and four columns representing time-series measurements: time (`t`), voltage (`v`), charge (`q`), and current (`i`).

Figure 3.1 shows the class distribution of the dataset. The leaky and non-leaky devices are nearly balanced, with 15,405 leaky samples (51.35%) and 14,595 non-leaky samples (48.65%). The distribution of label in our sample is very even.

Initial exploratory analysis showed that the `q` (charge) signal provides the most discriminative power for distinguishing leaky from non-leaky devices. As illustrated in Figure 3.2, leaky devices exhibit reduced saturation, increased asymmetry, and distortion in

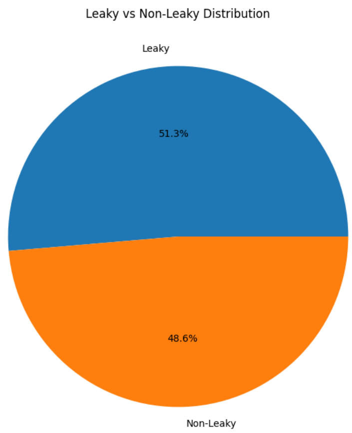

Figure 3.1: Distribution of leaky vs. non-leaky device samples in the synthetic dataset.

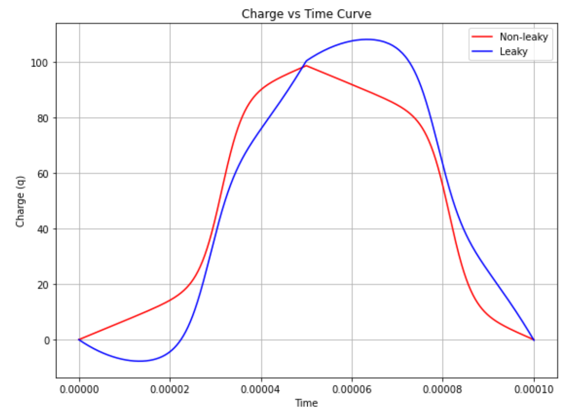

Figure 3.2: Synthetic charge-voltage curves. Red: non-leaky; Blue: leaky.

the charge-voltage response, whereas non-leaky devices follow a more symmetric and saturated polarization curve.

As shown in Figure 3.3, to prepare the data for model training and evaluation, we design a preprocessing pipeline consisting of three main steps:

- **Data Splitting:** We apply a stratified split at the device level to ensure that all measurements from a given device are assigned to a single subset. This avoids data leakage across the training, validation, and testing splits. The dataset is partitioned into 70% training, 15% validation, and 15% testing sets. And the category distribution of each one is relatively even.

- **Normalization:** We use `MinMaxScaler` normalization method to the column of the charge (`q`) measurements. The scaling parameters (min and max values) are computed exclusively from the training set and subsequently applied to the validation and testing sets as a global fitting to avoid information leakage. The normalization helps the model converge faster and improves performance by ensuring all features are on a similar scale. The scaler could be saved for future use.

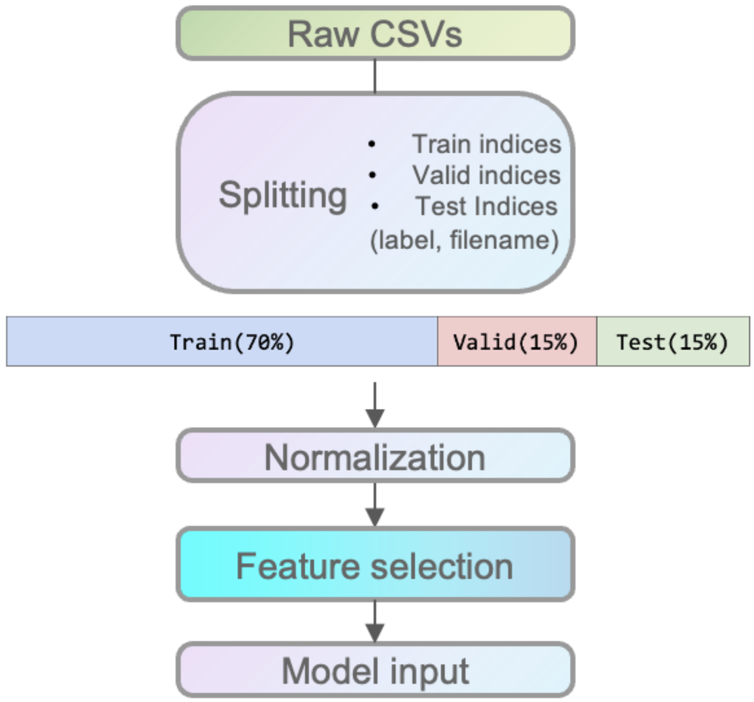

Figure 3.3: Overview of the data preprocessing pipeline: raw CSVs are split into train/valid/test subsets, normalized, and reduced to selected features for model input.

- **Feature Selection:** Based on domain knowledge of the polarization dynamics of the FeFFTs [20], we select the charge (q) response to applied voltage as the primary input feature. The Charge (q) capture polarization transfer and leakage process, so we focus on the charge fluctuations. Since our sequence length is consistent, each data file contains 1002 data points, we only use charge as our model input to dig insights from the charge dynamics, and finally achieve the goal of distinguishing between leaky and non-leaky devices from their electrical measurement data sequence. Each device is thus represented by a one-dimensional vector of length 1002, corresponding to the sampled charge values over time.

3.3 Model Selection

To detect the leakage of the semiconductor devices, we adopted the 1D Convolutional Neural Network (CNN) architecture originally developed by the GATECH-EIC Lab [21] for the 2022 TinyML contest [22]. Their model was designed for binary classification of Intracardiac Electroencephalography(IEGM) signals based on 1D temporal patterns.

Table 3.1: Comparison between IEGM Classification and Leakage Validation

Aspect	IEGM Classification	Leakage Detection
Signal Type	1D time-series cardiac signals	1D time-series measurements
Classification Task	Binary (VA vs. non-VA arrhythmias)	Binary (leaky vs. non-leaky devices)
Signal Length	1,250 samples	1,002 samples
Pattern Characteristics	Temporal morphological variations in waveform	Morphological distortions in charge response curves
Key Features	Waveform differences between normal and abnormal heart rhythms	Asymmetry, reduced saturation, and distortion in charge curves

As shown in Table 3.1, both applications involve analyzing time series data with subtle morphological differences that indicate different states, the TinyML problem analyzes the cardiac abnormalities in IEGM and our case is trying to find the leakage in semiconductor devices. In addition, the structure of our datasets consisting of univariate time series signals with significant pattern differences is relatively similar, so this architecture became the natural choice for our preliminary research. The 1D-CNN is well-suited for our case of distinguishing leaky from non-leaky semiconductor devices based on charge response sequences from electrical measurements.

3.4 Baseline Architecture

To establish a strong baseline for the validation task, we adapt the neural network architecture originally developed by the GATECH-EIC Lab for the 2022 TinyML contest above. Given the structural similarity between our dataset and the IEGM dataset and the same binary classification tasks, we employ this architecture with minimal modification. This decision enables us to quickly test feasibility and establish benchmark performance without introducing unnecessary model complexity in the early stages.

As illustrated in Figure Figure 3.4, the baseline model is a deep 1D convolutional neural network composed of the following elements:

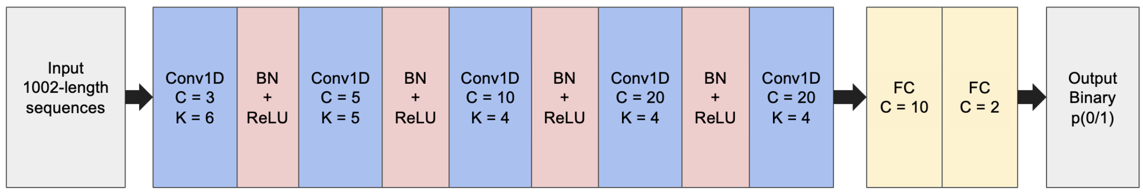

Figure 3.4: Baseline Conv1D architecture. C: Number of Channeld, K: Kernel Size

- **Five sequential 1D convolutional layers**, with filter counts increasing as (3, 5, 10, 20, 20) and kernel sizes of (6, 5, 4, 4, 4). These layers are used to gradually extract higher-level temporal features from the input charge signal.

- **Batch normalization** is applied after each convolutional layer to stabilize training and accelerate convergence.

- **ReLU activation functions** introduce nonlinearity after each layer to capture complex patterns in the signal.

- **Adaptive flattening layer** dynamically determines the correct dimensionality between the convolutional feature extractor and subsequent fully connected layers.

- **Two fully connected (dense) layers** finalize the classification pipeline: one with 10 neurons and the final output layer with 2 neurons corresponding to the binary classification labels (0 or 1).

In our case, the model takes as input a 1D signal of length 1002, it corresponds to the normalized charge response of the devices. And the model outputs a binary label (0 or 1) indicating whether the charge signal corresponds to a leaky or non-leaky device.

This multi-layer architecture follows the traditional approach of stacking multiple small convolutional kernels to build a deep feature extractor, it's similar to the VGG-style networks in image processing[23]. Although this method is effective for IEGM signal analysis, our case of e-test leakage validation and other similar tasks, it introduces a significant computational complexity with huge parameter count, which may limit deployment on resource constrained systems used in semiconductor testing environments.

3.5 Improved Architecture

The above baseline architecture has strong performance in binary classification, but it has multiple stacked layers, which results in approximately over 200000 parameters and introduces unnecessary computational complexity to our task of e-test validation.

From the analysis of the synthetic data, we think we can captured the charge patterns through simpler architectures. Through studies, we determined that a single convolutional layer with a larger kernel size (85) and increased stride (32) could achieve comparable discriminative ability. Rather than using the multi-layer CNN, we simplified the architecture to improve computational efficiency while maintaining classification accuracy. The improved architecture is shrunk from the multi-layer Conv-1D model which has stacked Conv layers.

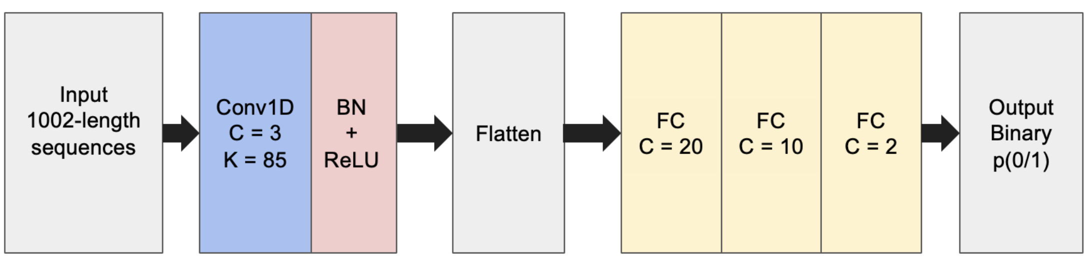

Figure 3.5: Improved Conv1D architecture. The model uses a single large-kernel convolutional layer. C: Number of Channel, K: Kernel Size

As illustrated in Figure 3.5, our optimized model consists of:

- A single 1D convolutional layer with 3 filters, a kernel size of 85, and a stride of 32
- Batch normalization for training stability
- ReLU activation function

- A flattening operation that converts the convolutional features to a 1D vector
- Two fully connected layers of sizes 20 and 10 neurons respectively, with ReLU activations
- Dropout layers (30% and 10% rates) for regularization
- A final output layer with 2 neurons corresponding to the binary classes (0 or 1)

The new architecture could significantly reduce the parameter count to below 3000, a decrease of over 90% compared to the baseline. It reduces the model size compared to the original 5-layer stack. This simplified architecture significantly reduces the computational cost while still capturing the essential temporal patterns in the charge response. The large kernel size and stride in the convolutional layer effectively downsample the signal while preserving the key features that distinguish leaky from non-leaky device measurements.

3.6 Training and Optimization Strategies

We implemented several training techniques to enhance model performance.

3.6.1 Data Augmentation

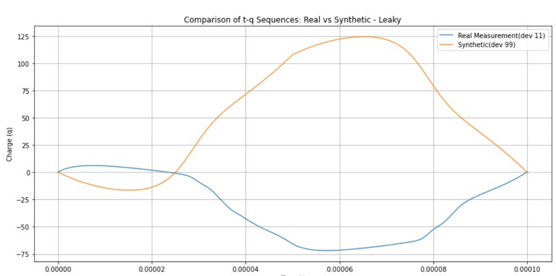

Figure 3.6: Comparison of real vs. synthetic charge-time curves for leaky devices.

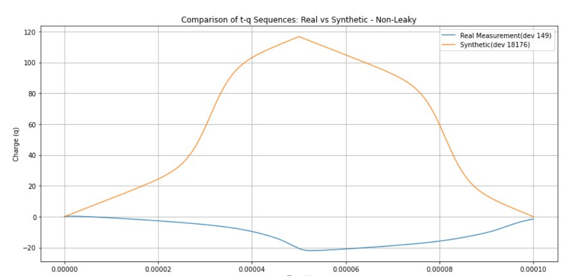

Figure 3.7: Comparison of real vs. synthetic charge-time curves for non-leaky devices.

We compared the patterns of the real data plots and the synthetic data plots of the charge-time sequences and checked the differences for both leaky and non-leaky devices.

As shown in Figure 3.6 and Figure 3.7, there are some variations in amplitude and shape between the overall curve structure of the synthetic samples and the actual measurement results. In order to improve generalization ability, we applied various specific transformations during the training process, this could help model learn patterns and tolerate minor signal fluctuations:

- **FlipSignal:** Randomly inverse signal (multiplication by -1) with the probability between 0.3 and 0.7
- **TimeReverse:** Randomly reverse the time series to regularize sequence directionality (horizontal flipping)
- **AddGaussianNoise:** Randomly add small noise to simulate measurement variability

We applied these augmentation techniques dynamically during the training process, with different random transformations for each epoch. This method effectively increasing the diversity of the training set.

3.6.2 Stochastic Weight Averaging (SWA):

We use SWA [24] to average multiple model checkpoints during the training process. This method helps the model converge to a flatter minima and improves generalization ability. Our implementation tracks the running average of model weights starting, helping to find a wider range of optimal solutions.

$$\theta_{SWA}^{(t)} = \frac{\theta_{SWA}^{(t-1)} \cdot n_{models} + \theta^{(t)}}{n_{models} + 1} \tag{3.1}$$

where $\theta_{SWA}^{(t)}$ is the SWA model after t epochs, $\theta^{(t)}$ is the current model, and n_{models} is the number of models included in the average so far. Our SWA starts from epoch 10, where we collect models every 5 epochs and include it in the average.

3.6.3 Optimization Algorithm

We used the Adam optimizer [25] for parameter optimization during model training. It combines the advantages of momentum based methods with adaptive learning rates. In our implementation, it is configured with an initial learning rate of 2×10^{-4} and weight decay of 1×10^{-4} for L2 regularization.

$$\text{optimizer} = \text{Adam}(\theta, \eta = 2 \times 10^{-4}, \text{weight\_decay} = 1 \times 10^{-4}) \tag{3.2}$$

The measurement data may exhibit different gradient amplitudes on different features, Adam's adaptive learning rate behavior helps the model converge more effectively.

In addition, our implementation combines the Adam with a cosine annealing scheduler [26] that gradually reduces the learning rate from an initial value (2e-4) to a minimum value (2e-6) following a cosine curve over epochs. This promotes helping the model converge to better minima.

$$\eta_t = \eta_{min} + \frac{1 + \cos\left(\frac{t\pi}{T}\right)}{2} \cdot (\eta_{max} - \eta_{min}) \tag{3.3}$$

where η_t is the learning rate at epoch t, T is the total number of epochs, and η_{max} and η_{min} are the maximum and minimum learning rates respectively.

3.6.4 Multiple Training Runs:

Considering the randomness of training, We used different random seeds and performed multiple training runs (usually 3-10 runs). Then we selected the best-performing model based on the validation F_β score and also stored the best accuracy model. For our application, we prioritizes recall over precision.

3.6.5 Training Procedure

As shown in Algorithm 1, the complete training process follows these steps.

Algorithm 1 Leakage Classification Training Algorithm

for each training run i in $1...N$ **do**
    Initialize model parameters θ randomly with seed s_i
    Initialize optimizer with learning rate $\eta_0 = 2 \times 10^{-4}$
    Initialize SWA model after epoch $E_{swa} = 10$
    Initialize best validation metric $FB_{best} = 0$
    for each epoch e in $1...E_{max}$ **do**
        // Training phase
        for each batch B in training set **do**
            Apply data augmentations to B
            Compute predictions $\hat{y} = f_\theta(B)$
            Compute loss $\mathcal{L}(\hat{y}, y)$
            Update parameters θ using Adam
        end for
        Update learning rate using cosine annealing schedule
        // Validation phase
        Compute validation metrics $FB_{val}, Acc_{val}, CM_{val}$
        if $FB_{val} > FB_{best}$ **then**
            $FB_{best} = FB_{val}$
            Save model parameters $\theta_{best} = \theta$
        end if
        if $e \geq E_{swa}$ and $e \mod f_{swa} = 0$ **then**
            Update SWA model with current parameters θ
        end if
    end for
    Perform final evaluation on test set using θ_{best}
    Update batch normalization for SWA model
    Evaluate SWA model on test set
    Save metrics and training curves
end for
Select best model across all runs

3.7 Performance Metrics

To evaluate the performance of the Conv1D model in binary leakage classification, we used a set of standard metrics based on the confusion matrix. These indicators include accuracy, precision, recall (sensitivity), specificity, F1 score, balanced accuracy (BAC), and F_β score emphasizing recall, where $\beta = 2$. Each metric is calculated based on the four basic components of the confusion matrix.

We used a comprehensive set of performance metrics based on the confusion matrix evaluate our model:

$$CM = \begin{bmatrix} TP & FP \\ FN & TN \end{bmatrix} \tag{3.4}$$

where TP (True Positives) represents the number of correctly identified leaky devices, TN (True Negatives) represents correctly identified non-leaky devices, FP (False Positives) indicates non-leaky devices misclassified as leaky, and FN (False Negatives) indicates leaky devices misclassified as non-leaky, as shown in Figure 3.8. The formal definations of all metrics are summarized in Table 3.2.

	Predicted Positive	**Predicted Negative**
Actual Positive	TP	FN
Actual Negative	FP	TN

Figure 3.8: Confusion matrix layout

Here, We choose F_β Score (with $\beta = 2$) as the main metric for model selection. Because in our case of quality validation, the cost of missing leaky devices(False negatives) is higher

Table 3.2: Evaluation metrics for leakage binary classification

Symbol	**Metric Name**	**Definition**
ACC	Accuracy	$\frac{TP+TN}{TP+TN+FP+FN}$
PPV	Precision	$\frac{TP}{TP+FP}$
NPV	Negative Predictive Value	$\frac{TN}{TN+FN}$
SEN	Recall (Sensitivity)	$\frac{TP}{TP+FN}$
SPE	Specificity	$\frac{TN}{TN+FP}$
F1	F1 Score	$2 \cdot \frac{\text{PPV}\cdot\text{SEN}}{\text{PPV}+\text{SEN}}$
FB	F_β Score ($\beta = 2$)	$(1 + \beta^2) \cdot \frac{\text{PPV}\cdot\text{SEN}}{\beta^2\cdot\text{PPV}+\text{SEN}}$
BAC	Balanced Accuracy	$\frac{\text{SEN}+\text{SPE}}{2}$

compared to missclassifying non-leaky devices (False positives). FB prioritizes recall over precision, so it is the most important indicator here.

CHAPTER 4
MODEL EVALUATION AND ANALYSIS

4.1 Experimental Results

All experiments were conducted on identical hardware with the same following training parameters:

- Batch size: 32
- Optimizer: Adam with initial learning rate of 0.0002
- Learning rate scheduling: Cosine annealing
- Training epochs: 50
- Dataset split: 70% training, 15% validation, 15% testing

4.1.1 Baseline Model Performance

We conducted 3 independent training runs on the baseline architecture to ensure the robustness of the results. The best performing model achieved a testing accuracy of 95.62% and an F-beta score of 0.9796, demonstrating strong foundational performance.

The total number of trainable parameters are 202499 in one run, the baseline model shows consistent performance across different runs, with an average training time of 79.3 minutes (4758 seconds) per run (on CPU), which reflects its considerable computational requirements. The high parameter count results in this computational cost.

4.1.2 Improved Architecture Performance

For our improved architecture, we use Stochastic Weight Averaging (SWA) and data augmentation compared to the baseline (Figure 4.1. To further understand the benefits of our

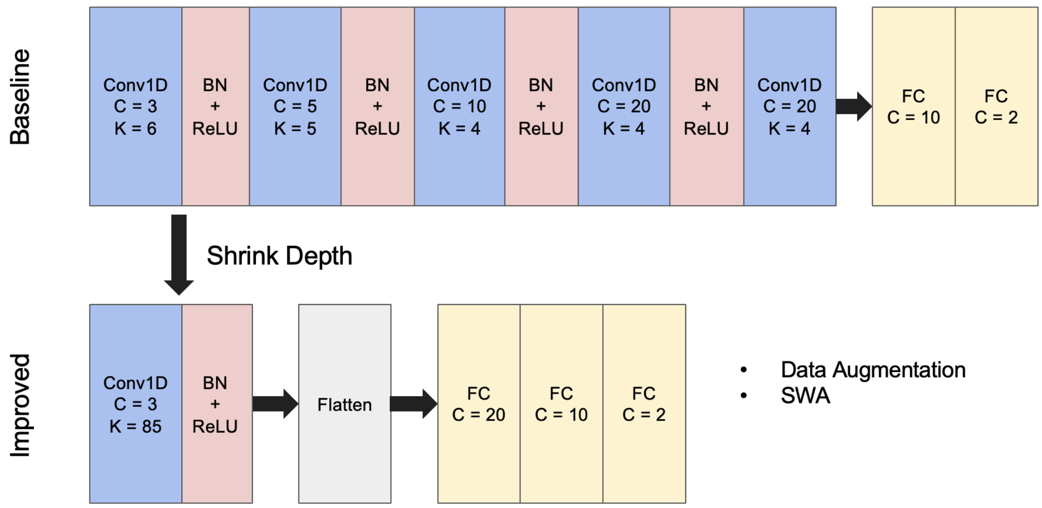

Figure 4.1: Comparison between the baseline and improved model architectures. C: number of Channels, K: Kernel Size

training methodology, we conducted additional experiments with the improved architecture, comparing performance with and without data augmentation and Stochastic Weight Averaging (SWA).

Firstly, we assessed the performance of our improved architecture without the application of data augmentation techniques or Stochastic Weight Averaging. Below shows the metrics from the best run in this configuration among 5 runs, this configuration achieved great performance.

- **Best validation epoch:** 34
- **Validation accuracy:** 95.13%
- **Validation F-beta score:** 0.9451
- **Test accuracy:** 95.82%
- **Test F-beta score:** 0.9558
- **Training time:** 2510.55 seconds (Approximately 41.8 minutes)

Then we conducted five independent training runs randomly with both SWA and data augmentation (including flip, reversal, and noise addition). The results of these five runs showed interesting variation in convergence patterns, as shown in Table 4.1:

Table 4.1: Detailed performance metrics across multiple runs of the improved architecture

Run ID	Best Epoch	Val Acc (%)	Val FB	Test Acc (%)	Test FB	Test Loss	Confusion Matrix TP	FN	FP	TN	Training CPU Time(s)
0	41	95.29	0.9480	95.51	0.9504	0.1506	2210	125	77	2088	2515.57
1	49	89.47	0.8488	90.04	0.8577	0.2679	1946	389	59	2106	2512.81
2	41	97.36	0.9701	97.67	0.9729	0.1099	2265	70	35	2130	2503.26
3	1	52.22	0.8453	51.89	0.8436	0.6924	2335	0	2165	0	2522.78
4	28	96.76	0.9829	96.76	0.9823	0.1098	2314	21	125	2040	2549.46

The third run (Run ID: 2) achieved the best overall performance with a test accuracy of 97.67% and an F-beta score of 0.9729. The fourth run (Run ID: 3) performed the worst with testing loss of 0.6924 and a curious confusion matrix showing the model predicted all samples as positive (leaky). So it is important to conduct multiple training runs with different initializations. Despite this outlier run3, runs 0, 2, and 4 all achieved excellent performance, in particular, run 4 showing strong F-beta scores (0.9823) despite a slightly lower accuracy than run 2.

The confusion matrix of one of our best model (Run ID: 2, Figure 4.2) provides a visualization of its classification performance. Among 4,500 test samples:

- 2,265 leaky devices were correctly identified (True Positives)
- Only 70 leaky devices were misclassified as non-leaky (False Negatives)
- Only 35 non-leaky devices were misclassified as leaky (False Positives)
- 2,130 non-leaky devices were correctly classified (True Negatives)

The improved pipeline achieves strong classification performance.

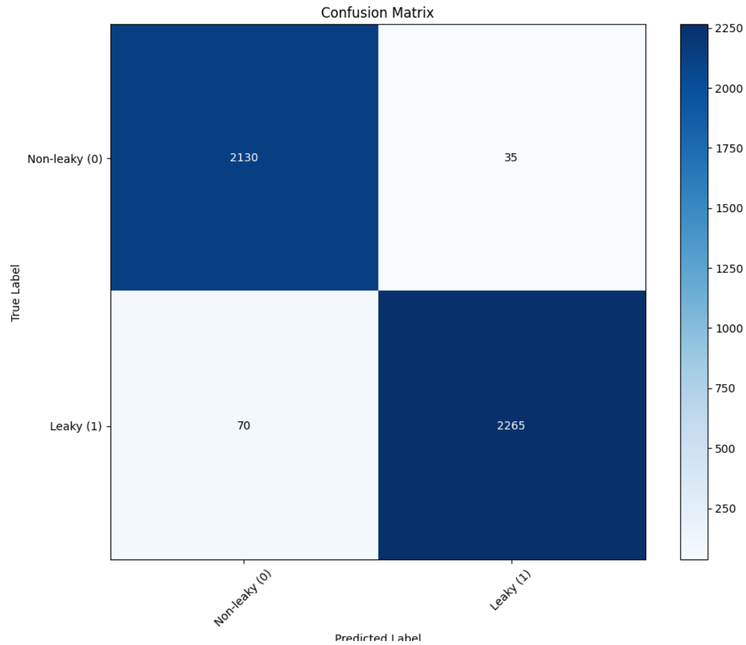

Figure 4.2: Confusion matrix of the best-performing model (Run ID: 2)

4.2 Analysis and Discussion

From the above experimental results, firstly comparing the improved architecture to the baseline, we can see that it outperforming the baseline model, at the same time, it requires approximately 47% less average training time (41.7 minutes versus 79.3 minutes) with less parameters. The significant improvement in computational efficiency proves the effectiveness of our architecture improvement.

The best run without data augmentation or SWA achieved a test accuracy of 95.82% and F-beta score of 0.9558 (best epoch: 34), compared to our best run with data augmentation and SWA which achieved 97.67% accuracy and 0.9729 F-beta score. This indicates an improvement of 1.85 percentage points in accuracy and an increase of 0.0171 in F-Beeta score. Although not significant, it also demonstrates the effectiveness of these techniques in enhancing model generalization.

These results confirm that our approach combining architecture optimization, data augmentation, and SWA has made substantial improvements over baseline architecture. The model achieves strong classification performance across all evaluation metrics, it is able to detect whether the device is leaky from measurement data, which can effectively increase efficiency during the characterization.

CHAPTER 5

CONCLUSION AND FUTURE WORK

5.1 Conclusion

This thesis consists of two projects. The first is the foundation model for predicting the film quality in ALD process, the second is automating semiconductor characterization.

For the first project, the initial stage of this study successfully focused on data collection and preliminary model selection. We systematically collected various multimodal datasets, including pressure sensor logs, and engineering process records related to Veeco ALD system. At the same time, we evaluated and identified a suitable multimodal foundation model that can handle high-dimensional data streams for predictive analysis.

For the second project which focus on the validation of the device leakage for the characterization based on the e-test measurement data, We have successfully developed and validated a deep learning method for detecting semiconductor flim device leakage based on charge (q) time sequence data.

By comparing the baseline and improved model architectures, we have demonstrated that deep convolutional neural networks can effectively learn subtle patterns in charge response data sequences that indicate potential leakage issues. This model lays a solid foundation for automated semiconductor testing systems with the ability to process 1D signal data with high precision, and the system can operate efficiently with minimal computational overhead.

5.2 Future Work

For the ALD prediction project, in the subsequent stages, we plan to use the collected ALD data to fine tune and validate these selected base models. We will focus on optimizing the model architecture, adjusting hyperparameters, and integrating time series and multimodal data to improve prediction accuracy, which ultimately contributes to improved process efficiency and sustainability in the ALD process.

For the validation project, our model performs well which helps us get the non-leaky devices from its e-test measurements. It is promising to extend this project to build the multi-agent automated characterization system for FeFFTs.

- We can fine-tune our model based on real electrical measurement data in the fabrication environment to address domain variations and measurement artifacts that do not exist in the synthetic dataset, and apply them to actual fabrication.

- Our model lay the foundation for the comprehensive digital twin of semiconductor devices, and these results will contribute to the later Preisach and NLS (Nucleation Limited Switching) models and characterization modeling.

- We can integrate our model into the whole autonomous testing framework, develop more agents, and adaptively automate the characterization.

REFERENCES

[1] M. Hinck, M. L. Olson, D. Cobbley, S.-Y. Tseng, and V. Lal, "Llava-gemma: Accelerating multimodal foundation models with a compact language model," *arXiv preprint arXiv:2404.01331v2*, Jun. 2024, Available: https://arxiv.org/abs/2404.01331v2.

[2] J.-Y. Park, *Leveraging gen ai for advanced equipment data analytics in semiconductor manufacturing at samsung*, YouTube video, Industrial AI Federation, Annual Industrial AI Conference at Stanford, Jan. 2024.

[3] O. Adeleke, S. Karimzadeh, and T.-C. Jen, *Machine Learning-Based Modelling in Atomic Layer Deposition Processes*. CRC Press, 2023.

[4] S.-H. Yoon *et al.*, "Extreme gradient boosting to predict atomic layer deposition for platinum nano-film coating," *Langmuir*, vol. 39, no. 14, pp. 4984–4992, 2023.

[5] A. Arunachalam *et al.*, "Machine learning approach to thickness prediction from in situ spectroscopic ellipsometry data for atomic layer deposition processes," *Journal of Vacuum Science & Technology A*, vol. 40, no. 1, p. 012 405, 2022.

[6] Y. Jiang, X. Lu, Q. Jin, Q. Sun, H. Wu, and C. Zhuo, "Fabgpt: An efficient large multimodal model for complex wafer defect knowledge queries," in *Proceedings of the International Conference on Computer-Aided Design (ICCAD)*, ACM, New York, NY, USA: Association for Computing Machinery, 2024, ISBN: 979-8-4007-1077-3.

[7] G. Dogan *et al.*, "Bayesian machine learning for efficient minimization of defects in ald passivation layers," *ACS Applied Materials & Interfaces*, vol. 13, no. 45, pp. 54 503–54 515, 2021.

[8] L. Han, X.-Y. Chen, H.-J. Ye, and D.-C. Zhan, "Softs: Efficient multivariate time series forecasting with series-core fusion," *38th Conference on Neural Information Processing Systems (NeurIPS)*, Nov. 2024, Available: https://github.com/Secilia-Cxy/SOFTS.

[9] M. Goswami, K. Szafer, A. Choudhry, Y. Cai, S. Li, and A. Dubrawski, "Moment: A family of open time-series foundation models," *arXiv preprint arXiv:2402.03885*, 2024.

[10] Y. Liang *et al.*, "Foundation models for time series analysis: A tutorial and survey," in *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '24)*, Barcelona, Spain: ACM, 2024.

[11] AutonLab, *Moment-1-large model*, https://huggingface.co/AutonLab/MOMENT-1-large, Accessed: 2025-04-24, 2025.

[12] K. Lee, B. R. Rhee, and C. Lee, "Leakage current-voltage characteristics of ferroelectric thin film capacitors," *Journal of the Korean Physical Society*, vol. 38, no. 6, pp. 723–728, 2001.

[13] Z. Wang, W. Yan, and T. Oates, "Time series classification from scratch with deep neural networks: A strong baseline," *arXiv preprint arXiv:1611.06455*, 2017.

[14] H. I. Fawaz, G. Forestier, J. Weber, L. Idoumghar, and P.-A. Muller, "Deep learning for time series classification: A review," *Data Mining and Knowledge Discovery*, vol. 33, no. 4, pp. 917–963, 2019.

[15] Y. Hiranaga, Y. Noguchi, T. Mimura, T. Shimizu, H. Funakubo, and Y. Cho, "Data-driven analysis of high-resolution hyperspectral image data sets through nanoscale capacitance–voltage measurements to visualize ferroelectric domain dynamics," *ACS Applied Nano Materials*, vol. 7, no. 8, pp. 8525–8536, 2024.

[16] S. M. Neumayer *et al.*, "To switch or not to switch – a machine learning approach for ferroelectricity," *Journal of Materials Chemistry C*, vol. 8, no. 20, pp. 6837–6846, 2020.

[17] Y. Liu, B. Yu, Z. Liu, D. Beck, and K. Zeng, "High-speed piezoresponse force microscopy and machine learning approaches for dynamic domain growth in ferroelectric materials," *ACS Applied Materials & Interfaces*, vol. 12, no. 8, pp. 9944–9952, 2020, Epub: February 17, 2020.

[18] Z. Xu, "An intelligent fault detection approach for digital integrated circuits through graph neural networks," *Mathematical Biosciences and Engineering*, vol. 20, no. 6, pp. 9992–10 006, 2023, Published: March 27, 2023.

[19] J. C. Agar *et al.*, "Revealing ferroelectric switching character using deep recurrent neural networks," *Nature Communications*, vol. 10, no. 1, p. 4809, 2019.

[20] I. R. Filgueira e Silva, O. Lipan, F. Hartmann, S. Höfling, and V. Lopez-Richard, "Microscopic modeling of polarization dynamics in leaky dielectrics: Insights into ferroelectric-like behavior," *Materials Science and Engineering B*, vol. 316, p. 118 089, 2025.

[21] GATECH-EIC, *Tinyml-contest-solution*, https://github.com/GATECH-EIC/TinyML-Contest-Solution, Accessed: March 10, 2025, 2022.

[22] ICCAD, *2022 acm/ieee tinyml design contest at iccad*, ACM/IEEE TinyML Design Contest at ICCAD 2022, Accessed: 1 April 2025, 2022.

[23] X. Ding, X. Zhang, N. Ma, J. Han, G. Ding, and J. Sun, "Repvgg: Making vgg-style convnets great again," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, arXiv preprint arXiv:2101.03697, 2021. arXiv: 2101.03697 `[cs.CV]`.

[24] P. Izmailov, D. Podoprikhin, T. Garipov, D. Vetrov, and A. G. Wilson, "Averaging weights leads to wider optima and better generalization," *arXiv preprint arXiv:1803.05407*, 2018.

[25] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *International Conference on Learning Representations (ICLR)*, 2015, Published as a conference paper at ICLR 2015.

[26] I. Loshchilov and F. Hutter, "Sgdr: Stochastic gradient descent with warm restarts," *arXiv preprint arXiv:1608.03983*, 2016.