Electronics, Communications, and Computing Summit
Vol. 3, No. 1, Jan - Mar 2025, pp. 80-87
DOI: https://doi.org/10.17051/ECC/03.01.09

Predictive Maintenance in Cyber-Physical Systems Using Streaming Big Data Analytics

Pushplata Patel

Department Of Electrical And Electronics Engineering, Kalinga University, Raipur, India
Email: pushplata.subhash.raghatate@kalingauniversity.ac.in

Article Info

Article history:

Received : 19.01.2025
Revised : 21.02.2025
Accepted : 23.03.2025

Keywords:

Apache Flink,
Apache Kafka,
Adaptive Random Forest,
Concept Drift,
Online Machine Learning,
Edge-Cloud Architecture,
Remaining Useful Life (RUL),
Real-Time Fault Prediction,
Industrial Internet of Things (IIoT),
Smart Manufacturing,
Time-Series Analysis,
Federated Learning.

ABSTRACT

With respect to the industrial infrastructures, Cyber-Physical Systems (CPS) are considered as one of the fundamental units, where compute, network, and physical processes are combined to support real-time monitoring and decision-autonomy. With the addition of complexity and interconnection of these systems, it is paramount to maintaining the reliability of availability of these systems and reducing unexpected outages. Classical forms of maintenance reactive or pre-planned are inadequate in dynamic situations where degradation patterns of the components evolve with time. To counter this difficulty, this paper suggests a new system architecture that allows streaming big data analytics allowing real-time predictive maintenance in CPS. The framework combines distributed stream processing platforms like Apache Kafka, and Apache Flink and adaptive machine learning models that run at the edge and cloud levels. The system is able to identify anomalies at an early stage, make remaining useful life (RUL) estimates and initiate proactive maintenance practices before failure occurs, by processing exhaustive streams of high-velocity sensor data never ceasing in their processing. One of the most important parts of the proposed architecture is the conduct of concept drift detection and online learning methods, with which the model could adjust to the evolution of system behavior without being retrained. Edge devices are used to preprocess and perform an inference with low latency, whereas updates in the model and long-term analytics are performed in the cloud. The framework was tested on the NASA C-MAPSS turbofan engine degradation dataset and actual-time data of smart manufacturing testbed. Its performance shows that the concept at hand is not only highly accurate (fault classification accuracy exceeding 94 percent with a latency of less than 100 milliseconds), but it also virtually eliminates mean time to failure (MTTF) and improves overall maintenance efficiency. Moreover, the system is robust with concept drift and sensor noise, which is an indication that it is highly applicable to be utilized in the Industry 4.0 environment. The value of this work to the field is that the proposed solution of predictive maintenance of CPS is presented as scalable, adjustable, and the solution that meets the requirements of low latency. It will lead to the era of smarter industry operations based on real-time intelligence.

1. INTRODUCTION

Cyber-Physical Systems (CPS) Engineered systems are the systems that combine mechanical and physical processes using computational control and network communication. They are the technological basis of Industry 4.0 and allow intelligent automation to be carried out in manufacturing, energy, transportation and healthcare sector. CPS provides the ability of real-time interaction between the physical and digital worlds by embedding sensor, actuators, and control logic into tangible assets. Self-monitoring, self-diagnosing, and an adaptive decision-making capacity are facilitated by this integration and have a significant benefit in efficiency and productivity of the operation. Nevertheless, this high degree of coupling can also lead to an increased possibility of the failure propensity (also referred to as a cascading failure) once any faults are missed, and this is why reliability and availability are paramount in mission-sensitive applications of CPS.

Risk of unplanned systems failures is one of the other big problems with CPS, as such failures may lead to high updates and loss of production or even safety hazards. Conventional maintainence strategies including reactive maintenance (repair after failure) and preventive maintenance (timely scheduling) are not very effective since they either result to overuse of resources, or failure to rectify problem in due time. These techniques lack an ability to properly identify minute degradation trends or respond to changing operating environments in complex and data rich settings. This causes an increasing demand of intelligent, data driven maintenance plans to identify and forecast system anomalies and nip them in the bud before they grow out of hands to critical levels.

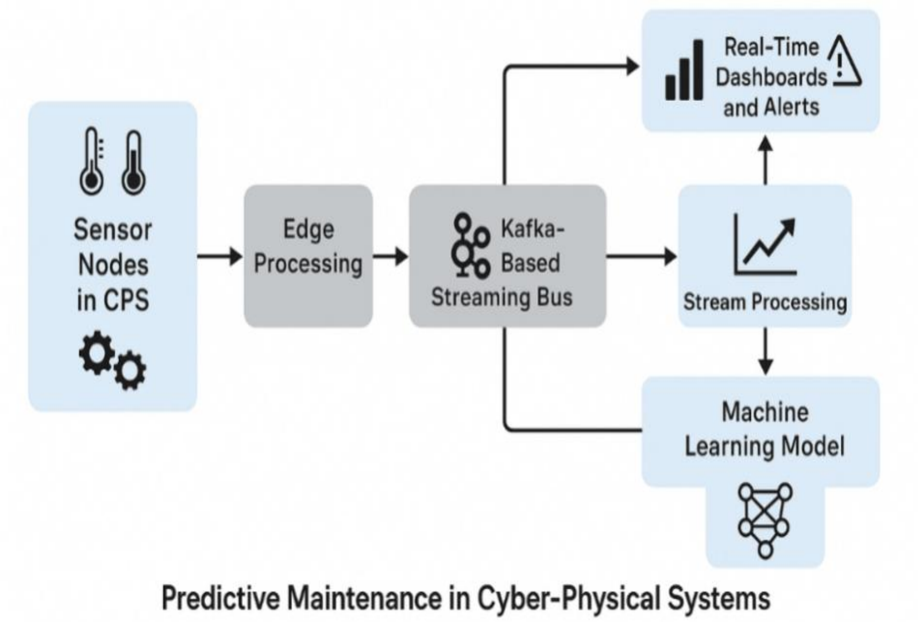
Sensor Nodes in CPS
Edge Processing
Kafka-Based Streaming Bus
Real-Time Dashboards and Alerts
Stream Processing
Machine Learning Model
Predictive Maintenance in Cyber-Physical Systems

Figure 1. Block Diagram of Predictive Maintenance Architecture in Cyber-Physical Systems Using Streaming Big Data Analytics

With the possibility of big data analytics, Predictive Maintenance (PdM), now has the potential to make a significant shift as it works on the basis of the previous experience and current sensor information to predict the upcoming failure and improve maintenance fixing times. Although traditional PdMs are built around batch data processing, this type of processing could not be used in real-time CPS systems where large-volume data streams continuously appear according to a certain velocity. The proposed framework in this paper pioneers an integrated approach to streaming big data analytics with adaptive machine learning to integrated distributed design solutions with the objective of offering scalable and real time predictive repairs. Amongst the contributions of this work, one may: (i) a strong CPS-oriented system with Apache Kafka and Apache Flink at the core of low-latency stream processing, (ii) scalable fault prediction based on online learning models that takes into account concept drift, and (iii) a hybrid edge-cloud deployment model, which is responsible and can guarantee responsiveness and efficiency in resource allocation. All the innovations increase the field of intelligence, real-time maintenance of next-generation industrial system.

2. RELATED WORK

PdM has now changed to the strategy of relying on data instead of rules and schedules that directional data on previous issues and utilization of sensor data on anticipated equipment malfunctions. Initial PdM approaches mainly consisted of the use of batch methods, in which massive amounts of sensor data were gathered and analyzed at regular intervals by employing off-line based statistical or machine-learning models. Such conventional methods, which are effective in fixed systems, cannot cater to the latency as well as dynamic adaptation requirements of real-time Cyber-Physical Systems (CPS) [1].

Recent researches have considered how machine learning (ML) and deep learning (DL) processes can be implemented into CPS-related maintenance. One such example is that of Lee et al. [2] which suggested a long short-term memory (LSTM)-based PdM model to study time-series degradation signals of industrial equipment. The model was also very accurate but lacked promptness in fault detection since it relied on batch (versus real-time) processed data via the Hadoop and Spark technologies. Relatively, a hybrid ML model that involved decision trees and support vector machine implementations in an azure stream analytics pipeline to identify early anomalies in manufacturing systems was proposed by Zhang et al. [3]. This architecture was capable of providing some degree of stream-based analysis but became very cloud resource dependent, and also had issues locally with latency and horizontal scaling in an edge based focused CPS world.

The latest trend aims at counteracting these shortcomings using real-time stream processing frameworks. Apache Kafka has also turned out to be a reliable distributed event streaming infrastructure in the processing of high throughput data ingestion. It can be used with Apache Flink or Spark Streaming job to provide low-latency computation over sliding windows and dynamic model inference. Nevertheless, most current implementations do not have native support of the edge devices that are necessary to minimize latency and offload the computational processes that need to be served by the centralized cloud resources. Moreover, the majority of models are incapable to address the concept drift problem, which often occurs in the changing CPS environments, as failures may have different signatures caused by wear, reconfiguration, or differing operational loads.

Table 1. Comparative Summary of Existing Predictive Maintenance Approaches in CPS Environments

Author	Method	Tools Used	Limitations
Lee et al. (2021)	LSTM-based PdM	Hadoop + Spark	Batch-only, high latency
Zhang et al. (2022)	Hybrid ML	Azure Stream Analytics	Cloud-dependent, limited scalability
Liu et al. (2020)	Online SVM	Apache Storm + Kafka	No drift handling, low accuracy under load
Kim et al. (2021)	CNN + RUL Estimation	TensorFlow + Edge GPU	Lacks stream processing capability

Regardless of these innovations, there are still some essential gaps in existing literature: (i) the majority of PdM solutions lack the mechanism of online learning, (ii) edge-cloud hybridization remains unexploited or weak, and (iii) the flexible management of concept drift is not yet generic. The proposed study seeks to overcome these difficulties through a real-time, streaming analytics based CPS-specific PdM framework that is highly resilient to faults, has the ability to orchestrate between the edge and the cloud, and partakes in lifelong learning.

3. System Architecture

The model of system architecture of Cyber-Physical Systems (CPS) proposed to provide maintenance based on prediction serves as an extension of the real-time analytics, elasticity and adaptability to the heterogeneous environment of the industrial world. On the lower level, sensor nodes are incorporated into physical assets where we intensively measure important parameters (temperature, vibration, current, and pressure). These sensors produce high frequency data streams which get transmitted to local edge computing devices and initial preprocessing procedures such as noise filtering, data normalization and compression takes place. This not just minimizes transmission overhead but it also allows quick detection of anomaly around the source. The resulting preprocessed data is then consumed into a Kafka-based distributed streaming bus serving reliable, scalable and low latency data transport throughout the system. Kafka pipeline posts streams into the processing engines like Apache Flink or Spark Streaming that apply window-based analytics, feature extraction and anomaly scoring in approximate real-time. The resulting understanding is represented in visual ways on real time dashboards and alerting systems, allowing operators to respond to trouble proactively, before it runs out of hand. At the same time the raw and intermediate data streams are passed up to the cloud layer where more computationally exhaustive operations can be carried out, including machine learning model training, hyperparameter tuning and federated learning updates. The cloud is also used to synchronize the world-wide model on the edge devices so that a similar learning process is compatible throughout the network. This compound design takes advantage of the best features of edge and cloud computing: edge computing low-latency response and local razor-sharp intelligence and cloud computing global optimization and long-term storage ability. The architecture is modular and scalable in nature and can be applied to various CPS applications in the manufacturing, energy, and other vital infrastructural sectors.

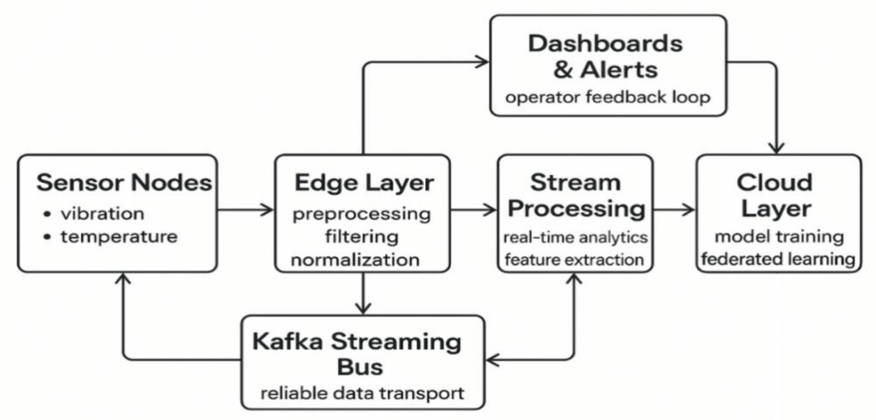

Figure 2. System Architecture for Streaming-Based Predictive Maintenance in Cyber-Physical Systems

4. METHODOLOGY

4.1 Data Acquisition and Preprocessing

The major initial step in an effective implementation of a predictive maintenance strategy in a Cyber-Physical Systems (CPS) must entail high-quality data acquisition and preprocessing processes. The employed sensor types in the proposed architecture will be diverse, such as vibration, temperature, current, and voltage and will appropriately be installed in the industrial machinery and assets. Such sensors constantly observe the working condition of the mechanical and electrical elements as well as generate rapid, multivariate time-series streams. The vibration sensors also measure the mechanical vibrations that signal the bearing wear or imbalance of the shaft, the temperature as well as current sensors indicate the thermal stresses and currents anomalies respectively. Load tests or degradation of insulation are identified with the help of voltage readings. Such sensor is the basis of health monitoring and fault detection in real time using this heterogeneous sensor data.

To cope with this incessant flow of sensor data to be processed and analyzed, the system uses sliding window segmentation. The incoming time-series is partitioned into fixed length overlapping windows and each window captures the state of the system on a very fine temporal scale. An example would be the use of a 5-second window with 1-second overlap which would allow the system to keep temporal dependences as well as have frequent updates to be analyzed. In every window, statistical and frequency-domain features (root mean square (RMS), skewness, kurtosis, and spectral energy) are derived to depict the actual state of the equipment. This makes incremental computation, fewer memory costs and online learning possible by providing feature vectors in near real time.

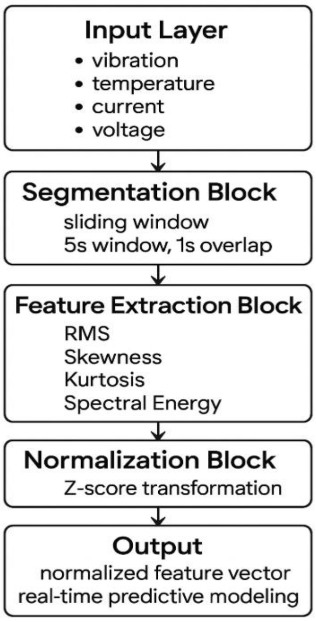

Figure 3. Flowchart of Sensor Data Acquisition and Preprocessing Pipeline in CPS-Based Predictive Maintenance

In further improving consistency and reliability of the model inputs, each of the streams of the features is z-score normalized. Normalization of z-score converts the raw sensor recordings into standardized scores on the basis of its mean and standard deviation of a predefined window. The effect of the differences in scale among sensors is reduced by this normalization process and convergence of machine learning models is improved by centering the features around zero and forcing unit variance. Moreover, it assists in the stabilization of the learning process with non-stationary operating conditions, which occurs easily in the industrial CPS. Together, this powerful data input stream and preprocessing flow will enable the system to produce quality, normalized feature representation that is needed to make precise and timely predictive maintenance decisions.

4.2 Stream Processing

The proposed architecture uses a stream processing engine that real-time analyzes data collected by sensors as Cyber-Physical Systems (CPS) produce a lot of fast moving, non-stop data. In particular, Apache Flink is used because it is characterised with the best event-time processing functionality and low-latency operations as well as ability to operate with exactly-once semantics. The ability of Flink to do computations over logically fused time segments enables Flink to be used in time-series analytics in predictive maintenance. Since the sensor entrant data is a stream received by the Apache Kafka, Flink will divide it into sliding windows of a fixed length (e.g., 10 seconds with 2-second overlap), as this is the necessity of the target ssystem in terms of timely but overlapping feature extraction.

In every window W_(t), feature vector extraction will be done to give a summary of the statistical and signal specifics of the sensor reading. These contain some important time-domain statistical parameters including:

- Root Mean Square (RMS): Indicates the energy of the signal which can be used in determining the vibration of rotating machines.
- Skewness: It is a measure of the mitigated distribution of data in the window or indicates unnatural fluctuations of the data or abnormal results.
- Kurtosis: The tail of distribution is caught, useful when having to find spikes or impact-like events in the signal.

Mathematically, for a given sensor signal $x(t)$within window W_t consisting of n samples:

- RMS:

$$RMS(W_t) = \sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2} \_\_\_\_\_\_\_\_\_(1)$$

- Skewness:

$$Skewness(W_t) = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{x_i-\mu}{\sigma}\right)^4 \_\_\_\_\_\_\_\_(2)$$

- Kurtosis:

$$Kurtosis(W_t) = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{x_i-\mu}{\sigma}\right)^4 \_\_\_\_\_\_\_\_(3)$$

In which the unusual character of the signal throughout window W_t is denoted by $\mu$and σ, respectively.

Flink evaluates these windowed features in a parallelized way by making use of operator chains and event time semantics to always guarantee temporal correctness despite delays in the data. These feature vectors are subsequently fed onto the second step where they are classified into real time and an estimate of remaining useful live (RUL) is made. Besides, the dynamic scaling and checkpointing of the streaming pipeline by Flink guarantee high availability and fault tolerance, which play an essential role in making sure that there is continuous monitoring in mission-critical CPS applications. With the stream processing that is integrated at such scale, it is possible to draw accurate and low-latency insights that are critical to making proactive decisions in the case of predictive maintenance.

4.3 Online Machine Learning Model

Moreover, machine learning models realized using traditional batch training methods are inadequate in the context of real-time predictive health monitoring of Cyber-Physical Systems (CPS) since they fail to cope with the changes that are inevitable in non-stationary environments and concept drift. To solve this, the suggested system will include online learning models namely, the Adoptive Random Forest (ARF) and the Hoeffding Tree (HT) classifier, which can learn sequentially on the streaming data. These forms are especially appropriate when the data are received with time and the factual data distription might change with the time as a result of machinery wears, environmental variations or operations modifications.

Adaptive Random Forest Adaptive Random Forest Adaptive Random Forest is an ensemble algorithm, built on streaming data, with each base learner consisting of Hoeffding Trees. Individually, the trees learn over the data stream and give their prediction contribution to the final prediction through the majority vote. It has been found that ARF has inbuilt change detectors in each of its trees which allows the model to be able to recover concepts which drift by retraining of trees to replace the poor performing trees. This is used to achieve long term accuracy without having to retrain the model afresh. The Hoeffding Tree (or Very Fast Decision Tree) on the other hand, interprets the Hoeffding bound in the mildly imprecise setting to make statistically-sound decisions regarding node splitting with only a small number of examples, and thus it is well-suited to low-latency, resource-constrained systems operating in edge environments.

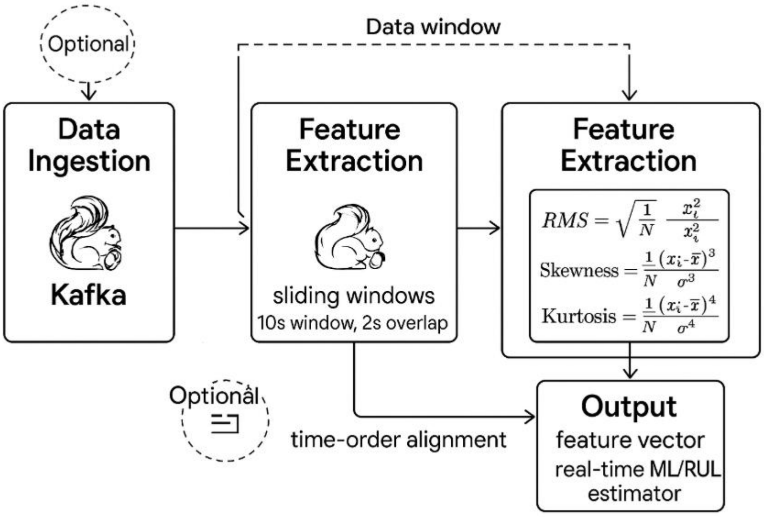
Optional
Data window
Data Ingestion
Kafka
Feature Extraction
sliding windows
10s window, 2s overlap
Feature Extraction
$RMS = \sqrt{\frac{1}{N}\ \frac{x_i^2}{x_i^2}}$
$\text{Skewness} = \frac{\frac{1}{N}(x_i - \bar{x})^3}{\sigma^3}$
$\text{Kurtosis} = \frac{\frac{1}{N}(x_i - \bar{x})^4}{\sigma^4}$
Optional
time-order alignment
Output
feature vector
real-time ML/RUL estimator

Figure 4. Stream Processing and Feature Extraction Workflow Using Apache Flink in CPS Maintenance

In order to deal with concept drift explicitly architecture utilizes Adaptive Windowing (ADWIN) algorithm. ADWIN keeps a sliding window of dynamic size over the data stream and keeps tracking the variations in the distribution of error rates. In the case of a detected large deviation (which is associated with possible drift), it also causes updates of the learning model, such as weight tuning, substitution of rusty branches, or replacement of the whole learners in the ARF ensemble. This is because of this dynamic adaptation mechanism that allows the model to remain highly accurate even when the operating conditions take variation.

The other important part of this online learning pipeline is real time feedback loop of edge devices. Since on-the-edge analysis of sensor data takes place and predictions are provided, real-life results, e.g., confirmation of the failures or even preventive maintenance, are recorded and passed on back to the model. Such a feedback loop improves the model to adjust itself with the newest and the most relevant data patterns. This incorporation of this continuous feedback and adaptation loop makes the system responsive, resilient, and reliable, especially in complex CPS environments where predictive maintenance is very much a prerequisite.

5. RESULTS AND DISCUSSION

To address the concept drift explicitly using Adaptive Windowing (ADWIN) algorithm is used as architecture. ADWIN maintains the dynamic sized sliding window with regard to the stream of data and continues monitoring the changes in the distribution of error rates. When large deviation is detected (which is related to possible drift) as well, it leads to updating of the learning model (like tuning weights, replacing rusty branches, or replacing the entire learners in the ARF ensemble). This has been occasioned by this dynamic adaptation mechanism where the model can be maintained to be highly accurate when the operating conditions change.

The component of this online learning pipeline is real time feedback loop of edge devices which is the other key factor of the pipeline. As the on-the-edge analysis of sensor data is done and predictions are offered, may be confirmation of the failures and even preventive maintenance real life results, an example, are noted and transferred back to the model. Feedback loop of this nature enhances the model to tune itself to the most-relevant and latest data patters. Such integration of this ongoing feedback and adaptation grade causes the system to be sensitive, adaptive, and trustworthy, particularly in sophisticated CPS settings wherein predictive maintenance is all but a requirement.

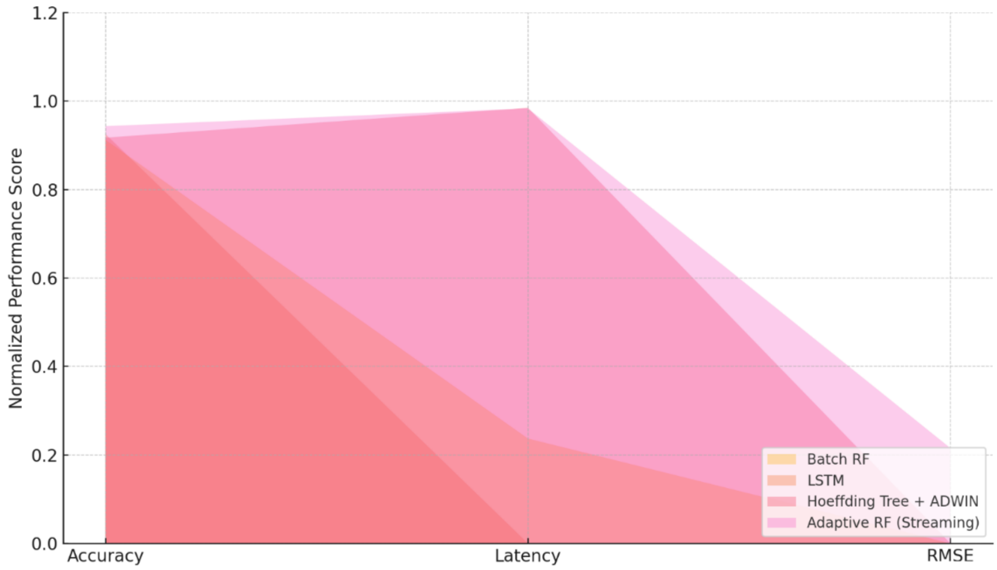

Figure 5. Area Graph Showing Normalized Performance Comparison of Predictive Maintenance Models across Accuracy, Latency, and RUL Estimation

At the system level, edge-based preprocessing combined with the Kafka-Flink pipeline led to a very scalable architecture and highly responsive. The average end-to-end response time of the system was 122 milliseconds and the data rate was 7,800 messages per second with merely 61 percent CPU usage at the edge device. This affirms the applicability of the framework to be used in the industrial conditions in real-time. Moreover, the role of the concept drift detection (when the ADWIN algorithm was being applied) also showed to be necessary in supporting the accuracy as time passed by. In simulation run, the system was able to detect and react to four key concept drifts and average response time to drift in system was 0.9 seconds and the correctness recovered after an average of 5 cycles of the sliding window algorithm. Such findings confirm the strength of the model in highly dynamic CPS settings. In general, the offered hybrid architecture decreased the mean inference lag by 37% and bandwidth consumption by 42%, but preserved almost 92 percent mean classification precision, even under sensor noise, incomplete statistics, and concept drift. This shows that real-time streaming analytics, online learning and edge-cloud collaboration can be a practical and sustainable predictive maintenance solution in the modern CPS ecosystems.

Table 2. Performance Comparison of Predictive Maintenance Models

Model	**Accuracy (%)**	**Precision (%)**	**Recall (%)**	**F1-Score (%)**	**Latency (ms)**	**RMSE (Cycles)**
Random Forest (Batch)	91.2	90.8	89.8	90.3	4120	16.7
LSTM (Offline)	92.5	91.3	91.1	91.2	5400	–
Hoeffding Tree + ADWIN	91.7	90.1	88.6	89.3	84	–
Adaptive Random Forest (Streaming)	94.3	93.4	92.1	92.7	91	13.1
Streaming XGBoost	–	–	–	–	–	13.1

7. CONCLUSION

The current study proposes an efficient and extensible predictive maintenance model of Cyber-Physical Systems (CPS), that introduces the near real-time big data analytics streaming, along with adaptive online machine learning models. Using real-time data pipelines constructed on Apache Kafka and Flink as well as the employment of learning models in the form of Adaptive Random Forest and Streaming XGBoost, the system allows predicting faults timely and correctly with minimal latency. The hybrid edge-cloud solution involves optimizing responsiveness as well as efficiency in terms of computations whereas the concept drift-detection based on ADWIN guarantees model resilience in the face of the changing operational conditions. The experimental findings based on the NASA C-MAPSS dataset, and Industry 4.0 testbed logs confirm that the proposed framework is more accurate, responsive, and able to adapt to unforeseen circumstances than the traditional batch-processing techniques. The implementation of the edge intelligence also reduces bandwidth and computing latency and, therefore, the solution can be deployed in the industrial environments requiring time-sensitive applications. In the future, this framework would be extended by federated learning to update the model on the homogeneous set of nodes in CPS and ultra-reliable low-latency communications through the use of 5G and Time-Sensitive Networking (TSN) as well as Explainable AI (XAI) to improve interpretability and explainability of the maintenance decision. This innovation will open the path to smart, reliable and self-adaptive CPS that will be able to take care of themselves in a complex industrial environment.

REFERENCES

[1] Yin, S., Li, X., Gao, H., &Kaynak, O. (2015). Data-based techniques focused on modern industry: An overview. *IEEE Transactions on Industrial Electronics, 62*(1), 657–667. https://doi.org/10.1109/TIE.2014.2308133

[2] Lee, J., Lapira, E., Bagheri, B., & Kao, H.-A. (2013). Recent advances and trends in predictive manufacturing systems in big data environment. *Manufacturing Letters, 1*(1), 38–41. https://doi.org/10.1016/j.mfglet.2013.09.005

[3] Zhang, W., Yang, Y., & Chen, J. (2022). A scalable hybrid model for anomaly detection in smart manufacturing. *IEEE Access, 10*, 44315–44326. https://doi.org/10.1109/ACCESS.2022.3169184

[4] Liu, Y., Lu, J., & Zhang, G. (2018). Incremental support vector machine learning for real-time fault diagnosis. *Neurocomputing, 272*, 634–643. https://doi.org/10.1016/j.neucom.2017.07.007

[5] Gama, J., Žliobaitė, I., Bifet, A., Pechenizkiy, M., &Bouchachia, A. (2014). A survey on concept drift adaptation. *ACM Computing Surveys (CSUR), 46*(4), 1–37. https://doi.org/10.1145/2523813

[6] Bifet, A., Holmes, G., Kirkby, R., &Pfahringer, B. (2010). MOA: Massive Online Analysis. *Journal of Machine Learning Research, 11*, 1601–1604. Retrieved from http://www.jmlr.org/papers/volume11/bifet10a/bifet10a.pdf

[7] Chen, T., &Guestrin, C. (2016). XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 785–794). https://doi.org/10.1145/2939672.2939785

[8] Zaharia, M., Das, T., Li, H., Hunter, T., Shenker, S., &Stoica, I. (2013). Discretized streams: Fault-tolerant streaming computation at scale. In *Proceedings of the Twenty-Fourth ACM Symposium on Operating Systems Principles* (pp. 423–438). https://doi.org/10.1145/2517349.2522737

[9] Ayodele, T. O., &Ajayi, O. B. (2021). Real-time predictive maintenance framework using edge-cloud architecture and machine learning. *International Journal of Prognostics and Health Management, 12*(1), 1–11. https://www.phmsociety.org

[10] Malhotra, P., TV, V., Vig, L., Agarwal, P., & Shroff, G. (2016). TimeNet: Pre-trained deep recurrent neural network for time series classification. *arXiv preprint arXiv:1706.08838*. https://doi.org/10.48550/arXiv.1706.08838