International Journal Of
Industrial Electronics Control and Optimization
Print ISSN: 3517-2645
Online ISSN: 3568-2645
Homepage: https://ieco.usb.ac.ir/

Secure PDM: A novel Byzantine Fault Tolerant federated learning framework using a robust PCA-based anomaly detection approach

Khalil Jahani[1] | Behzad Moshiri[2] | Babak Hossein Khalaj[3]

[1]Department of Computer Science, kish International Campus, University of Tehran, Tehran, Iran
[2]School of ECE, College of Engineering, University of Tehran, Tehran, Iran
[3]Department of Electrical Engineering, Sharif University of Technology, Tehran, Iran
Corresponding author: e-mail: moshiri@ut.ac.ir

Article Info

Article type:
Original Article

Article history:
Received: 23-February-2025
Received in revised form: 17-May-2025
Accepted:30-May-2025
Published online: 22-Dec-2025

Keywords:
Federated Learning,
Byzantine fault tolerant,
Predictive Maintenance,
Anomaly detection.

ABSTRACT

With the proliferation of federated learning programs as a suitable framework for protecting user privacy and reducing the computational overhead of AI algorithms, various industries have also turned to the widespread use of this framework in industrial applications such as improving predictive maintenance (PDM). However, despite its increasing applications, several security challenges, such as Byzantine attacks, make the application of federated learning in industries questionable. Byzantine attacks in FL can degrade model performance by injecting malicious updates, causing model divergence or biased learning. This reduces accuracy, and can introduce security vulnerabilities such as backdoors. To address this problem, we propose a Byzantine Fault Tolerant (BFT) federated learning algorithm designed to improve PDM in industrial applications. Our proposed approach uses a PCA-based anomaly detection algorithm to detect and mitigate local Byzantine updates. Also, a game theory-based reward mechanism is designed to promote honest participation and discourage malicious behavior among federated users. The proposed framework is evaluated using the predictive maintenance datasets "AI4I 2020" and "NASA Acoustics and Vibration". The results show that our proposed framework effectively detects and mitigates Byzantine attacks, enhancing the overall reliability of PDM in industrial applications.

I. Introduction

Today, federated learning has been a promising computational field that enables decentralized training of models across multiple devices while preserving data privacy [1, 2]. One of the main capabilities of federated learning is the distributed learning capability. It allows the use of distributed data sources without centralized data processing or storage. The main advantage of this capability is that this practical feature of federated learning reduces concerns about the data privacy of network users. It also reduces the computational overhead. Despite its advantages, one of the significant challenges this FL capability brings is ensuring the integrity and reliability of the learning process [3]. One of the most critical challenges is the presence of Byzantine faults—malicious or faulty updates from compromised participants that can degrade the performance of the global model or even lead to catastrophic failures [4, 5].

In federated learning, one of the main assumptions that can be made about local Byzantine nodes is that they are malicious or faulty participants that intentionally change the weights or gradients of their local updates before sending them to the central server for aggregation [6, 7]. These nodes can significantly disrupt the training process by injecting corrupted or misleading data, compromising the global model's integrity and accuracy. Because the central federated learning server relies on collecting local updates from multiple distributed nodes to refine the global model repeatedly, the presence of Byzantine nodes distorts the aggregated results, leading to a corrupted or biased global model that may perform poorly in the overall data distribution [7, 8].

To handle this problem, we propose a BFT federated learning framework using a PCA-based anomaly detection algorithm. Our proposed algorithm prevents local Byzantine updates after detection and ensures the reliability of the federated learning process. The proposed algorithm

identifies anomalous patterns in local model updates that exhibit potential Byzantine behavior. Then, it filters them before they affect the global model.

Enhancing PDM with our proposed PCA-based anomaly detection algorithm improves the accuracy and reliability of predictive maintenance models. It also leads to more effective maintenance plans, reduced downtime, and extended equipment lifetime.

This paper describes our proposed BFT-FL framework and how to integrate it with PDM applications. Then, we evaluate our framework and demonstrate its effectiveness in detecting and mitigating Byzantine attacks. We also discuss the impact of our proposed framework on improving PDM parameters (precisely, the Remaining Useful Life criterion).

A. Our contributions

This paper proposes a Byzantine fault-tolerant federated learning framework that can be integrated with PDM systems to enhance predictive maintenance. Our approach proposes a novel PCA-based anomaly detection algorithm, which detects anomalous local updates sent by byzantine local nodes. The anomaly detection algorithm also specifies the anomaly severity of each local update, indicating the potential impact of that malicious update if it has not been prevented. After the byzantine updates are detected and prevented, the benign updates will be given to an aggregation module, obtaining the new global model, which will be sent to local nodes as the updated global model.

To encourage the local nodes to behave honestly, a game theory-based reward/penalty module has been designed that uses historical trust scores of nodes and assigns them rewards/penalties based on their scores. This module also uses the weights given by the anomaly detection algorithm to update the historical scores of local nodes. Briefly, our main contributions are:

- A novel byzantine fault tolerant federated learning framework with appropriate performance for use in PDM system to improve the PDM technology.
- Designing a PCA-based anomaly detection system that detects and prevents anomalous local updates.
- Using a Game Theory-based incentivization and reward/penalty mechanism encourages local nodes to behave honestly and minimizes the impact of byzantine local nodes on the global model.
- Designing an interactive weighted aggregation mechanism that takes the weight vector of local updates from the Game Theory-based incentivization module and then implements a weighted aggregation of local updates.
- Ability to detect and prevent dynamic byzantine nodes on each round.

B. Paper organization

The remainder of this paper is organized as follows. In section 2, we explain some related works. Section 3 explains our proposed federated learning architecture by describing the designed anomaly detection and incentivization/reward-penalty modules. In Section 4, we present and discuss the achieved results. Finally, Section 5 concludes the paper and proposes possible future works.

II. Related work

In this section, we review several of the most critical secure aggregation techniques to mitigate the impact of Byzantine attacks. The goal of these methods is to detect and prevent malicious updates and ensure that the aggregation process is robust to adversarial attacks. For example, Byzantine- resistant Cosine Similarity Aggregation [9], Krum [10], and Trimmed Mean [11] are designed to filter out or mitigate the impact of outlier updates that deviate significantly from the majority. In addition, anomaly detection mechanisms can identify and remove updates that exhibit suspicious patterns indicative of Byzantine behavior. By combining these strategies, FL systems can increase their robustness and reliability and maintain global model integrity even in the presence of adversarial participants. This section briefly describes previous work on secure aggregation mechanisms and BFT ML algorithms.

The work in [12] introduces “BytoChain”, a framework that integrates blockchain technology with federated learning to enhance security and integrity, especially at the network’s edge. BytoChain uses a Byzantine-resistant consensus mechanism, Proof of Accuracy (PoA). This approach also decentralizes the aggregation process to reduce the influence of malicious nodes.

In [13], a blockchain-based FL framework is proposed to address Byzantine challenges. It uses fully homomorphic encryption to ensure all data is encrypted during aggregation, preventing data manipulation. Blockchain technology records and verifies transactions to increase the transparency of the learning process.

In [14], an approach called “BDFL” focuses on the specific needs of autonomous vehicles by developing a centralized, decentralized, fault-tolerant, Byzantine learning method. It combines Peer-to-Peer (P2P) learning with strong Byzantine fault-tolerant protocols, including publicly verifiable secret sharing, to enhance the safety of data exchange and model aggregation.

The FLTH algorithm introduced in [15] uses reliable data and historical performance metrics to assess the reliability of participating nodes. The method dynamically adjusts the influence of each node based on its reputation and filters out malicious or untrustworthy contributions.

The research in [16] presents the BREA framework, which incorporates stochastic quantization, verifiable outlier detection, and secure aggregation to defend against Byzantine users in federated learning environments. BREA

is designed to detect and exclude anomalous updates, ensuring the integrity of the model aggregation process.

The study in [17] explores a decentralized approach to federated learning that enhances resistance to Byzantine faults by distributing the aggregation function across multiple nodes. Using consensus mechanisms similar to those in blockchain technology, the method ensures that multiple parties validate all updates before integration.

The research in [18] proposes a federated learning framework that ensures Byzantine fault tolerance even under semi-honest and Byzantine behaviors among participants. The approach leverages the Expectation-Maximization algorithm to distinguish between benign and Byzantine participants. The framework can reliably identify malicious participants by evaluating the performance of randomly generated candidate models using all participants' datasets.

The study in [19] addresses the challenge of Byzantine resilience in edge-based clustering. The authors introduce a distributed gradient descent algorithm with Byzantine resilience optimized for convex and non-convex stochastic problems. In this study, gradient compression is also used to increase communication efficiency and maintain an optimal statistical error rate in the presence of Byzantine adversaries.

The study in [20] introduces a local gradient descent algorithm with a comparative elimination filter to effectively counter Byzantine attacks. It is assumed that agents may provide incorrect data intentionally or due to errors caused by system failures in the considered system. The study distinguishes between deterministic settings (exact error tolerance) and stochastic settings (approximate error tolerance) and examines the algorithm's robustness under different adversarial conditions.

Despite the numerous advantages of previously proposed schemes, they suffer from certain challenges and limitations that drive us toward designing a new secure aggregation mechanism. One of the main limitations associated with the implementation of most of these schemes is the low speed of the aggregation process due to the use of blockchain-based, peer-to-peer, secret sharing, and distributed architectures. Moreover, it should be noted that the use of blockchain-based and peer-to-peer architectures potentially leads to the violation of the confidentiality of clients' local models.

Our proposed scheme addresses the challenges and limitations mentioned in previous works by introducing a lightweight and computationally efficient secure aggregation mechanism that ensures minimal information disclosure among clients while maintaining high accuracy in detecting and mitigating Byzantine nodes in FL environments.

III. The proposed approach

In this section, we explain our proposed approach. The system architecture of our method comprises a PCA-based anomaly detection module, a Game Theory-based incentivization/reward-penalty mechanism, and an aggregation integrator module, which takes the outputs of those modules as inputs and sends the final benign updates and their weights to the central server for aggregation. Figure 1 shows the proposed architecture.

The PCA-based anomaly detection module sits between the end nodes and the federated learning server, taking local updates sent by end devices to the central server. It detects anomalous updates and prevents them from reaching the server. Finally, it sends a severity vector of local updates to the aggregation integrator module. The severity of each local update indicates the value of its outlierness and anomaly. Severity values near 1 show a high probability of the client being anomalous, while values near 0 show a potentially normal client.

The Game Theory-based incentivization/reward-penalty mechanism upgrades the historical status of local devices, using their historical records and the current severity vector of their updates calculated by the anomaly detection module. It then sends the upgraded historical status of local devices to the aggregation integrator module.

Finally, the aggregation integrator module sends the upgraded historical status of local devices and their updates to the server. This process repeats at each global training round.

Here, we explain the architecture of our PCA-based anomaly detection, Game Theory-based incentivization/reward- penalty mechanism, and global model, which is updated using this architecture.

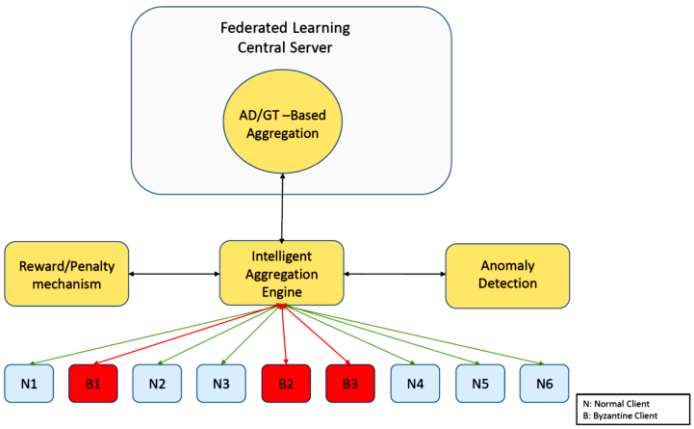

Fig. 1. The architecture of proposed Byzantine Fault Tolerant federated learning framework

A. PCA-Based anomaly detection

In this section, we describe the proposed anomaly detection algorithm. This algorithm works as follows:

1. Preprocessing: converting local models' weights to vectors with fixed length
2. Feeding the PCA algorithm with an initial set of benign local models' weights
3. Transforming the local updates using the PCA algorithm
4. Calculating the upper and lower bounds of the principal components with eigenvalues less than a

fixed threshold (in this case, below 1% of total variance)

5. Finding the outliers and calculating the severity of each one, according to their outliers' values on each of the principal components
6. Sending the index of anomalous updates and their severity values as the output

Figure 2 shows an example of principal components (PC) 1 and 15 without Byzantine nodes. The transformed data are divided into time intervals for a specific end device. Since the smoothed updated weight vector of the model resulting from each local device has more than 3500 elements, we divided it into vectors with lengths equal to 50 to facilitate the analysis. Therefore, each end device has about 70 rows of data.

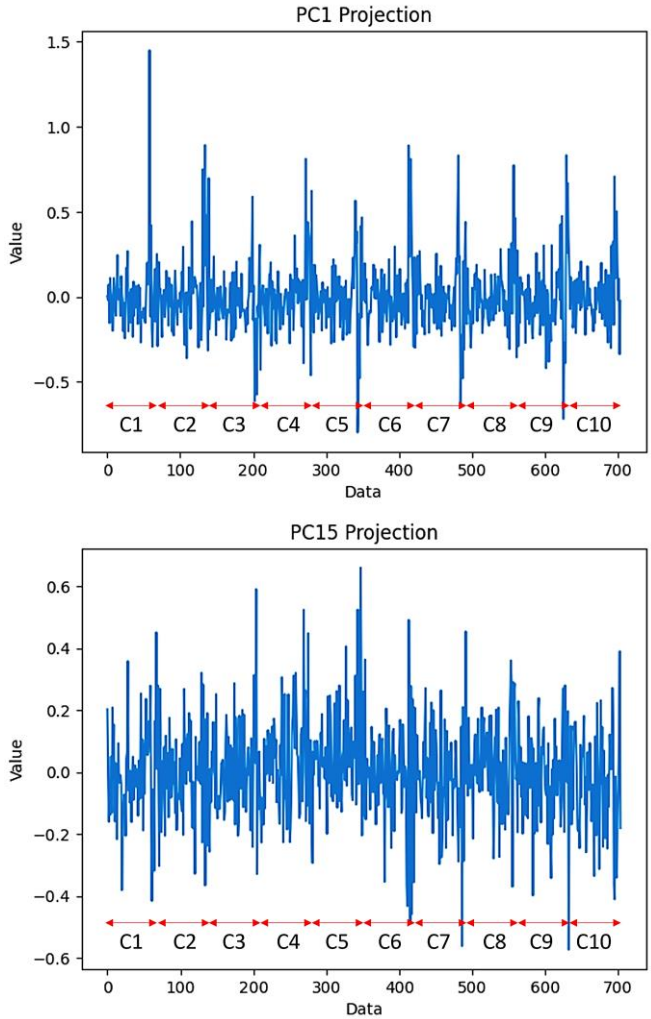

Fig. 2. The principal components PC1 and PC15 of benign local updates. In this example, we have ten end devices.

B. Incentive mechanism

In this section, we describe the exponential weighting incentive mechanism. This mechanism is a practical approach used in the proposed BFT federated learning framework to assign weights to local updates from different local devices. It uses the historical score of each local device to determine the impact of its new update in the network.

At the beginning, each device is assigned an initial score (denoted S_0). We assume this initial score to be uniform across all devices.

$$s_{t+1} = s_t + \gamma \Delta s_t \tag{1}$$

For each device, this score is continuously updated over time, based on the validity of its submitted updates. In the next step, valid updates increase the score, while malicious updates decrease it. We formulate the score update rule as follows:

where s_t is the score at time t, γ is a constant coefficient, which we set to 1 for simplicity. Δs_t is the change in the score at time t, determined by whether the update is valid or malicious and is formulated as follows:

$$\Delta s_t = \begin{cases} +\beta_1 & \text{If update is valid} \\ -\beta_2 & \text{If update is malicious} \end{cases} \tag{2}$$

where β_1 is the positive reward increment for valid updates, and β_2 is the penalty for malicious updates. Since we want our mechanism to be as strict as possible with Byzantine nodes and ensure that the impact of any malicious update from a client remains negligible on the global model updates for a long time, we choose the values of β_1 and β_2 such that $\beta_1 \ll \beta_2$. In this case, we assume β_1=1 and β_2=10.

These calculated historical scores show the reliability of each local client at each timestamp. Our mechanism then uses these scores to calculate new weights for each user through an exponential function. This weighting mechanism ensures that local users with higher scores receive higher exponential weights. The proposed method makes a significant distinction between trusted users and Byzantine users of trust, which increases the overall robustness of the federated learning model against Byzantine attacks.

We calculate a normalized exponential function for each historical score to calculate its weight in the final exponential weighting. We normalize these values so that their sum stands equal to one. Mathematically, this is expressed as:

$$\alpha_i(t) = \frac{\exp(s_i(t))}{\sum_{j=1}^{N} \exp(s_j(t))} \tag{3}$$

This approach also amplifies the differences in scores and makes the influence of trusted local devices more prominent. The progressive weighting gives local devices a high incentive to maintain or improve their performance over time because, given the scoring mechanism, even a slight improvement in their scores can significantly increase their weights. This approach causes a better performance of the BFT federated learning framework.

C. Global model update

The global model update module in a Byzantine Fault Tolerant (BFT) federated learning framework is essential for integrating local updates into a single, cohesive global model. This module uses weights obtained from the Exponential Weighting incentivization mechanism, ensuring that participants with a history of reliable contributions significantly influence the global model. The weighted aggregation of the local updates is mathematically expressed as:

$$w_{\text{global (t)}} = \sum_{i=1}^{N} \alpha_{i(t)} w_{i(t)} \tag{4}$$

where $w_{\text{global (t)}}$ is the global model, $w_{\text{i(t)}}$ represents the local update from local client i, and $a_{\text{i(t)}}$ denotes the weight assigned to local client i at time t. As said before, this weight is derived from the mentioned incentivization mechanism.

In addition to the weighting mechanism, the global model update module also utilizes the scores of each client in another way. This algorithm, in each round, removes the updates of clients with negative scores from the global aggregation process before it is performed. This pipeline design helps the final global model by allowing it to correct local updates before weighting and aggregation, ensuring greater integrity of the global model.

IV. Results and discussion

This section evaluates our model by implementing several byzantine scenarios and comparing federated learning performance with and without our proposed architecture.

Byzantine scenarios

In a federated learning scenario, imagine a network of local devices in a distributed system participating in collaborative training of a machine learning model. These local devices are responsible for updating their weights based on their local data and sending them to a central server for aggregation and model updating. However, in this scenario, some local devices are compromised or malfunctioning, exhibiting Byzantine behavior.

Byzantine local devices intentionally modify their updated weights before sending them to the central server. This can include injecting spurious gradients by manipulating the weight updates and producing incorrect weight values. As a result, the central server needs to receive consistent and correct information from these Byzantine local devices during the aggregation process.

Unaware of Byzantine behavior, the central server incorporates these falsified weights into its model aggregation process, leading to erroneous global model updates. This can have significant repercussions, mainly if the central server heavily relies on the accuracy of the weights provided by the local devices for global model training.

Datasets

In the following sections, we evaluate our proposed model on two useful predictive maintenance datasets:

- AI4I 2020 Predictive Maintenance: This is a labeled sensory dataset that contains values for air temperature, process temperature, rotational speed, torque, and tool wear sensors for various industrial equipment. Each row of this dataset is labeled with a binary value indicating machine failure.
- NASA Acoustics and Vibration: This is an unlabeled sensory dataset that records the simulated vibrational behavior of four bearings over two weeks, capturing their normal and abnormal behavior until the end of their lifespan.

The reason for choosing the AI4I 2020 dataset is its suitable structure for partitioning it among different clients. In this dataset, the sensor data associated with each machine is listed in different situations along with its failure or non-failure status. This makes it an ideal candidate for federated learning environments where each client can process its respective machine's data. The reason for choosing the NASA Acoustics and Vibration dataset is that it has a time series structure to analyze the failure status of 4 bearings throughout the entire life of the equipment, from the initial anomaly observation to their failure. Therefore, using this dataset in the analyses provides a good visual and numerical view of the impact of our proposed system on system performance in estimating the remaining life of each equipment.

A. Evaluation on AI4I 2020 Dataset:

This section evaluates our Federated Learning model on the "AI4I 2020" dataset. To evaluate this labeled dataset, we assess the system under three scenarios:

- A normal scenario without Byzantine behavior.
- Scenarios with Byzantine ratios of 0.2, 0.4, and 0.6, but without the anomaly detection module.
- Scenarios with Byzantine ratios of 0.2, 0.4, and 0.6, including the anomaly detection module.

We compare the accuracy and loss of the federated learning framework across these three scenarios to demonstrate the importance and effectiveness of our proposed BFT architecture. Under the first scenario, where there are no Byzantine nodes, the federated learning algorithm achieves an accuracy of 90% and a loss of 0.22. In the second situation, where there are Byzantine nodes in 3 different scenarios (Byzantine portions equal to 0.2, 0.4, and 0.6) but no anomaly detection module, Table I indicates the accuracy and loss values of the federated learning algorithm. The results indicate the adverse impact of Byzantine nodes on

system performance, causing, for instance, a 17% reduction in accuracy in one scenario. Finally, the third scenario repeats the Byzantine cases, but our anomaly detection module is applied to the architecture this time. The results show that our approach significantly improves the performance of the FL algorithm by detecting and reducing all Byzantine updates. For example, in this scenario, our approach improves the accuracy by approximately 16% and reduces the loss by approximately 0.53, which emphasizes the effectiveness of our proposed architecture.

We have also compared the results of our proposed method with those of two well-known approaches, Krum aggregation and Trimmed-Mean. As shown in Table I, our proposed method achieves better results in all three Byzantine scenarios.

TABLE I COMPARISON OF THE RESULTS OF THE FL ALGORITHM EVALUATION USING AI4I 2020 DATASET WITH DIFFERENT PORTIONS OF BYZANTINE NODES WITH AND WITHOUT ANOMALY DETECTION MODULE

Scenarios	Metrics	Byzantine = 0.2	Byzantine = 0.4	Byzantine = 0.6
Without anomaly detection	Accuracy	86.5%	84.1%	73.56%
	Loss	0.28	0.33	0.78
Krum aggregation [10]	Accuracy	88.92%	88.70%	88.61%
	Loss	0.258	0.269	0.277
Trimmed-Mean [11]	Accuracy	89.48%	89.4%	84.3%
	Loss	0.238	0.261	0.497
One-Shot Federated Learning with PCA-based detection	Accuracy	82.57%	82.35%	78.29%
	Loss	0.326	0.367	0.511
Our Proposed Framework	**Accuracy**	**89.90%**	**89.83%**	**89%**
	Loss	**0.237**	**0.245**	**0.256**

Figures 3 and 4 show the performance of the anomaly detection module in two different scenarios. In the first scenario, nodes 2 and 8 are Byzantine, while nodes 4 and 6 are Byzantine in the second scenario. As the figures show, our proposed anomaly detection can successfully detect all Byzantine nodes due to their outlierness.

B. Evaluation on NASA Acoustics Dataset:

In this section, we evaluate our Federated Learning model using the "NASA Acoustics and Vibration" dataset under similar conditions discussed in the previous section. We extend our evaluation by calculating each scenario's RUL and

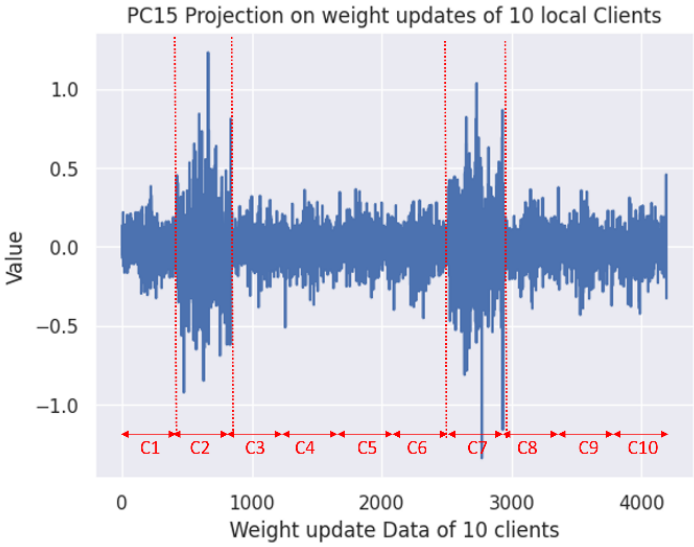

Fig. 3. The principal components PC15 of local updates. In this example, we have ten end devices. In this example, nodes 2 and 7 are byzantine.

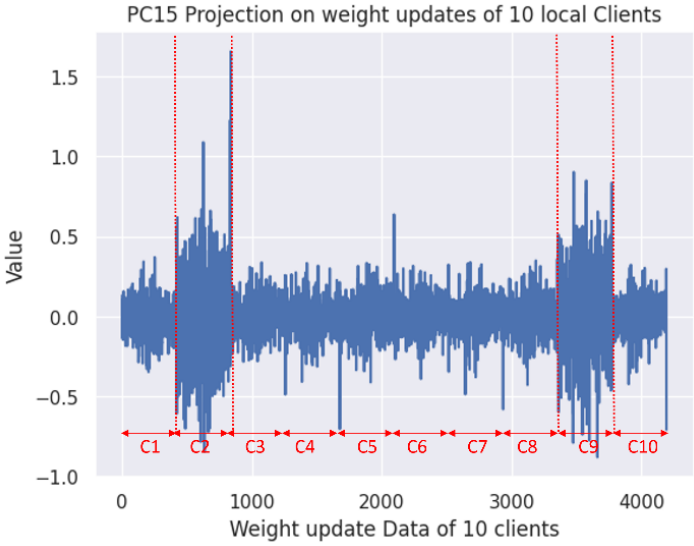

Fig. 4. The principal components PC15 of benign local updates. In this example, we have ten end devices. In this example, nodes 2 and 9 are byzantine.

RMSE parameters. We demonstrate the negative impact of Byzantine attacks on calculating the RUL criterion, one of the most critical parameters in PDM. The byzantine attack may deceive industry experts by suggesting incorrect RUL values. We investigate this in two main cases: producing overestimated or underestimated RUL values.

Overestimated RUL values lead to incorrect planning for equipment replacement and repair. This planning error occurs when maintenance experts mistakenly believe the equipment has more remaining life than it actually does. As a result, they schedule repairs or replacements too late after the equipment has already failed, which leads to equipment failure before repairs are scheduled. It causes significant damage and financial losses for the organization. Such failures may result

in irreparable damage to the equipment, damage to related machinery, and costly production line shutdowns during the time needed for repairs.

Conversely, underestimated RUL values cause maintenance specialists to schedule repairs or replacements much earlier than necessary. This leads to premature repairs or replacements, significantly increasing maintenance costs. In this case, in a fixed period, specialists are called more often to repair and replace the equipment, and secondly, more equipment is purchased because the entire life of the equipment is not used.

To evaluate the effectiveness of our proposed framework under the aforementioned Byzantine scenarios, we implemented and trained an LSTM-Autoencoder (LSTM-AE) as the core anomaly detection model. This model is utilized to construct a robust Health Index (HI) based on its reconstruction error, which reflects the deviation of input data from learned normal behavior.

The selection of the LSTM-AE architecture is motivated by its synergistic integration of two powerful components:

- Long Short-Term Memory (LSTM) networks, which are well-established for their ability to capture and model complex temporal dependencies inherent in time-series data.
- Autoencoders, which are proficient at learning compressed latent representations of the input data and identifying anomalous patterns by detecting deviations from the training distribution.

Together, these characteristics make the LSTM-AE model particularly well-suited for anomaly detection in sequential industrial data, enabling effective identification of Byzantine behaviors within the federated learning environment.

Table II summarizes the architecture of the LSTM-Autoencoder model. The Health Index (HI), derived from the reconstruction error, is passed through a Kernel Density Estimation (KDE) function to analyze the distribution of training loss values. Based on this distribution, an anomaly detection threshold is determined, set to exceed 95% of the training error values. This threshold is used to identify anomalous behavior. Following anomaly detection, the resulting reconstruction error sequence is used to train an LSTM regression model, which estimates the Remaining Useful Life (RUL) at each time step.

The results of our implementation are shown in Table III. Figure 5 shows the RUL curve in a normal scenario with no byzantine behavior. The RMSE value calculated in this scenario is equal to 63.47. Figure 6 shows the RUL curve in the first byzantine scenario in which there 20% of nodes have byzantine behavior. The RMSE values calculated in this scenario where there is no byzantine detection module and where our proposed method is used are equal to 149.85 and 78.21, respectively. Finally, Figure 7 shows the RUL curve in the second byzantine scenario in which 40% of nodes have byzantine behavior. The RMSE values calculated in this scenario where there is no byzantine detection module and where our proposed method is used are equal to 409.06 and 97.01, respectively. The results indicate the effectiveness of our proposed Byzantine detection model, which reduces the RMSE in Byzantine scenarios by up to 76%.

TABLE II COMPARISON OF THE RESULTS OF THE FL ALGORITHM EVALUATION USING NASA ACOUSTICS AND VIBRATION DATASET WITH DIFFERENT PORTIONS OF BYZANTINE NODES WITH AND WITHOUT ANOMALY DETECTION MODULE

Layer (type)	Output Shape	Params
input1(Input Layer)	(None, 1, 4)	0
lstm (LSTM)	(None, 1, 32)	4736
Leaky relu (Leaky ReLU)	(None, 1, 32)	0
Batch normalization (Batch Normalization)	(None, 1, 32)	128
dropout (Dropout)	(None, 1, 32)	0
lstm1(*LST M)*	(None, 16)	3136
leakyrelu1(Leaky ReLU)	(None, 16)	0
Batch normalization1(Batch Normalization)	(None, 16)	64
Repeat vector (Repeat V ector	(None, 1, 16)	0
lstm2(*LST M)*	(None, 1, 16)	2112
leakyrelu2(Leaky ReLU)	(None, 1, 16)	0
batchnormalization2(Batch Normalization)	(None, 1, 16)	64
dropout1(*Dropout*)	(None, 1, 16)	0
lstm3(*LST M)*	(None, 1, 32)	6272
leakyrelu3(Leaky ReLU)	(None, 1, 32)	0
batchnormalization3(Batch Normalization)	(None, 1, 32)	128
time distributed (Time Distributed)	(None, 1, 4)	132

Similar to the previous part, we have also compared the results of our proposed method with those of two aggregation methods, Krum and Trimmed-Mean. As shown in Table III, our proposed method achieves better results in all three scenarios: without Byzantine nodes, with Byzantine nodes at a ratio of 0.2, and with Byzantine nodes at a ratio of 0.4.

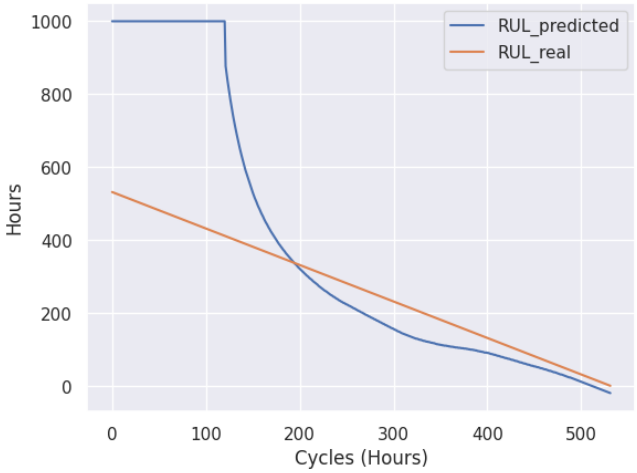

Fig. 5. The RUL curve is estimated in case no byzantine behavior exists.

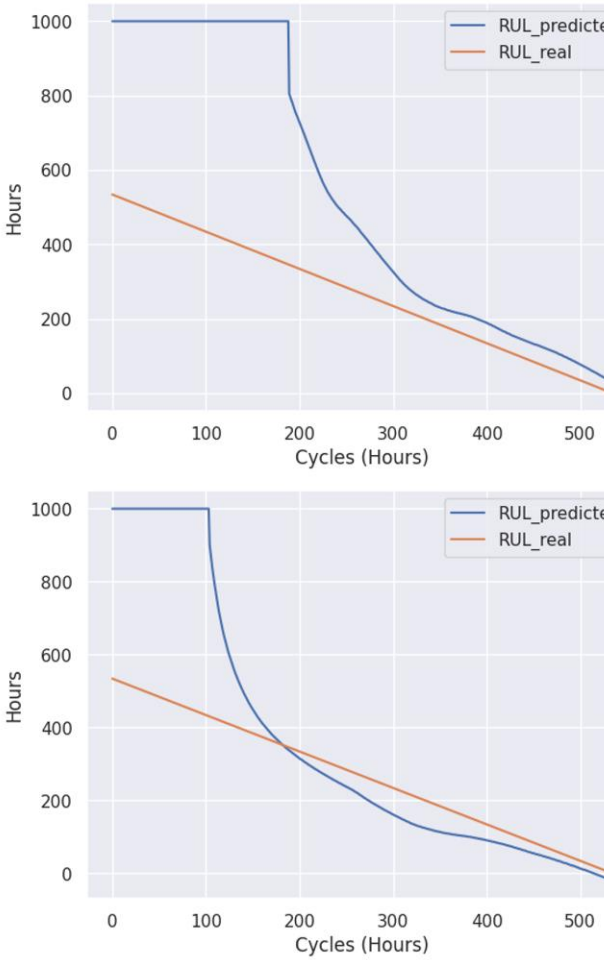

Fig. 6. Comparison of RUL estimations in two scenarios. When 20% of nodes are byzantine without any detection module(above), and when our detection module is used (below).

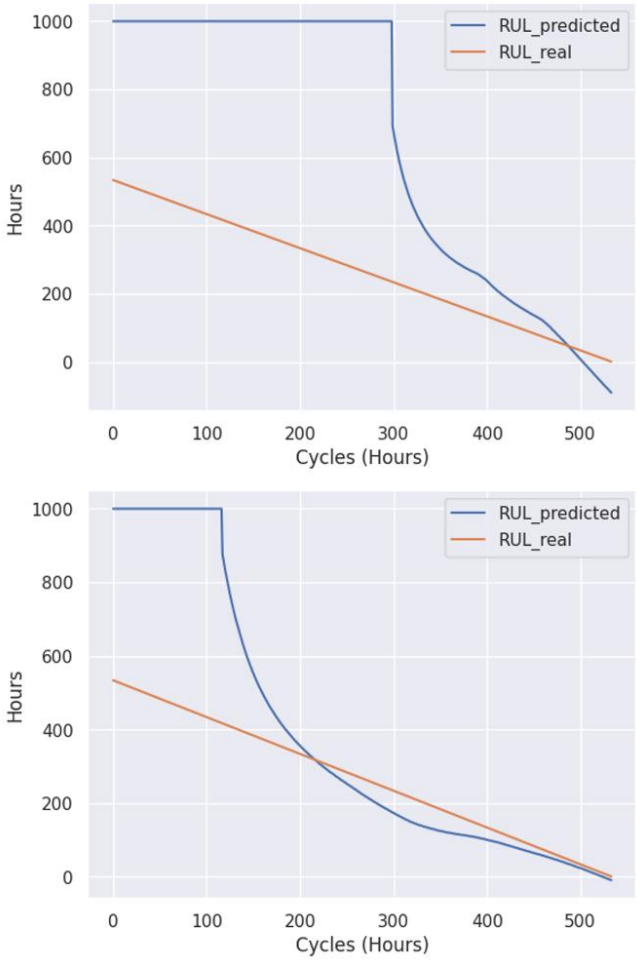

Fig. 7. Comparison of RUL estimations in two scenarios. When 40% of nodes are byzantine without any detection module(above), and when our detection module is used (below).

TABLE III COMPARISON OF THE RMSE RESULTS OF THE FL ALGORITHM EVALUATION WITH THE NASA ACOUSTICS AND VIBRATION DATASET FOR DIFFERENT PORTIONS OF BYZANTINE NODES, WITH AND WITHOUT THE ANOMALY DETECTION MODULE.

Scenarios	Byzantine = No Byzantine	Byzantine = 0.2	Byzantine = 0.4
Without anomaly detection	67.43	149.85	409.06
Krum aggregation [10]	67.43	116.19	166.98
Trimmed-Mean [11]	67.43	163.41	175.31
One-Shot Federated Learning with PCA-based detection	67.43	158.1	173.9
Our Proposed Framework	**67.43**	**78.21**	**97.01**

C. Comparison with One-Shot PCA-FL

To provide a comprehensive performance comparison, we also implemented a one-shot federated learning approach that uses PCA-based anomaly detection only in a single round—without historical scoring or incentivization. This method performs anomaly detection at the first round, filters out suspected updates, and aggregates the remaining updates once to form the global model.

The results showed that while the one-shot approach is faster due to its non-iterative structure, its performance in terms of accuracy and Byzantine resilience was significantly lower than our proposed multi-round BFT-FL framework. For example, in the AI4I 2020 dataset with 40% Byzantine nodes, the one-shot method achieved an accuracy of ~82.3%, compared to 89.83% for our proposed approach. Similarly, on the NASA Vibration dataset, the RMSE in RUL estimation was ~173.9 for the one-shot method, whereas our framework achieved 97.01.

These results confirm that while one-shot PCA-based FL may offer slight computational advantages, it lacks the robustness and adaptability of our iterative design, especially in dynamic or adversarial environments.

D. Convergence Analysis

We also examined the impact of our proposed BFT-FL framework on the convergence time of the federated learning process. Convergence was defined as the point at which the global model's validation accuracy stabilized with minimal fluctuation. Figure 8 shows the convergence time analysis of our proposed method. In the scenario with no Byzantine nodes on the AI4I 2020 dataset, the baseline FL without any anomaly detection converged in 8 rounds with an average accuracy of ~90%. Our proposed method reached convergence in 9 rounds but achieved a significantly the accuracy of ~89.8% in the scenario with 40% Byzantine nodes. Although the convergence required one additional round, the trade-off is justified by the improved model robustness and reliability. This indicates that the proposed enhancements introduce only a negligible delay in convergence while providing substantial gains in resilience.

E. Computational Cost Analysis

To support our claim that the proposed framework is computationally lightweight, we measured the average runtime of the federated training process across five global rounds with ten clients. The total runtime with the PCA-based anomaly detection module enabled was 985.0410 seconds, while the runtime without the detection module was 980.7393 seconds. This demonstrates that the additional computational cost introduced by our detection mechanism is minimal—approximately 4.3 seconds over five rounds, or less than 1% overhead. Such a negligible increase confirms the efficiency of our design and shows that the anomaly detection process can be integrated into industrial PDM systems without introducing significant latency or resource burden.

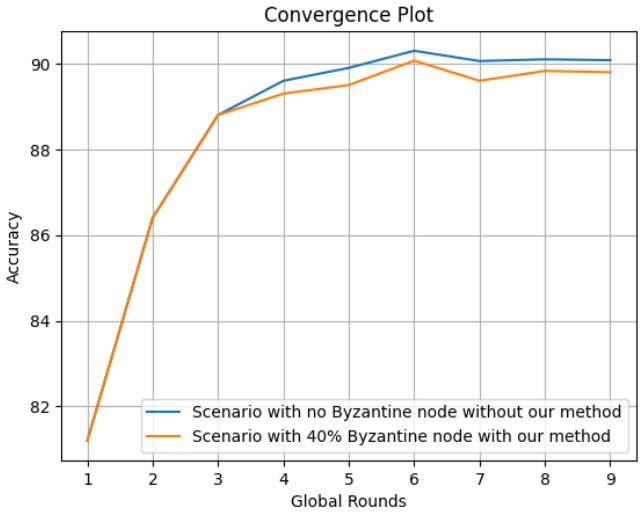

Fig. 8. The convergence time analysis of our proposed method in comparison with the scenario which our method is not used.

V. CONCLUSION AND FUTURE WORK

In this paper, we proposed a novel Byzantine Fault-Tolerant Federated Learning framework, designed improve the reliability of PDM application in industry. The proposed frame- work employed a PCA-based anomaly detection algorithm to detect and mitigate Byzantine local updates, ensuring the integrity and accuracy of the global model. Additionally, we incorporated a game theory-based incentive mechanism to motivate normal behavior and deter malicious behavior among local clients.

The results of evaluating our framework using the "AI4I 2020" and "NASA Acoustics and Vibration Predictive Maintenance" Datasets showed significant improvements in fault tolerance and so model performance.

Our approach provides a secure and scalable solution for maintaining federated learning models' performance, security, and integrity in critical industrial applications. By ad- dressing the challenges associated with Byzantine faults and promoting honest participation through incentivization, our framework ensures that the benefits of federated learning can be realized even in adversarial environments.

While our proposed framework has shown promising results, several avenues for future research and improvements exist, including extending to more complex datasets, enhancing anomaly detection, scalability, and efficiency, real-world implementation, exploring other incentive mechanisms, and security enhancements.

The current study assumes an ideal communication channel between clients and the central server. In practice, communication noise (e.g., due to fading or interference) can corrupt transmitted model updates. We plan to investigate how such non-idealities affect the integrity of weight transmission and evaluate whether our PCA-based anomaly

detection mechanism can also mitigate channel-induced distortions. Exploring the use of error-correcting codes or robust compression techniques will be an important step in enhancing the framework's resilience.

By addressing these future directions, we can continue to advance secure federated learning and its applications in industrial predictive maintenance, ensuring that these systems are resilient, reliable, and capable of delivering accurate and timely insights.

Strengthening the PCA-based anomaly detection module to detect more complex Byzantine scenarios and exploring more advanced reward/penalty mechanisms can be considered as our future work.

It is worth noting that this approach has been able to lead to significant performance improvements in a real industrial PDM scenario. Specifically, the proposed model has shown a 16% improvement in prediction accuracy and a 76% reduction in RMSE compared to the baseline methods. These results indicate the high potential of the proposed method for application in industries to more accurately and quickly detect potential equipment failures.

REFERENCES

[1] H. Brendan McMahan, E. Moore, D. Ramage, S. Hampson, B. Agüera y Arcas, "Communication-Efficient Learning of Deep Networks from Decentralized Data," Proceedings of the 20 th International Conference on Artificial Intelligence and Statistics (AISTATS) 2017. JMLR: W&CP volume 54, last revised 26 Jan 2023 (v4), doi: https://doi.org/10.48550/arXiv.1602.05629.

[2] L. Li, Y. Fan, M. Tse, and K.-Y. Lin, "A review of applications in federated learning," Computers & Industrial Engineering, vol. 149, p. 106854, 2020, doi: https://doi.org/10.1016/j.cie.2020.106854.

[3] J. Zhang, H. Zhu, F. Wang, J. Zhao, Q. Xu, and H. Li, "Security and Privacy Threats to Federated Learning: Issues, Methods, and Challenges," Security and Communication Networks, vol. 2022, no. 1, p. 2886795, 2022, doi: https://doi.org/10.1155/2022/2886795.

[4] K. Jahani, B. Moshiri and B. Hossein Khalaj, "PPFL: Privacy-Preserving Techniques in Federated Learning", JAIAI, vol. 1, no. 3, pp. 49–67, Jul. 2024, doi: 10.61838/jaiai.1.3.6.

[5] Q. Xia, W. Ye, Z. Tao, J. Wu, and Q. Li, "A survey of federated learning for edge computing: Research problems and solutions," High-Confidence Computing, vol. 1, no. 1, p. 100008, 2021, doi: 10.1016/j.hcc.2021.100008.

[6] J. Shi, W. Wan, S. Hu, J. Lu, L. Y. Zhang, "Challenges and Approaches for Mitigating Byzantine Attacks in Federated Learning," vol. cs. CR,2022, doi: https://arxiv.org/abs/2112.14468.

[7] A. Gouissem, K. Abualsaud, E. Yaacoub, T. Khattab and M. Guizani, "Federated Learning Stability Under Byzantine Attacks," 2022 IEEE Wireless Communications and Networking Conference (WCNC), Austin, TX, USA, 2022, pp. 572-577, doi: 10.1109/WCNC51071.2022.9771594.

[8] M. Fang, X. Cao, J. Jia, and N. Z. Gong, "Local model poisoning attacks to byzantine-robust federated learning," in 29th USENIX Security Symposium, ed: USENIX Association, 2020, pp. 1623–1640.

[9] T. Zhu Z. Guo, C. Yao, J. Tan, S. Dou, W. Wang, Z. Han, "Byzantine-robust Federated Learning via Cosine Similarity Aggregation," Computer Networks, vol. 254, p. 110730, 2024, doi: https://doi.org/10.1016/j.comnet.2024.110730.

[10] P. Blanchard, E. M. El Mhamdi, R. Guerraoui, and J. Stainer, "Machine Learning with Adversaries: Byzantine Tolerant Gradient Descent," NIPS'17: Proceedings of the 31st International Conference on Neural Information Processing Systems Pages 118 - 128, 2017.

[11] D. Yin, Y. Chen, R. Kannan, and P. Bartlett, "Byzantine-Robust Distributed Learning: Towards Optimal Statistical Rates," presented at the Proceedings of the 35th International Conference on Machine Learning, Proceedings of Machine Learning Research, 2018. [Online]. Available: https://proceedings.mlr.press/v80/yin18a.html.

[12] Z. Li, H. Yu, T. Zhou, L. Luo, M. Fan, Z. Xu, G. Sun "Byzantine Resistant Secure Blockchained Federated Learning at the Edge," IEEE Network, vol. 35, no. 4, pp. 295–301, 2021, doi: 10.1109/MNET.011.2000604.

[13] Y. Miao, Z. Liu, H. Li, K. K. R. Choo, and R. H. Deng, "Privacy-Preserving Byzantine-Robust Federated Learning via Blockchain Systems," IEEE Transactions on Information Forensics and Security, vol. 17, pp. 2848–2861, 2022, doi: 10.1109/TIFS.2022.3196274.

[14] J. H. Chen, M. R. Chen, G. Q. Zeng, and J. S. Weng, "BDFL: A Byzantine-Fault-Tolerance Decentralized Federated Learning Method for Autonomous Vehicle," IEEE Transactions on Vehicular Technology, vol. 70, no. 9, pp. 8639–8652, 2021, doi: 10.1109/TVT.2021.3102121.

[15] X. Luo and B. Tang, "Byzantine Fault-Tolerant Federated Learning Based on Trustworthy Data and Historical Information," Electronics, vol. 13, no. 8, 2024, doi: 10.3390/electronics13081540.

[16] J. So, B. Güler, and A. S. Avestimehr, "Byzantine-Resilient Secure Federated Learning," IEEE Journal on Selected Areas in Communications, vol. 39, no. 7, pp. 2168–2181, 2021, doi: 10.1109/JSAC.2020.3041404.

[17] X. Lin, Y. Li, X. Xie, Y. Ding, X. Wu, and C. Ge, "SF-CABD: Secure Byzantine fault tolerance federated learning on Non-IID data," Knowledge-Based Systems, vol. 296, p. 111851, 2024, doi: https://doi.org/10.1016/j.knosys.2024.111851.

[18] X. Tang, H. Gu, L. Fan, and Q. Yang, "Achieving Provable Byzantine Fault-tolerance in a Semi-honest Federated Learning Setting," in Advances in Knowledge Discovery and Data Mining, Cham, H. Kashima, T. Ide, and W.-C. Peng, Eds., 2023: Springer, https://doi.org/10.1007/978-3-031-33377-4_32.

[19] Y. Tao, S. Cui, W. Xu, H. Yin, D. Yu, W. Liang, X. Cheng, "Byzantine-Resilient Federated Learning at Edge," IEEE Transactions on Computers, vol. 72, no. 9, pp. 2600–2614, 2023, doi: 10.1109/TC.2023.3257510.

[20] N. Gupta, T. T. Doan, and N. Vaidya, "Byzantine Fault-Tolerance in Federated Local SGD Under 2f-Redundancy," IEEE Transactions on Control of Network Systems, vol. 10, no. 4, pp. 1669–1681, 2023, doi: 10.1109/TCNS.2023.3237489.

Khalil Jahani holds a B.Sc. (2012) and M.Sc. (2014) in Information Technology Engineering – Computer Networks from IUST, Tehran. He is currently a Ph.D. candidate in Computer Science – Artificial Intelligence at the University of Tehran, having commenced in 2020. An experienced researcher and developer, he possesses extensive expertise in algorithm design for machine learning, AI, signal processing, and computer vision. As a Senior Algorithm Developer in aviation, he specializes in deep learning applications for anomaly detection, condition monitoring, PHM, and predictive maintenance. He earned AML CAT.B1 (A&P) and CAT.B2 (Avionics) licenses from Iran's Civil Aviation Organization in 2019.

Behzad Moshiri (IEEE Senior Member) received his B.Sc. degree in mechanical engineering from Iran University of Science and Technology (IUST) in 1984 and M.Sc. and Ph.D. in control systems engineering from the University of Manchester, Institute of Science and Technology (UMIST), U.K. in 1987 and 1991, respectively. He has been senior member of IEEE since 2006. He is the author/co-author of more than 360+ articles. He has been an adjunct professor of the Department of ECE at the University of Waterloo since May 2014. He has been a member of "Waterloo AI Institute" since 2018. His research fields include advanced industrial control, advanced instrumentation systems, data fusion theory, and feasibility studies on applications and implementations of sensor/data fusion.

Babak Hossein Khalaj (IEEE Senior Member) received the B.Sc. degree in electrical Engineering from the Sharif University of Technology, Tehran, Iran, in 1989, and the M.Sc. and Ph.D. degrees in electrical engineering from Stanford University, Stanford, CA, USA, in 1993 and 1996, respectively. He has been with the pioneering team at Stanford University, where he was involved in adopting multi-antenna arrays in mobile networks. Since 1999, he has been a Senior Consultant in data communications and a Visiting Professor with CEIT, San Sebastian, Spain, from 2006 to 2007. He has coauthored many papers in signal processing and digital communications and holds four U.S. patents. He received the Alexander von Humboldt Fellowship from 2007 to 2008 and the Nokia Visiting Professor Scholarship in 2018.