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:

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.

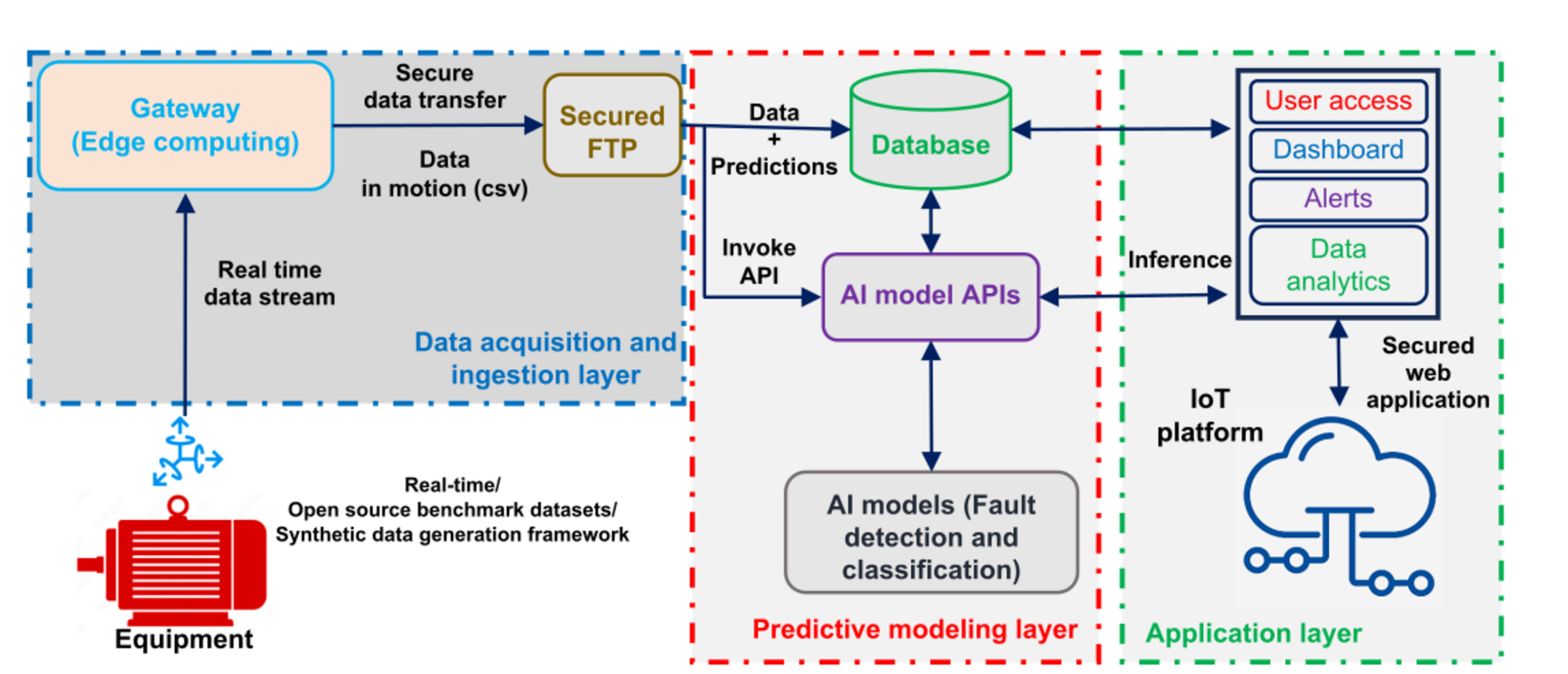
Network Instability in Industrial Environments

- 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.

ARCHITECTURE OF INTELLIPDM



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.

IntelliPdM Layer Objectives

The framework's goal is to prevent production disruptions in automotive factories by detecting equipment failures early. It uses a three-layer architecture to intelligently schedule analysis tasks.

1. Data Acquisition and Ingestion Layer

This layer uses edge gateways to continuously collect and preprocess raw data, like vibration and temperature readings, directly from factory machinery. This initial processing creates a standardized and reliable data stream, preparing the workload for efficient analysis and scheduling.

2. Predictive Modeling Layer

This layer acts as the system's intelligence hub, executing a clear task scheduling policy to balance speed and power.

- Edge Execution: For immediate insights, lightweight models like 1D-CNN and Isolation Forest are scheduled to run directly on the edge gateway, ensuring real-time fault detection.
- Cloud Execution: For deep diagnostics, computationally intensive models like Random Forest and Autoencoders are offloaded and scheduled for execution in the cloud, leveraging its superior processing power.

3. Application Layer

This final layer delivers the output from all scheduled tasks to the maintenance teams. It aggregates the real-time alerts generated at the edge with the in-depth diagnostics from the cloud, presenting a complete operational view with actionable recommendations on a unified web dashboard.

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.

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.
- For fault detection we use RMS and kurtosis.
- 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.

Synthetic Data Generation:

- Synthetic data generation(SMARThome) framework in designed using Python scripts and necessary packages was used to generate synthetic data for predictive maintenance in manufacturing assets
- The synthetic data generation module produces real-time operational data based on benchmark PdM datasets, limited real-time data, or defined fault scenarios.

Challenges:

- Benchmark datasets contain ‘normal’ and ‘faulty’ data samples generated for specific machinery specifications
- Operational schedule and faulty scenarios differ with respect to the nature of the machinery

Implementation:

- InteliPdM implemented using Python libraries, Apache Spark, Apache Kafka, MongoDB, Power BI, and AWS cloud services.
- 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

Predictive Maintenance Application:

- Phase 1 -Fault Detection(For finding out the probability of anomaly detection in the machine data)

- Phase2-Fault Classification(Finding the probability of the anomalyshould be the particular fault)

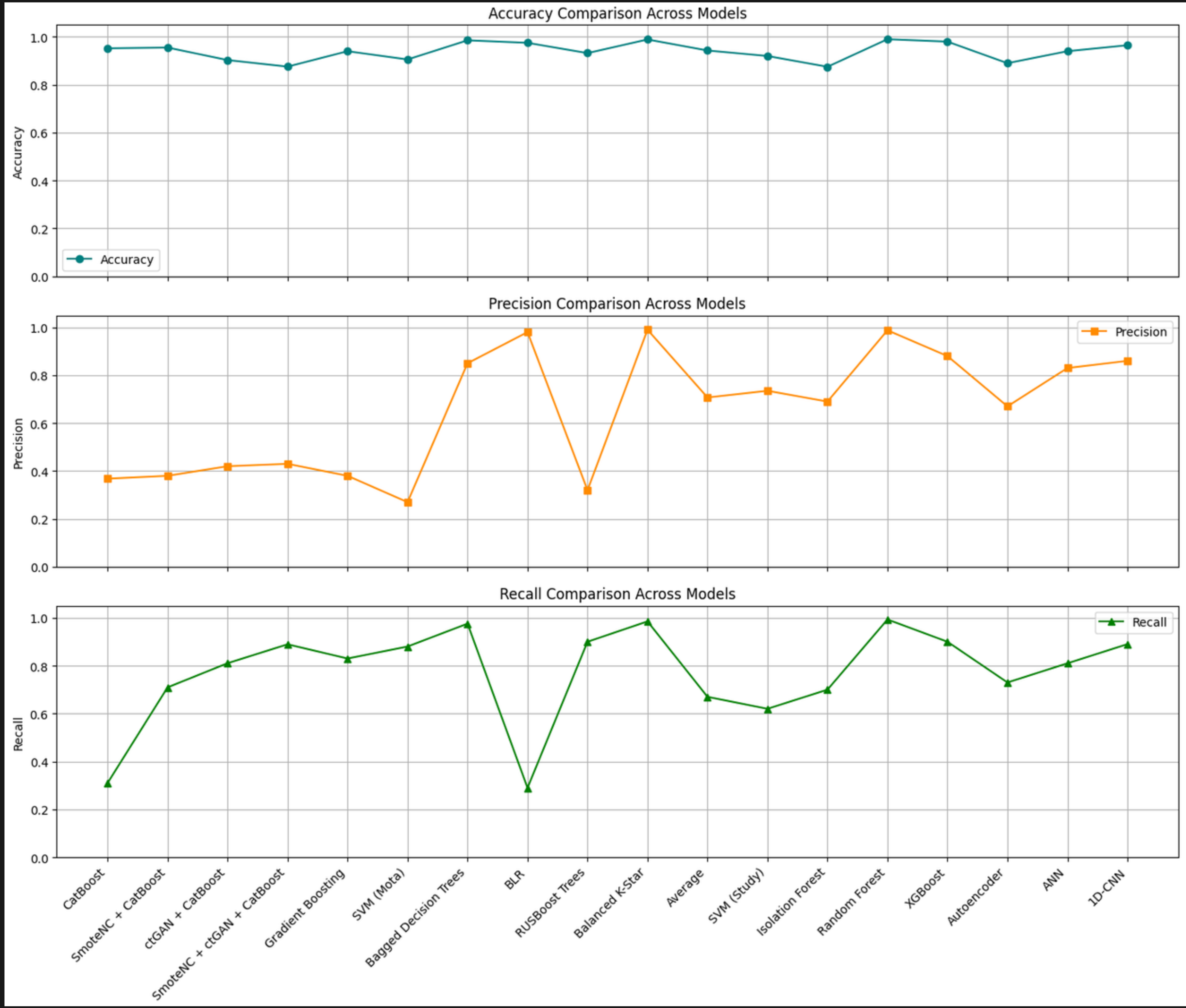
Artificial And Neural Networks used:

- 1D-CNN
- Isolation Forest
- Auto Encoders
- Random Forest(More Accurate)

(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



Accuracy Comparison Across Models
Accuracy
Precision Comparison Across Models
Precision
Recall Comparison Across Models
Recall
1.0
0.8
0.6
0.4
0.2
0.0
CatBoost
SmoteNC + CatBoost
ctGAN + CatBoost
SmoteNC + ctGAN + CatBoost
Gradient Boosting
SVM (Mota)
Bagged Decision Trees
BLR
RUSBoost Trees
Balanced K-Star
Average
SVM (Study)
Isolation Forest
Random Forest
XGBoost
Autoencoder
ANN
1D-CNN

STATE OF THE ART LITERATURE

	An Edge-Cloud 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

THANK YOU