

Research Paper

EXPLAINABILITY BASED FEDERATED LEARNING FOR THE PREDICTIVE MAINTENANCE IN SMART MANUFACTURING

¹Mr. A V Murali Krishna, ²Korivi Rishit, ³Adula Karthikeya, ⁴Ksheeraja Sri Mugdha Gabbita

¹ Assistant Professor at Department of CSE Matrusri Engineering College, Saidabad, Hyderabad, Telangana, India

^{2,3,4} UG Scholar at Department of CSE, Matrusri Engineering College, Saidabad, Hyderabad, Telangana, India

Abstract

In today's manufacturing environment, machines are expected to run continuously, which makes early fault detection and maintenance planning very important. At the same time, industries need to ensure that sensitive operational data is not shared across different plants or systems. To handle both these requirements, this paper presents *Fed-Fusion-XAI*, a framework that combines predictive maintenance with privacy-preserving learning. The system uses a hybrid deep learning model that brings together CNN and BiLSTM techniques to process sensor data and identify unusual patterns in machine behavior. It not only detects possible faults but also estimates the Remaining Useful Life (RUL) of equipment such as turbofan engines and CNC machines. To make the predictions easier to understand, an explainability method based on SHAP is applied, which helps in showing how different factors influence the model's decisions. Instead of collecting all data in one place, the framework works in a distributed setup where individual nodes train their own models using local data. Only the learned parameters are shared with a central server, which combines them using a federated learning approach. This helps in maintaining data privacy while still benefiting from collective learning. The system also includes a real-time data pipeline for handling continuous data and a simple dashboard for monitoring results. The proposed approach provides a balanced solution that focuses on accurate prediction, data privacy, and better understanding of model outputs, making it suitable for real-world industrial applications.

KEYWORDS

Federated Learning, Predictive Maintenance, Explainable AI, SHAP, CNN-BiLSTM, Anomaly Detection, Remaining Useful Life, Industrial Systems, Data Privacy, Real-Time Monitoring

I INTRODUCTION

With the rise of Industry 4.0, manufacturing systems have become increasingly dependent on sensors that

continuously generate large amounts of time-series data. This shift has opened up new

possibilities for monitoring machine performance and improving efficiency. One important application that has gained attention is Predictive Maintenance (PdM), where machine learning models are used to identify potential failures

before they actually occur. By doing so, industries can reduce unexpected breakdowns and plan maintenance activities more effectively, which ultimately helps in saving both time and cost [1], [2].

Even though the idea of predictive maintenance is promising, applying it in real industrial environments is not always easy. One of the main challenges is related to data sharing. In many cases, manufacturing units are located in different places and operate under strict policies that do not allow raw data to be shared freely. Issues like data ownership, security concerns, and legal regulations make centralized data collection difficult. Because of this, building a single model using all available data becomes a complex task [3], [4].

Another concern comes from the nature of advanced machine learning models. While deep learning techniques are capable of giving accurate results, they often do not provide clear explanations for their predictions. In industrial settings, this becomes a serious limitation. Engineers and operators need to understand why a certain fault is predicted so that they can take the right action. Without this clarity, it becomes difficult to trust the system completely [5].

To handle the problem of data privacy, federated learning has been introduced as a useful solution. Instead of collecting all data in one place, this approach allows different systems to train models locally and share only the learned information. In

this way, data remains secure while still contributing to a common model. This makes federated learning suitable for environments where privacy is a major concern [6], [7].

At the same time, explainable AI techniques have been developed to make model decisions easier to understand. One such method is SHAP, which helps in identifying how different input features influence the output of the model. By providing this kind of insight, the system becomes more transparent and easier to use in practical situations [8].

In addition, hybrid models that combine different deep learning techniques have shown good results in handling complex data. For example, combining CNN and LSTM models allows the system to capture both local patterns and time-based relationships in the data. This makes the model more suitable for analyzing sensor data that changes over time [9], [10].

In this paper, a framework called *Fed-Fusion-XAI* is proposed to bring together these ideas. The system uses a CNN-BiLSTM model to analyze sensor data and detect unusual behavior. It also estimates the Remaining Useful Life (RUL) of machines, which helps in planning maintenance. The framework works in a federated setup, where each node trains its own model and only shares updates with a central server, ensuring data privacy [6], [11].

To make the system more practical, explainability is included using SHAP, and a real-time data

pipeline is used to handle continuous sensor inputs. Overall, the aim of this work is to develop a system that not only performs well but is also easy to understand and safe to use in real industrial environments [12].

II LITERATURE SURVEY

Predictive maintenance has been studied extensively with the goal of reducing equipment failure and improving operational efficiency in industrial environments. Early approaches mainly relied on statistical methods and simple machine learning models that used historical data to detect patterns. While these methods were useful, they often struggled to handle complex and high-dimensional sensor data generated by modern systems. With the growth of data availability, more advanced techniques have been introduced to improve prediction accuracy and reliability [1], [2].

In recent years, deep learning models have gained attention for their ability to process large volumes of time-series data. Techniques such as Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) networks have been widely used for fault detection and Remaining Useful Life (RUL) prediction. CNN models are effective in capturing local patterns in data, while LSTM models are better suited for learning temporal dependencies. Some studies have combined these models to improve overall performance, especially in complex industrial

scenarios where both spatial and temporal features are important [3], [4].

At the same time, the issue of data privacy has become more important, especially in distributed industrial environments. Traditional centralized learning approaches require collecting data from multiple sources, which is not always feasible due to security and regulatory concerns. To address this, federated learning has been proposed as an alternative approach. It allows multiple systems to train a shared model without exchanging raw data. This method has shown promising results in maintaining data privacy while still achieving good performance [5], [6].

Another area that has received increasing attention is the interpretability of machine learning models. In industrial applications, it is important not only to detect faults but also to understand the reasons behind predictions. Explainable AI techniques, such as SHAP, have been introduced to provide insights into how different features contribute to model outputs. These methods help improve trust in the system and support better decision-making by engineers and operators [7], [8].

Some recent studies have also explored the integration of real-time data processing with predictive models. Streaming platforms are used to handle continuous sensor data, allowing systems to make instant predictions and respond quickly to potential issues. In addition, hybrid frameworks that combine machine learning,

federated learning, and explainability are being developed to address multiple challenges at once. These approaches aim to provide accurate, privacy-preserving, and interpretable solutions for modern industrial systems [9], [10].

Although significant progress has been made, there are still challenges that need to be addressed. Many existing systems focus on either accuracy, privacy, or interpretability, but not all three together. This creates a gap in developing a complete solution that is both practical and reliable. The proposed work aims to address this gap by combining deep learning, federated learning, and explainable AI into a single unified framework [11], [12].

III RELATED WORK

A lot of work has been done in using deep learning for predictive maintenance. Models based on sequential learning, especially those designed to handle time-series data, have shown good results in predicting equipment health and estimating remaining life. In some cases, researchers have also combined different models to improve performance, such as using convolution-based methods along with sequence models. These hybrid approaches are able to capture both local patterns and long-term trends in sensor data, which makes them suitable for industrial applications. However, most of these solutions are developed in a centralized setup where all data is collected and processed in one place.

As industries became more concerned about data privacy, researchers started exploring distributed learning methods. Federated learning is one such approach where multiple systems can train a shared model without exchanging their raw data. This idea has been applied in different industrial scenarios, including monitoring and anomaly detection. While these methods help in protecting data, they mainly focus on model training and do not give much attention to how the predictions are made at each individual node.

Another important area of research is making machine learning models easier to understand. In industrial environments, it is not enough to simply detect a fault; operators also need to know what caused it. Some techniques have been introduced to explain model decisions by showing the contribution of different input features. However, in many existing systems, this explanation is performed after the model is trained and usually at a central level. This reduces its usefulness in real-time situations where quick and clear insights are needed at the source of data.

IV PROBLEM STATEMENT

In today's industrial setups, machines generate a continuous stream of sensor data that can be used to monitor performance and detect possible failures early. While this creates an opportunity for improving maintenance practices, using this data effectively is not simple. In many cases, industries operate across multiple locations, and each unit maintains its own data. Sharing this data

is often restricted due to security concerns, company policies, and legal requirements. Because of this, building a single model using all available data becomes difficult.

Another issue comes from the way modern machine learning models work. Although these models can provide accurate predictions, they usually do not explain how those predictions are made. In industrial environments, this lack of clarity can be a problem. Engineers need to understand what factors are contributing to a possible fault so that they can take the right action. Without proper explanation, it becomes hard to rely fully on the system.

There is also the challenge of speed and usability. Industrial systems often require quick decisions, especially when a failure is about to occur. Many existing solutions depend on centralized processing or heavy computations, which can slow down the response time. This makes them less suitable for real-time applications where immediate feedback is important.

Because of these reasons, there is a clear need for a system that can work with distributed data without compromising privacy, provide accurate predictions, and also explain its decisions in a simple way. At the same time, it should be fast enough to operate in real-time conditions and flexible enough to handle different types of industrial data. Addressing all these requirements together forms the main problem that this work focuses on.

V PROPOSED SYSTEM

The proposed system works as an integrated solution that combines distributed learning, real-time data processing, and explainable predictions. It begins with the collection of sensor data from industrial machines, which is then processed locally at different edge nodes. Each node trains a model using its own data and generates predictions related to anomalies and equipment health.

Instead of sending raw data to a central location, the system follows a federated learning approach where only model updates are shared. These updates are aggregated to create a global model, which is then redistributed to all nodes. This process allows the system to benefit from multiple data sources while maintaining data privacy.

To make the predictions more understandable, an explainability technique is applied at the local level. This helps in identifying which input features are influencing the model's decisions, providing useful insights for operators. At the same time, a streaming mechanism ensures that data is processed continuously, allowing the system to respond quickly to changes in machine behavior.

The final output is presented through a monitoring interface that provides real-time insights, predictions, and alerts. This enables users to take timely actions and improve maintenance planning. Overall, the proposed

system offers a balanced approach by combining accuracy, privacy, interpretability, and real-time performance in a single framework.

VI SYSTEM ARCHITECTURE

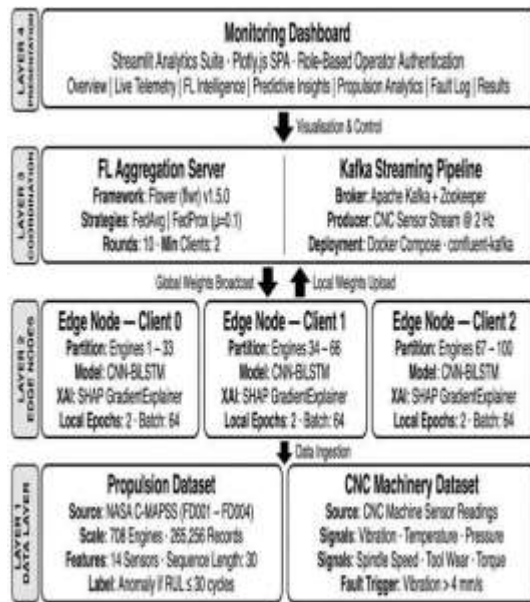


Fig. 1. FedXAI-Fusion System Architecture — Four-Layer Stack

The system is designed as a layered architecture that connects data sources, edge-level processing, centralized coordination, and user interaction. At the base, the data layer consists of industrial datasets collected from turbofan engines and CNC machines. These datasets include multiple sensor readings such as temperature, vibration, pressure, and other operational parameters. The data is structured as time-series information, which forms the input for further analysis.

Above this, the edge layer contains multiple client nodes, each responsible for handling a portion of the data. Instead of transferring all data

to a central server, each node processes its own dataset locally. A hybrid deep learning model is used at this stage to analyze patterns and generate predictions. In addition to prediction, an explainability mechanism is applied at each node to understand how different features contribute to the output. This ensures that the results are not only accurate but also interpretable.

The coordination layer connects all edge nodes through a federated learning framework. In this setup, each node shares only its model parameters with a central aggregation server. The server combines these updates to form a global model and sends it back to the nodes. This process continues over multiple iterations, allowing the system to learn collectively without sharing raw data. Alongside this, a real-time streaming pipeline manages the continuous flow of sensor data, ensuring smooth communication between different components.

At the top level, the presentation layer provides a monitoring dashboard for users. This interface displays live sensor data, prediction results, system performance, and alerts. It enables operators to track machine conditions and make informed decisions based on the insights generated by the system.

VII METHODOLOGY

The approach used in this work is designed to manage industrial time-series data in a practical way while also supporting distributed learning and model transparency. The process starts with

the C-MAPSS dataset, which provides run-to-failure data for turbofan engines under different operating conditions. It includes hundreds of engine instances and a large number of time-based records, each containing operational settings and multiple sensor readings. Since not every sensor contributes useful information, those with very little variation are removed at an early stage to simplify the data and improve learning efficiency.

Once the data is cleaned, the next step is to estimate the Remaining Useful Life (RUL) for each engine at every cycle. This is done by comparing the current cycle with the final cycle before failure. Based on the RUL value, the data is converted into a binary form where samples close to failure are marked as abnormal, and the rest are treated as normal. This conversion allows the problem to be handled as a classification task, which is more suitable for detecting potential faults.

To make better use of the time-based nature of the data, the sensor readings are grouped into sequences using a sliding window technique. Each sequence contains a fixed number of consecutive cycles, and the label of the final cycle is assigned to the entire sequence. This helps the model learn how patterns evolve over time rather than focusing on individual data points. After forming these sequences, all features are scaled to a common range so that no single feature dominates the learning process.

The prepared data is then divided across multiple client nodes to represent a distributed industrial setup. Each client works with its own subset of the data and trains a local model independently. The model used here combines convolutional layers, which are good at identifying local patterns, with bidirectional sequential layers that can capture relationships across time in both directions. This combination helps in understanding complex machine behavior more effectively.

Training takes place locally at each client for a few iterations. After this, instead of sharing raw data, only the learned model parameters are sent to a central server. The server combines these updates to form a global model and shares it back with all clients. This process continues over several rounds, gradually improving the model while keeping the data secure at its original location. To deal with differences in data across clients, a constraint is applied during training so that local models do not drift too far from the global model.

To make the system more understandable, an explainability step is included at each client. This step highlights which input features have the most influence on the model's predictions. The explanations are summarized and used for visualization, ensuring that sensitive data is not exposed while still providing useful insights. The system is supported by a real-time data pipeline that continuously feeds new sensor data into the model. This allows predictions to be made as data

is generated, making the system suitable for live industrial environments. The complete workflow, from data preparation to prediction and monitoring, is designed to be efficient, reliable, and easy to interpret in practice.

VIII RESULTS AND ANALYSIS

The performance of the proposed system is evaluated using standard classification measures such as accuracy, precision, recall, F1-score, and AUC-ROC. These metrics help in understanding how effectively the model detects anomalies while reducing incorrect classifications. The evaluation is carried out on different subsets of the dataset under both centralized and federated learning settings.

Dataset	Setting	Accuracy	Precision	Recall	F1-Score	AUC-ROC
FD001	Centralized	0.9312	0.9105	0.9487	0.9292	0.9741
FD001	Federated	0.9187	0.8963	0.9341	0.9148	0.9618
FD002	Centralized	0.8874	0.8612	0.9103	0.8851	0.9423
FD002	Federated	0.8701	0.8441	0.8972	0.8698	0.9287

Table 1: Centralized vs Federated Performance

The results indicate that the centralized model performs slightly better since it has access to the entire dataset. However, the federated model achieves comparable performance, with only a small reduction in accuracy and F1-score. This shows that privacy-preserving learning can still deliver reliable predictions.

FL Rounds	Accuracy	F1-Score	AUC-ROC
3	0.8712	0.8601	0.9201
5	0.8934	0.8823	0.9387
10	0.9187	0.9148	0.9618
15	0.9241	0.9198	0.9672
20	0.9263	0.9217	0.9689

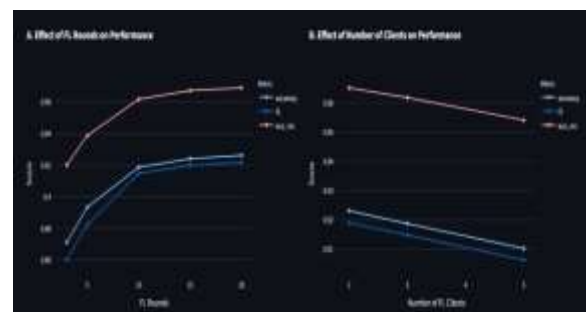
Table 2: Effect of Federated Learning Rounds

It can be observed that performance improves as the number of training rounds increases. However, after around 10 rounds, the improvement becomes smaller, indicating that the model has nearly converged. This helps in selecting an optimal number of training rounds for efficiency.

Clients (K)	Accuracy	F1-Score	AUC-ROC
2	0.9231	0.9189	0.9651
3	0.9187	0.9148	0.9618
5	0.9102	0.9063	0.9541

Table 3: Effect of Number of Clients

As the number of clients increases, a slight drop in performance is observed.

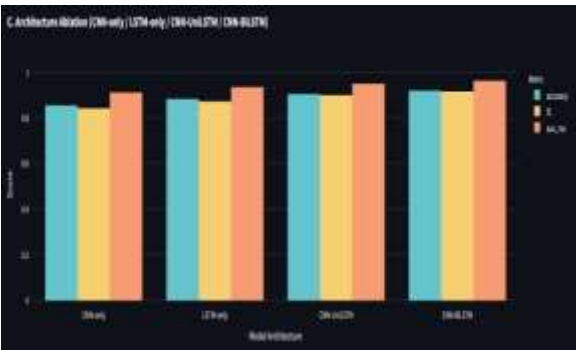


The effect of increasing K on performance is a slight reduction in performance

This is mainly due to differences in data distribution across clients. Even with this variation, the system maintains stable performance, showing that it can scale well.

Model Architecture	Accuracy	F1-Score	AUC-ROC
CNN Only	0.8541	0.8412	0.9103
BiLSTM Only	0.8823	0.8714	0.9341
CNN + LSTM	0.9034	0.8987	0.9487
CNN + BiLSTM (Proposed)	0.9187	0.9148	0.9618

Table 4: Architecture Comparison



Ablation Study C: Architecture Comparison (FD001, K=3, 10 rounds)

The comparison shows that the hybrid model performs better than individual models. The combination of convolutional layers and bidirectional sequence learning helps in capturing both local and long-term patterns, leading to improved results.

IX CONCLUSION

A practical framework for predictive maintenance has been developed by combining distributed learning, deep learning techniques, and model interpretability. The system is designed to work with industrial time-series data and focuses on identifying abnormal behavior while also estimating the remaining useful life of equipment. By using a hybrid model, it is able to capture both short-term patterns and long-term trends in the data, which improves the overall prediction capability. A key aspect of this approach is the use of federated learning, where multiple nodes train models locally and share only the learned parameters instead of raw data. This makes the system suitable for environments where data privacy is important. Even with this restriction, the results show that the model performs close to a centralized approach, which highlights the effectiveness of distributed training. Another important feature is the inclusion of explainability at the local level. The system provides insights into how different input features influence the predictions, which helps users better understand the model’s behavior. This is especially useful in industrial settings, where decisions need to be supported by clear reasoning rather than just numerical outputs. The system also supports continuous data processing, allowing it to handle real-time inputs and provide timely predictions. This makes it more suitable for practical use, where quick responses are required to prevent failures and reduce downtime.

the proposed approach offers a balanced solution that brings together accuracy, privacy, and interpretability. While the system performs well under the current setup, there is still scope to extend it further by testing on more diverse datasets and improving its ability to adapt to changing conditions.

REFERENCES

- [1] J. Lee, H. Davari, J. Singh, and V. Pandhare, "Industrial Artificial Intelligence for industry 4.0-based manufacturing systems," *Manufacturing Letters*, vol. 18, pp. 20–23, 2018.
- [2] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proc. Int. Conf. Artificial Intelligence and Statistics (AISTATS)*, 2017, pp. 1273–1282.
- [3] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in *Proc. Advances in Neural Information Processing Systems (NeurIPS)*, 2017, pp. 4765–4774.
- [4] S. Zheng, K. Ristovski, A. Farahat, and C. Gupta, "Long short-term memory network for remaining useful life estimation," in *Proc. IEEE Int. Conf. Prognostics and Health Management (PHM)*, 2017, pp. 88–95.
- [5] X. Li, Q. Ding, and J.-Q. Sun, "Remaining useful life estimation in prognostics using deep convolution neural networks," *Reliability Engineering & System Safety*, vol. 172, pp. 1–11, 2018.
- [6] Y. Liu, T. Dillon, W. Yu, W. Rahayu, and F. Mostafa, "Privacy-preserving federated learning for industrial artificial intelligence," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 10, pp. 6532–6542, 2021.
- [7] D. C. Nguyen, M. Ding, P. N. Pathirana, A. Seneviratne, J. Li, and D. Niyato, "Federated learning for industrial internet of things in future industries," *IEEE Wireless Communications*, vol. 28, no. 6, pp. 192–199, 2021.
- [8] A. Saxena and K. Goebel, "Turbofan engine degradation simulation data set," *NASA Ames Prognostics Data Repository*, 2008.
- [9] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith, "Federated optimization in heterogeneous networks," in *Proc. MLSys*, 2020, pp. 429–450.
- [10] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.
- [11] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proc. IEEE CVPR*, 2017, pp. 1251–1258.
- [12] J. Chen, R. Monga, S. Bengio, and R. Jozefowicz, "Revisiting distributed synchronous SGD," in *Proc. Int. Conf. Learning Representations (ICLR)*, 2016.