IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 29 March 2023, accepted 7 April 2023, date of publication 14 April 2023, date of current version 26 April 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3267089

RESEARCH ARTICLE

A Parallel Ensemble Learning Model for Fault Detection and Diagnosis of Industrial Machinery

MOHAMMED NASSER AL-ANDOLI[1], SHING CHIANG TAN[2], KOK SWEE SIM[1], (Senior Member, IEEE), MANJEEVAN SEERA[3], AND CHEE PENG LIM[4]

[1]Faculty of Engineering and Technology, Multimedia University, Melaka 75450, Malaysia
[2]Faculty of Information Science and Technology, Multimedia University, Melaka 75450, Malaysia
[3]Department of Econometrics and Business Statistics, School of Business, Monash University Malaysia, Subang Jaya 47500, Malaysia
[4]Institute for Intelligent Systems Research and Innovation, Deakin University, Waurn Ponds, VIC 3216, Australia

Corresponding author: Shing Chiang Tan (sctan@mmu.edu.my)

This work was supported in part by the Fundamental Research Grant Scheme (FRGS) from the Malaysia Ministry of Higher Education under Project FRGS/1/2019/ICT02/MMU/02/2, and in part by the Multimedia University (MMU) Grant under Project MMUI/220154.

ABSTRACT Accurate fault detection and diagnosis (FDD) is critical to ensure the safe and reliable operation of industrial machines. Deep learning has recently emerged as effective methods for machine FDD applications. However, the gradient descent optimization method that is commonly used in deep learning suffers from several limitations, such as high computational cost and local sub-optimal solutions. Accordingly, this paper proposes a new parallel ensemble model comprising hybrid machine and deep learning for undertaking FDD tasks. Composed of three levels of learning, the proposed ensemble model employs two base learners and a meta-learner, and is executed in parallel processing platform to achieve efficient computation. The base learners adopt a hybrid Back-Propagation (BP) and Particle Swarm Optimization (PSO) algorithms to exploit the corresponding local and global optimization capabilities for identifying optimal features and improving FDD performance. The proposed model is validated through a series of experiments using two benchmark data sets, i.e., CWRU and MAFaulD. The results demonstrate a high performance with accuracy rates of 98.45% and 99.79% for CWRU and MAFaulD, respectively. Its parallel implementation is able to reduce the computation time, resulting in a speed-up of 5.9 time and 7.17 time, respectively. These findings indicate that the proposed model is effective and efficient for FDD of industrial machinery, making it a promising solution for implementation in real-world environments.

INDEX TERMS Fault detection and diagnosis, deep learning, machine learning, ensemble learning, parallel computing, hybrid optimization.

I. INTRODUCTION

Machinery Fault detection and diagnosis (FDD) is a critical area that involves identifying the relationships between the health of a machine and the sensory data samples collected from the machine. Historically, experts have relied on their engineering experience and knowledge to identify these relationships [1]. However, this traditional approach is time-consuming, as experts need to spend a significant amount of time to gather data, manually analyze them, and make decisions based on their expertise. In addition, the traditional approach relies on human judgement, which is subject to human errors. As a result, there is a growing interest to develop efficient and accurate methods for FDD of industrial machinery. The advent of Artificial Intelligence (AI) methodologies, which include machine learning (ML) and deep learning (DL), has opened up the way to design intelligent systems for various data analytics tasks in different domains, e.g. financial fraud detection [2], [3], medical data classification [4], [5], power systems monitoring [6], [7], and many others. In FDD, AI-based methods are useful for automating the underlying process, reducing the time and resources, and increasing the performance [1], [8].

The associate editor coordinating the review of this manuscript and approving it for publication was Mehul S. Raval.

This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 License.
For more information, see https://creativecommons.org/licenses/by-nc-nd/4.0/

Non-invasive methods such as vibration analysis of machines are useful for monitoring and predicting the maintenance needs of machines. Vibration monitoring is a commonly used method for analyzing a machine's state, which can be used to detect incipient failures, identify the failure location, and estimate the failure time. Monitoring the state of machines is critical to reduce or avoid shutdowns of industrial systems or processes [9]. By combining vibration signal detection with an intelligent system, it is possible to identify the machine's condition and predict the possibility of failures. This can help maintenance teams to predict and repair machines in a timely manner, avoiding unplanned maintenance, long downtimes, and costs [10], [11], [12]. Advances in technology, including the reduction in sensor and data storage costs and increased computer power, have made digitized and intelligent FDD systems possible. Indeed, many data-driven solutions utilizing big data and computational intelligence models such as support vector machine (SVM) [13] and artificial neural network (ANN) [14] have been applied to fault detection, classification and prediction in industrial environments.

The use of ML/DL for identifying and diagnosing failures in industrial motors has become more prevalent in recent years. In this regard, DL models have become widely used in FDD of machinery [15], [16], [17], as they can address the limitations of traditional ML methods in performing automated feature extraction. On the other hand, ensemble methods, which combine the strengths of multiple techniques to overcome their individual limitations, have been adopted for solving FDD problems [17], [18]. However, many existing DL-based solutions employ single models, which are often not efficient and effective in handling large-scale data and complex systems [19]. Furthermore, the learning procedure of many DL models hinges on the backpropagation (BP) algorithm with gradient descent optimization. BP is known to suffer from several limitations, such as local suboptimal solutions, difficulty in deciding the initial parameter settings, and absence of parallelization implementation that can the expedite computation process [20], [21], [22].

To cope with the above-mentioned shortcomings, a new parallel ensemble model comprising heterogeneous ML - DL models for machinery FDD is proposed in this study. The proposed model is a powerful and efficient approach for identifying and addressing FDD tasks. This approach utilizes multiple ML and DL models that operate in parallel to rapidly detect and diagnose faults in industrial machinery with high accuracy. Owing to its parallel implementation, this approach is highly scalable and can be applied to large-scale industrial systems in a cost-effective manner. In summary, the contributions of this paper are as follows:

- an ensemble-based method to leverage the strengths of multiple ML/DL models;
- a hybrid PSO-BP algorithm to optimize the DL parameters;
- a parallel processing platform to reduce computation time and make the developed method suitable for undertaking large-scale FDD tasks.

The rest of this paper is organized as follows: Section II presents a literature review on FDD and related areas. Section III explains the proposed method in detail, Section IV presents and discusses the experimental study. Concluding remarks are given in Section V.

The rest of this paper is structured as follows. Section II describes the related studies in the literature. Section III explains the ECDLP method in detail. Experimental results, analyses, and discussion are presented in Section IV, while concluding remarks are given in Section V.

II. LITERATURE REVIEW

FDD plays a crucial role in identifying the relationships between machine health states and measured sensor data. Traditionally, experts rely on their engineering experience and knowledge to detect these relationships. However, this approach is time-consuming, labor-intensive, and subject to human errors. To address these issues, ML and DL methods have been utilized to automate FDD tasks [1].

ML has been successfully used various FDD problems, e.g. in [13], an SVM combined with Continuous Wavelet Transform (CWT) was applied to bearing fault detection in an induction motor. The CWT extracted features associated with various types of faults, and the SVM classified them with good results. In [23], a hybrid approach using a Random Forests (RF) classifier was developed for FDD of rolling bearings, which leveraged wavelet packet decomposition to extract fault features. In [24], the K-Nearest Neighbour (KNN) and self-organizing map models were adopted for condition monitoring of cooling fan bearings, while in [25], the ability of a Bayesian Network Classifier (BNC) to infer probabilities for FDD of chiller faults in single step was demonstrated. In [26], a framework for predictive maintenance based on key performance indicators and a cluster-based hidden Markov model was developed for machinery deterioration estimation. These ML methods require significant effort in feature extraction from raw signals, which is costly in terms of labour and expertise.

Recently, DL methods have been shown useful for FDD of rotating machinery [15], as they can address the limitations of ML methods by automatically performing feature extraction. In [27], a parse autoencoder was applied to extract representative features from statistical values of bearing signals and a Deep Belief Network (DBN) was used to classify the health conditions. In [14] a normalize sparse autoencoder (NSAE) was developed for intelligent fault diagnosis, and a local connection network constructed by NSAE was devised to minimize misclassification of mechanical health conditions. Additionally, in [28], a self-adaptive optimized DBN was proposed for FDD of rolling bearings, which was pre-trained by a mini-batch stochastic gradient descent method. In [11], an adaptive residual Convolutional Neural Networks (CNN) approach was proposed and evaluated using fault data collected from a Chinese lead-based nuclear reactor and nuclear platform. In [12], Deep CNNs with variational mode decomposition (VMD) algorithms were combined for FDD of wind turbines, while in [10] deep adversarial networks were developed for machinery fault diagnosis and prediction. Bayesian DL approach was also proposed in [29], which utilized the prediction uncertainty to improve FDD by using a risk function reflecting the cost of misclassifications.

Ensemble methods, which combine the strengths of multiple techniques to overcome their individual limitations, have been used in the literature for FDD tasks. In [30], an ensemble multi-objective optimization framework comprising DBN, deep autoencoder, and CNN was employed for effective diagnosis of rotor and bearing faults in rotating machinery. In [31], an ensemble DL method composed of DBNs and stacked denoising autoencoders was presented for transformer FDD using an internet-of-thing (IoT)-based condition monitoring system. Furthermore, in [32], various ensemble learning and ML methods were evaluated for FDD of photovoltaic systems, with a focus on detecting complex faults. An ensemble DL-based data fusion approach was proposed in [33] for fault detection in industrial IoT settings. The DNN, CNN, and Long Short Term Memory Network (LSTM) models were utilized to produce good FDD results. Overall, based on the literature, ensemble-learning methods are effective in improving performance in undertaking FDD tasks.

This study aims to enhance ensemble learning approaches by addressing some of the limitations of current DL-based methods. Many current DL-based methods employ single classifiers. Besides, DL models are often not suitable for handling large-scale data due to their complex architectures. Additionally, DL methods typically rely on gradient optimization with BP, which can result in suboptimal generalization performance. To overcome these issues, this research proposes an ensemble of heterogeneous ML-DL models that are optimized using a hybrid PSO-BP algorithm. To improve processing efficiency, a parallel processing platform is designed to implement the ensemble model for FDD tasks.

III. METHODOLOGY

In this study, we introduce a new ensemble-based ML/DL method for FDD of machinery, and is denoted as EMDL-FDD, this proposed method covers preparing and cleaning data (Section III-A), ensemble learning and training of DNN models with a hybrid PSO-BP method (Section III-B) and parallel implementation of the resulting ensemble-based method (Section 2.3).

A. PRE-PROCESSING

The first step in the proposed method is pre-processing the data samples. The raw data samples are in the form of multivariate time-series signals. These signals are transformed into a set of features that are relevant for FDD. Specifically, statistical features are extracted from the time and spectral domains of each signal [34]. The extracted features form a new representation that reduces the dimensionality of the original features as well as reduces the computational costs. The feature data samples are then normalized using z-score normalization [35] (Equation 1) before being used in the ensemble of DNN and ML models.

$$X = \frac{(x-\mu)}{\sigma}, \tag{1}$$

where variables x, σ, μ are the original value, standard deviation, and mean value, respectively.

B. ENSEMBLE LEARNING OF EMDL-FDD

Fig. 1 illustrates the proposed EMDL-FDD framework, which adopts a three-tiered learning approach. The first level (*L0*) comprises a number of DNNs used as base learners for performing initial classification. The second level (*L1*) involves a combination of DNNs and various ML models, which include SVM, RF, KNN, Decision Tree (DT), and XGBoost (XGR), for further classification. The third level (the meta-learner) utilizes single ANN model to make the final classification decision.

In *L0*, the base learners (i.e., DNNs models) operate independently on a parallel processing platform. The input data are processed by each DNN model, resulting in a prediction at the output layer ($n * d$), where d and n indicate the target class and the number of samples, respectively. The hidden layers between the input layer and the ouput layer use functions $h = f\,(W_1 x + b_1)$, $h_l = f1\,(W_2 h_l + b_2)\,, \ldots o = f_l\,(W_{l+1} h_l + b_{l+1})$, where x, h, o, l denote the data sample, the hidden layer output, the final layer output, and the number of layers, respectively. The DNN layers are dense, and a Rectified Linear Unit (*ReLU*) activation function is used in the hidden layers. The ensemble DNN model employs two different activation functions in the last hidden layer, either sigmoid or Softmax.

The DNN models in *L0* are trained and optimized using a combination of metaheuristic optimization (i.e., PSO) and gradient optimization (i.e., BP) techniques. PSO is simple and has demonstrated its capability in solving complex optimization problems in various domains. It supports constructive collaboration among all particles to search for a global optimal solution, while BP performs non-linear mapping through local search. The combination of both techniques yields better results than using them individually.

BP employs the Weighted Cross Entropy (WCE) as a loss function to adjust the DNN trainable parameters, θ, in a minimization mode. The loss function (J_θ) is computed using Equation 2:

$$J_\theta = \frac{1}{n}\sum\nolimits_{i=1}^{n}\left[X_i \log(Y_i) + (1-X_i)\log(1-X_i)\right], \tag{2}$$

where n is the number of samples, X_i and Y_i are the original and predicted class labels respectively. Parameters θ are updated during local search with BP using

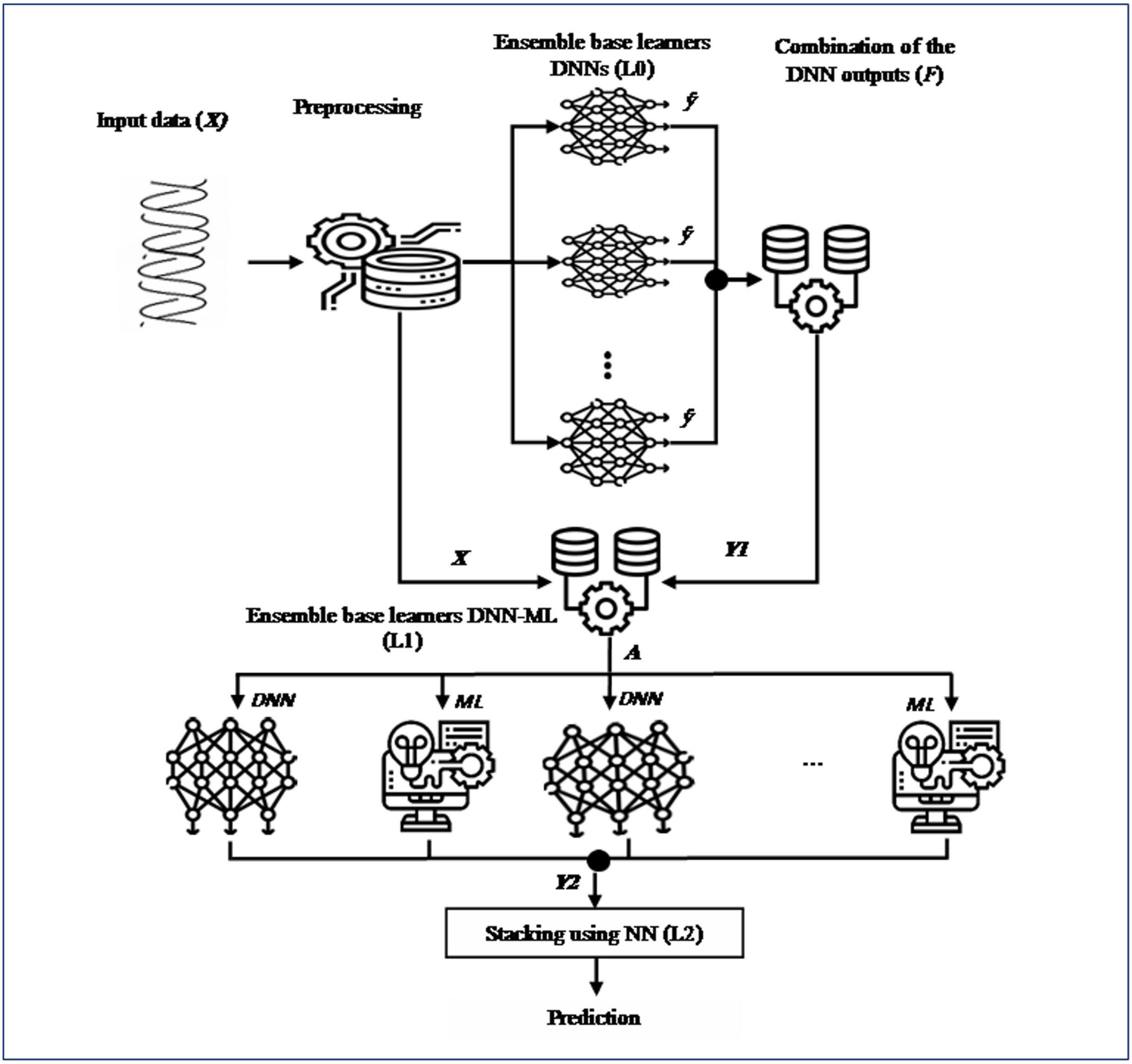

FIGURE 1. A schematic diagram of the proposed ensemble-based ML-DL method for fault detection (EMDL-FDD).

Equations 3 and 4:

$$\theta_\alpha^{ij} = \theta_\alpha^{ij} - \gamma \frac{\partial}{\partial \theta_\alpha^{ij}} J_\theta(X, Y), \tag{3}$$

$$\begin{aligned}\frac{\partial}{\partial \theta_\alpha^{ij}} J_\theta\left(X, g(f(X))\right) &= \sum_{i=1}^{N} \frac{\partial}{\partial \theta_\alpha^{ij}} J_\theta\left(X_i, g(f(X_i))\right) \\ &= \sum_{i=1}^{N} \frac{\partial}{\partial z_\alpha^{j}} J_\theta\left(X_i, g(f(X_i))\right) \frac{\partial}{\partial \theta_\alpha^{ij}} z_\alpha^{j} = \sum_{i=1}^{N} \delta_\alpha^{j} X_i^T,\end{aligned} \tag{4}$$

where γ is the learning rate, N is the number of neurons, and α refers to the activation value of the hidden layer, H, and the output layer, Y. The contribution of each data sample in the optimization process leads to the total error: $\delta_\alpha^j = \frac{\partial}{\partial z_\alpha^j} J_\theta\left(X_i, g(f(X_i))\right)$.

The output of local optimization with BP, computed using Equations 3 to 5, is denoted as M_θ. This output is then utilized in the PSO algorithm. Specifically, M_θ is used to generate a set of particles P_s, which is distributed throughout different regions of the search space. Each particle P_i then moves with velocity $V_{i\theta}$ over a number of iterations (t) according to Equations 5 and 6 to update its position.

$$V_{\theta i}^{(t+1)} = \lambda v_{\theta i}^{(t)} + c_1 r_1 \left[p_{\theta i}^{best(t)} - M_{\theta i}^{(t)}\right] + c_2 r_2 \left[g_{\theta}^{best(t)} - M_{\theta i}^{(t)}\right], \tag{5}$$

$$i \in [1, P_s],$$

$$M_{\theta i}^{(t+1)} = M_{\theta i}^{(t)} + V_{\theta i}^{(t+1)}, i \in [1, P_s], \tag{6}$$

For each particle, P_i, the best local solution is updated with Equation 7.

$$P_{\theta i}^{best} = M_{\theta i} | f\left(M_{\theta i}\right) = \min_{c=1,2,3,\ldots,t+1} \left\{f\left(M_{\theta i,c}\right)\right\}, \tag{7}$$

In Equations 5 to 7, r_1 and r_2 are two randomly chosen numbers in the interval [0, 1]; $M_{\theta i}$ is the output of a local replica;λ determines the P_s movement inertia; c_1 is the cognitive parameter and c_2 is the social parameter, while t, v, g^{best}, *and* p^{best} are the number of iterations, velocity, global best particle and local best particle, respectively. The global best particle is computed by averaging p^{best} or the best local particle with the highest rank. The global best particle is computed using Equation 8.

$$g_{\theta}^{best} = \min \left(\begin{bmatrix} P_{\theta i}^{best} | f(P_{\theta i}^{best}) = \min_{i=1,2,3,\ldots,P_s} \left\{ f(P_{\theta i}^{best}) \right\} \end{bmatrix}, \atop \left[\frac{1}{P_s} \sum_{i=1}^{P_s} P_{\theta i}^{best} \right] \right). \tag{8}$$

The fitness function is calculated by the average values of the loss function (i.e., WCE) from local BP optimization $J_{\theta M} = (J_{\theta 1}, J_{\theta 2}, J_{\theta 3}, \ldots, J_{\theta m})$ in Equation 3, as indicated in Equation 9.

$$f(J_{\theta M}) = \frac{1}{m} \sum_{i=1}^{m} J_{\theta i}, \tag{9}$$

where P_s refers to the number of particles. Once the fitness function is improved in a minimization mode, $P_{i\theta}^{best}$ and g_{θ}^{best} are updated using Equations (6) to (9).

The optimization of DNN models is repeated to achieve a high performance in the first-level classification. The output of each DNN base learner ($\hat{y}$) is then combined into one vector (*Y1*), where $Y1 = (\hat{y}_1, \hat{y}_2, \ldots, \hat{y}_{b1})$, *b1* indicates the number of base learners in *L0*. New feature representations (A) are created by combining the original features (X) and the predictions of the first base learners *(Y1)*, where $A = X \cup Y1$.

In the second level of learning (i.e. L1), the new features are represented as A. A set of heterogeneous ML and DL models that include DT, RF, KNN, XGB, and SVM as well as DNN models with either softmax or sigmoid activation functions is formed. Fig.2 illustrates the learning process of *L1*. The DNN training process in *L1* uses the same PSO-BP optimization method as that in *L0*. Other ML models, i.e., DT, RF, KNN, XGB, and SVM, are trained using their respective default methods provided in the scikit-learn library.

Once a high performance has been achieved in *L1*, the output ($\hat{y}$) from each base learner (DNN or ML) are combined into one vector *Y2*, where $Y2 = (\hat{\hat{y}}_1, \hat{\hat{y}}_2, \ldots, \hat{\hat{y}}_{b2})$, and *b2* represents the number of base learners in *L1*. This merged output is passed on to single ANN, known as the meta-learner, to perform the final classification. In this study, the meta learner is a feedforward ANN with a hidden layer and an output layer activated using a softmax function and trained using BP.

C. PARALLEL PROCESSING IN EMDL-FDD

A parallel processing platform is utilized to enhance the efficiency of EMDL-FDD. This platform operates on two levels: high-level parallelism (e.g. multi-processors and machines) and low-level parallelism (e.g. multi-threads). Data parallelism is used to construct the platform. Parallel processing is implemented for the base learners and the tasks of each DNN base learner. In other words, parallelization is performed among individual base learners and among the tasks of single DNN base learner. This parallelization includes dividing the dataset into partitions and training the model. Moreover, parallel processing is used to update the local solution for a pool of particles simultaneously. The Python's Ray library is employed, which is one of the most up-to-date libraries in the field of parallel processing to train and optimize ML and DL models [36]. It is capable of processing large amount of data and of supporting data sharing concepts.

In *L0*, each DNN model and its training operations are executed in a high-level parallelism mode, i.e., multi machines or processors or cores. In addition, the tasks of the DNN base learners are carried out in a low-level parallelism mode, i.e. multi-threads, with CPU cores. This parallelization is conducted for two reasons. Firstly, parallel tasks consume most of the execution time of EMDL-FDD. Secondly, independent processing occurs among individual base learners, as well as among DNN tasks, such as the operations of PSO particles. Therefore, the synchronization and communication operations are limited. The same parallelization is also carried out in *L1*, i.e., among the base learners, as well as among the tasks of each DNN base learner.

The parallelism operation of EMDL-FDD starts by establishing several Ray workers according to the maximum number of base learners. Each worker receives its allocated resources, i.e., machines and processors, in a ratio of 1: m, where m is the number of base learners. In each DNN, multi-thread Ray workers are established according to the number of particles P_s, where each individual particle, P_i, carries out a search process and updates its local solution p^{best} in parallel processing. All individual particles are synchronized to collect the results of local replicas M_θ. Next, the fitness function of PSO is computed, and the global best solution g_θ^{best} is updated. Once the learning process in *L0* has been performed, a synchronized step is carried out to collect the output of the respective base learners and send to the base learners in *L1*. In *L1*, the parallelization procedure is carried out in the same way as that in *L0*. Finally, the output of *L1* is sent to the meta-learner to perform the final classification. Fig.2 illustrates the parallel processing platform of EMDL-FDD, while Algorithm 1 shows the steps of EMDL-FDD.

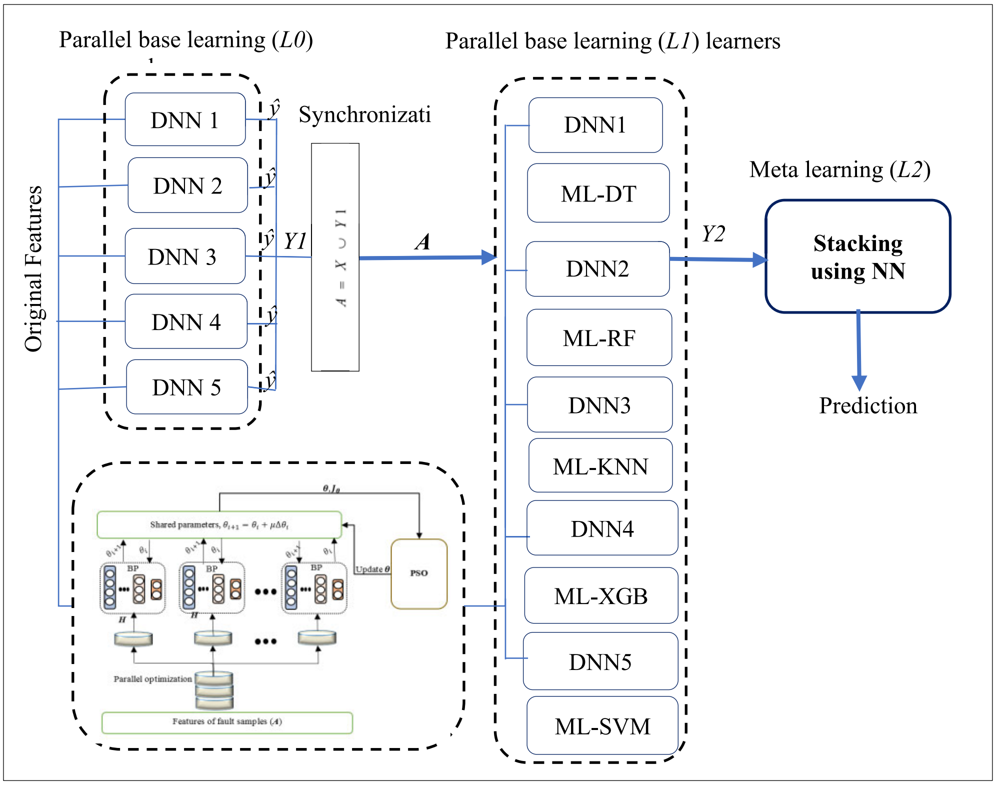

FIGURE 2. Parallel Processing Platform of EMDL-FDD.

IV. EXPERIMENTAL STUDIES AND RESULTS

A series of experiments was conducted to evaluate the performance of EMDL-FDD. The results are analyzed and discussed in detail.

A. DATA SETS

In this study, we use five malware datasets for performance evaluation, as follows:

1) Case Western Reserve University (CWRU) [37], [38]: This is a popular and easily accessible data set. It was collected from the driving system of a test motor. The motor has 2 HP power, a torque transducer, a dynamometer, and control electronics. The data set contains 2300 samples, which cover three conditions: (1) 1 HP load applied to the motor, (2) 48 kHz sampling frequency of the accelerometers, and (3) shaft rotating speed of 1772 rpm. The data set contains three defects at different diameters (0.007, 0.014, and 0.021 inches) in one of three parts of the bearing: inner race, outer race, and ball.
2) Machinery Fault Database (MAFaulD) [39]. This data set contains 1951 multivariate time-series sensory data acquired from a SpectraQuest's Machinery Fault Simulator (MFS) Alignment-Balance-Vibration (ABVT) system. The data samples cover six simulated states: normal function, imbalance fault, horizontal and vertical misalignment faults, and inner and outer bearing faults. The data set contains 561655 samples from 8 sources (6 accelerometers, a microphone, and a tachometer).

B. EVALUATION METRICS

Accuracy, precision, recall, and F1-score metrics were adopted for performance evaluation and comparison. These metrics are computed based on the True Positive (*TP*), False Positive (*FP*), True Negative (*TN*), and False Negative (*FN*) rates, as in Equations (10) to (13):

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}. \tag{10}$$

$$Precision = \frac{TP}{TP + FP}. \tag{11}$$

$$Recall = \frac{TP}{TP + FN}. \tag{12}$$

$$F1 - score = \frac{\mathrm{Re}\,call \times \mathrm{Pr}\,ecision}{\mathrm{Re}\,call + \mathrm{Pr}\,ecision}. \tag{13}$$

To evaluate efficiency and scalability of EMDL-FDD, runtime speedup and Percentage Improvement (PI) indicators were used, as defined in equations 14 and 15.

$$Speed - up(A, B) = \frac{Method(A)}{Method(B)}(time). \tag{14}$$

$$PI(A, B) = \frac{Method(A) - Method(B)}{Method(A)} \times 100. \tag{15}$$

Algorithm 1 EMDL-FDD

Input: Data fault samples X
Output: Fault detection and classification in data samples
Processing data:
  Pre-processing with equation 1.
Ensemble learning and classification:
  a) First base learners (L0)
    For all (base learners $b1$) ***do in parallel***
      Generate DNN models, which represent base classifiers (See Section III-B)
      Train DNN models:
        for j=1 to t
          For all ($P_s$inP_m): ***doinparallel***
            Optimize DNN models with PSO-BP (Refer to Section III-B)
          end parallel
        end for
      Obtain fault data detection and classification $\hat{y}$
    end parallel
    Send the output to the fusion step
  b) Fusion step:
    Combine the outputs of the first base learners
    Combine *Y1* to the original features X *to* produce A, $A = X \cup Y$
  c) Second base learners (L1)
    For all (base learners $b1$) ***do in parallel***
      Generate DNN and ML models, which represent base classifiers (See Section III-A)
      Train DNN models in parallel with PSO-BP likewise (a) step:
      Obtain fault data detection and classification $\hat{\hat{y}}$
    end parallel
    Send the output to the fusion step
  d) Fusion step:
    Combine the outputs of the second base learners $Y2 = (\hat{\hat{y}}_1, \hat{\hat{y}}_2, \ldots, \hat{\hat{y}}_{b2})$
  e) Meta classifier:
    Receive the prediction values *Y2* from the second base learners
    Generate meta learner (NN)
    Train the meta learner and make the final prediction
Return fault data detection samples.

C. EXPERIMENTAL SETTINGS

Two machines were employed to analyze and verify the parallel implementation of EMDL-FDD. The processors used were Intel Core-i7, and the operating system was Ubuntu 22.04. One machine had 8 GB Random Access Memory (RAM), and the other had 4 GB. The Python Ray Library [36] was used to implement parallelization. For the base learners (i.e., DNNs), the number of iterations, batch size, and learning rate were set to 100, 256, and 0.001, respectively. Five DNN models were used in the *L0*, and eight heterogeneous models (i.e., four DNNs, RF, DT, KNN, and XGB) were used in *L1* to form an ensemble method. The PSO parameters for DNN optimization were set as follows: *c1*, *c2*, the inertia weight, the number of particles were 0.5, 0.3, 20, 0.9, respectively. Each particle's position was initialized randomly between 0 and 1. Two ensemble learning models were developed for performance comparison: an ensemble of heterogeneous models (denoted as EML) consisting of ML-RF, ML-DT, ML-KNN, ML-XGB, and ML-SVM, and an ensemble of several DNNs (denoted as EDNN). EDNN served as a variation of EMDL-FDD, which excluded the use of ML models. The data set was divided into a training set and a test set with a ratio of 4 to 1. The experiment was repeated ten times, and the average results were computed.

D. RESULTS AND DISCUSSION

1) PERFORMANCE ANALYSIS OF EMDL-FDD METHOD ON CWRU

The CWRU data set contains 10 faults classes (C0, C1, C2, C3, C4, C5, C6, C7, C8, C9): three for ball defects with 0.007, 0.014, 0.021 inches, three for Inner race faults with 0.007, 0.014, 0.021 inches, three for Outer race faults with 0.007, 0.014, 0.021 inches, and one for normal state. The results were obtained from three levels of fault classification: the first layer of base learners (*L0*), the second layer of base learners (*L1*), and the meta-learner (*L2*).

The performance of individual DNNs in *L0* is shown in Table 1. The results indicate that individual DNNs can produce high accuracy, precision, recall, and F1-score rates of 95.78%, 95.79%, 95.78%, and 95.77%, respectively.

Table 2 displays the performance of individual DNNs and MLs models in *L1* using the original features and outputs from *L0*. Table 2 (a) shows that DNNs demonstrate high performance with the four metrics (accuracy, precision, recall, and F1-score rates) of 97.04%, 97.05%, 97.04%, and 97.03%, respectively. Table 2 (b) shows that the best ML model, i.e., ML-XGB, achieves the accuracy, precision, recall, and F1-score rates of 95.78%, 96.1%, 95.0%, and 96.09%, respectively. The results in Tables 1 and 2 indicate that DNNs yield better performance in *L1* as compared with those in *L0*.

TABLE 1. Performance of individual DNN models in L0 on the CWRU data set with original features.

Method	Accuracy	Precision	Recall	F1-score
DNN1	95.78	95.77	95.78	95.77
DNN2	95.22	95.4	95.22	95.24
DNN3	95.65	96.06	95.65	95.77
DNN4	95.78	95.79	95.78	95.77
DNN5	94.74	94.73	94.74	94.72

TABLE 2. Performance of individual DNN models and ML models in L1 on the CWRU data set with original features and outputs from L0.

(a) DNNs				
Method	Accuracy	Precision	Recall	F1-score
DNN1	97.04	97.05	97.04	97.03
DNN2	96.43	96.45	96.43	96.43
DNN3	96.65	96.66	96.65	96.65
DNN4	96.74	96.77	96.74	96.73
DNN5	96.57	96.58	96.57	96.56
(b) MLs				
Method	Accuracy	Precision	Recall	F1-score
ML-RF	93.48	94.12	93.48	94.0
ML-DT	92.0	94.0	92.17	92.17
ML-KNN	93.04	93.04	93.04	93.04
ML-XGB	95.78	96.1	95.0	96.09
ML-SVM	93.04	94.0	93.04	93.04

Table 3 shows the experimental results of EMDL-FDD for each fault type, which lead to the final classification and fault detection in the meta-learner level (*L2*). The average accuracy, precision, recall, and F1-score rates of EMDL-FDD are 98.45%, 98.42%, 98.59%, 98.27%, respectively.

A performance comparison among EMDL-FDD, EML (comprising ML-RF, ML-DT, ML-KNN, ML-XGB, and ML-SVM) and EDNN (five DNNs with ML models excluded) was conducted. The results of EML, EDNN, and EMDL-FDD are listed in Table 4. It is clear that the proposed EMDL-FDD method outperforms EML and EDNN in all performance indicators. EDNN ranks the second, while EML has the worst performance.

TABLE 3. Performance of EMDL-FDD for each fault in the CWRU data set.

Class	Accuracy	Precision	Recall	F1-score
C0	96.12	96.97	95.43	94.72
C1	97.98	96.96	99.76	96.68
C2	95.67	95.03	96.56	97.8
C3	99.98	99.91	100	100
C4	99.56	99.9	100	100
C5	98.43	98.76	99.93	97.88
C6	99.89	100	100	100
C7	96.88	96.7	94.23	95.66
C8	100	100	100	100
C9	100	100	100	100
Average	98.45	98.42	98.59	98.27

TABLE 4. Performance comparison of three ensemble methods on the CWRU data set.

Method	Accuracy	Precision	Recall	F1-score
EML	96.52	96.63	96.52	96.48
EDNN	97.39	97.51	97.39	97.42
EMDL-FDD	**98.45**	**98.42**	**98.59**	**98.27**

2) PERFORMANCE ANALYSIS OF EMDL-FDD METHOD ON MAFAULD

The performance of EMDL-FDD was also evaluated using the MAFaulD data set. There are six fault classes (C0, C1, C2, C3, C4, C5): normal, imbalance, horizontal misalignment, vertical misalignment, overhang bearing, and underhang bearing.

Tables 5 to 7 show the results from three levels of fault classification: *L0*, *L1*, and *L2*. Based on Table 5, the results of individual DNNs in *L0* using the original features are high, with the best accuracy, precision, recall, and F1-score rates of 95.58%, 95.61%, 95.58%, and 95.6%, respectively. Table 6 shows the performance of individual DNNs and MLs models in *L1* using the original features and outputs from *L0*. DNNs produced high accuracy, precision, recall, and f1-score rates of 97.68%, 97.88%, 97.32%, and 97.9%, respectively. Again, ML-XGB yielded the best ML results with accuracy, precision, recall, and f1-score rates of 95.49%, 95.64%, 95.49%, and 95.48%, respectively. The results in Tables 5 and 6 indicate that DNNs yield better performance in *L1* as compared with those in *L0*.

TABLE 5. Performance of individual DNN models in L0 on the MAFaulD data set using original features.

Method	Accuracy	Precision	Recall	F1-score
DNN1	94.37	94.38	94.37	94.4
DNN2	94.01	94.12	94.01	94.32
DNN3	93.98	94.03	93.98	94.00
DNN4	94.46	94.46	94.46	94.46
DNN5	95.58	95.61	95.58	95.6

TABLE 6. Results of individual DNN models and ML models in L1 on the MAFauID data set with original features and outputs from L0.

(a) DNNs				
Method	Accuracy	Precision	Recall	F1-score
DNN1	97.68	97.88	97.32	97.9
DNN2	98.12	98.45	98.35	98.45
DNN3	97.87	97.88	97.87	97.87
DNN4	97.92	98.01	97.97	98.0
DNN5	96.6	96.96	96.81	96.96
(b) MLs				
Method	Accuracy	Precision	Recall	F1-score
ML-RF	94.46	94.46	94.46	94.46
ML-DT	94.21	94.3	94.01	94.01
ML-KNN	93.76	93.98	93.84	94.01
ML-XGB	95.49	95.64	95.49	95.48
ML-SVM	92.25	92.78	92.54	92.64

Table 7 shows the performance of EMDL-FDD for each fault class using the MAFaulD data set, depicting an average accuracy of 99.79%, precision of 99.745%, recall of 99.75%, and F1-score of 99.63%. Notice that classes C0 and C1 produce 100% in all metrics. Table 8 shows a comparison among EML, EDNN, and EMDL-FDD using the MAFaulD data set. Again, EMDL-FDD yields the best overall performance.

TABLE 7. Performance of EMDL-FDD for each fault in the MAFauID data set.

Class	Accuracy	Precision	Recall	F1-score
C0	100	100	100	100
C1	100	100	100	100
C2	99.8	100	99.72	100
C3	100	99.35	100	99.41
C4	99.3	99.12	99.04	99.25
C5	99.64	100	99.2	99.12
Average	99.79	99.745	99.75	99.63

TABLE 8. Performance comparison of three ensemble methods on the MAFauID data set.

Method	Accuracy	Precision	Recall	F1-score
EML	96.28	96.30	96.00	96.12
EDNN	98.6	98.96	98.81	98.96
EMDL-FDD	**99.79**	**99.745**	**99.75**	**99.63**

3) REMARKS

The proposed EMDL-FDD method has demonstrated exceptional performance in identifying various fault classes in both CWRU and MAFaulD data sets, as shown in Tables 1 to 8. The ensemble approaches, namely EML and EDNN, are able to improve the FDD performance of individual ML and DNN models. The effectiveness of EMDL-FDD is attributed to the combination of heterogeneous ML and DNN models for classification. Clearly, the strength of using two levels of base learners and a meta-learner leads to improved classification performance by utilizing the output of the first level and the original features. Additionally, the use of hybrid optimization for devising DNN models, which combines metaheuristic optimization with PSO and gradient optimization with BP, has allowed EMDL-FDD to find the best solution and achieve the best performance, as compared with other methods. Overall, the results positively demonstrate that EMDL-FDD is a promising approach to FDD of industrial machinery.

4) COMPARISON WITH RELATED STUDIES

We conducted two additional empirical studies to further evaluate the effectiveness of EMDL-FDD as compared with other state-of-the-art methods for FDD.

Table 9(a) presents a comprehensive comparison of EMDL-FDD with six state-of-the-art techniques on the CWRU data set, namely a stacked denoising autoencoder (SDAE) [40], hierarchical CNN (HCNN) [41], multi-scale recursive semi-supervised DL method (MRAE-AG) [16], hierarchical adaptive CNN (HACNN) [42], ensemble DBN (EDBN) [43], and predictive maintenance with CNN (PdM-CNN) [9]. Among these six methods, HACNN produces the highest accuracy rate of 97.9%, as shown in Table 9(a). Comparatively, EMDL-FDD yields an accuracy of 98.45%, outperforming all other techniques.

Table 9(b) presents a comparison of EMDL-FDD with six state-of-the-art methods on the MAFaulD data set, namely Fourier domain features and ANN (FANN) [44], feature vector (kurtosis and entropy) with ANN (KE-ANN) [45], similarity-based models (SBM) [34], synthetic minority oversampling and DNN (SMOTE-DNN) [46], improved SBM (ISBM) [47], and PdM-CNN [9]. Again, EMDL-FDD yields the highest accuracy rate of 99.79%, outperforming the best compared result of 99.58% from PdM-CNN. These results further indicate the effectiveness of the proposed EMDL-FDD method for FDD of machineary.

TABLE 9. Comparison of EMDL-FDD with existing studies for FDD of machinery.

CWRU data set		**MAFaulD data set**	
Method	Accuracy	Method	Accuracy
SDAE [40]	91.79%	FANN [44]	93.5%
HCNN [41]	92.60%	KE-ANN [45]	95.8%
MRAE-AG [16]	94.38%	SBM [34]	96.4%
HACNN [42]	97.90%	SMOTE-DNN [46]	97.0%
EDBN [43]	96.95%	ISBM [47]	98.5%
PdM-CNN [9]	97.3%	PdM-CNN [9]	99.58%
EMDL-FDD	**98.45%**	**EMDL-FDD**	**99.79%**

5) PARALLEL IMPLEMENTATION AND COMPUTATIONAL TIME ANALYSIS OF EMDL-FDD

The time analysis of various methods on the CWRU and MALFauID data sets was conducted, as illustrated in Fig. 3 and Fig. 4. The results indicate that the DNNs at *L1* have

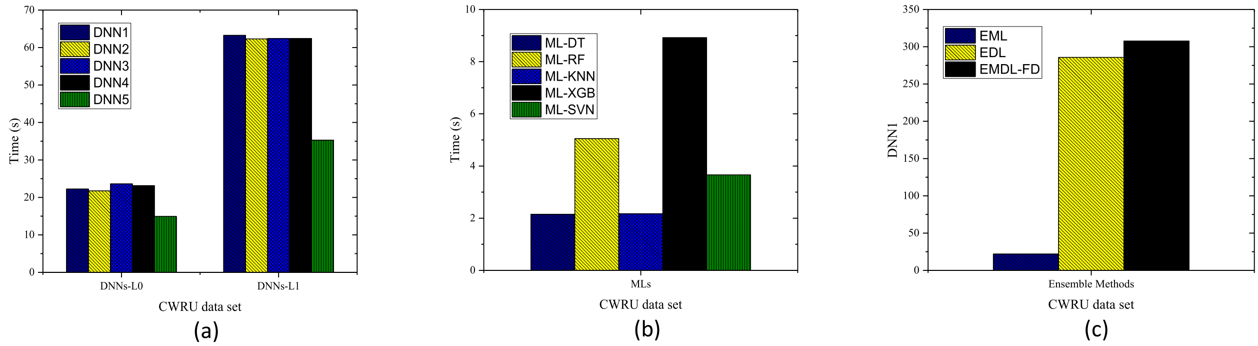

FIGURE 3. Computation time comparison of different methods on the CWRU data set.

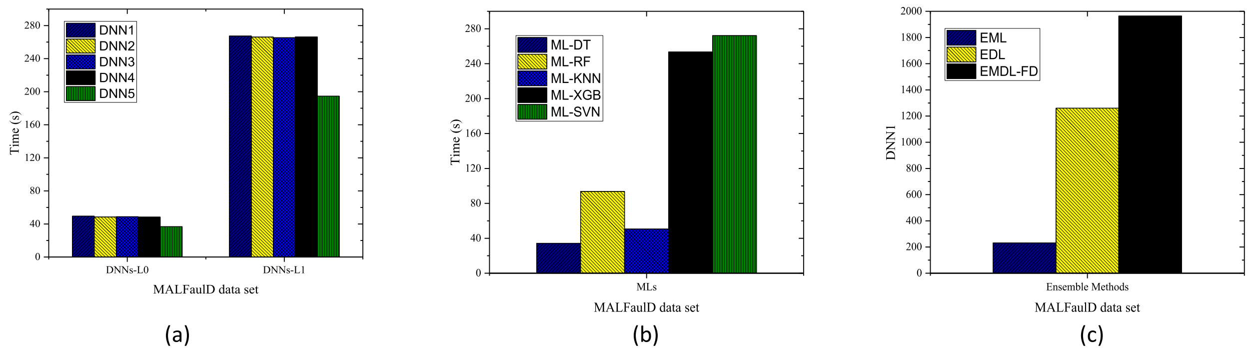

FIGURE 4. Computation time comparison of different methods on the MALFaulD data set.

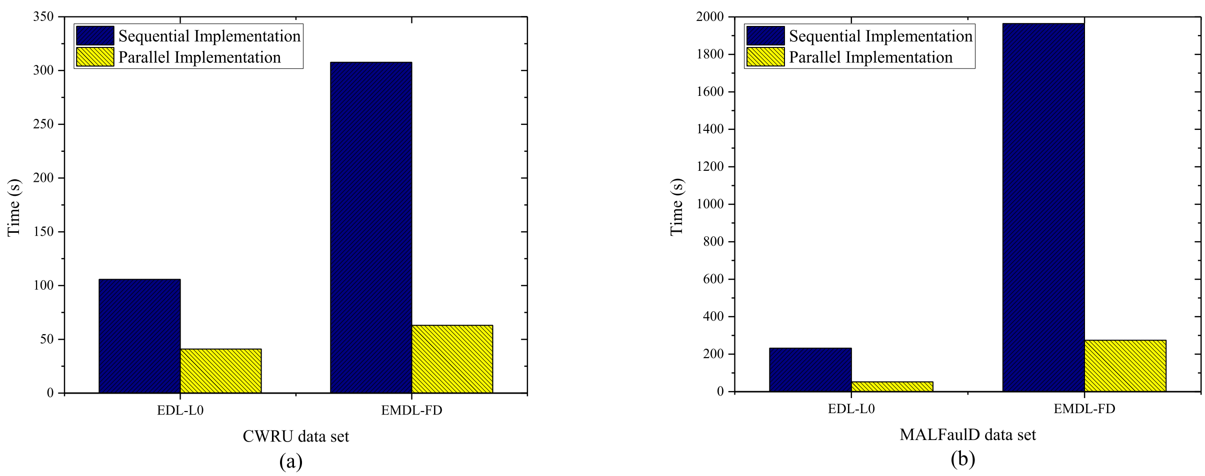

FIGURE 5. Computation time of EMDL-FDD with sequential and parallel implementations on the CWRU and MALFaulD data sets.

longer computation times than those at *L0*, due to the higher number of input features. Among the ML methods, ML-DT and ML-KNN have the shortest computation times, while XGBoost and SVM have the longest. In terms of ensemble methods, EML has the fastest computation time, followed by EDNN, and EMDL-FDD has the slowest computation time. While EMDL-FDD requires a longer computation time, the use of parallel implementation can significantly reduce the computation time by 80% to 63 seconds on CWRU (Fig. 5(a)) while preserving its high accuracy, precision, recall, and F1-score results. Similarly, on the MALFaulD data set (Fig. 5(b)), EMDL-FDD in parallel implementation can reduce the computation time by 86% to 274 seconds, while maintaining its high level of performance. Additionally, parallel implementation also improves the computation time of EDNN by 61% from 105 to 41 seconds on CWRU and 77.5% from 231 to 53 seconds on MALFaulD. The results show that parallel implementation of EMDL-FDD is more

Nodes List

	Host/Cmd Line	State	ID	IP/PID	Actions	CPU Usage	Memory	Object Store Memory
1	Alandoli-Lenovo-Z50-70	ALIVE	7f6397..	192.168.×.× (Head)	Log	100%	3.17GB /7.66GB(41.4%)	89.47MB
2	Alandoli-X555LJ	ALIVE	70463e..	192.168.×.×	Log	99.9%	1.4GB /3.74GB(37.6%)	35.17MB

FIGURE 6. Machine view of the distributed and parallel system of EMDL-FDD, indicating the resource utilization information.

effective when dealing with large data sets, as it can improve the computation time by up to 81% on CWRU and 86% on MALFauID.

Fig. 5 shows a comparison of the computation time of EMDL-FDD with sequential and parallel implementations on the CWRU and MALFauID data sets. Parallel implementation has significantly reduced the computation time on both data sets. On the CWRU data set, the computation time has been reduced by 80% from 315 seconds to 63 seconds (Fig. 5 (a)), achieving a speedup up to 5.9X. On the MALFauID data set, the computational time has been reduced by 86% (Fig. 5(b)), yielding a speedup up to 7.17X. These results demonstrate the effectiveness of parallel implementation in reducing the computation time for EMDL-FDD, especially for large data sets. It is also worth noting that the parallel implementation does not compromise the performance of EMDL-FDD.

Although EMDL-FDD has longer computation times as compared with those from some MLs methods, such as ML-DT and ML-KNN, it achieves higher levels of accuracy, precision, recall, and F1-score. Therefore, there is a trade-off between computation time and performance. To further improve the computation time, more powerful hardware or increasing the number of cores used in parallel implementation can be considered.

Fig. 6 presents the machine view of EMDL-FDD as a distributed and parallel system, which displays resource utilization information per worker. It shows information on how resources, specifically CPUs, have been utilized by each worker in the system. From Fig. 6, it can be observed that the developed distributed parallel framework for EMDL-FDD is efficient in utilizing the available resources. Overall, the use of parallel implementation and the high level of performance achieved make EMDL-FDD a suitable choice for FDD tasks.

V. CONCLUSION

In this paper, we have designed and developed a new ensemble method consisting of heterogeneous ML and DL models. Denoted as EMDL-FDD, this proposed ensemble method is cable to effectively and efficiently perform FDD of industrial machinery, as demonstrated through empirical evaluations. Specifically, EMDL-FDD covers data preparation and cleaning, and it has three levels of learning, two base learners and one meta-learner, and is executed in a parallel processing platform for efficient computation. The first level adopts several DNNs as the base learners for initial classification. The second level forms a combination of DNNs and various ML models for further classification. The final level is the meta-learner that utilizes single NN model to make the final classification decision. During the training process, EMDL-FDD uses a hybrid BP and PSO algorithm that combines both local and global optimization capabilities to identify optimal features for improving the classification performance. The performance evaluation using two benchmark machinery FDD data sets, i.e., CWRU and MAFauID, has indicated that EMDL-FDD outperforms state-of-the-art methods, yielding the highest accuracy rates of 98.45% and 99.79% for both data sets, respectively. The results also demonstrate the efficiency of parallel implementation in reducing the computation time of EMDL-FDD, achieving a high speedup up of 5.9X and 7.17X on both data sets.

In summary, the proposed EMDL-FDD method has been demonstrated to be robust, efficient, and highly accurate for FDD of industrial machinery. Further research will focus on real-world implementation of EMDL-FDD in actual industrial environments. In addition, other AI and related techniques such as deep reinforcement learning or generative models, will be investigated for integration into EMDL-FDD, to further improve its capabilities in FDD of industrial machinery.

REFERENCES

[1] Z. Zhao, J. Wu, T. Li, C. Sun, R. Yan, and X. Chen, "Challenges and opportunities of AI-enabled monitoring, diagnosis & prognosis: A review," *Chin. J. Mech. Eng.*, vol. 34, no. 1, pp. 1–29, Dec. 2021, doi: 10.1186/s10033-021-00570-7.

[2] K. Randhawa, C. K. Loo, M. Seera, C. P. Lim, and A. K. Nandi, "Credit card fraud detection using AdaBoost and majority voting," *IEEE Access*, vol. 6, pp. 14277–14284, 2018, doi: 10.1109/ACCESS.2018.2806420.

[3] M. Seera, C. P. Lim, A. Kumar, L. Dhamotharan, and K. H. Tan, "An intelligent payment card fraud detection system," *Ann. Oper. Res.*, pp. 1–23, Jun. 2021, doi: 10.1007/s10479-021-04149-2.

[4] M. Seera and C. P. Lim, "A hybrid intelligent system for medical data classification," *Expert Syst. Appl.*, vol. 41, no. 5, pp. 2239–2249, 2014, doi: 10.1016/j.eswa.2013.09.022.

[5] M. Seera, C. P. Lim, W. S. Liew, E. Lim, and C. K. Loo, "Classification of electrocardiogram and auscultatory blood pressure signals using machine learning models," *Expert Syst. Appl.*, vol. 42, no. 7, pp. 3643–3652, May 2015, doi: 10.1016/j.eswa.2014.12.023.

[6] M. Seera, C. P. Lim, C. K. Loo, and H. Singh, "Power quality analysis using a hybrid model of the fuzzy Min–Max neural network and clustering tree," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 12, pp. 2760–2767, Dec. 2016, doi: 10.1109/TNNLS.2015.2502955.

[7] S. C. Tan and C. P. Lim, "Application of an adaptive neural network with symbolic rule extraction to fault detection and diagnosis in a power generation plant," *IEEE Trans. Energy Convers.*, vol. 19, no. 2, pp. 369–377, Jun. 2004, doi: 10.1109/TEC.2003.821826.

[8] J. Chen, L. Zhang, Y. Li, Y. Shi, X. Gao, and Y. Hu, "A review of computing-based automated fault detection and diagnosis of heating, ventilation and air conditioning systems," *Renew. Sustain. Energy Rev.*, vol. 161, Jun. 2022, Art. no. 112395, doi: 10.1016/j.rser.2022.112395.

[9] R. M. Souza, E. G. S. Nascimento, U. A. Miranda, W. J. D. Silva, and H. A. Lepikson, "Deep learning for diagnosis and classification of faults in industrial rotating machinery," *Comput. Ind. Eng.*, vol. 153, Mar. 2021, Art. no. 107060, doi: 10.1016/j.cie.2020.107060.

[10] W. Zhang, X. Li, H. Ma, Z. Luo, and X. Li, "Open-set domain adaptation in machinery fault diagnostics using instance-level weighted adversarial learning," *IEEE Trans. Ind. Informat.*, vol. 17, no. 1, pp. 7445–7455, Nov. 2021, doi: 10.1109/TII.2021.3054651.

[11] Y. Yao, J. Wang, and M. Xie, "Adaptive residual CNN-based fault detection and diagnosis system of small modular reactors," *Appl. Soft Comput.*, vol. 114, Jan. 2022, Art. no. 108064, doi: 10.1016/j.asoc.2021.108064.

[12] Z. Xu, C. Li, and Y. Yang, "Fault diagnosis of rolling bearing of wind turbines based on the variational mode decomposition and deep convolutional neural networks," *Appl. Soft Comput.*, vol. 95, Oct. 2020, Art. no. 106515, doi: 10.1016/j.asoc.2020.106515.

[13] P. Konar and P. Chattopadhyay, "Bearing fault detection of induction motor using wavelet and support vector machines (SVMs)," *Appl. Soft Comput.*, vol. 11, no. 6, pp. 4203–4211, Sep. 2011, doi: 10.1016/j.asoc.2011.03.014.

[14] F. Jia and Y. Lei, "A neural network constructed by deep learning technique and its application to intelligent fault diagnosis of machines," *Neurocomputing*, vol. 272, pp. 619–628, Jan. 2018, doi: 10.1016/j.neucom.2017.07.032.

[15] O. AlShorman, M. Irfan, N. Saad, D. Zhen, N. Haider, A. Glowacz, and A. AlShorman, "A review of artificial intelligence methods for condition monitoring and fault diagnosis of rolling element bearings for induction motor," *Shock Vibrat.*, vol. 2020, pp. 1–20, Nov. 2020, doi: 10.1155/2020/8843759.

[16] S. Tang, C. Wang, F. Zhou, X. Hu, and T. Wang, "Multi-scale recursive semi-supervised deep learning fault diagnosis method with attention gate," *Machines*, vol. 11, no. 2, p. 153, Jan. 2023, doi: 10.3390/machines11020153.

[17] O. Kullu and E. Cinar, "A deep-learning-based multi-modal sensor fusion approach for detection of equipment faults," *Machines*, vol. 10, no. 11, p. 1105, Nov. 2022, doi: 10.3390/machines10111105.

[18] L. Qian, Q. Pan, Y. Lv, and X. Zhao, "Fault detection of bearing by Resnet classifier with model-based data augmentation," *Machines*, vol. 10, no. 7, p. 521, Jun. 2022, doi: 10.3390/machines10070521.

[19] M. N. Al-Andoli, S. C. Tan, K. S. Sim, C. P. Lim, and P. Y. Goh, "Parallel deep learning with a hybrid BP-PSO framework for feature extraction and malware classification," *Appl. Soft Comput.*, vol. 131, Dec. 2022, Art. no. 109756, doi: 10.1016/j.asoc.2022.109756.

[20] B. Akay, D. Karaboga, and R. Akay, "A comprehensive survey on optimizing deep learning models by metaheuristics," *Artif. Intell. Rev.*, vol. 55, pp. 829–894, Mar. 2021, doi: 10.1007/s10462-021-09992-0.

[21] M. N. Al-Andoli, S. C. Tan, W. P. Cheah, and S. Y. Tan, "A review on community detection in large complex networks from conventional to deep learning methods: A call for the use of parallel meta-heuristic algorithms," *IEEE Access*, vol. 9, pp. 96501–96527, 2021, doi: 10.1109/ACCESS.2021.3095335.

[22] M. N. Al-Andoli, S. C. Tan, and W. P. Cheah, "Distributed parallel deep learning with a hybrid backpropagation-particle swarm optimization for community detection in large complex networks," *Inf. Sci.*, vol. 600, pp. 94–117, Jul. 2022, doi: 10.1016/j.ins.2022.03.053.

[23] Z. Wang, Q. Zhang, J. Xiong, M. Xiao, G. Sun, and J. He, "Fault diagnosis of a rolling bearing using wavelet packet denoising and random forests," *IEEE Sensors J.*, vol. 17, no. 17, pp. 5581–5588, Sep. 2017, doi: 10.1109/JSEN.2017.2726011.

[24] J. Tian, M. H. Azarian, and M. Pecht, "Anomaly detection using self-organizing maps-based k-nearest neighbor algorithm," in *Proc. PHM Soc. Eur. Conf.*, vol. 2, no. 1, 2014, pp. 1–9, doi: 10.36001/phme.2014.v2i1.1554.

[25] S. He, Z. Wang, Z. Wang, X. Gu, and Z. Yan, "Fault detection and diagnosis of chiller using Bayesian network classifier with probabilistic boundary," *Appl. Thermal Eng.*, vol. 107, pp. 37–47, Aug. 2016, doi: 10.1016/j.applthermaleng.2016.06.153.

[26] Z. Wu, H. Luo, Y. Yang, P. Lv, X. Zhu, Y. Ji, and B. Wu, "K-PdM: KPI-oriented machinery deterioration estimation framework for predictive maintenance using cluster-based hidden Markov model," *IEEE Access*, vol. 6, pp. 41676–41687, 2018, doi: 10.1109/ACCESS.2018.2859922.

[27] Z. Chen and W. Li, "Multisensor feature fusion for bearing fault diagnosis using sparse autoencoder and deep belief network," *IEEE Trans. Instrum. Meas.*, vol. 66, no. 7, pp. 1693–1702, Jul. 2017, doi: 10.1109/TIM.2017.2669947.

[28] S. Gao, L. Xu, Y. Zhang, and Z. Pei, "Rolling bearing fault diagnosis based on intelligent optimized self-adaptive deep belief network," *Meas. Sci. Technol.*, vol. 31, no. 5, May 2020, Art. no. 055009, doi: 10.1088/1361-6501/ab50f0.

[29] A. Maged and M. Xie, "Uncertainty utilization in fault detection using Bayesian deep learning," *J. Manuf. Syst.*, vol. 64, pp. 316–329, Jul. 2022, doi: 10.1016/j.jmsy.2022.07.002.

[30] S. Ma and F. Chu, "Ensemble deep learning-based fault diagnosis of rotor bearing systems," *Comput. Ind.*, vol. 105, pp. 143–152, Feb. 2019, doi: 10.1016/j.compind.2018.12.012.

[31] C. Zhang, Y. He, B. Du, L. Yuan, B. Li, and S. Jiang, "Transformer fault diagnosis method using IoT based monitoring system and ensemble machine learning," *Future Gener. Comput. Syst.*, vol. 108, pp. 533–545, Jul. 2020, doi: 10.1016/j.future.2020.03.008.

[32] A. Mellit and S. Kalogirou, "Assessment of machine learning and ensemble methods for fault diagnosis of photovoltaic systems," *Renew. Energy*, vol. 184, pp. 1074–1090, Jan. 2022, doi: 10.1016/j.renene.2021.11.125.

[33] S. Venkatasubramanian, S. Raja, V. Sumanth, J. N. Dwivedi, J. Sathiaparkavi, S. Modak, and M. L. Kejela, "Fault diagnosis using data fusion with ensemble deep learning technique in IIoT," *Math. Problems Eng.*, vol. 2022, pp. 1–8, Jun. 2022, doi: 10.1155/2022/1682874.

[34] F. Ribeiro, M. Marins, S. Netto, and E. Silva, "Rotating machinery fault diagnosis using similarity-based models," in *Proc. Anais de XXXV Simpósio Brasileiro de Telecomunicações e Processamento de Sinais*, 2017, pp. 3–6.

[35] D. G. Zill, *Advanced Engineering Mathematics*. Burlington, MA, USA: Jones & Bartlett, 2020.

[36] P. Moritz, "Ray: A distributed framework for emerging AI applications," in *Proc. 13th USENIX Symp. Operating Syst. Design Implement. (OSDI)*, 2018, pp. 561–577.

[37] CWRU. *Case Western Reserve University Bearing Data Center*. Accessed: Dec. 22, 2022. [Online]. Available: https://engineering.case.edu/bearingdatacenter

[38] D. Neupane and J. Seok, "Bearing fault detection and diagnosis using case western reserve university dataset with deep learning approaches: A review," *IEEE Access*, vol. 8, pp. 93155–93178, 2020, doi: 10.1109/ACCESS.2020.2990528.

[39] F. M. L. Ribeiro. (2018). *Machinery Fault Database*. Accessed: Jan. 3, 2023. [Online]. Available: https://www02.smt.ufrj.br/~offshore/mfs/page_01.html

[40] C. Lu, Z.-Y. Wang, W.-L. Qin, and J. Ma, "Fault diagnosis of rotary machinery components using a stacked denoising autoencoder-based health state identification," *Signal Process.*, vol. 130, pp. 377–388, Jan. 2017, doi: 10.1016/j.sigpro.2016.07.028.

[41] C. Lu, Z. Wang, and B. Zhou, "Intelligent fault diagnosis of rolling bearing using hierarchical convolutional network based health state classification," *Adv. Eng. Inform.*, vol. 32, pp. 139–151, Apr. 2017, doi: 10.1016/j.aei.2017.02.005.

[42] X. Guo, L. Chen, and C. Shen, "Hierarchical adaptive deep convolution neural network and its application to bearing fault diagnosis," *Measurement*, vol. 93, pp. 490–502, Nov. 2016, doi: 10.1016/j.measurement.2016.07.054.

[43] T. Liang, S. Wu, W. Duan, and R. Zhang, "Bearing fault diagnosis based on improved ensemble learning and deep belief network," *J. Phys., Conf. Ser.*, vol. 1074, Sep. 2018, Art. no. 012154, doi: 10.1088/1742-6596/1074/1/012154.

[44] A. A. de Lima, T. de M Prego, S. L. Netto, E. A. B. da Silva, R. H. R. Gutierrez, U. A. Monteiro, A. C. R. Troyman, F. J. da C Silveira, and L. Vaz, "On fault classification in rotating machines using Fourier domain features and neural networks," in *Proc. IEEE 4th Latin Amer. Symp. Circuits Syst. (LASCAS)*, Feb. 2013, pp. 1–4, doi: 10.1109/LASCAS.2013.6518984.
[45] D. Pestana-Viana, R. Zambrano-Lopez, A. A. de Lima, T. De M. Prego, S. L. Netto, and E. A. B. da Silva, "The influence of feature vector on the classification of mechanical faults using neural networks," in *Proc. IEEE 7th Latin Amer. Symp. Circuits & Syst. (LASCAS)*, Feb. 2016, pp. 115–118, doi: 10.1109/LASCAS.2016.7451023.
[46] M. Messaoudi, S. S. Refaat, M. Massaoudi, A. Ghrayeb, and H. Abu-Rub, "Classification of mechanical faults in rotating machines using SMOTE method and deep neural networks," in *Proc. IECON 48th Annu. Conf. IEEE Ind. Electron. Soc.*, Oct. 2022, pp. 1–6, doi: 10.1109/IECON49645.2022.9968875.
[47] M. A. Marins, F. M. L. Ribeiro, S. L. Netto, and E. A. B. da Silva, "Improved similarity-based modeling for the classification of rotating-machine failures," *J. Franklin Inst.*, vol. 355, no. 4, pp. 1913–1930, Mar. 2018, doi: 10.1016/j.jfranklin.2017.07.038.

MOHAMMED NASSER AL-ANDOLI received the B.Sc. degree in computer information systems from Mutah University, Jordan, in 2011, the M.Sc. degree in computer science from the Jordan University of Science and Technology, Jordan, in 2016, and the Ph.D. degree in information technology from Multimedia University (MMU), Malaysia, in 2022. He is currently a Postdoctoral Researcher with MMU. His main research interests include malware analysis, complex networks analysis, machine learning, high-performance computing, deep learning, and parallel computing.

SHING CHIANG TAN received the dual degrees from Universiti Sains Malaysia, Malaysia, in 1999 and 2002, respectively, and the Ph.D. degree from Multimedia University, Melaka, Malaysia, in 2008. He is currently a Professor with the Faculty of Information Science and Technology, Multimedia University. His current research interests include computational intelligence, deep learning and their applications, which include data classification, condition monitoring, fault detection and diagnosis, stroke rehabilitation, malware detection, and biomedical disease classification. He was a recipient of the Matsumae International Foundation Fellowship, Japan, in 2010.

KOK SWEE SIM (Senior Member, IEEE) is currently a Professor with Multimedia University, Bukit Beruang, Melaka, Malaysia. He is also working closely with various local and overseas institutions and hospitals. He has filed many patents and copyrights. He has received many international and national awards, such as the Academic Science Malaysia (ASM) as Top Research Scientists Malaysia (TRSM); the 2005, 2006, and 2011 World Conference in Applied Computing, USA; the 2008 IEEE Conference at U.K. for national level achievements; and the Korean Innovation and Special Awards, in 2013, 2014, and 2015. He received the Gold Medal Award in the Invention, Innovative Technology Exhibition (ITEX), in 2008, 2009, 2010, 2013, and 2014; the Bio Malaysia Award, in 2009 and 2010; the MMU Best Staff Award, in 2009, 2010, and 2015; the Malaysia Technology Expo 2011, AIK2011, and AIK2012; and the APICTA Gold Medal Award, in 2014 and 2015. He received awards for the TM Kristal Award and two International Championships of World Summit on the Information Society (WSIS) Prizes in the category ICT applications, such as E-science, during the event held in conjunction with WSIS 2016, Geneva, Switzerland, in 2016. He was a recipient of the Japan Society for the Promotion of Science (JSPS) Fellowship, Japan, in 2018.

MANJEEVAN SEERA received the Ph.D. degree in computational intelligence from Universiti Sains Malaysia. He is currently an Associate Professor of business analytics with the School of Business, Monash University Malaysia. He has over 17 years of experience in both academia and industry. His research interests include machine learning principles, applications in finance and engineering, and the design and development of advanced machine learning models for fintech applications, particularly, the detection and prediction of fraudulent financial transactions.

CHEE PENG LIM received the Ph.D. degree from the University of Sheffield, U.K., in 1997. He is currently a Professor with Deakin University, Australia. He has published more than 550 technical papers in books, international journals, and conference proceedings. His research interests include computational intelligence, data analytics, pattern classification, and multi-objective optimization.