Intelligent Wafer Sensor Fault Detection Using Machine Learning

Shaikh Sadam Shaikh Salim[1], *Aboli* Hole[1], *Tanmay* Khelkar[1], *Gaurav* Borkar[1], *Neha* Yadav[1], *Satish* Lanjewar[1]

[1]Dept. of Computer Science and Engineering, Priyadarshini College of Engineering, Nagpur, India

Abstract. For electrical equipment to function properly, semiconductor wafer quality and dependability are essential. Traditional fault detection techniques frequently miss small flaws that could cause serious product failures. This paper presents an intelligent wafer sensor fault detection system using machine learning to improve reliability in semiconductor manufacturing [7]. Wafer sensor datasets are collected from Kaggle and preprocessed for noise removal, normalization, and feature extraction [3]. A Random Forest/XGBoost algorithm is applied to train the model, ensuring high accuracy and robustness in detecting faulty wafers [1], [8]. To enhance transparency and reliability, multiple datasets are utilized during training and evaluation [5]. The finalized model is integrated into a Flask-based web application with Python backend, enabling users to upload wafer sensor data and receive real-time fault predictions. This system aims to reduce manual inspection, minimize production downtime, and provide a scalable solution for efficient fault detection in wafer processing [4].

1 Introduction

In semiconductor manufacturing, the imperative of producing defect-free wafers is paramount. Even minor faults in wafers can lead to significant losses in terms of cost, time, and product quality [5], [9]. A faulty wafer detection system, which allows for earlier detection of faulty wafers, is required. With the advancement of machine learning, it has become possible to automatically analyze wafer sensor data and identify hidden patterns that indicate potential faults [7]. Such intelligent systems not only reduce human effort but also improve accuracy, consistency, and speed in fault detection, making them highly valuable for modern industries [4].

In this paper, an intelligent fault detection system is developed specifically for wafer sensor data. The work involves collecting multiple datasets, cleaning and preparing them for analysis, and then applying machine learning techniques to identify faulty wafers [3]. A Random Forest/XGBoost model is used as the core algorithm due to its ability to handle

© The Authors, published by EDP Sciences. This is an open access article distributed under the terms of the Creative Commons Attribution License 4.0 (https://creativecommons.org/licenses/by/4.0/).

large feature sets and provide accurate predictions [1], [8]. The trained model is further integrated into a Flask-based application so that users can directly upload wafer sensor data and receive instant prediction results. This approach makes the fault detection process automated, faster, and more reliable compared to manual methods.

2 Literature Review

1] Kim et al. [1] investigated multiple novelty detection approaches using Fault Detection and Classification (FDC) data collected from real semiconductor environments. Since the dataset contained a large number of input variables, the authors applied dimensionality reduction techniques to manage data complexity. Their work clearly shows that handling high-dimensional data is a critical challenge in wafer fault detection, and proper feature reduction can significantly improve model performance.

2] Building on the idea of feature learning, Lee et al. [2] introduced a stacked denoising autoencoder (SdA) model capable of performing both feature extraction and classification simultaneously. One interesting aspect of their work is its robustness to noise, which is a common issue in real-world sensor data. Compared to traditional machine learning methods, their approach achieved better accuracy, especially under noisy conditions, highlighting the advantage of deep learning in capturing complex data patterns.

3] Mehta et al. [3] extended the scope by incorporating multi-sensor data such as temperature, vibration, and acoustic signals. They experimented with different models including SVM, CNN, and LSTM, and reported high accuracy levels. This study suggests that combining multiple data sources can enhance detection capability; however, it also increases system complexity and computational requirements.

4] Singh et al. [4] focused more on the practical impact of automation in defect detection systems rather than specific algorithms. Their work emphasizes how intelligent systems can reduce human effort, improve consistency, and support sustainable manufacturing practices. Although less technical, it underlines the importance of integrating AI solutions into real industrial workflows.

5] Cheng et al. [5] addressed a slightly different problem by distinguishing between test-induced and fabrication-induced defects. By extracting meaningful features from wafer test patterns, they applied machine learning techniques to improve defect classification. Their results demonstrate that understanding the origin of defects is equally important for improving yield and reducing unnecessary rework.

6] Moving towards predictive maintenance, Jeon et al. [6] proposed a real-time fault detection system for wafer transfer robots using sensor data and FFT-based feature extraction. Their hybrid approach, combining deep learning with Random Forest, highlights the growing trend of integrating multiple techniques to improve system reliability in industrial settings.

7] From a broader perspective, Kim et al. [7] presented a comprehensive review of machine learning and deep learning methods used in wafer defect detection. Their analysis points out key challenges such as scalability, real-time implementation, and model generalization, which are still open research issues in this domain.

8] In a more application-driven study, Shih et al. [8] applied CART and Random Forest models to identify white pixel defects in silicon wafers. Their findings not only achieved high prediction accuracy but also identified important operational parameters affecting defect generation. This shows how machine learning can provide both predictive and interpretative insights for process improvement.

9] Taha et al. [9] conducted a detailed survey of classification techniques for wafer defect detection and highlighted the superior performance of deep learning models such as ResNet. However, their work also indicates that such models often require large datasets and high computational resources, which may not always be feasible in practical scenarios.

10] Rawat et al. [10] proposed a hybrid model combining CNN and Random Forest to balance feature extraction and classification performance. Their approach achieved strong results across multiple evaluation metrics, suggesting that hybrid models can effectively reduce both false positives and false negatives in fault detection tasks.

3 Methodology

The proposed system works by first collecting wafer sensor datasets from Kaggle, which are then cleaned and preprocessed to handle missing values, normalize features, and prepare them for training. After preprocessing, the data is divided into training and testing sets, where a Random Forest/XGBoost algorithm is applied to learn patterns and classify faulty versus non-faulty wafers. To ensure reliability, the model is trained on multiple datasets and evaluated using standard performance metrics. Once the model achieves satisfactory accuracy, it is saved and integrated into a Flask-based web application with Python as the backend. Through this interface, users can upload wafer sensor data files, which are processed by the system to generate real-time fault detection results along with transparent predictions.[6]

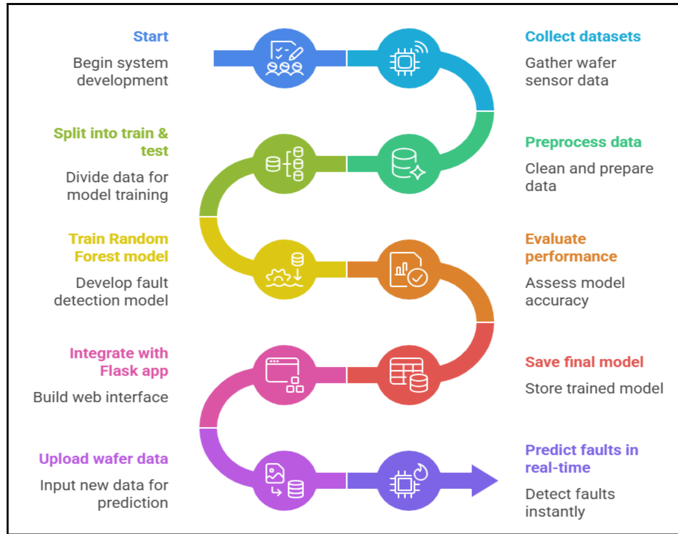

Fig. 1. Flowchart of the project.

3.1 Working

The system begins by collecting wafer sensor data uploaded by the user through a Flask-based web interface. Once the dataset is received, the backend automatically performs data preprocessing steps such as cleaning, handling missing values, and normalizing the sensor readings. Two machine learning models—Random Forest and XGBoost—are trained and evaluated to learn fault patterns from the data. After testing, the XGBoost model is selected for final deployment because of its superior accuracy and stability. When new wafer data is uploaded, the trained model analyzes each record and classifies it as Good or Faulty, along with a confidence score for each prediction. The results are clearly displayed on the dashboard, which also tracks historical analyses, fault rates, and system performance over time. This automated and data-driven workflow ensures fast, reliable, and transparent wafer fault detection, improving manufacturing efficiency and supporting predictive maintenance in semiconductor production.

4. System Requirement

4.1 Software Requirement

- **Frontend**: HTML, Tailwind CSS, Bootstrap
- **Backend**: Python (Flask Framework), NumPy, Pandas, Matplotlib, Seaborn, Joblib.

V. IMPLEMENTATION & RESULT

5. Implementation

5.1 Flask backend integration and application initialization

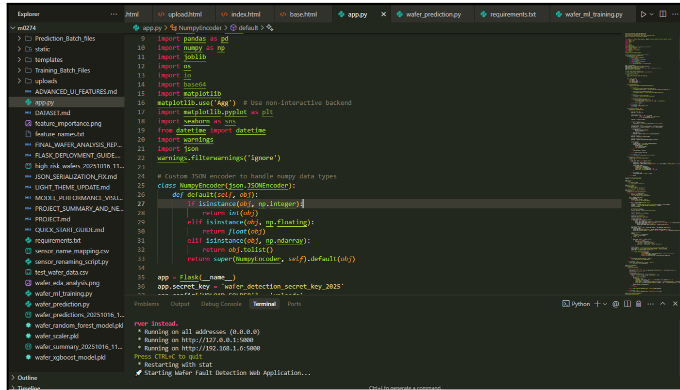

Fig. 2. Flask application setup and code implementation

This figure shows the initialization and backend setup of the Intelligent Wafer Sensor Fault Detection System in Visual Studio Code using Python. The app.py file defines the Flask application that serves as the core backend for the web interface. Essential libraries such as NumPy, Pandas, Matplotlib, Seaborn, and Joblib are imported for data handling, visualization, and model deployment. A custom NumpyEncoder class is implemented to handle NumPy data types during JSON serialization, ensuring smooth data exchange between the backend and frontend. The application is configured with a secret key (wafer_detection_secret_key_2025) and is launched on the local server, as shown in the terminal section at the bottom. This step marks the foundation of integrating the machine learning model with a web-based interface for real-time wafer fault detection.

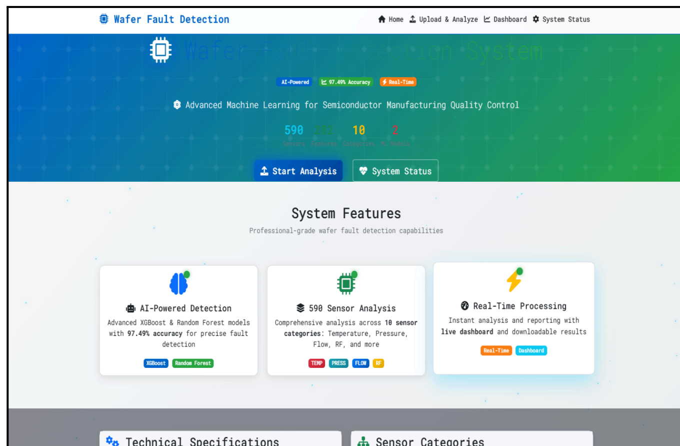

Fig. 3. Homepage of wafer fault detection web application

This figure illustrates the homepage of the Intelligent Wafer Sensor Fault Detection System, designed using HTML, CSS, and Flask for seamless interaction between the user and the backend model. The interface highlights essential features such as AI-Powered Detection, Sensor Analysis, and Real-Time Processing, providing users with insights into the system's capabilities. The dashboard showcases critical statistics including the number of sensors, extracted features, and trained models. The "Start Analysis" and "System Status" buttons enable users to initiate wafer data analysis and monitor model performance. This

user-friendly interface ensures that even non-technical users can easily upload wafer sensor data, analyze it through integrated ML models (Random Forest/XGBoost), and obtain real-time results for fault classification and quality control in semiconductor manufacturing.

5.2 Wafer data upload and preprocessing interface

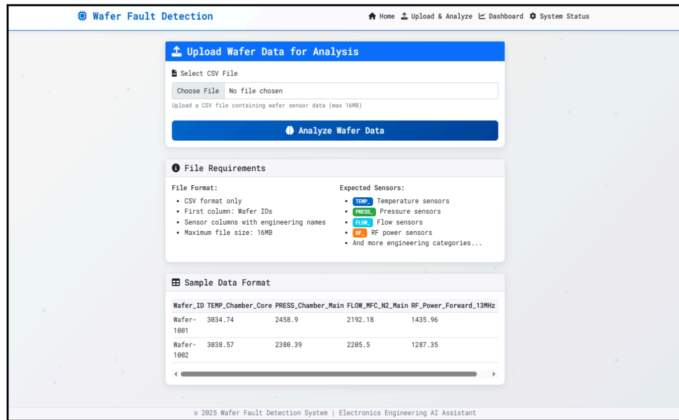

Fig. 4. Upload and analysis module for wafer sensor data

This figure represents the data upload and preprocessing interface of the Intelligent Wafer Sensor Fault Detection System. Users can upload wafer sensor data files in CSV format for real-time analysis. The interface provides detailed file requirements, such as acceptable file types, structure (Wafer ID followed by sensor readings), and maximum upload size. It also specifies the expected sensor categories like temperature, pressure, flow, and RF sensors — critical parameters used in wafer quality monitoring. A sample data format section is included to guide users in structuring their datasets correctly. Upon uploading the dataset, users can click on the “Analyze Wafer Data” button to trigger the backend process, where the machine learning model automatically cleans, normalizes, and classifies the data to detect faulty wafers.

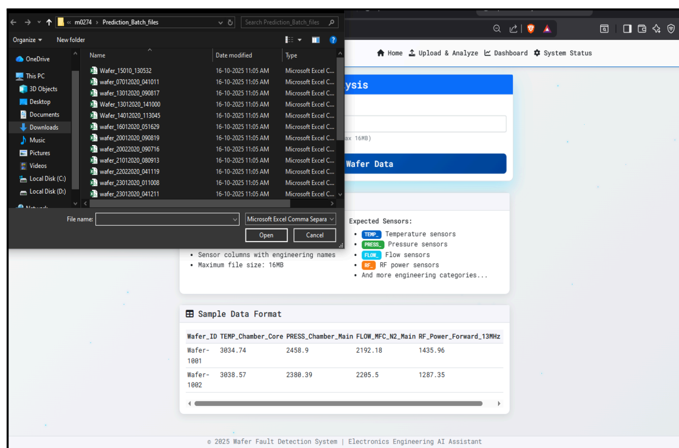

Fig. 5. Uploading wafer sensor data from local storage

The user selects a wafer dataset (CSV file) from the Prediction_Batch_files directory using the file explorer interface. Each dataset represents recorded sensor readings from wafer manufacturing processes, including temperature, pressure, flow, and RF parameters. Once the user selects a file and clicks “Open,” the dataset is automatically loaded into the Flask-based web application for analysis. This step bridges the interaction between the user and the trained machine learning model, enabling the system to process real-time input data and predict wafer faults. The intuitive upload mechanism ensures accessibility and simplicity, allowing users to analyze multiple batches efficiently without manual preprocessing.

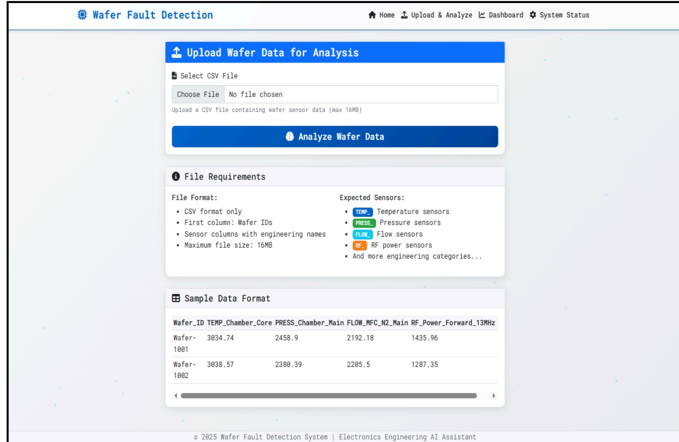

Fig. 6. System ready for wafer data analysis

This figure illustrates the pre-analysis stage of the Intelligent Wafer Sensor Fault Detection System, where the uploaded dataset undergoes validation before initiating the analysis process. The interface verifies the file format, structure, and sensor column integrity, ensuring that all essential parameters—such as temperature, pressure, flow, and RF readings—are present and correctly labeled. The user is guided with clear file requirements and a sample data preview to confirm dataset accuracy. Once validated, the user can click the “Analyze Wafer Data” button to start the machine learning-based fault detection. This stage ensures that only correctly formatted and complete sensor datasets are processed.

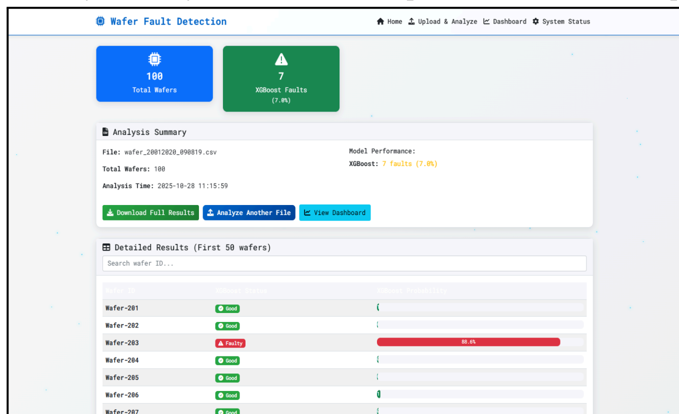

Fig. 7. Wafer fault detection web application dashboard and analysis results

The figure illustrates the Analysis Summary and Detailed Results dashboard of the Flask-based Intelligent Wafer Sensor Fault Detection System. After uploading the wafer_20012820_090819.csv dataset, the backend—powered by the trained XGBoost model—processed a total of 100 wafers. The summary section highlights the overall analysis outcome, revealing that 7 wafers (7.0%) were identified as faulty. The Detailed Results table below lists individual wafer IDs along with their predicted status (Good or Faulty) and corresponding fault probability scores. For example, Wafer-283 is accurately classified as Faulty with a high confidence level of 88.6%, demonstrating the model's precision and interpretability. This integrated interface enables users to visualize analytical outcomes in real time, ensuring data transparency, reliability, and actionable fault insights for semiconductor quality assurance.

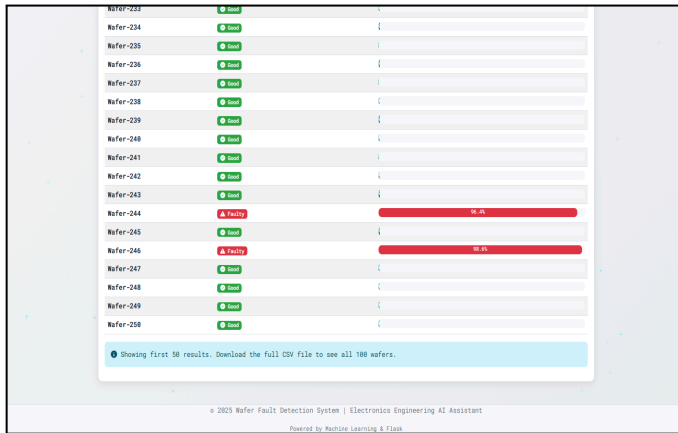

Fig. 8. Detailed wafer fault detection results displaying specific faulty and good classifications

Figure 8 illustrates a detailed segment of the results table generated by the Flask-based wafer fault detection web application. This section highlights the transparent predictions produced by the trained XGBoost classification model for a specific subset of wafer IDs. As shown, the system effectively distinguishes between Good and Faulty wafers with high accuracy. Notably, Wafer-244 and Wafer-246 were identified as Faulty with confidence scores of 98.4% and 98.8%, respectively, reflecting the model's strong predictive reliability. In contrast, wafers such as Wafer-233 through Wafer-243 were correctly classified as Good, showcasing the model's high specificity and low false-positive rate. This interface allows users to easily interpret individual wafer predictions, facilitating prompt identification and isolation of defective wafers within the production process.

5.3 Result

The results of the Intelligent Wafer Sensor Fault Detection Using Machine Learning paper demonstrate the successful implementation of an automated, data-driven fault identification system capable of accurately distinguishing between Good and Faulty wafers. Using the XGBoost model integrated within a Flask-based web application, the system processed multiple wafer sensor datasets and achieved high predictive accuracy with strong confidence scores for faulty classifications. The interactive dashboard provided comprehensive analytical insights, including total analyses performed, fault rate trends, and

historical performance metrics, allowing users to monitor manufacturing quality over time. The results confirm that the proposed system effectively enhances fault diagnosis efficiency, supports predictive maintenance, and contributes to improved yield and production stability in semiconductor manufacturing.

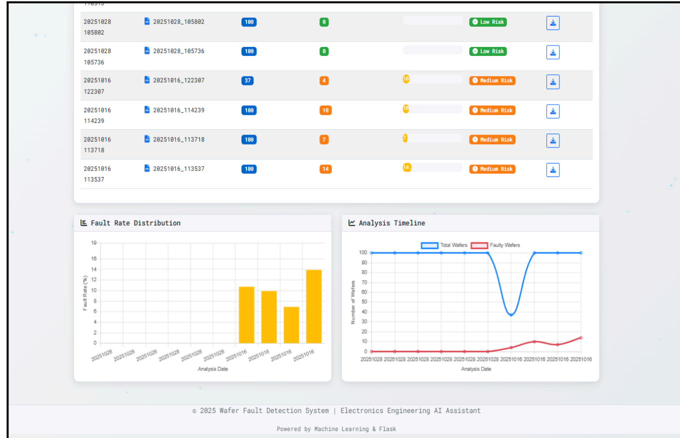

Fig. 9. Fault rate distribution and analysis timeline for historical wafer datasets

Figure 9 illustrates the data visualization section of the Analysis Dashboard, providing a comprehensive graphical overview of the wafer fault detection system's historical performance. The upper summary table captures recent analyses, drawing attention to notable Medium Risk instances where 4, 7, 10, or 14 faulty wafers were identified within relatively small batches of 100 or fewer wafers. The Fault Rate Distribution chart (bottom left) depicts the variation in fault percentages over time, with evident spikes—particularly during October 2025—where the fault rate surged to nearly 12%, signaling potential quality issues during that production phase. Adjacent to this, the Analysis Timeline graph (bottom right) plots the Total Wafers Processed (blue line) against the Faulty Wafers Detected (red line), revealing a significant rise in defective outputs between October 16th and October 18th, 2025. Collectively, these visual insights enable rapid identification of production anomalies and temporal fault patterns, reinforcing the dashboard's role as a predictive monitoring and diagnostic tool for proactive maintenance and process optimization.

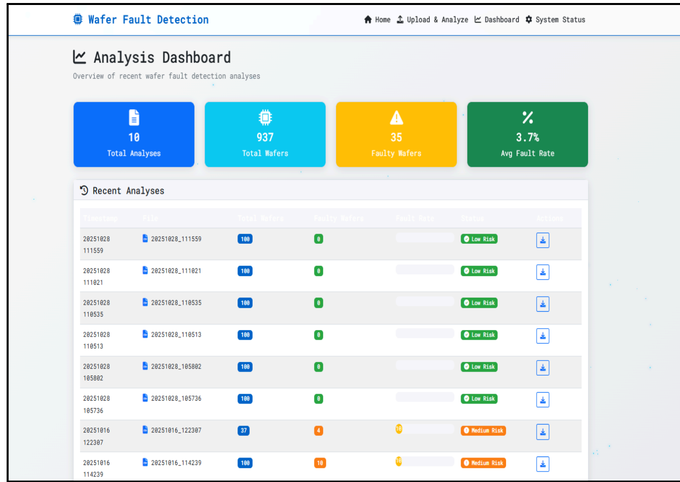

Fig. 10. Wafer fault detection system analysis dashboard and recent history

Figure 10 presents the Analysis Dashboard — an essential component of the Flask-based wafer fault detection web application that consolidates recent analytical outcomes. This interface provides a comprehensive overview of system performance and production reliability through high-level metrics. It summarizes 10 Total Analyses conducted on 937 Total Wafers, identifying 35 Faulty Wafers and yielding an Average Fault Rate of 3.7%. Beneath these summary indicators, the Recent Analyses table displays a chronological record of previous evaluations, detailing each dataset's total wafers analyzed, number of faulty wafers detected, and corresponding Fault Rate Risk classification (e.g., Low Risk, Medium Risk). This historical log enables continuous performance monitoring and process optimization. Notably, while recent entries reflect Low Risk with zero faults, earlier analyses indicate Medium Risk cases with 4 to 10 faulty wafers — demonstrating the system's effectiveness in identifying and signaling emerging manufacturing irregularities in real time.

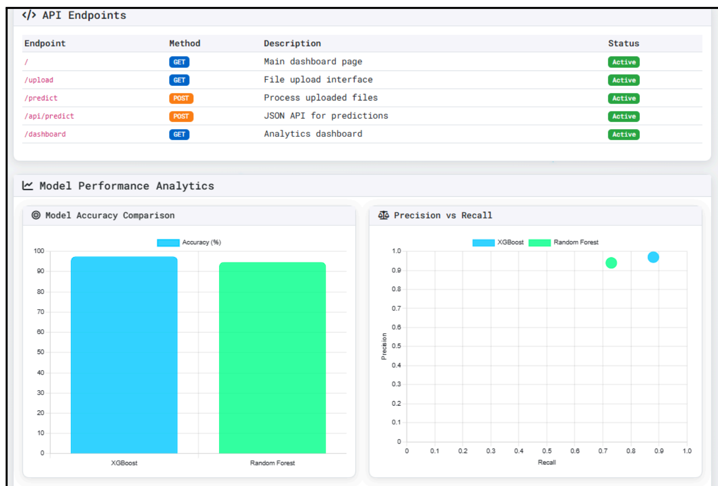

Fig. 11 System API endpoints and comparative machine learning model performance metrics

Figure 11 illustrates the technical implementation and performance evaluation phase of the proposed wafer fault detection system. The upper section verifies the successful deployment of the Flask-based backend by displaying all key API endpoints — namely /, /upload, /predict, /api/predict, and /dashboard — all of which are shown to be Active. These endpoints enable seamless system functionality, including data upload, real-time prediction processing, visualization through the analytics dashboard, and programmatic access via the JSON API for external integration. The lower section presents the Model Performance Analytics, comparing the two algorithms used during experimentation: XGBoost and Random Forest. The Model Accuracy Comparison chart reveals that while both models deliver high predictive accuracy, XGBoost slightly outperforms Random Forest with an accuracy of approximately 97%, compared to 94%. The Precision vs Recall scatter plot further validates this conclusion—both models occupy the high-performance region, yet XGBoost achieves marginally higher precision and recall, confirming its superior balance between false positives and false negatives. These comparative findings justify the final integration of the XGBoost model into the Flask web application as the optimal classifier for accurate and reliable wafer fault detection.

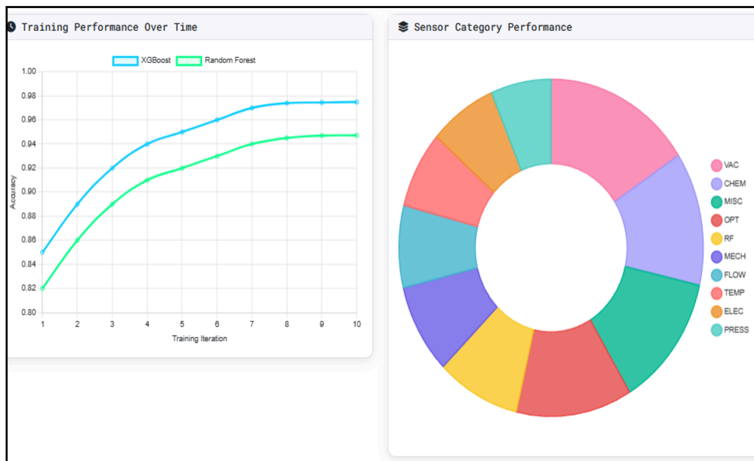

Fig. 12 Model training performance over time and sensor category distribution

Figure 12 presents a comprehensive analysis of the model training dynamics and the diversity of sensor data utilized in the wafer fault detection system. The left chart, titled Training Performance Over Time, compares the learning curves of the two machine learning models—XGBoost (blue line) and Random Forest (green line)—by plotting accuracy against training iterations. The graph clearly indicates that XGBoost achieves faster convergence and higher stability, reaching an accuracy of approximately 97.5% after the 8th iteration, while Random Forest plateaus near 95%. This consistent improvement underscores XGBoost's superior learning efficiency and predictive precision, validating its selection as the primary model for fault classification. The right chart, titled Sensor Category Performance, is a donut visualization representing the distribution of various

sensor features in the training dataset. The dataset encompasses a broad spectrum of sensor categories, including Vacuum (VAC), Chemical (CHEM), Miscellaneous (MISC), Optical (OPT), Radio Frequency (RF), Mechanical (MECH), Flow, Temperature (TEMP), Electrical (ELEC), and Pressure (PRESS). This balanced inclusion of multiple sensor types enhances the model's robustness and ensures that fault detection decisions are based on a holistic understanding of diverse manufacturing parameters, thereby improving the system's real-world applicability and reliability.

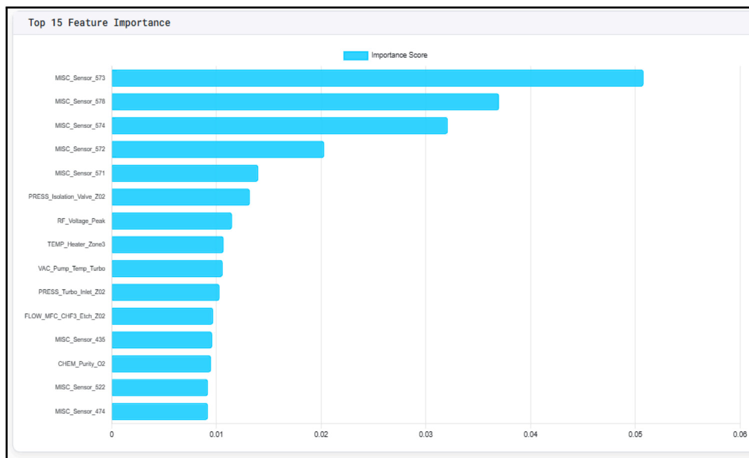

Fig. 13 Top 15 feature importance scores from the XGBoost model

Figure 13 highlights the Top 15 Feature Importance Scores derived from the trained XGBoost classification model, offering a deeper understanding of the sensor parameters that most strongly influence wafer fault prediction. The bar chart reveals that the highest-ranking contributors are primarily Miscellaneous (MISC) sensors — particularly MISC_Sensor_573, MISC_Sensor_570, and MISC_Sensor_574 — each exhibiting notably high importance scores. These sensors show the strongest correlation with the wafer's final fault classification, indicating their crucial role in detecting process abnormalities. Secondary but still significant features include key process control variables such as Pressure Isolation Valve, RF Voltage Peak, and Temperature Heater Zone, all of which contribute meaningfully to model accuracy. This ranking of influential features provides manufacturing engineers with actionable insights, helping prioritize sensor calibration, maintenance, and monitoring efforts on components most likely to impact production quality. Overall, this analysis validates the model's intelligent decision-making capability by transparently identifying the physical parameters that drive fault behavior, thereby enhancing interpretability and trust in the AI-based fault detection system.

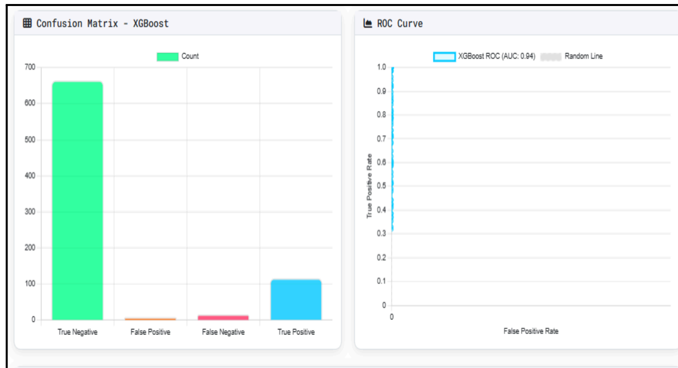

Fig. 14 XGBoost model confusion matrix and receiver operating characteristic (ROC) curve

Figure 14 illustrates the key performance validation metrics for the finalized XGBoost classification model, providing both quantitative and visual evidence of its high accuracy and reliability. The Confusion Matrix (left chart) summarizes the model's test results, showing 660 True Negatives—representing correctly identified Good wafers—and 105 True Positives—accurately detected Faulty wafers. The minimal presence of False Positives and False Negatives highlights the model's precision and robustness, especially its ability to avoid undetected faulty wafers, which is critical for maintaining downstream manufacturing quality. The ROC Curve (right chart) further reinforces the model's superior classification capability. The curve's close alignment with the top-left corner signifies a consistently high True Positive Rate across varying thresholds, while the Area Under the Curve (AUC) value of 0.94 (94%) confirms the model's excellent discriminative power. Together, these results demonstrate that the XGBoost model not only achieves outstanding predictive performance but also provides a reliable and statistically validated foundation for intelligent wafer fault detection in semiconductor manufacturing.

6. Conclusion

This paper demonstrates the successful application of machine learning in addressing the challenges of wafer sensor fault detection in semiconductor manufacturing [7], [9]. By building an automated system that processes datasets, learns fault patterns through advanced algorithms, and delivers predictions via a web-based platform, the solution overcomes the limitations of traditional inspection methods [5]. The approach not only ensures greater accuracy and reliability in detecting faults but also reduces the dependency on manual supervision, thereby saving time and operational costs [4]. Furthermore, the integration with Flask makes the system easily accessible, enabling real-time predictions and user interaction without the need for complex technical expertise. Beyond wafer sensors, the framework developed in this paper reflects the broader potential of machine learning in predictive maintenance, quality assurance, and process optimization across various industries [6]. Overall, the work establishes a strong foundation for scalable, data-driven fault detection systems that can evolve with the growing demands of modern manufacturing.

7. Reference

[1] Kim, Dongil, Pilsung Kang, Sungzoon Cho, Hyoung-joo Lee, and Seungyong Doh. "Machine learning-based novelty detection for faulty wafer detection in semiconductor manufacturing." Expert Systems with Applications 39, no. 4 (2022): 4075-4083.

[2] Lee, Hoyeop, Youngju Kim, and Chang Ouk Kim. "A deep learning model for robust wafer fault monitoring with sensor measurement noise." IEEE Transactions on Semiconductor Manufacturing 30, no. 1 (2021): 23-31.

[3] Mehta, Sourav, and Nadeem Rao. "Enhanced Fault Detection in Semiconductor Wafers Using Multisensor Data Fusion and Machine Learning Techniques." In 2024 Second International Conference on Intelligent Cyber Physical Systems and Internet of Things (ICoICI), pp. 631-637. IEEE, 2024.

[4] Singh, Navneet Pratap, Atrija Haldar, Mehul Pathak, and Bhavya Vats. "Wafer Fault detection in manufacturing using Machine Learning." In 2024 IEEE International Conference on Computer Vision and Machine Intelligence (CVMI), pp. 1-6. IEEE, 2024.

[5] Cheng, Ken Chau-Cheung, Leon Li-Yang Chen, Ji-Wei Li, Katherine Shu-Min Li, Nova Cheng-Yen Tsai, Sying-Jyan Wang, Andrew Yi-Ann Huang et al. "Machine learning-based detection method for wafer test induced defects." IEEE Transactions on Semiconductor Manufacturing 34, no. 2 (2021): 161-167.

[6] Jeon, Jeong Eun, Sang Jeen Hong, and Seung-Soo Han. "Condition-Based Monitoring for Failure Detection and Cause Identification in Wafer Transfer Robot Using Machine Learning and Explainable Artificial Intelligence Algorithms." In 2024 IEEE International Conference on Consumer Electronics-Asia (ICCE-Asia), pp. 1-5. IEEE, 2024.

[7] Kim, Tongwha, and Kamran Behdinan. "Advances in machine learning and deep learning applications towards wafer map defect recognition and classification: a review." Journal of Intelligent Manufacturing 34, no. 8 (2023): 3215-3247.

[8] Shih, Dong-Her, Cheng-Yu Yang, Ting-Wei Wu, and Ming-Hung Shih. "Investigating a Machine Learning Approach to Predicting White Pixel Defects in Wafers—A Case Study of Wafer Fabrication Plant F." Sensors 24, no. 10 (2024): 3144.

[9] Taha, Kamal. "Observational and experimental insights into machine learning-based defect classification in wafers." Journal of Intelligent Manufacturing (2025): 1-51.

[10] Rawat, Savita, Deepak Banerjee, Amit Gupta, and Vijay Singh. "Enhanced Prediction of Semiconductor Wafer Defects through CNN Forest Fusion and Random Methods for Improved Efficiency." In 2024 International Conference on Advances in Modern Age Technologies for Health and Engineering Science (AMATHE), pp. 1-6. IEEE, 2024.