Novel Ensemble Domain Adaptation Methodology for Enhanced Multi-class Fault Diagnosis of Highly-Connected Fleet of Assets

Takanobu Minami[1], Alexander Suer[2], Pradeep Kundu[3], Shahin Siahpour[4], and Jay Lee[5]

[1,5]*Center for Industrial Artificial Intelligence, Department of Mechanical Engineering, University of Maryland, MD, USA*
minamitu@umd.edu, leejay@umd.edu

[1]*Komatsu Ltd., Tokyo, Japan*

[2,4]*Center for Intelligent Maintenance Systems, Department of Mechanical and Materials Engineering, University of Cincinnati, OH, USA*
suerad@mail.uc.edu, siahposn@mail.uc.edu

[3]*Department of Mechanical Engineering, KU Leuven, 8200 Bruges, Belgium*
pradeep.kundu@kuleuven.be

ABSTRACT

This paper proposes a novel methodology for enhancing multi-class classification accuracy in fault diagnosis problems among domains with highly-connected fleets of assets using time series data. The approach involves appending specially tailored models to an initial model and incorporating domain adaptation techniques to account for domain variations. The methodology is demonstrated through a case study on fault diagnosis of a fleet of hydraulic rock drills, which presents challenges due to variations in sensor data between different fault classes and individual machines. Results show significant improvements in classification accuracy, both in validation and testing, upon employing ensemble models and applying domain adaptation. While the study is limited to one case study, it lays the groundwork for exploring the applicability of the proposed methodology in other contexts.

1. INTRODUCTION

In the operation of complex mechanical systems, addressing multi-class classification problems poses a significant challenge. The need to differentiate between various fault classes and account for data variability across different domains, such as individual machines and entire fleets, calls for a robust and adaptable methodology. Consequently, researchers and engineers are motivated to develop and apply innovative strategies to tackle these issues in a practical and efficient manner.

Numerous studies have endeavored to address the multi-class classification problem in fault diagnosis for complex mechanical systems. Deep learning (DL) has emerged as a major trend in this field. For instance, Duan, Xie, Wang, and Bai (2018) reviewed the effectiveness of DL models, such as convolutional neural network (CNN), Deep Belief Network (DBN), or autoencoder (AE), in handling multi-class fault diagnosis of machinery problems compared to conventional machine learning (ML) techniques. Furthermore, researchers are working on improving DL performance through ensemble methods. Chen, Gryllias, and Li (2019), for example, integrate CNN with Extreme Learning Machine (ELM) and demonstrate outstanding performance both on a gearbox and a motor bearing classification problem. Ma and Chu (2019) utilized an ensemble learning method with that integrated multiple classifiers and demonstrated superior performance on rotor-bearing fault diagnosis

Despite these advancements, several limitations persist in the aforementioned methodologies. DL methods require more labeled data per class, risk overfitting, and are computationally expensive. Ensemble learning, on the other hand, may result in less interpretable models and increased computational complexity. A method that balances these drawbacks and benefits is needed.

In this study, we propose a novel methodology that addresses the drawbacks mentioned above by combining the strengths of existing approaches while mitigating their limitations. We focus on classes with low accuracy in the conventional ML model and adapt DL techniques to them. This strategy improves the data-intensive, overfitting risk, and computationally expensive aspects of DL. Furthermore, the

DL model handles a small part of the problem, minimizing the deterioration of the model's interpretability. Additionally, by incorporating domain adaptation (DA) in DL models, our methodology effectively accounts for individual differences and ensures the generalization of the model. Our approach integrates ensemble learning, DA, and DL for binary classification, which simplifies the problem and potentially increases accuracy.

2. METHODOLOGY

This paper proposes a methodology, depicted in Figure 1, that aims to improve the precision of conventional ML models for multi-class classification problems with diverse data sources. In this method, the baseline model, the initial model, is built with conventional ML methods, and then sub-models that utilize DL are added. Specifically, the sub-models focus solely on binary-class classification for target classes in which the baseline model performs poorly, while the baseline model handles the entire multi-class classification. Furthermore, DA techniques are employed to bridge discrepancies among different domains in datasets, utilizing healthy class samples available in all domains. The next section provides a detailed explanation of this methodology.

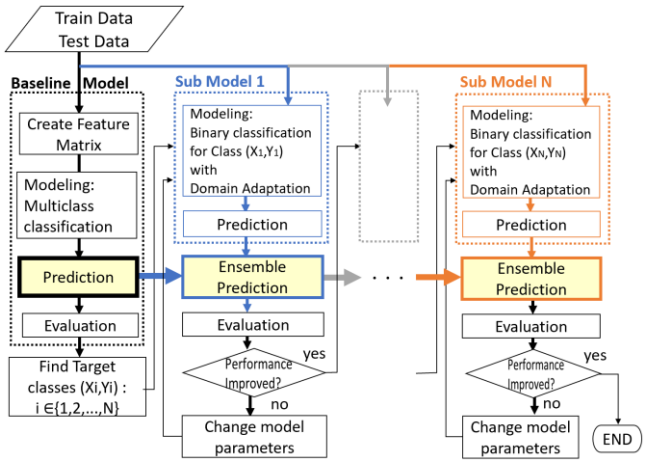

Figure 1 Proposed Methodology

2.1. Problem statement

This paper addresses a challenging semi-supervised multi-class and multi-domain fault diagnosis problem based on time series data. The problem is given as a training dataset with n_{train} labeled samples $\{(\boldsymbol{x}_i^{train}, y_i^{train}, d_i^{train})\}_{i=1}^{n_{train}}$ and a testing dataset with n_{test} labeled samples in which only a part of healthy class data are labeled $\{(\boldsymbol{x}_i^{test}, y_i^{test}, d_i^{test})\}_{i=1}^{n_{test}}$,where $\boldsymbol{x}$, y and d stand for data samples, associated labels, and domains to which data belong, respectively. The training and testing data are potentially sourced from different domains, such as machines or working conditions, making classification challenging.

2.2. Baseline Modeling with Conventional DL

The baseline model, often employed in research papers for comparison, refers to a model built using well-established traditional machine learning methods. It is constructed through a process that includes feature extraction, feature selection, and the application of machine learning libraries. Steps include transforming raw data into features, eliminating low-importance features using the recursive feature elimination method (Guyon, Weston, Barmhill, and Vapnik, 2002), building models with popular algorithms like XGBoost (Chen & Guestrin, 2016) and Support Vector Machine (SVM), and selecting the best model based on performance evaluation. Feature engineering includes time-domain statistical features, frequency-domain features, and Dynamic Time Warping (DTW) for time-series data analysis (Berndt & Clifford, 1994). After feature extraction, a feature selection process is conducted to find the best feature subset. Modeling algorithms are compared, and the best one is selected based on a scoring metric.

2.3. Ensemble Modeling with Domain Adaptation

The proposed methodology enhances the baseline model by constructing specialized sub-models for underperforming classifications. This approach simplifies the problem and potentially boosts accuracy by creating binary classification sub-models for target classes identified through cross-validation. These sub-models use raw data as input and apply domain adaptation techniques to address individual differences. The process continues until all sub-models for target classes are built, with predictions updated using an ensemble rule.

The first step in our ensemble approach is to find classes that need to be improved, classes for which the baseline model performs poorly. Target classes for sub-models are determined from the results of the cross-validation of the baseline. The following algorithm selects these targets.

Algorithm 1: Target Classes

INPUT Sum of all confusion matrix of validation; Conf (x,y)
Number of Classes; Nc, Threshold value; α min
For i from 1 to Nc do:
  j = $\underset{j,j\neq i}{\operatorname{argmax}}\,\mathrm{Conf}(j,i)$
  Dataframe[X1(i)] = min(i,j)
  Dataframe[X2(i)] = max(i,j)
  Dataframe[α (i)] = Conf(j,i)
  Dataframe[C(i)] = i
For i from 1 to Nc-1 do:
  For j from i+1 to Nc do:
    If (X1(i) = X1(j) and X2(i) = X2(j)) then:
      α (i) = α (i) + α (j)
      Delete row j from Dataframe
      Add j to C(i)
Sort Dataframe in descending order with respect to α
Reset index as No. of sub-models
For i from 1 to Nc do:
    In Dataframe, if α (i) < α min then delete row i
END

Here, the pair of two classes $(X1_i, X2_i)$ are the target classes, and sub-models will be built for them. α_i indicates the upper limit of accuracy that can be improved by the sub-model for $(X1_i, X2_i)$. A pair of two target classes with high α has a high possibility of improving accuracy, while a sub-model with a low α has a low chance of improving accuracy. Therefore, in this proposed methodology, a lower limit α_{min} is set, and sub-models below this limit (models with a low possibility for improvement) are eliminated from the candidates. This threshold will save computational resources as well as time to tune the sub-models, making our method efficient.

Once the target classes are calculated, then related sub-models are created. Figure 2 shows the structure of a sub-model: the sub-model inputs both raw training data and testing data into the feature extractor. Extracted features are inputted into the classifier, which then outputs the predictions for each data. A classification loss is calculated using the predictions from training data of target classes. The classification loss is a measure of how far away from the true labels the predicted labels are, with smaller values indicating more accurate predictions.

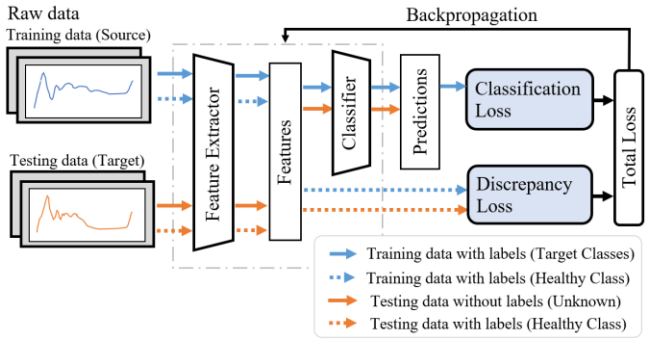

Figure 2 Structure of sub-models

Domain adaptation techniques are also used to account for differences between domains. This involves minimizing classification loss, such as cross-entropy loss, commonly used as a loss function in multi-class classification, and discrepancy loss, such as maximum mean discrepancy (MMD) (Gretton, Borgwardt, Rasch, Schölkopf, and Smola, 2012), to extract domain-independent features. The total loss is calculated by combining these two losses, enabling the classifier to effectively handle data from different domains.

Once a sub-model is built, the prediction from the model is obtained, and then the new prediction is given by the ensemble with the old prediction. This updated prediction is evaluated, and if the accuracy improves, then the next sub-model is created.

3. Case Study: Fault Diagnosis of Rock Drills

3.1. Data Description and Problem Statement

The proposed methodology is validated by solving a multi-class classification problem, specifically the fault diagnosis of multiple rock drills using data from PHM North America Data Challenge 2022 (PHM Society, 2022). The dataset provided for this data challenge consists of six individual drills, which can be considered as different domains, with eleven distinct operating conditions, including one no-fault and ten different faulty conditions.

This case study aims to diagnose faults in unknown drills using data from known ones, but the challenge is that there are variations in sensor data between both different fault classes and domains. Pressure oscillation patterns differ more significantly among drills in the same fault class than among different fault classes in the same individual. As a reference, some of no-fault data are available among the unknown drill, and the model is expected to learn the differences between drills using this data

3.2. Modeling

As the first procedure of modeling, pressure signals are divided into two working condition regimes based on pressure surge, and then features are extracted from both of these, statistical features, frequency domain features with FFT, and wavelet analysis. Also, the DTW technique has been used (Jakobsson, Krysander, and Pettersson, 2022)

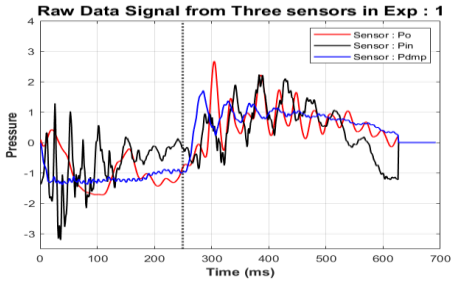

Figure 3 Raw signal split for feature extraction

SVM, Random Forest, and XGBoost algorithms are selected as baseline models, and the best one is chosen. CNN is utilized as a feature extractor and classifier for sub-models. Table 1 displays the network structure, which consists of two convolutional layers.

Table 1 Network Architecture

Module	Layer	Operator and Parameters
Feature Extractor	Conv 1	filter:30, kernel:5, strid:1, ReLu
	Conv 2	filter:30, kernel:5, strid:1, ReLu
	MaxPooling	2x2
	Flatten	-
	FC 1	Note: 128, ReLu
Classifier	FC 2	Note: Number of Classes

3.3. Evaluation Metric

The data from one of the six drills are used as test data, while the remaining five are used as training data. The model is evaluated using 5-fold cross-validation on the training data, and the average accuracy is calculated to compare performance. The accuracy of the test data is also compared.

4. Results and Discussion

4.1. Features and Model Selection

The best features are selected based on backward feature elimination with feature importance given by XGBoost. Table 2 presents the final list of sixteen features. Then, model selection has proceeded. Table 3 displays the results from three algorithms. The XGBoost classifier is selected as the baseline model, as it exhibits the best validation accuracy.

Table 2 Best selected features

Rank	Sensor	Feature	Types of features
1	Pin	DTW: Peak to Peak of $\delta_{ts}(x_{NF}, x_{sample})$	DTW
2	Pin	Median of the magnitude of the FFT	Frequency domain
3	Pdmp	Percentage energy of the 16th wavelet node	Wavelet
4	Pin	DTW: Average distance p_{amp} for the class 1	DTW
5	Pin	Minimum of the sample	Time domain
6	Po	Median of the sample	Time domain
7	Po	DTW: Average distance p_{ts} for the class 2	DTW
8	Pdmp	DTW: Average distance p_{ts} for the class 6	DTW
9	Pdmp	Maximum of the sample	Time domain
10	Pin	Maximum of the sample	Time domain
11	Pin	Maximum of the magnitude of the FFT	Frequency domain
12	Pdmp	RMS of the second half of the signal	Time domain
13	Po	Maximum of the magnitude of the FFT	Frequency domain
14	Pdmp	Minimum of the sample	Time domain
15	Po	Percentage energy of the 1st wavelet node	Wavelet
16	Po	DTW: Average distance p_{amp} for the class 8	DTW

Table 3 Conventional ML accuracies

Feature set	SVM	Random Forest	XGBoost
All	0.824	0.868	0.900
Best	0.901	0.920	**0.923**

4.2. Model Performance

Figure 4 shows the detailed results of all validations by the baseline model. Five subfigures show the confusion matrices and the elements, the amount of data. Overall, the accuracies are a minimum of 0.867 and a maximum of 0.958.

Here some mislabeled patterns can be observed. One primary mislabel pattern is that the data with true label 8 is predicted as label 7. There are 47 to 355 mislabels in validations. Similarly, the data with true label 7 predicted as label 8 is also an obvious mislabel pattern. From 19 to 394 mislabels exist in the validations. Thus, it is possible to improve the overall performance by reducing these errors.

To determine the target class for sub-models, the impact of reducing these errors is assessed. Fig. 5 shows how the targets are determined by the algorithm 1. Firstly, confusion matrixes from all validations are summed up, and then each element of the matrix is divided by the total number of data (Fig. 5(a)). Maximum elements are selected from each column which we can see as red bold outside borders. These percentages indicate how much the prediction accuracy improved if the mislabels completely solved. For instance, if one overcomes the mislabels that data with true label 7 predicted as label 8, the accuracy would be improved by 2.1 %. Secondly, based on these percentages, the target classes of the sub-models are determined. Here, we set α_{min}= 1% ,and then two pairs of target classes (7,8) and (1,5) are selected (Fig. 5(b))

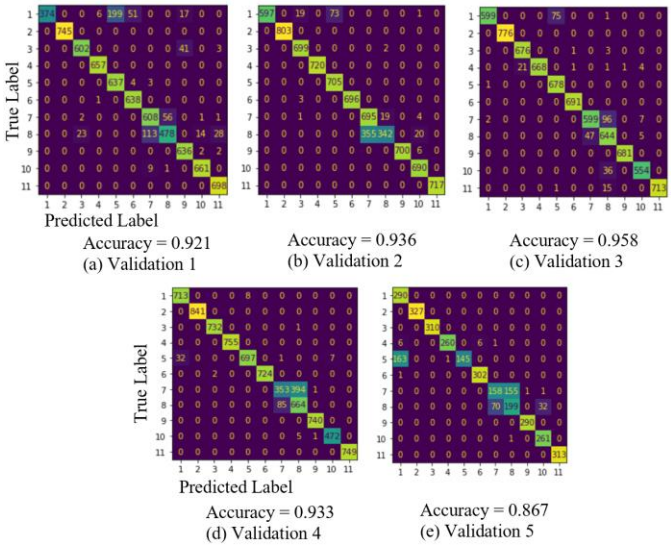

Figure 4 Validation Results of Baseline Model

(a) Confusion Matrix (Average of Five Validations)

True lables \ Predicted labels	1	2	3	4	5	6	7	8	9	10	11
1	7.6%	0.0%	0.1%	0.0%	1.0%	0.1%	0.0%	0.0%	0.0%	0.0%	0.0%
2	0.0%	10.3%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
3	0.0%	0.0%	8.9%	0.0%	0.0%	0.0%	0.0%	0.0%	0.1%	0.0%	0.0%
4	0.0%	0.0%	0.1%	9.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
5	0.6%	0.0%	0.0%	0.0%	8.4%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
6	0.0%	0.0%	0.0%	0.0%	0.0%	9.0%	0.0%	0.0%	0.0%	0.0%	0.0%
7	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	7.1%	2.1%	0.0%	0.0%	0.0%
8	0.0%	0.0%	0.1%	0.0%	0.0%	0.0%	2.0%	6.8%	0.0%	0.2%	0.1%
9	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	8.9%	0.0%	0.0%
10	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.1%	0.0%	7.7%	0.0%
11	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	9.4%

(b) Selection of Sub-models

Sub-model No.	j	1	2	3	4	5	6	7	8	9
Target Classes	X1j	7	**1**	8	1	3	8	3	1	1
	X2j	**8**	**5**	10	6	9	11	4	4	2
Max. improve	αj	**4.1%**	**1.6%**	0.2%	0.1%	0.1%	0.1%	0.1%	0.0%	0.0%
Target labels	Cj	7,8	1,5	10	4	9	11	3	4	2

Figure 5 Process of Selecting Target Classes

Table 4 provides an example of how the sub-models reduce the number of mislabels, illustrating the results from validation 4. It demonstrates that predictions with the

addition of sub-model 1 yield fewer mislabels compared to using only the baseline model. Likewise, incorporating sub-model 2 leads to the correct classification of mislabels between classes 1 and 5, futher enhancing overall accuracy.

Table 5 presents the comprehensive results of both validations and testing. The inclusion of sub-models increases all accuracy metrics. Consequently, the model developed using the proposed methodology performed superior performance. This result validated the effectiveness of our methodology.

Table 4 Mislabels reduced by adding sub-models

True Label	Predicted Label	Baseline model	Add Sub-model 1	Add Sub model 1,2
'7'	'8'	394	**0**	0
'8'	'7'	85	**8**	8
'1'	'5'	8	8	**0**
'5'	'1'	32	32	**0**

Table 5 Accuracy of validation and testing
(Average accuracy ± Standard deviation)

	Model	Validation Accuracy	Test Accuracy
1	Baseline model	0.923±0.034	0.932
2	Base+Sub-model 1	0.947±0.046	0.992
3	Base+Sub-model 1,2	**0.959**±0.038	**0.993**

4.3. Discussion

The proposed methodology incorporates DA to enhance prediction accuracy while considering individual differences. A study is conducted to evaluate the effectiveness of this approach. Table 6 presents a comparison between utilizing DA in the sub-models and not using it. The accuracies with DA are equal to or better than those without it at any level. This finding implies that DA effectively mitigates bias between training and testing data, resulting in a more generalized model. Based on these outcomes, we conclude that DA is effective for addressing the problem addressed in this paper.

Table 6 Sub-models with/without DA

No	Model	DA	Validation	Test
1	Baseline model	—	0.923±0.034	0.932
2	Base + Sub-model 1,2	Yes	**0.959**±0.038	**0.993**
3		No	0.940±0.059	**0.993**

A limitation for this study is that the proposed methodology was tested solely on a hydraulic rock drill system, leaving its effectiveness on other machine systems uncertain. Further research is necessary to evaluate the performance of the proposed methodology in other scenarios. Furthermore, this case study employs a simple CNN as the algorithm for the sub-models, though more advanced deep learning architectures might offer improved performance. Developing an efficient process for building and selecting appropriate deep learning algorithms could potentially yield even higher accuracy.

5. CONCLUSION

In conclusion, this paper proposed a practical methodology for achieving elevated precision in a multi-class classification problem by appending DL based sub-models to a conventional ML based baseline model and incorporating DA techniques. The approach was validated through a case study on the fault diagnosis of rock drills, where the dataset included variations in sensor data between different fault classes and different domains. Overall, the proposed methodology offers a practical solution for improving classification accuracy in semi-supervised multi-class fault diagnosis problems based on time series data.

REFERENCES

Berndt, D. J., & Clifford, J. (1994, July). Using dynamic time warping to find patterns in time series. *In KDD workshop* (Vol. 10, No. 16, pp. 359-370).

Chen, T., & Guestrin, C. (2016, August). Xgboost: A scalable tree boosting system. *In Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining* (pp. 785-794).

Chen, Z., Gryllias, K., & Li, W. (2019). Mechanical fault diagnosis using convolutional neural networks and extreme learning machine. *Mechanical systems and signal processing*, 133, 106272.

Duan, L., Xie, M., Wang, J., & Bai, T. (2018). Deep learning enabled intelligent fault diagnosis: Overview and applications. *Journal of Intelligent & Fuzzy Systems*, 35(5), 5771-5784.

Gretton, A., Borgwardt, K. M., Rasch, M. J., Schölkopf, B., & Smola, A. (2012). A kernel two-sample test. *The Journal of Machine Learning Research*, 13(1), 723-773.

Guyon, I., Weston, J., Barnhill, S., & Vapnik, V. (2002). Gene selection for cancer classification using support vector machines. *Machine learning*, 46, 389-422.

He, J., Li, X., Chen, Y., Chen, D., Guo, J., & Zhou, Y. (2021). Deep transfer learning method based on 1d-cnn for bearing fault diagnosis. *Shock and Vibration*, 2021.

Jakobsson, E., Frisk, E., Krysander, M., & Pettersson, R. (2022). Time Series Fault Classification for Wave Propagation Systems with Sparse Fault Data. *arXiv* preprint arXiv:2203.16121.

Ma, S., & Chu, F. (2019). Ensemble deep learning-based fault diagnosis of rotor bearing systems. Computers in industry, 105, 143-152.

PHM Society. 2022 PHM Conference Data Challenge - PHM Society Data Repository. (2022, July 30). https://data.phmsociety.org/2022-phm-conference-data-challenge

Takanobu Minami received his B.S. and M.S. degree in mechanical engineering from Kyoto University in 2008 and in 2011, respectively. Currently, he is pursuing his Ph.D. degree in mechanical engineering with the University of Maryland, College Park, MD, USA and employed as engineer in Komatsu Ltd. His research interests include machine learning, deep learning, prognostics and health management, and industrial A.I.

Alexander Suer received his B.S. in mechanical engineering from the University of Cincinnati, Cincinnati, OH, USA in 2022. He is pursuing his M.S. and Ph.D. degree in mechanical engineering with the University of Cincinnati. His research interests include machine learning, prognostics and health management, industrial A.I., and robotics.

Pradeep Kundu has been working as an Assistant Professor at the Department of Mechanical Engineering at KU Leuven, Belgium. Before joining KU Leuven, he worked as a Post Postdoctoral Fellow and Research Associate at the IMS Center, University of Cincinnati, USA and DMEM, University of Strathclyde, UK, respectively. Pradeep completed his PhD in 2020 in the specialization of Prognosis and Health Management from the Department of Mechanical Engineering at the Indian Institute of Technology (IIT) Delhi, India. In 2019, he received a Visiting Research Fellow grant from SERB, Govt. of India, to carry out his PhD research work at the University of Alberta, Canada. Pradeep has published more than 30 articles in reputed academic journals and conferences. His research interests are Industrial Artificial Intelligence, Digital Twins, Smart Manufacturing, Fault Diagnosis, Fault Prognosis and Reliability Engineering.

Shahin Siahpour received the B.S. degree in mechanical engineering from Shiraz University, Shiraz, Iran, in 2015, and the M.S. degree in mechanical engineering from the University of Tehran, Tehran, Iran, in 2017. He is currently pursuing the Ph.D. degree in mechanical engineering with the University of Cincinnati, Cincinnati, OH, USA. His research interests include deep learning, prognostics, health management, and industrial artificial intelligence.

Jay Lee is Clark Distinguished Professor and Director of Industrial AI Center in the Mechanical Engineering Dept. of the Univ. of Maryland College Park. His research is focused on intelligent analytics of complex systems including highly-connected industrial systems including energy, manufacturing, and healthcare/medical, etc. He has been working with medical school in Traumatic Brain Injury (TBI) using multi-dimension data for predictive assessment of patient in ICU with funding from NIH and NSF. Previously, he served as an Ohio Eminent Scholar, L.W. Scott Alter Chair and Univ. Distinguished Professor at Univ. of Cincinnati. He was Founding Director of National Science Foundation (NSF) Industry/University Cooperative Research Center (I/UCRC) on Intelligent Maintenance Systems during 2001-2019. IMS Center pioneered industrial Ai-augmented prognostics technologies for highly-connected industrial systems and has developed research memberships with over 100 global company since 2000 and was selected as the most economically impactful I/UCRC in the NSF Economic Impact Study Report in 2012. He is also the Founding Director of Industrial AI Center. He mentored his students and developed a number of start-up companies including Predictronics through NSF iCorp in 2013 and has won 1st Place for PHM Society Data Challenges competition 5 times. He was on leave from UC to serve as Vice Chairman and Board Member for Foxconn Technology Group (ranked 26th in Global Fortune 500) during 2019-2021 to lead the development of Foxconn Wisconsin Science Park (~$1B investment) in Mt. Pleasant, WI. In addition, he advised Foxconn business units to successfully receive five WEF Lighthouse Factory Awards since 2019. He is a member of Global Future Council on Advanced Manufacturing and Production of the World Economics Council (WEF), a member of Board of Governors of the Manufacturing Executive Leadership Council of National Association of Manufacturers (NAM), Board of Trustees of MTConnect, as well as a senior advisor to McKinsey. Previously, he served as Director for Product Development and Manufacturing at United Technologies Research Center (now Raytheon Technologies Research Center) as well as Program Director for a number of programs at NSF. He was selected as 30 Visionaries in Smart Manufacturing in by SME in Jan. 2016 and 20 most influential professors in Smart Manufacturing in June 2020, SME Eli Whitney Productivity Award and SME/NAMRC S.M. Wu Research Implementation Award in 2022