EDGE-BASED PREDICTIVE MAINTENANCE IN SMART FACTORIES

Title of the Paper : An edge-cloud IIoT framework for predictive maintenance in manufacturing systems

Authors : Nivethitha Somu , Nirupam Sannagowdara Dasappa

Journal : Advanced Engineering Informatics

Team Members:

Team No : 6

Pradeep S - CB.SC.U4CSE23232

Rishiikesh S K - CB.SC.U4CSE23236

Sri Jayaram J S - CB.SC.U4CSE23255

Sathya Roopan M - CB.SC.U4CSE23256

PROBLEM STATEMENT

- In automotive component manufacturing factories, where continuous production and high equipment availability are essential to meet strict delivery timelines and quality standards, unexpected equipment failures can severely disrupt the production flow.

- The machines used in these factories are subject to various fault types including bearing wear, belt misalignment, hydraulic leakage, motor imbalance, rotor/stator faults, and excessive vibration or heat due to high operational loads.

- To address these challenges, the IntelliPdM framework is designed to be deployed within such automotive factories, enabling continuous monitoring and intelligent fault detection.

WHY IS IT AN EDGE PROBLEM?

Real-Time Decision Requirement

- Faults like belt slippage or overheating in CNC machines need to be detected and acted upon within milliseconds to prevent damage or production halts.
- Cloud latency is too high for such time-sensitive responses; edge computing enables decisions close to the machine.

High-Frequency, High-Volume Sensor Data

- Vibration sensors and temperature sensors generate large volumes of data every second.
- Continuously transmitting all this raw data to the cloud is bandwidth-intensive and inefficient. Edge processing reduces data transmission needs by analyzing data locally.

Network Instability in Industrial Environments

- Industrial factories often face unreliable or restricted network connectivity.
- Edge devices can continue functioning independently of constant cloud access, ensuring fault detection doesn’t pause due to network issues.

Scalability Across Machines and Units

- Large factories may have hundreds of machines. Sending all their data to a central cloud system becomes a bottleneck.
- Distributing the load across edge devices makes the system more scalable and responsive.

Our paper presents solutions to four key problems.

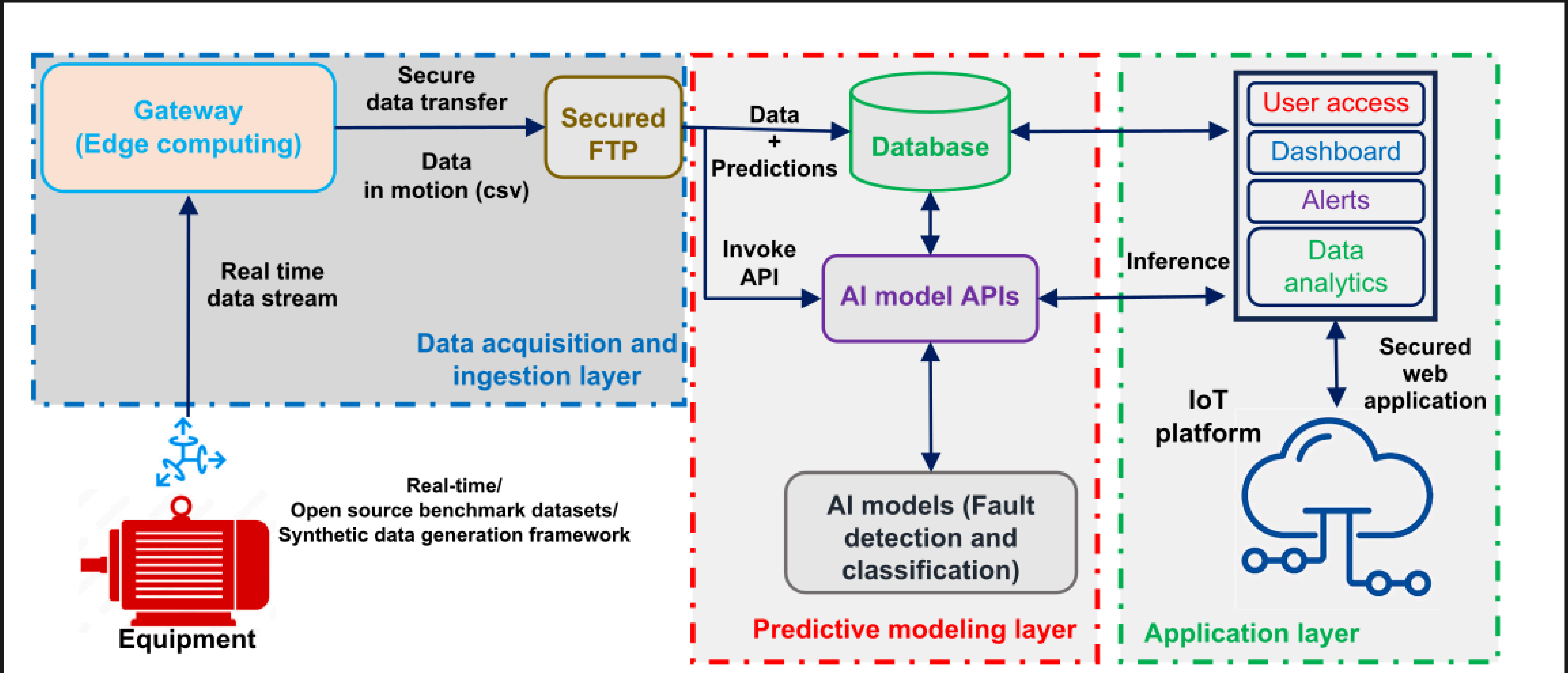
1. What can be a ‘one-stop solution’ for data challenges related to real-time operational data and open-source datasets?

2. What is the effective strategy to handle heterogeneous data from multiple sensors?

3. How to effectively consider multi-fault aspects of fault diagnosis in manufacturing assets?

4. What are the potential deployment options for resource-constrained computing environment?

ARCHITECTURE OF INTELLIPDM



Task Scheduling and Solution for Q4

An edge gateway performs immediate fault detection on sensor data using low-compute AI models, while securely offloading more intensive tasks to the cloud for deeper analysis. The cloud runs advanced models, and the combined predictions are then delivered to users as alerts and visualizations through a secured web application.

Implementation Strategy

1. Environment Setup

- iFogSim toolkit to model the edge-cloud computing architecture.
- Required Python libraries like Pandas, Scikit-learn, and TensorFlow

2. Data & Scenario Formulation

- The SMARTHome synthetic data generation framework will be used to produce the operational dataset.
- We will model fault scenarios for bearing defects, motor imbalance, and belt misalignment.

3. Model Implementation & Training

- We will implement and train Random Forest, ANN, and 1D-CNN models using the synthetic data.
- These models will perform Fault Detection and Fault Classification.

4. Simulation & Validation

The trained models will be deployed as modules within the iFogSim environment.
Simulations will be executed to validate the end-to-end performance from data generation to fault prediction.

Synthetic Data Generation:

Q1. What can be a ‘one-stop solution’ for data challenges related to real-time operational data and open-source datasets?

- **Synthetic data generation** is a framework that addresses data scarcity issues and a lack of benchmark datasets. It generates artificial data that simulates real-world conditions, including various fault scenarios, to augment or replace real-time operational data.

How Synthetic Data Generation Works

- **Problem and Solution**: Synthetic data generation solves the problem of insufficient real-time operational data for predictive maintenance. By creating realistic, artificial datasets that cover all possible fault scenarios, it allows for comprehensive training and validation of AI models, which is often not possible with limited real-world data.

- **Usage and Generation**: The synthetic data is used to train and validate machine learning models. This is achieved by simulating machine behaviors and failure patterns. The data is generated using software code, such as Python scripts and libraries, which take limited real-time data or open-source datasets and specific scenarios as inputs. The reliability of this generated data is then verified by domain experts.

Models used in Edge:

- We use Random Forest and if needed less computing resource ANN also used here.
- Random Forest collects data from vibration, temperature, pressure, acoustic sensors.
- ANN,Useful for both fault detection and basic classification of fault types (if features are already extracted).Can run on Edge if it’s a small model, but Cloud deployment may be needed for bigger ANN architectures.

Models used in cloud:

- We employ a hybrid deep learning strategy for highly accurate predictive maintenance. While 2D-CNNs analyze image-based data like thermal scans, our system's core utilizes a cloud-deployed 1D-CNN specifically designed for processing sequential time-series data from vibration and acoustic sensors. This architecture is exceptionally effective at learning from ordered data streams, allowing it to identify intricate patterns over time.

Why and what are FFT’s:

- FFT is a mathematical algorithm that converts vibration data from time domain to frequency domain.
- After converting to the frequency domain we can able to identify the faults in the machines by seeing the frequency.FFT helps identify the exact frequencies where faults might be occurring.
- RMS can detect the overall energy in the signal or the overall level of vibration(like an imbalance and looseness).
- Kurtosis can measure the spiky in the signal,that quantifies the outliers.

Implementation:

- Data Preprocessing is done for handling challenges in Synthetic data(Outlier Frequency).
- Z-score normalization was applied using StandardScaler from Scikit-learn to standardize features.
- Principal Component Analysis (PCA) from sklearn.decomposition.PCA was used to reduce dimensions.
- Each data sample (normal or faulty) was labeled manually using a supervised learning approach.
- These labels help train the AI models to learn the difference between healthy and faulty machine behavior.

Fault Types:

Machine Faults	Additional Data Inputs	Sensor Type
Bearing Defects	Bearing Specification	Vibration Sensor
Motor Imbalance	Motor Specification	Current and Vibration Sensors
Hydraulic Leaks	Hydraulic System specs	Presuure Sensor

Model Selection and Tuning:

- HyperParameters-Each model has settings (hyperparameters) like,Number of layers, learning rate, number of trees, etc.
- A smart and efficient method to find the best parameters is required to filter the models,whichis called hyper parameter tuning.
- The method used to filter the model is...

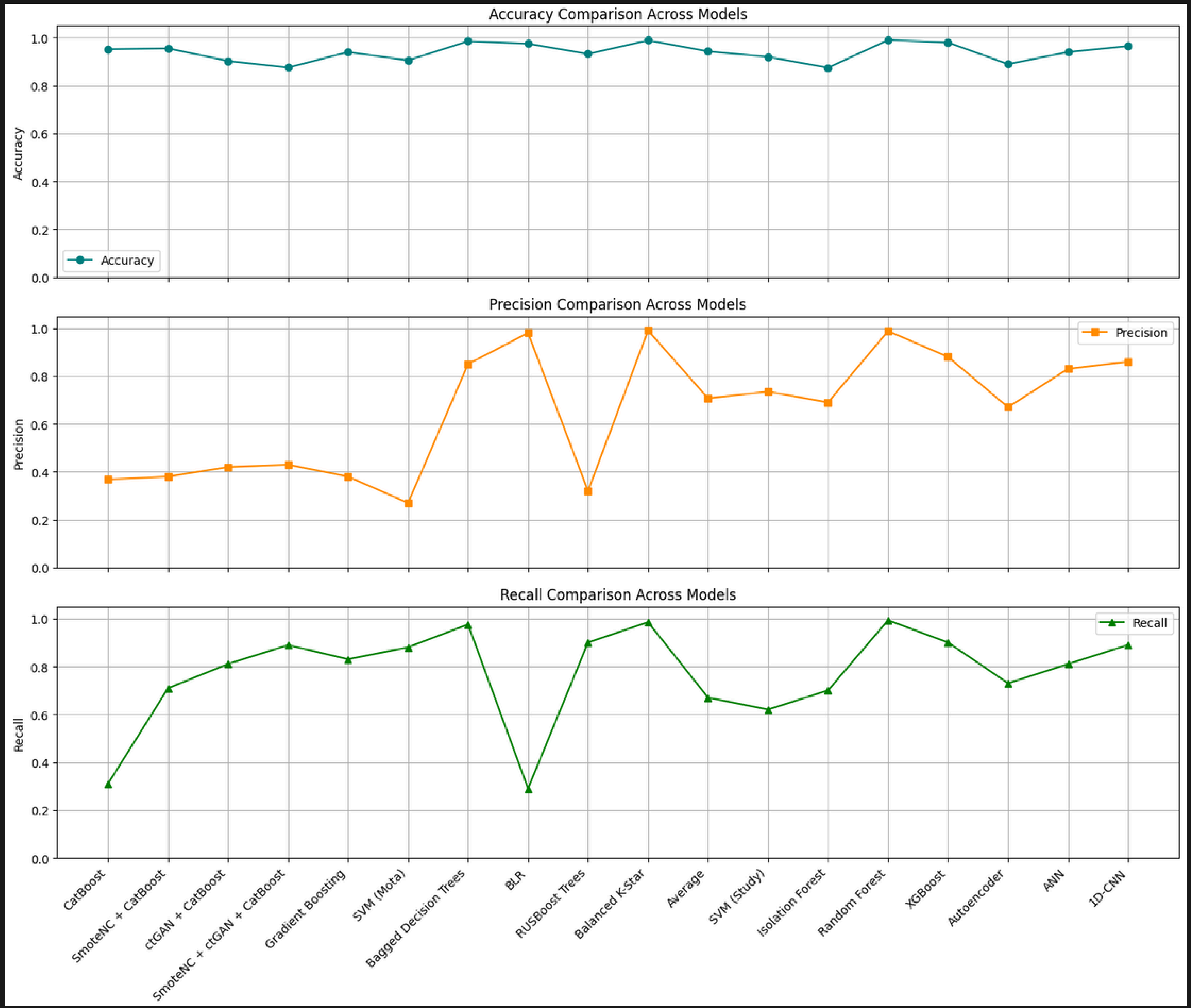
HalvingGridSearchCV

- Initially every model is trained with small amount of data and their performance is evaluated,and 50% of the models which doesnt perform well is eliminated.
- Then in next round the models is given with doubled data,and the process repeats until only one stands out.

(Accuracy,Precision,Recall) vs Models

Comparative analysis – State-of-the-art PdM frameworks.

Authors	Models	Accuracy	Precision	Recall	F1-Score
Chen et al. [38]	CatBoost	0.9521	0.3680	0.3102	0.3359
	SmoteNC + CatBoost	0.9553	0.3800	0.7905	0.5141
	ctGAN + CatBoost	0.9030	0.4200	0.8104	0.5532
	SmoteNC + ctGAN + CatBoost	0.8756	0.4300	0.8893	0.5857
Mota et al. [39]	Gradient Boosting	0.9400	0.38	0.83	0.50
	SVM	0.9055	0.27	0.88	0.41
Matzka et al. [39]	Bagged Decision Trees	0.9855	0.8500	0.9756	0.9100
Iantovics et al. [40]	BLR	0.9750	0.9800	0.2900	0.4500
Torcianti et al. [41]	RUSBoost Trees	0.9321	0.3200	0.9000	0.4700
Ghasemkhani et al. [42]	Balanced K-Star	0.9890	0.9900	0.9850	0.9870
	Average	0.9430	0.7070	0.6700	0.6030
Considered models for the study	SVM	0.9200	0.7350	0.6200	0.6730
	Isolation Forest	0.8750	0.6900	0.7000	0.6950
	Random Forest	**0.9901**	**0.9876**	**0.9930**	**0.9901**
	XGBoost	0.9800	0.8800	0.9000	0.8890
	Autoencoder	0.8900	0.6700	0.7300	0.6970
	ANN	0.9400	0.8300	0.8100	0.8200
	1D-CNN	0.9650	0.8600	0.8900	0.8740



STATE OF THE ART LITERATURE

	An Edge-Based IIoT Framework for Predictive Maintenance in Manufacturing Systems by Somu & Dasappa (2025)	Application of Sensor Data Based Predictive Maintenance and Artificial Neural Networks to Enable Industry 4.0	Machine Learning based Real Time Predictive Maintenance at the Edge for Manufacturing Systems
Architecture	Combines local quick responses with powerful cloud analytics	Relies on sending all data to the cloud for analysis	Processes directly on machines without cloud support
Data Handling	Works with sensor data (vibration, temp) and images/videos	Handles sensor data but not images or video	Focused on vibration signals
Accuracy	Consistent across different fault scenarios	Very high for its single machine	Strong results
Efficiency	Edge saves bandwidth and energy, cloud handles heavy tasks	High bandwidth use; less energy-efficient	**Very energy-friendly**
Scalability	Can expand to many sites easily	Suitable for small sites, but harder to scale broadly	Scaling means adding more edge devices manually

The solutions offered by the IntelliPdM for the research questions formulated based on the challenges in the state-of-the-art IIoT based predictive maintenance framework for manufacturing systems are :

1. What can be a ‘one-stop solution’ for data challenges related to real-time operational data and open-source datasets?

Solution : SMARTHome, a synthetic data generation framework was properly configured to generate real-time operational machine health data for various manufacturing machinery

2. What is the effective strategy to handle heterogeneous data from multiple sensors?

Solution: A separate data pipeline is designed to ingest structured and unstructured data into the appropriate data store

3. How to effectively consider multi-fault aspects of fault diagnosis in manufacturing assets?

Solution: The real-time operational machine health data for multi-faults in different machinery was generated by the synthetic data generation framework based on the well-defined faulty scenarios or use cases

4. What are the potential deployment options for resource-constrained computing environment?

Solution: Edge-cloud deployment architectures are best suited for resource- constrained environments, wherein sensitive data are processed locally on the edge; insights from the locally processed data, and complex data (audio, video, and images) are processed and staged on the cloud computing infrastructure ensuring efficient use of bandwidth,security and privacy, near real-time data analysis and decision-making & minimal cost and latency.

TASK SPLIT

Team Members	Task Split
Pradeep & Rishiikesh	1. Data Pipeline Implementation – Setting up sensor data collection, preprocessing workflows, and storage systems.
Sathya Roopan & Jayaram	2. Application/Interface Implementation − Building the dashboard, alert system, or reporting tool that uses the processed data for monitoring and decision-making.

THANK YOU