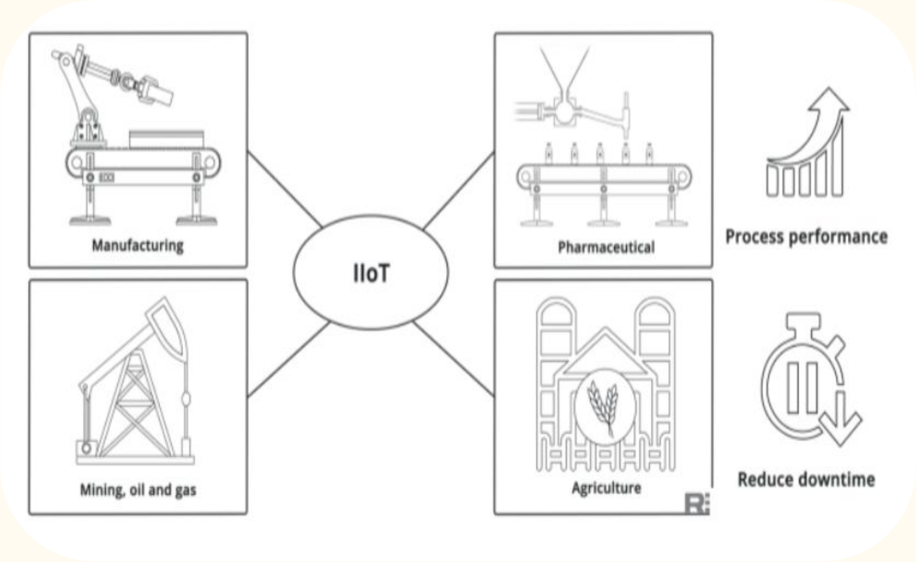
Smart Manufacturing with Predictive Quality and Defect Detection using IIOT

Group 3:
Sameerjeet Chhabra,
Vishavjit Singh Khinda,
Jose Moreno-Virgil

Agenda:

- Problem Overview
- System Architecture
- Layer by Layer Breakdown
- Results
- Conclusion
- Live Demonstration

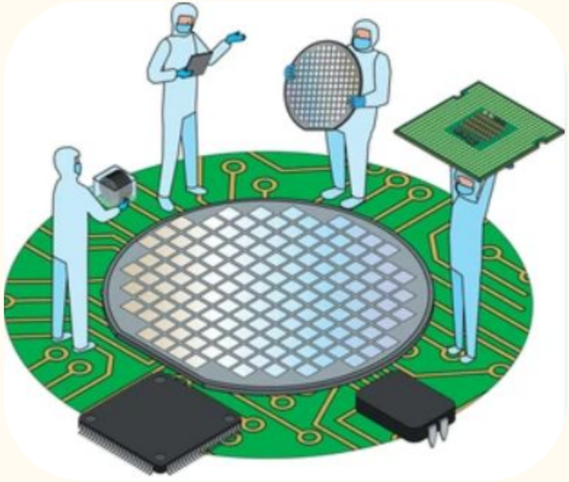


Problem Overview & Objective:

Real-World Problem:

Semiconductor wafer manufacturing faces critical quality challenges

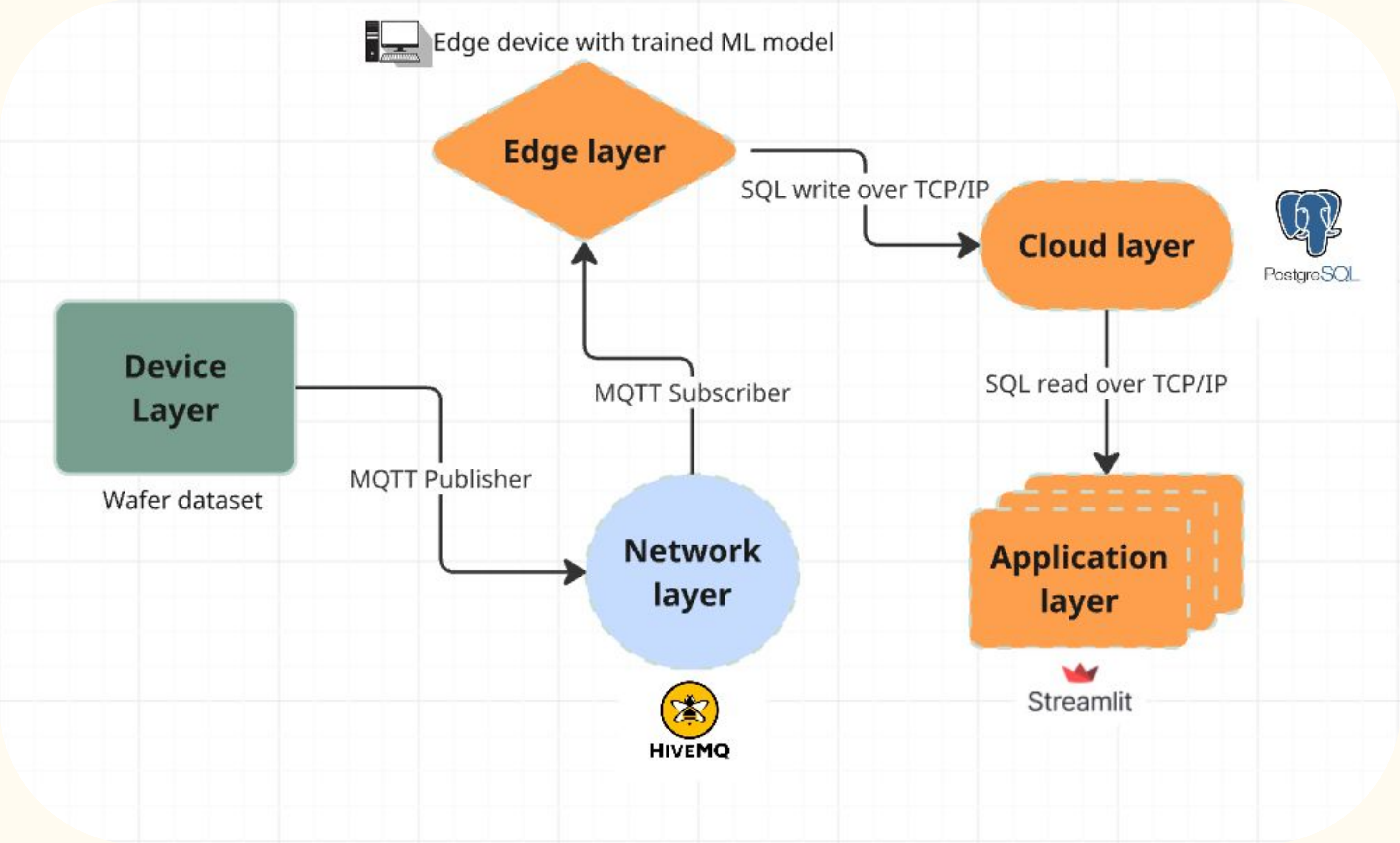
- Defects detected too late in production cycle, leading to wastage of material and resources, causing huge revenue loss.
- Manual inspection is slow, inconsistent and labor-intensive.
- Lack of real-time visibility into production line health.



Project Objectives:

- Implement end to end IIOT architecture to make the wafer production process smart with data-driven decision making .
- Deploy a trained ML model at edge to do wafer quality prediction in real-time (Edge analytics) before optical/e-beam inspection.
- Develop an interactive dashboard with production line status , real-time alert display and acknowledgement system.
- Demonstrate quality improvement, cost savings.

System Architecture Overview



Layer	Technology used
Application	Streamlit dashboard: Real-time process monitoring, alerts and analytics
Cloud	PostgreSQL database: Tables with time-series sensor data, alerts, production line status
Edge	Python gateway: XGBoost trained ML model, real-time wafer quality prediction, alert generation, send data to database
Network	MQTT protocol (HiveMQ): Lightweight, publish-subscribe messaging
Physical	Wafer Dataset simulator: 3 production lines (Lithography, Etching, Deposition) streams process data

Data Flow:

Dataset simulator → MQTT → Edge ML Processing → Cloud Database → Real-time Dashboard

Physical Device Layer

Arizona State University

- Used semiconductor wafer dataset (4219 wafers) from Kaggle for this project.
- Sorted the dataset by timestamp and divided the dataset into 3 parts
- Oldest 60% of data (2531 wafers) was used to train the ML model.
- Next 20% of data (844 wafers) to validate/test the trained ML model.
- Recent 20% of data (844 wafers)was used in streaming/simulation using python script.
- Next the streaming data divided into 3 production lines depending on tool type.

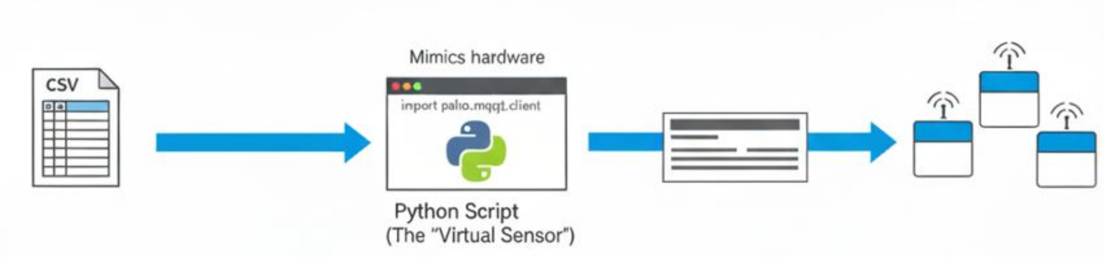
- data_simulation.csv
- data_train.csv
- data_validation.csv

Process_ID	Timestamp	Tool_Type	Wafer_ID	Chamber_Ter	Gas_Flow_Rat	RF_Power	Etch_Dept	Rotation_	Vacuum_P	Stage_Alig	Vibration_	UV_Expos	Particle_C	Defect	Join_Status
P1087	01-07-2025 00:00	Lithography	WAF90799	77.8243207	39.40402495	355.5155	538.2439	1446.394	0.437551	2.279001	0.014463	122.7394	108	0	Joining
P1270	01-07-2025 00:00	Etching	WAF62509	79.64036	53.16852339	379.3958	566.864	1245.579	0.488671	1.562643	0.010686	116.8369	259	0	Joining
P1102	01-07-2025 00:00	Etching	WAF87755	80.2001336	54.86533303	287.8643	441.8853	1494.25	0.513109	1.497935	0.00712	103.9026	800	0	Joining
P1614	01-07-2025 00:00	Etching	WAF58402	72.9006292	38.04743201	404.0273	486.8517	1557.893	0.524697	3.020189	0.00498	116.2112	407	0	Joining
P1466	01-07-2025 00:00	Lithography	WAF21066	84.5608902	57.99518849	317.4137	539.4488	1257.981	0.523497	0.396944	0.015819	110.8216	702	1	Non-Joining
P1130	01-07-2025 00:00	Deposition	WAF62092	78.873867	45.79913995	305.3594	439.4161	1696.002	0.52067	2.249753	0.016154	100.924	264	0	Joining
P1330	01-07-2025 00:00	Etching	WAF17008	82.9305707	53.00286892	320.5211	543.5892	1676.892	0.467081	2.915342	0.010306	106.1587	258	0	Joining
P1871	01-07-2025 00:00	Deposition	WAF27153	78.2524482	39.13710826	291.5288	388.0412	1308.372	0.4053	2.500453	0.006322	115.8376	147	0	Joining
P1252	01-07-2025 00:01	Deposition	WAF22279	75.6907378	54.50414161	367.88	747.5116	1888.711	0.480828	1.021195	0.016414	139.8958	297	0	Joining
P1474	01-07-2025 00:01	Lithography	WAF14474	78.6610904	70.84071274	300.5008	579.8646	1233.656	0.609697	0.68699	0.015078	139.0523	735	0	Joining
P1313	01-07-2025 00:01	Etching	WAF71274	74.6527243	54.86031602	262.3935	591.4513	1461.403	0.441875	1.194716	0.010419	128.4602	275	0	Joining
P1160	01-07-2025 00:01	Lithography	WAF51534	73.2969796	59.11213104	344.7014	634.2476	1568.734	0.472307	1.767846	0.020008	129.8144	622	0	Joining
P1955	01-07-2025 00:01	Etching	WAF14355	80.2083337	50.75498009	214.142	624.7094	1636.987	0.520017	1.565729	0.013059	136.4802	413	0	Joining

How we ensured realism of device layer?

- For each part: training, validation and streaming, different data was used. So, there is no data leakage.
- Industries also follow this approach: use their past year data to make prediction on current data.

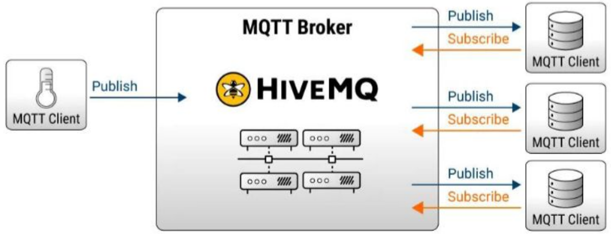
- The “Virtual Sensor”:
  - Python script(device_publisher.py) functions as the MQTT publisher with 3 MQTT topics.
  - Iterates through (data_simulation.csv) to simulate real-time data generation (does not pass the target variable)
  - Frequency of data streaming is at a rate of 8 seconds for demonstration purpose.
  - Can be changed to mimic the frequency and structure of physical industrial hardware.



Network/Communication Layer

MQTT Protocol:

- Lightweight protocol ideal for IoT (minimal bandwidth)
- Publish-subscribe pattern enables scalable architecture
- 3 MQTT topics:
- Python dictionary converted into JSON payload format for structured simulated sensor data




```python
MQTT_TOPICS = {
    'Lithography': 'factory/line1/lithography',
    'Etching': 'factory/line2/etching',
    'Deposition': 'factory/line3/deposition'
}
```

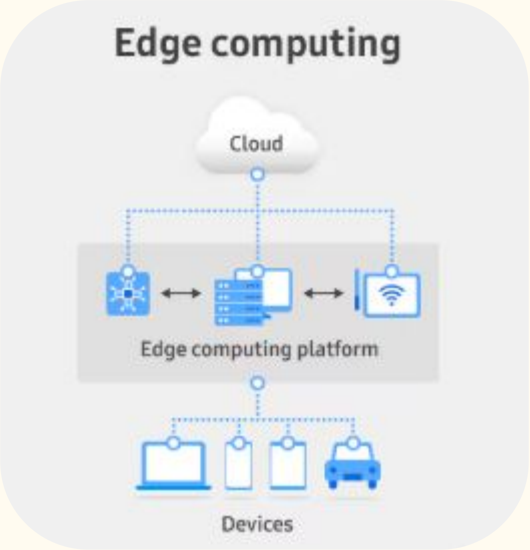

```json
{
  "wafer_id": "WAF12345",
  "production_line": "Lithography",
  "chamber_temperature": 245.3,
  "vacuum_pressure": 0.0023,
  "gas_flow_rate": 152.8,
  "rf_power": 485.2,
  "deposition_time": 185.4,
  "etch_rate": 42.7,
  "thickness": 1.85,
  "timestamp": "2025-11-30 19:45:32"
}
```

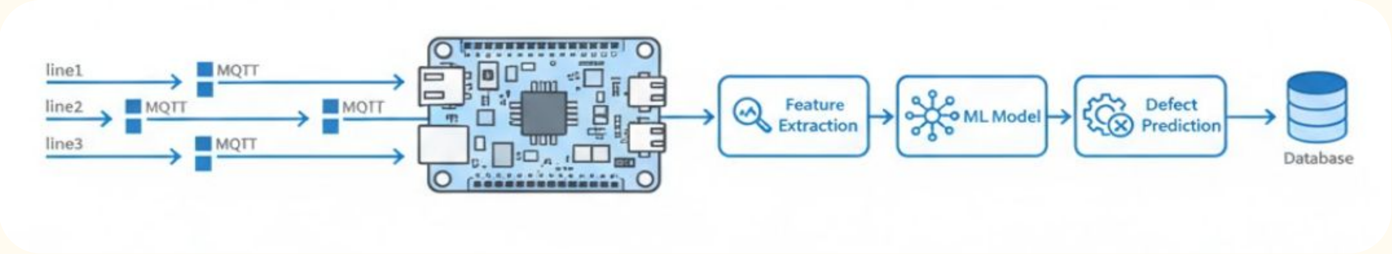
Edge/Gateway Layer

Edge Device: Python MQTT Gateway (edge_gateway.py) for edge computing and prediction analytics

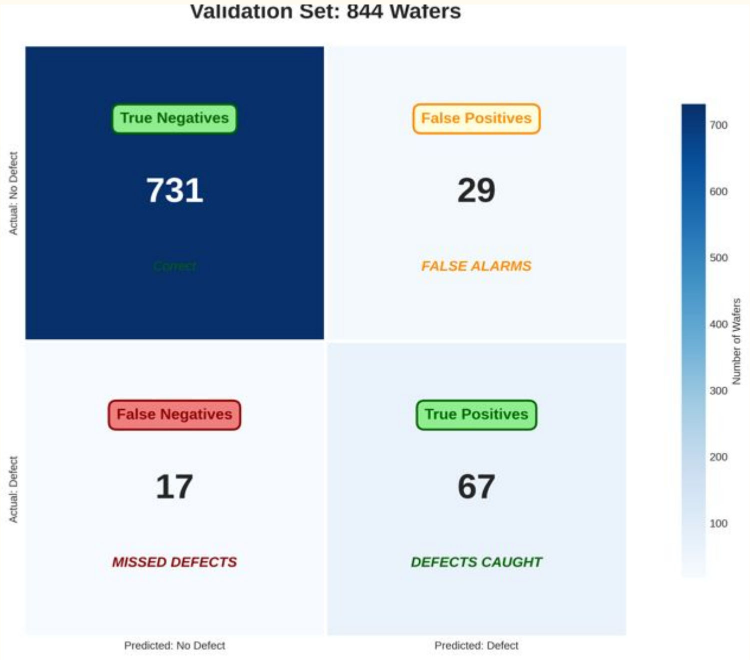
Function Implemented:

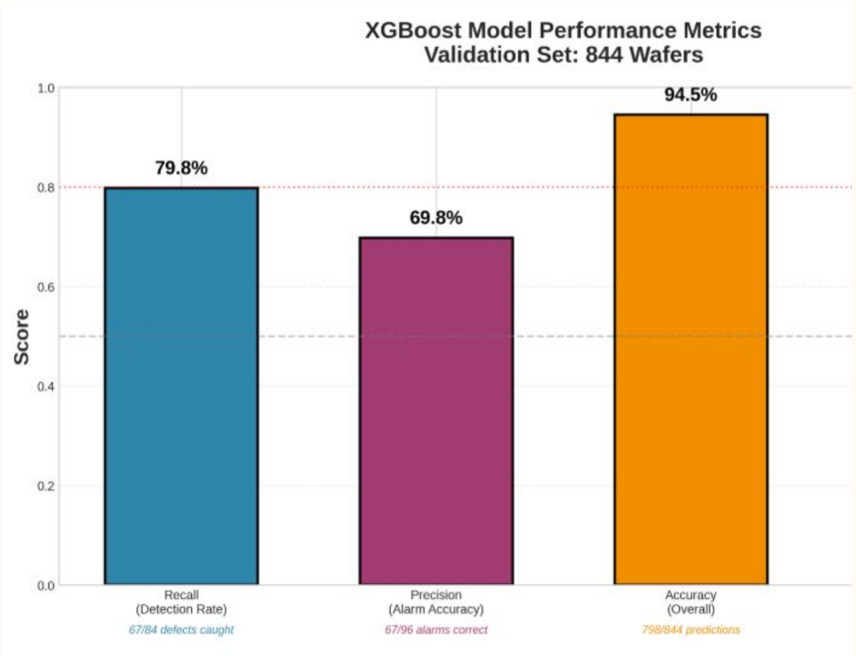
- Acts as MQTT subscriber and receive messages from 3 topics: line1, line2, line3
- Contains pre-trained ML model with ~80% recall rate
- Performs ML prediction using incoming simulated sensor data locally
- Generates defect probability score and alert for bad wafer
- Forwards process data to PostgreSQL database for storage over TCP/IP





ML Model Validation Results





Recall = TP/(TP+FN) = 67/(67+17) = 79.8% | Precision = TP/(TP+FP) = 67/(67+29) = 69.8%

Advantages of doing edge analytics

1. **Low Latency ⚡**

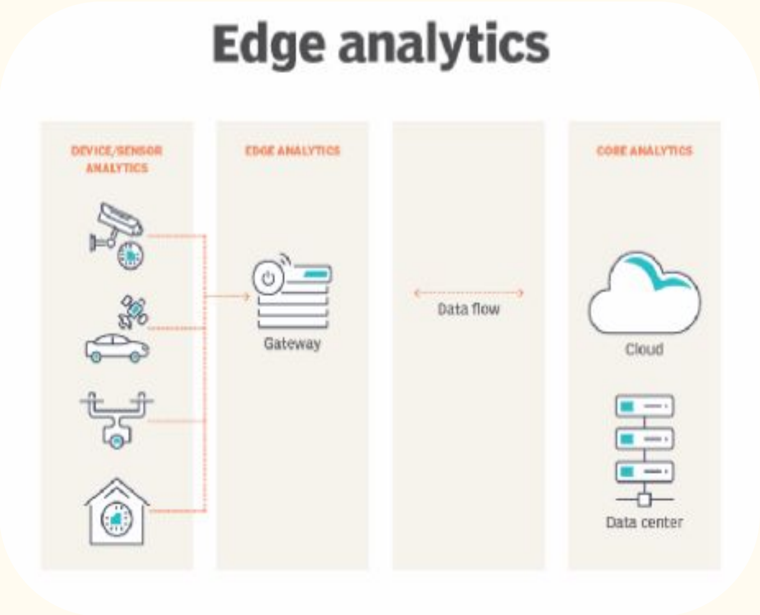
- Edge inference takes < 50ms
- Real-time defect detection for immediate action

2. **Bandwidth Efficiency 💾**

- Sending all raw sensor data to cloud wastes bandwidth
- Only predictions, important process parameters and alerts sent to cloud storage

3. **Reliability 🔒**

- Cloud connection loss stops predictions, if ML model is deployed in cloud layer
- Edge continues to work offline

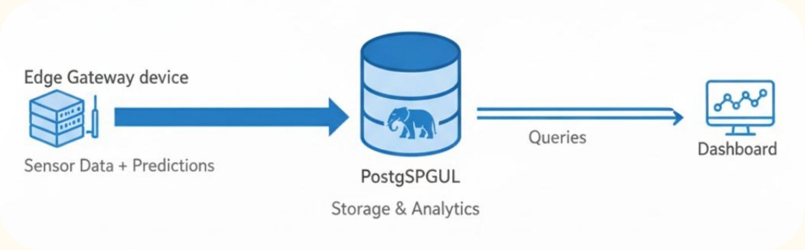


Cloud Platform Layer

Platform: PostgreSQL Database

Data Pipeline Architecture:

Edge Gateway ⟶ PostgreSQL Database ⟶ Dashboard



Database Storage:

- Time series data storage using 3 tables
- Table 1: Simulated sensor data
- Table 2: ML prediction results and alerts with defect probability
- Table 3: Production line status updates
- Analytics: Defect rates, production statistics, trend analysis

Application Layer

Streamlit

Arizona State University

Platform: Python-based Streamlit Dashboard (dashboard.py)

Visualization Features:

- Production line status display
- Real-time process parameter trends (Temperature, Pressure, Gas flow..)
- Active defect notifications on detecting bad wafer by edge ML model
- Alert management interface with one click acknowledgement
- Overall system statistics: Total wafer processed, Defect detected, Total alerts.

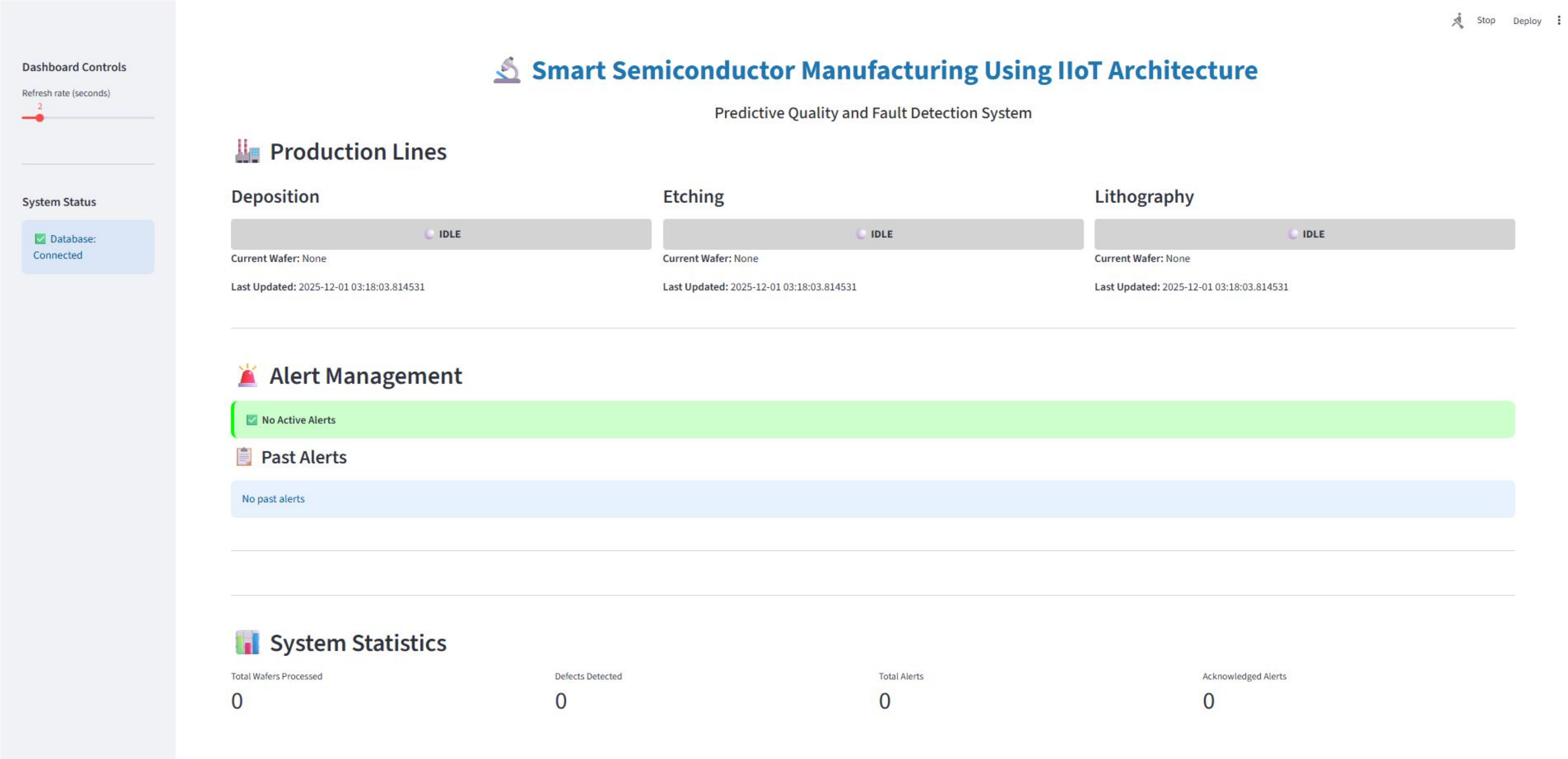
Data Retrieval:

- SQL queries to PostgreSQL database over TCP/IP
- 2-second refresh rate

User Interactions:

- Alert acknowledgment

Dashboard (Idle state)



Results, Challenges & Lessons Learned

Results:

- Successfully demonstrated the 5 layered IIOT architecture for smart manufacturing in semiconductor industry
- Edge ML inference achieved $<$50ms latency—critical for manufacturing where every second counts in preventing defective wafers from advancing to expensive downstream processes
- Real-time detection reduces scrap costs by identifying defects before optical investigation, further value-added processing, preventing waste of materials, energy, and production time

Technical Challenges:

- Balancing existing imbalanced dataset to develop a properly trained machine learning model
- Balancing precision vs recall tradeoff—higher recall (better defect detection) comes at cost of false alarms(moderate precision), requiring operational judgment on acceptable balance

Lessons Learned:

- IIOT can be used to transform dumb objects into smart systems by connecting them to a device network and using data analytics to turn raw data into useful business information
- Edge computing is not just faster—it fundamentally changes operational economics by enabling immediate intervention
- MQTT is ideal for Smart Manufacturing, is lightweight, uses low bandwidth and scales easily with more machines and production lines

Business Value Calculation (Rough Estimate)

Cost Savings Example: (Data from the web)

Assumptions:

- Investigation(eg: optical) time per false alarm: **5 minutes**
- Labor cost: **$60/hour** (\$60 per hour $\div$ 60 minutes = \$1 per minute)
- 5 minutes $\times$ \$1 per minute = **$5 per investigation**

Monthly Impact (assuming 10,000 wafers/month, 10% defect rate):

Without IIoT System:

- Assuming 1,000 defects detected during optical inspection
- Cost of inspection: 1,000 $\times$ \$5 = **$5,000**

With this IIoT System:

- 800 defects caught early (~80% of 1,000) before optical inspection, as our model has recall rate of ~80%
- Savings: 800 $\times$ \$5 = **$4,000/month**
- False alarm cost: 300 alarms $\times$ \$5 = **$1,500/month** (~30% of 1000)
- **Net savings:** (4000 - 1500) = **$2,500/month**

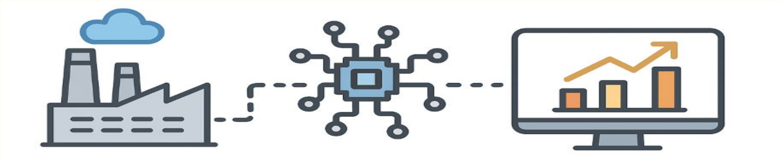
Conclusion & Future Extension

Conclusion:

- Our prototype demonstrated a functional end-to-end IIOT prototype that connects data generation → MQTT → edge ML prediction → database → real-time dashboard
- This system demonstrates that IIoT is not just about technology—it's about transforming manufacturing economics. By catching defects early, companies save money on every wafer that would otherwise consume expensive processing resources before being scrapped

Future Improvements:

- **Enhanced Machine Learning Model**: Improve recall and reduce false alarms by expanding the dataset, adding more process parameters, and incorporating advanced models such as gradient boosting or neural networks
- **Integration of Additional Sensors / Real Equipment**: Replace simulated data with actual hardware inputs (temperature, vibration, pressure sensors, or PLC signals) to further validate performance and improve prediction reliability in a real manufacturing environment
- **Security Implementation:** Add TLS/SSL encryption, authentication, and role-based access control



Yay!! We saved some wafers today!

Thank you !