High-Level Design (HLD)

WAFER FAULT DETECTION

Version 1.0
Last date of revision: 02/05/2024

JENNA RAVEENKUMAR

Document Version Control

Date Issued	Version	Description	Author
02nd May 2024	1.0	First Version of Complete HLD	Jenna Raveenkumar

Table of Contents

1. Introduction

1.1 Purpose

The purpose of this document is to provide a high-level design for a machine learning-based system to predict the condition of wafers used in semiconductor manufacturing. The system will determine whether a wafer needs to be replaced or can continue to be used, based on sensor inputs. The aim is to improve manufacturing efficiency by predicting wafer faults early, thereby reducing downtime and costs associated with faulty wafers.

1.2 Scope

The scope of this project includes the development of a predictive model, data preprocessing, model training, and validation. The project also includes the integration of the model into an existing system or a newly proposed system to automate the prediction process. The final deliverable is a deployable model along with the necessary documentation.

2. General Description

2.1 Product Perspective

The Wafer Fault Detection System is designed to be a predictive tool integrated into the semiconductor manufacturing process. The system will take sensor data from wafers as input and output a prediction indicating whether the wafer is faulty (-1) or functional (+1). The model will be part of a larger quality control and monitoring system, helping engineers make informed decisions about wafer replacement.

2.2 Existing System

Currently, wafer inspection in semiconductor manufacturing may involve manual checking, rule-based systems, or basic threshold alerts based on sensor data. These methods are often inefficient and prone to human error, leading to either excessive replacement of functioning wafers or missed identification of faulty ones.

2.3 Proposal System

The proposed system will use machine learning to automatically predict the condition of each wafer based on historical and real-time sensor data. The system will reduce the need for manual inspection, increase accuracy, and allow for early detection of faults, thereby reducing production downtime and material waste.

2.4 Tools Used

- **Programming Language**: Python
- **Matplotlib**, **Seaborn**, **and Plotly** For creating visual representations of data and model performance metrics.
- **Machine Learning Libraries**: Scikit-learn, imblearn
- **Data Processing**: Pandas, NumPy
- **Model Evaluation**: Cross-validation techniques, confusion matrix, accuracy, precision, recall
- **Deployment**: FastAPI for web integration, Docker for containerization. Deployed in AWS
- **Version Control**: Git
- **Data Storage**: MongoDB NoSQL databases, AWS S3 for large data storage.

2.5 Constraints

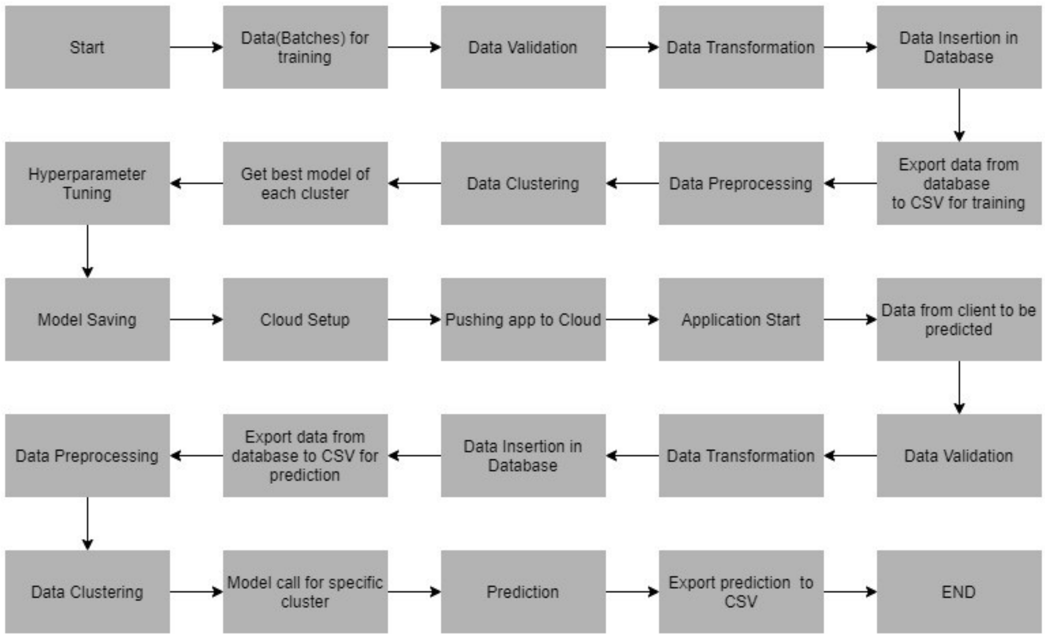
- The model must process real-time data with minimal latency.
- The system should be scalable to handle varying volumes of wafer sensor data.
- The system needs to be highly accurate to minimize false positives/negatives.
- Integration with existing manufacturing systems may require custom interfaces or APIs.

2.6 Assumptions

- Sufficient historical sensor data is available for training the model.
- The sensor data is preprocessed and normalized before input into the model.
- The environment in which the system will be deployed has the necessary infrastructure (e.g., servers, cloud services) to support real-time prediction and storage.

3.Design Details

3.1 Functional Architecture



The system architecture consists of the following components:

- **Data Ingestion**: Collects sensor data from wafers and stores it in a centralized database.
- **Data pre-processing**: Removed unwanted features, impute nan values apply data transformation methods for transform the data.
- **Model Training**: Trains the machine learning model on historical data to learn the patterns associated with faulty and functional wafers.
- **Prediction Engine**: Takes real-time sensor data as input and predicts the wafer's condition.
- **User Interface**: A dashboard for engineers to view predictions they will do batch prediction by providing batch file path.

3.2 Event Logs

The system will maintain logs for all significant events, including:

- Data ingestion (timestamp, data source)
- Model training (parameters, training duration, model accuracy)
- Predictions (timestamp, input data, prediction result)
- Errors (timestamp, error type, system state at error)

3.3 Error Handling

The system will include mechanisms for:

- Graceful handling of missing or corrupted data.
- Alerts for any prediction errors or inconsistencies.
- Fall-back mechanisms to use a previous model version if the current one fails.

3.4 Performance

Performance benchmarks will focus on:

- Prediction latency: Ensuring real-time processing.
- Model accuracy: Targeting a minimum accuracy of 95% on test data.
- Scalability: The system should scale to handle large volumes of sensor data without performance degradation.

3.5 Reusability

The design will emphasize modularity, allowing components like data pre-processing, model training, and prediction engines to be reused across different predictive maintenance scenarios. The system will also support adding new sensor data types with minimal code changes.

4. Conclusion

The Wafer Fault Detection System is poised to significantly enhance the efficiency of semiconductor manufacturing by automating the detection of faulty wafers. The use of machine learning ensures high accuracy and scalability, making it a robust solution for the industry. This document has outlined the high-level design, covering the purpose, scope, architecture, and key components of the system. Future steps will involve detailed design, implementation, testing, and deployment.