IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received May 31, 2021, accepted June 8, 2021, date of publication June 10, 2021, date of current version June 17, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3088237

Light-Weight CNN Enabled Edge-Based Framework for Machine Health Diagnosis

INDRANI MUKHERJEE AND SIDDHARTH TALLUR
Department of Electrical Engineering, Indian Institute of Technology (IIT) Bombay, Mumbai 400076, India
Corresponding author: Siddharth Tallur (stallur@ee.iitb.ac.in)

ABSTRACT Condition Based Monitoring (CBM) leverages sensor measurements for measuring state of health of an asset and autonomous diagnosis of faults to trigger remedial actions. Countless deep learning architectures are available for cloud-based feature engineering, feature extraction and classification of data for CBM models. However, complex models pose memory and processing speed constraints for edge implementation, while cloud-based computing poses high latency, high cost of data transmission and storage and privacy threats. This calls for a data driven machine health diagnosis system that is effective yet edge-compatible and secure. In this work, we present a model that is light-weight and non-redundant by optimizing the model size and complexity for edge implementation onto resource constrained, low-cost hardware. The model is based on a light-weight, edge implementable Convolutional Neural Network (CNN) algorithm that utilizes vibration sensor measurements for fault event estimation of machines. The model was trained and tested on two publicly available and widely studied vibration datasets for rolling element bearing faults. This CNN based classification system using spectrograms is able to achieve near perfect accuracy for both binary as well as multi-class fault classification. Lastly, the trained model was implemented and tested on a Raspberry Pi single-board computer, thus realizing an edge-compatible implementation which is cost-saving, secure and reliable.

INDEX TERMS Condition based monitoring, convolutional neural network, machine health monitoring, edge computing, vibration analysis.

I. INTRODUCTION

Reliability of machinery is indispensable on a plant floor, not just for the safety of operators but for sustained economic benefits of having the production up and running. Condition Based Monitoring (CBM) focuses on being able to monitor the current state of a machine and draw future inferences on its health without disrupting normal operation and simultaneously avoiding non-essential maintenance and associated downtime [1]. CBM is a step towards predictive maintenance through continuous monitoring and hence is extensively data-driven. Recent market developments with increasing commodification of sensors and rapid growth in Artificial Intelligence (AI) and Industrial Internet of Things (IIoT) applications have led to a surge in exploration of deep learning techniques of various forms on raw sensor data from machines for fault diagnosis.

The associate editor coordinating the review of this manuscript and approving it for publication was Mohammad Shorif Uddin.

Machine fault diagnosis essentially is a pattern recognition problem. Machine faults can occur over wide spectral and temporal ranges, and the model must account for the scale to monitor and diagnose critical machinery continually and reliably. Moreover, migration of condition based monitoring models to edge devices is naturally desired, to do away with costs associated with transmission and storage of sensor raw data on the cloud for running diagnostic algorithms. Extensive research has gone into improving algorithms for machine health monitoring viz. classifying wear level of bearings and gears and estimating their remaining useful life starting from simple neural networks and regression based techniques such as Support Vector Machines (SVM) [2], [3] to complex fusion networks with near-perfect accuracy such as deep belief network [4], stacked residual convolutions [5], ensemble deep Convolutional Neural Networks (CNNs) [6] etc. CNNs are the most preferred model architecture in such cases as they combine feature abstraction and localization with reduced number of operations [7], [8]. With an appropriate network architecture with gradient based back-propagation,

CNNs synthesize a complex decision surface and are capable of classifying high-dimensional patterns [9]. Researchers are exploring ensemble transfer CNNs [10], [11], including implementations based on stochastic pooling and Leaky Rectified Linear Unit (LReLU) on multichannel signals for fault diagnosis [12]. However, such complex models are incompatible with edge devices due to memory and processing speed constraints [13] and hence most diagnostic solutions utilize cloud-based post-processing. This cloud-based computing approach has several downsides namely high latency, connectivity dependence, high cost of data transmission and privacy threats [14]. However, the shift from cloud processing to edge computing for CBM has been slow, due to a number of practical challenges. Misra *et al.* discuss the problems faced by VLSI-HNN (Very large-scale integration - hardware neural networks) designers in mapping highly irregular interconnection topologies on regular two dimensional surfaces as they contain complex computations with distributed communication [15]. Embedded Graphics Processing Units (GPUs), Application Specific Integrated Circuits (ASICs), and reconfigurable devices such as Field Programmable Gate Arrays (FPGAs) are being explored as target devices for deep learning, but are far from becoming reliable and cost-effective devices compatible with rugged, harsh industrial plant floor set-up [16]. Zhang *et al.* present various strategies to realize fast, parallel processing designs to develop CNN accelerators but mention the challenge of encountering mismatch of computation throughput with the memory bandwidth provided by an FPGA platform [17]. Apart from such design strategies, another direction that has been explored is the development of compact, compression resilient and non-redundant neural network architectures. Edge devices are limited in terms of performance (memory), cost, and energy while a CNN architecture is an end-to-end approach and hence computationally intensive. This presents an inherent trade-off between the network size and precision. Iandola *et al.* have presented an architecture that has 50× fewer parameters than AlexNet with comparable accuracy [18]. Hadidi *et al.* compared in-the-edge inference of deep neural nets with execution times, energy consumption, and temperature performance and noted that hardware and software co-designs with custom support on edge devices can bring down the inference time [19].

In this work, we propose an extremely light-weight CNN with softmax classifier and gradient-descent backpropagation for bearing fault identification and classification of the fault type. The motivation underlying our work is the realization of a model that is light-weight and non-redundant by optimizing the model size and complexity for edge implementation onto resource constrained, low-cost hardware. Figure 1 shows the framework of our proposed machine health classification system. To test the effectiveness of our system, it was first tested offline (on a workstation) and found to perform with near perfect accuracy (98 %-100 %) on two rich, publicly available and heavily cited bearing fault datasets from Case Western Reserve University and Center for Intelligent Maintenance Systems, University of Cincinnati. Next, the trained model

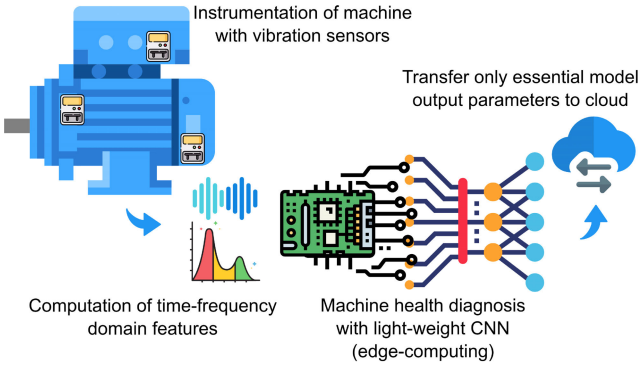

FIGURE 1. Illustration of CNN-enabled edge-based machine health classifier.

was implemented on a Raspberry Pi single-board computer and found to perform efficiently with an inference time of approximately 1 s per test. The paper is structured as follows: section II presents a brief review of feature engineering for vibration signals and CBM algorithms reported in literature, along with a brief description of the datasets used in this study. The architecture of the model developed in this work and results based on validation of the model using the datasets are reported in sections III and IV respectively. Finally, sections V and VI present results obtained by running the trained model on a Raspberry Pi 4 Model B board and conclusions based on this work respectively.

II. BRIEF REVIEW OF MACHINE LEARNING CBM ALGORITHMS

A. TIME-FREQUENCY ANALYSIS OF VIBRATIONS

Bearings are core components of any rotating mechanical machinery as they provide support to the shaft. Bearing malfunction is catastrophic and almost always bring significant downtime with huge repair costs. Bearing fault diagnosis is therefore an active research area in the AI-enabled CBM community [20]. Vibration signature generated from rolling element bearings is often chosen as the raw data for application of prognostic techniques since it is one of the earliest indicators of incipient wear, as compared to other parameters such as temperature, power consumption, and lubricant analysis [1]. Representative features from vibrations are extracted through various signal processing methods and processed using machine learning models for fault identification and classification [21], [22]. The sequential nature of 1-D time series poses additional complexity for feature extractions leading to heavy pre-processing to evaluate time domain parameters such as mean value, kurtosis, skewness, envelope, mean square deviation etc. Fault diagnosis models using 1-D time series for vibrations requires insight and application specific expert knowledge for useful feature selection [23]. Even then, such continuous monitoring systems possess unique challenges because the fault signatures appear at totally different temporal and spectral scales with no definite feature sets to identify all possible faults and

anomalies [24]. Garcia *et al.* [25] have presented six different temporal to image encodings (Gramian Angular Field, Markov Transition Field, recurrence plot, grey scale encoding, spectrogram, and scalogram) to implement deep learning image processing algorithms for unsupervised machine anomaly detection. Wang *et al.* [26] explore eight such time-to-image encodings, including spectrogram, constant-Q Gabor transform, Hilbert Huang transform etc. to identify most suitable encoding scheme. Several studies emphasize that time or frequency domain feature alone cannot explain the changes in vibration data as it is non-stationary for all practical purposes. Spectrogram is the traditional and most widely adopted time-frequency analysis tool for temporal signals across various domains [27]. Mean Peak Frequency (MPF) is a representative feature of spectrograms, that is obtained by finding the mean of the frequencies corresponding to the peak (maximum) Power Spectral Density (PSD) value of every column. In a spectrogram PSD image $P(t, \omega)$, the MPF is an average of the peak frequency at each time instant t (denoted as $f_{peak}(t)$):

$$f_{peak}(t) = \arg\max_{\omega} P(t, \omega)$$

$$MPF = \int_t f_{peak}(t)dt$$

Yan *et al.* have reported that the MPF increases sharply with advancing defects in machines [28]. Therefore, when working with unlabeled machine vibration datasets, MPF could be used as a fault labeling parameter.

B. CONVOLUTIONAL NEURAL NETWORKS FOR BEARING VIBRATION ANALYSIS

Machine health classifier models based on vibration signal analysis are commonly benchmarked using two publicly available and popular datasets: (i) labeled bearing fault dataset from Case Western Reserve University (CWRU) [37], and (ii) unlabeled dataset from Center for Intelligent Maintenance Systems, University of Cincinnati (IMS) [38]. Table 1 shows a comparison of the model presented in this work with other CBM reports on fault diagnosis, failure detection, Remaining Useful Life (RUL) estimation, wear level estimation etc. using deep learning techniques implemented on the CWRU and IMS datasets. The models mentioned in Table 1 are representative of the most popular architectures used in the deep learning community for fault diagnosis, such as fuzzy Adaptive Resonance Theory MAPs (ARTMAPs) [3], CNN-State Vector Machines (CNN-SVMs) [34], stacked auto-encoder (Auto encoders) [36], deep transfer learning and residual networks, to name a few. These architectures range from heavy (multiple hidden layers) to light-weight (residual connections, with depth separable convolutions) and their algorithmic complexities range from high (heavy pre-processing or large floating point computations or both) to low (no significant pre-processing). A brief summary of the key operations involved in these methods are also reported, to provide readers an idea of the level of complexity involved in bearing fault detection and diagnosis. The model sizes reported in literature vary from 17 kB [31] to as large as 50 317 kB [35] with as high as 1.45×10^8 FLoating-point OPerations (FLOPs). To the best of our knowledge, these models utilized workstation/cloud-based pre-processing (i.e. training and testing) and unlike this work, have not been tested for performance on any embedded platform. Recently, Lu *et al.* discussed implementation of fault detection system on a single board computer, namely Raspberry Pi [39]. The work presented by Lu *et al.* was based on their own experimental dataset for bearing vibrations, and not baselined on these popular publicly available machine vibration datasets. Next, we provide a brief introduction to the nature of these datasets and details of their subsets used in our study.

1) CENTER FOR INTELLIGENT MAINTENANCE SYSTEMS BEARING DATASET (IMS)

This bearing dataset provided by the Center for Intelligent Maintenance Systems (IMS), University of Cincinnati, is downloadable from the NASA Ames Prognostics Data Repository [38]. Four bearings were installed on a shaft, and high sensitivity quartz ICP accelerometers were installed on each bearing housing for health monitoring. Three datasets are included in the data packet, wherein each set describes a run-to-failure experiment. The rotation speed was maintained at 2000 rpm and a radial load of 6000 lb was applied onto the shaft and bearing with a spring mechanism. All failures occurred after exceeding designed life time of the bearing (>100 million revolutions). Each dataset consists of individual files corresponding to 1 s long vibration signal time-series recorded at specific intervals. Each file consists of 20 480 samples obtained at 20 kHz sampling rate [40]. For further details on the experimental setup, please refer [41]. Out of the three datasets provided, data for channels 3 and 4 from test number 3 have been used in this study, since this set has the maximum number of data files, and thus suitable for training and testing of the CNN. Since the IMS bearing dataset is an unlabelled dataset, the MPF of the spectrograms (generated for each of the 1 s long signal recordings) has been used to distinguish and label the healthy and faulty data series.

2) CASE WESTERN RESERVE UNIVERSITY BEARING DATASET (CWRU)

Among many publicly available bearing vibration datasets, the CWRU dataset has been widely used to detect and diagnose bearing faults and is accepted as a standard reference for validating models [42]. Figure 2 shows a photograph of the test setup [37]. Single point faults were introduced to the test bearings using electro-discharge machining, resulting in different bearing fault diameters at two locations - Drive End (DE) and Fan End (FE). Sensors (accelerometers) were placed at DE, FE and Base End (BE) of the motor. Table 2 shows the parameters that were varied for the data collection and Table 3 shows the subset of this dataset used to test various cases in this paper.

TABLE 1. Comparison of various CBM methods using deep learning. Glossary – SFAM: Simplified Fuzzy ARTMAP, SVM: State Vector Machine, AE: Auto-Encoder, CNN: Convolutional Neural Network, IDS-CNN: Improved Dempster–Shafer CNN, LDR-CNN: Lightweight Deep Residual CNN, DFCNN: Dropout plus Fully connected CNN, DL: Deep learning, TL: Transfer Learning, IMS: University of Cincinnati, Center for Intelligent Maintenance Systems, CWRU: Case Western Reserve University.

Reference	Architecture	Dataset	Goal	Key operations	Accuracy
Ali et al. [3]	SFAM	IMS	Remaining useful life	Weibull distribution fitting of input feature, optimum SFAM structure selection, SFAM classification, smoothing classification	65.46 %
Lu et al. [29]	SVM	IMS	Wear level	Data segmentation, bootstrap resampling, calculation of CEEMD parameters, CEEMD-IMF parameter calculation, Kernel SVM for health condition classification	98.5 %
Chen et al. [30]	Stacked AE	CWRU	Fault diagnosis	Deep hierarchical structure with three layers of auto encoders, three hidden layers, one classifier	91.67 %-99.83 %
Wang et al. [26]	CNN/Alexnet	CWRU	Failure detection	Time-frequency input image formation, modified AlexNet classifier with five convolutional, three pooling, two normlization and three fully connected layers followed by a softmax layer	94.34 %-100 %
Pinedo et al. [22]	CNN/AlexNet	IMS	Wear level	Feature extraction, Shannon's entropy calculations, feature clustering, signal to image transformation, Alexnet classifier	99.25 %
Li et al. [6]	IDS-CNN	CWRU	Failure detection	Sub-bands and RMS value extraction from time series, 2-D image mapping, CNN classifier with three convolution, three pooling, one fully connected and one softmax layer, IDS evidence theory implementation from evidences and modified Gini Index, classified label allocation	89 %-100 %
Ma et al. [31]	LDR-CNN	CWRU, IMS	Fault diagnosis	Wavelet packet generation, LDR-CNN with one convolutional+CReLU unit followed by six units with one each of pointwise convolutional layer, depth separable convolutional layer, CReLU activation layer,batch normalization layer and an identity shortcut, followed by an average pooling, fully connected and softmax activation layer	100 %
Min et al. [32]	CNN	CWRU	Fault diagnosis	2-D represntaion by data fusion from multiple sensors, CNN classifier with two layers each of convolution, pooling, dropout, and one layer each of fully-connected and softmax	96.67 %-100 %
Zhang et al. [33]	DFCNN	CWRU	Fault diagnosis	2-D feature extraction including Wavelet, FFT and EMD, followed by classification using four convolution, four maxpool,two dropout and two fully connected layers	99 %-100 %
Yuan et al. [34]	CNN-SVM/ Resnet-18-SVM	CWRU	Fault diagnosis	2-D wavelet transform, input to Resnet-18 (18 layers with skip connections) followed by SVM for classification	98.2 %-99.2 %
Wen et al. [35]	CNN/LeNet-5	CWRU	Fault diagnosis	2D image formation from raw signals as input to 4 conv layers followed by 2 fully connected layers	99.79 %
Vikas et al. [36]	AE with DL, TL	CWRU, IMS	Fault diagnosis	Relevant feature selection and pre-training of sparse AE, fine-tuning of DNN, deep transfer learning	86 %-100 %
This work	Basic CNN	CWRU, IMS	Fault diagnosis	Time-frequency input image formation, classifier with single convolution, pooling and fully connected layer with softmax activation, implementation of trained model on Raspberry Pi	98 %-100 %

TABLE 2. CWRU bearing dataset overview.

Parameter	Types
Sampling rate	12 kHz and 48 kHz
Fault location	DE, FE
Sensor locations	DE, FE, BE
Fault diameters	7 mil, 14 mil, 21 mil, 28 mil, 40 mil
Fault classes	Healthy, Ball Bearing (BB), Inner Raceway (IR), Outer Raceway (OR)
Motor speeds	1720 rpm and 1797 rpm

TABLE 3. Subset of CWRU dataset used in this study. All fault classes and motor speeds are considered.

Label	Fault	Sensor	Sampling rate	Diameter
A	DE	DE	12 kHz	7 mil, 14 mil, 21 mil, 28 mil
B	DE	FE	12 kHz	7 mil, 14 mil, 21 mil
C	DE	BA	12 kHz	7 mil, 14 mil, 21 mil
D	DE	DE	48 kHz	7 mil, 14 mil, 21 mil
E	DE	DE, FE, BE	12 kHz	7 mil, 14 mil, 21 mil

III. PROPOSED MODEL - CNN BASED MACHINE HEALTH STATE CLASSIFIER

In this study, we use spectrograms as the input feature to our light-weight CNN. The three hyper-parameters considered for STFT are window type: Hanning window, window size: 1024 points, overlap factor: 50 %, and number of points in FFT: 1024. The total number of time series

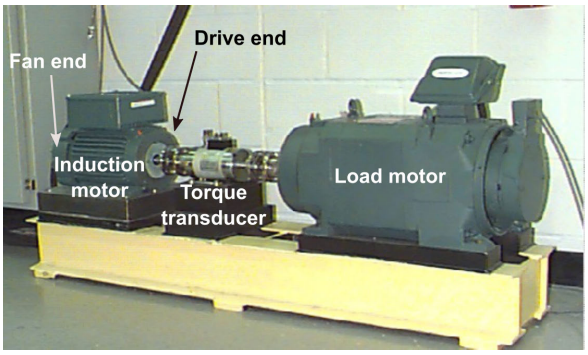

FIGURE 2. CWRU bearing test setup [37] comprising of a 2 hp induction motor, torque transducer/encoder, load motor, and control electronics (not shown). Faulty bearings (with various faults engineered) were placed at the drive end and fan end and vibration sensors located at drive end (DE), fan end (FE) and base end (BE) (not shown) of the motor.

data samples taken to generate spectrograms is maintained as 20 480 throughout the study. The resulting spectrograms are of size 513×39. Since the resolution (quality) of input image directly influences feature abstraction and the classification accuracy achieved by the CNN model, the size of the spectrogram is maintained across all cases investigated in this work. It was verified that this resolution was satisfactory and increasing the spectrogram resolution (either in time or frequency) beyond this size did not result in noticeable improvements in image quality for both the IMS and CWRU datasets. However, since the CNN model is agnostic to input image dimensions, the spectrogram size can be modified as per the raw data quality when working with other datasets. The structure of the proposed classifier is that of a basic CNN consisting of a single convolution, pooling and fully-connected (FC) layer as shown in figure 3. Softmax has been used as the activation function in the FC layer, as it produces outputs that add up to 1, and therefore can be interpreted as probability (likelihood) of an image belonging to a particular class. The CNN model is implemented in python, without using any deep learning programming platform to avoid package and library dependencies so as to optimize the program size for subsequent edge-implementation of the trained model. The 1-D time series vibration signals are converted into spectrograms of size 513×39 as part of pre-processing for input feature generation. The spectrograms are provided as input to the convolution layer. Across literature, CNNs for image processing and computer vision applications typically have odd-sized filters with depth typically ranging from 8 to 512. The odd size of a filter provides symmetry around the output pixel. Since our key idea is to keep the model light-weight yet efficient, the convolution layer has 8 filters of size 3×3. The input to this layer is a spectrogram image of size 513×39 and output is of size 511×37×8, since the convolution operation uses valid padding. The convolution layer output is then provided as input to the pooling layer of type maxpool and size =2, that subsamples the convolved images and provides an output image of size 255×18. The last layer is the FC layer with softmax activation. The maxpool layer output is vectorized to a size of 36 720×1 (255×18×8) prior to plugging into the FC layer. This is passed through the non-linear softmax activation, with two or four likelihood outputs (depending on whether the model is to be trained for binary or multi-class classification e.g. CWRU). For binary classification, each spectrogram is labeled/classified as 'Healthy' or 'Faulty', whereas for multi-class classification for CWRU dataset, each spectrogram is labeled/classified into one of 'Healthy', 'Ball bearing fault', 'Inner raceway fault' or 'Outer raceway fault' type of faults. The total number of learnable parameters in the proposed model for multiclass classification is given by:

$$w_{conv} + w_{FC} + b_{FC} = 3\times3\times8 + 36720\times4 + 4 = 146956$$

where w_{conv} denotes the number of weights in the convolution layer ($3\times3\times8$), w_{FC} denotes the number of weights in FC layer (36720×4) and $b_{FC} = 4$ is the number of bias nodes in FC layer. This results in a model size of approximately 587.82 kB. The computational complexity in the model is primarily determined by multiplication FLOPs in the convolutional and FC layers. The number of multiplication FLOPs per filtering instance (n_f) is obtained as:

$$n_f = r_f \times c_f \times ch_f \times N_f = 3\times3\times1\times8 = 72$$

where r_f, c_f and ch_f denote the number of rows, columns and channels per filter respectively and N_f denotes the total number of filters in the convolutional layer. The total number of instances in the convolutional layer ($n_{instance}$) is expressed as: $n_{instance} = (r_f - r_i + 1)\times(c_f - c_i + 1) = 511\times37 = 18907$, where r_i and c_i denote the number of rows and columns in the input image. The number of multiplication FLOPs in the convolutional layer is thus expressed as:

$$n_{conv} = n_f \times n_{instance} = 72 \times 18907 = 1361304$$

Similarly, the number of multiplication FLOPs in the FC layer of size $s_{FC} = 36720$ and $l_{output} = 4$ outputs for multiclass classification is obtained as:

$$n_{FC} = s_{FC} \times l_{output} = 36720 \times 4 = 146880$$

The total number of multiplication FLOPs is thus specified as $n_{conv} + n_{FC} = 1361304 + 146880 = 1508184 \approx 1.5\times10^6$. For binary classification, the total number of learnable parameters is 73 514, resulting in a model of size 294.05 kB, and the total number of FLOPs is obtained as 1.4×10^6. Note that the number of FLOPs is dependent on the input size and thus, scaling the input scales the FLOPs. In this work, the input spectrogram size is constant across all test cases. However, one can scale down the input image size to as extent where the resolution of difference in spectrograms across different classes is not compromised.

IV. EXPERIMENTAL VALIDATION AND RESULTS

The model was trained and tested multiple times using different combinations of both CWRU and IMS datasets.

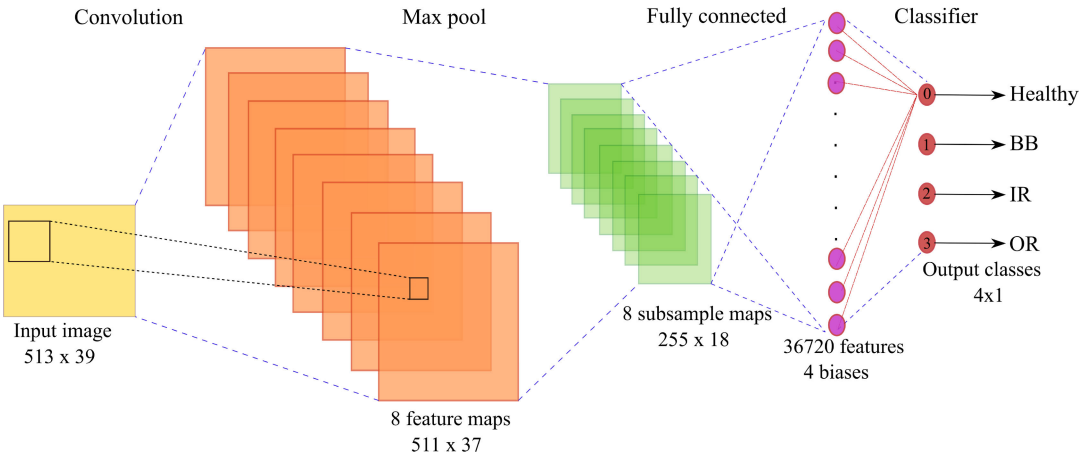

FIGURE 3. Architecture of the light-weight CNN model that comprises of convolution layer consisting of 8 filters of size 3×3, followed by maxpool layer and a fully connected layer with softmax activation. Each time-series with 20 480 samples is converted to spectrogram image of size 513×39 that is then used as input to the model. The model produces likelihoods for each machine health state as the output.

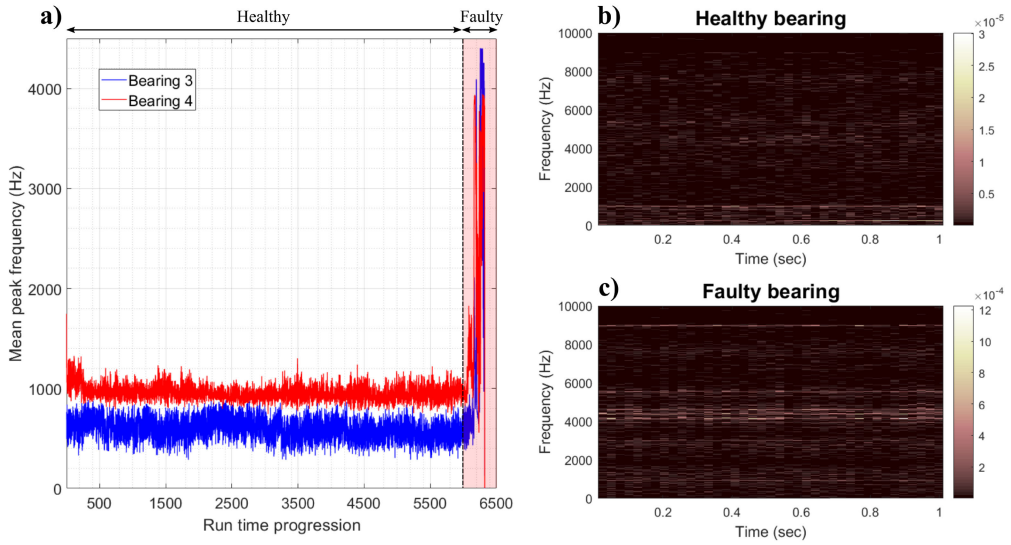

FIGURE 4. a) Mean peak frequencies (MPF) of 6324 data files from IMS dataset, as the machine run time progresses towards failure. The MPF progression for vibration sensor data collected from both Bearing 3 and Bearing 4 show a sharp increase with onset of failure, and is therefore used to label the data as 'Healthy' and 'Faulty' as shown. b) and c) Spectrograms of vibration data from IMS dataset corresponding to Bearing 3 recorded at 20 kHz sampling rate for b) healthy bearing, and c) faulty bearing.

For training, a standard workstation with 16 GB RAM and an Intel(R) core i7-8700 processor was used. Before generating the input features, spectrograms and corresponding MPF for all cases were investigated to understand underlying data trends. The test accuracy of the model was examined on data from the same distribution as the training set as well as on unseen data from a sensor at a different location, sampling rate or fault location not included in the original training set.

A. DATA PRE-PROCESSING FOR INPUT FEATURE GENERATION

1) IMS

To test the model on IMS dataset, the third test set in which outer race fault has occurred on the third bearing at the end of run-time was selected. Hence, data corresponding to sensor placed on the third bearing (Bearing 3) were used to visualize the spectrograms and the underlying trend in MPF as the motor run-time progressed towards failure. Since most faults occur near the end of machine run time, time series recordings corresponding to healthy data are always invariably large in number while those for faulty data are scant. Hence, data from the sensor placed on the fourth bearing (Bearing 4) were also used to increase the number of faulty data series, so as to have balanced healthy and faulty classes for model training. Figure 4(a) shows the variation in MPF of spectrograms of all 6324 vibration data files corresponding to Bearing 3 and Bearing 4 as the machine continued to run till failure. It can be observed that the MPF does not vary significantly for

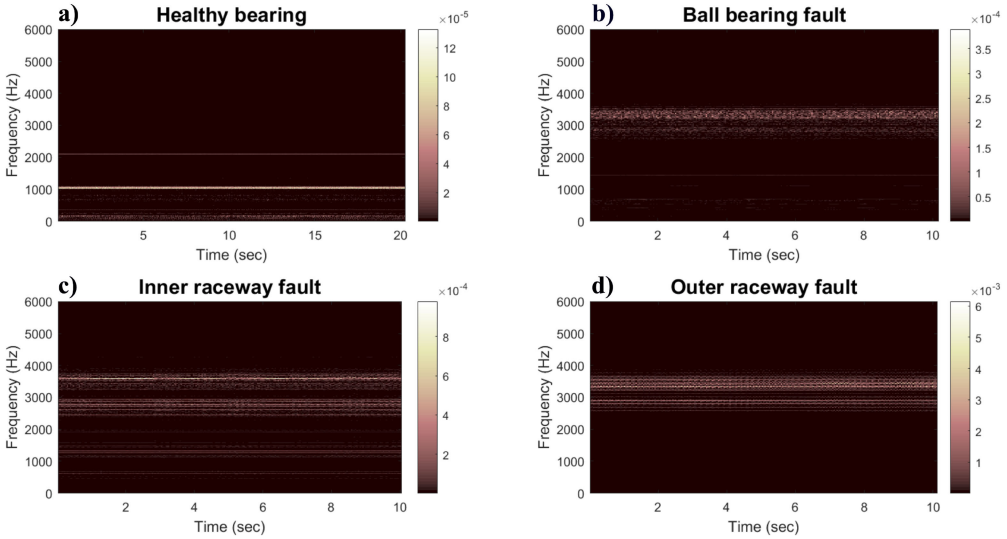

FIGURE 5. Spectrograms of vibration data for CWRU dataset for sensor and fault located at drive end sampled at 12 kHz for a) healthy bearing, b) ball bearing fault, c) inner raceway fault, and d) outer raceway fault.

most of the machine run-time since commencement of data collection, and increases drastically towards the end due to progression of failure, starting from approximately the 5900^{th} record out of 6324 total records. This information is used for labeling the spectrograms into healthy and faulty classes by marking the 1^{st} to 5899^{th} spectrograms as belonging to the 'Healthy' class while 5900^{th} to 6324^{th} spectrograms are labeled as belonging to the 'Faulty' class. Similarly, the MPF of data from Bearing 4 were observed to show sharp increase from the 6000^{th} spectrogram, and therefore the 6000^{th} to 6324^{th} spectrograms for Bearing 4 are added to the 'Faulty' class. Thus, combining the data from Bearing 3 and Bearing 4 provides a total of 750 spectrograms (425 from Bearing 3 and 325 from Bearing 4) labeled as 'Faulty'. To balance both classes, 750 spectrograms were chosen at random from the 'Healthy' class. Figure 4(b) and 4(c) show spectrograms of two vibration data files of the third bearing corresponding to healthy and faulty states respectively.

2) CWRU

Figure 5(a) shows example spectrograms obtained from samples of vibration data corresponding to the 'Healthy' bearing and 5(b), (c) and (d) from data subset labeled A in Table 3. Note that most of the spectral content of a healthy bearing is restricted to lower frequency bands. The spectrograms of BB and OR faults have significant similarities in the frequency bands with major differences only in PSD amplitude. The IR fault spectrogram has signal spread over a large spectral range, with significant overlap with the other three cases. Figure 6 shows example spectrograms for CWRU dataset for 7 mil diameter, ball bearing fault chosen from data subsets A (6a), B (6b), C (6c) and D (6d) respectively. The data for the subset D is recorded with 48 kHz sampling rate and was downsampled by a factor of 4 (Figure 6(d)). We observe that a change in sensor location causes significant changes in the time-frequency representation of a vibration signal.

B. CNN CLASSIFICATION

For the CWRU dataset, six different combinations of train+test cases have been examined in this work, whereas for IMS dataset, one case has been examined. The details of the training, test and model performance in terms of overall accuracy as well as classwise precision (proportion of positive identifications that are actually correct) and recall (proportion of actual positives that are identified correctly) are detailed below. All trainings were performed for 3 epochs with a learning rate of 0.05. When input batch size for training is set to 100, the average training time per batch in an epoch was found to be approximately 0.82 s. The stability is confirmed through observation of saturation of average loss within 10 to 15 batches of training in all cases. Five iterations of model testing were performed for each case. The precision P and recall R are expressed in terms of number of true positives TP, false positives FP, true negatives TN and false negatives FN as follows:

$$P = \frac{TP}{TP + FP}, R = \frac{TP}{TP + FN}$$

The F_1 score of a class (defined as the harmonic mean of the precision and recall) is used to analyze the class-wise classification efficiency of a model and has been reported for all our test cases. Table 4 provides a summary of the performance of the model performance on all test cases for both datasets.

1) IMS

Binary classification was performed on the IMS dataset using data files for Bearings 3 and 4 as described above. A total of 1500 spectrograms (healthy and faulty) were randomly

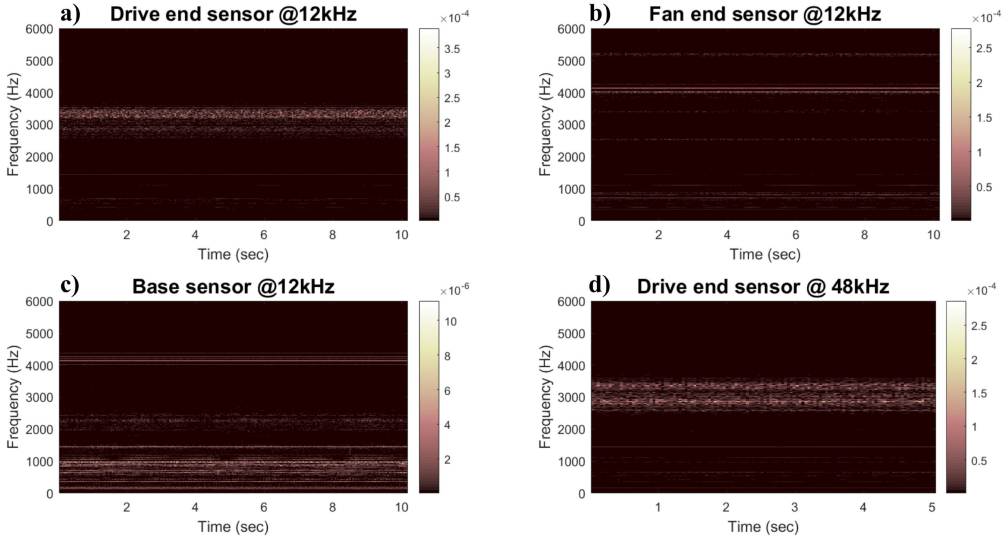

FIGURE 6. Spectrograms from CWRU dataset for a ball bearing fault (7 mil diameter) at the drive end of motor captured by a) drive end sensor, 12 kHz sampling rate, b) fan end sensor, 12 kHz sampling rate, c) base sensor, 12 kHz sampling rate, and d) drive end sensor, 48 kHz sampling rate.

TABLE 4. Summary of model performance for different cases. Accuracy range reported based on five iterations of testing. Cases III.A, III.B, III.C correspond to instances where the model is trained and tested on different sensors, and it is not surprising that the accuracy is significantly lower. Including data from all sensors in training results in perfect accuracy (case IV).

Case	Dataset	Train set	Test set	Classification type	Accuracy
IMS	IMS	3	3	Binary	98.33 %
I	CWRU	A	A	Binary	100 %
II	CWRU	A	A	Multi-class	100 %
III.A	CWRU	A	B	Multi-class	49 %-65 %
III.B	CWRU	A	C	Multi-class	50 %-64 %
III.C	CWRU	A	D	Multi-class	72 %-76 %
IV	CWRU	A, B, C	A, B, C	Multi-class	100 %

split in the ratio of 80:20 for train and test respectively, resulting in 1200 spectrograms for training and 300 for testing. This was repeated five times and test accuracy varied in the range of 97.33 % - 99.00 %. Figure 7(a) shows the confusion matrix for one such trial, the F_1 scores for which are 0.985 for 'Healthy' class and 0.982 for 'Faulty' class. The F_1 scores of 'Healthy' class for all the five test cases lie in the range of $0.972 - 0.991$ whereas for 'Faulty' class, the F_1 score lies in the range $0.974 - 0.989$ Higher number of faulty spectrograms were observed as getting classified as healthy. It should be noted that this is a run-to-failure, unlabeled dataset for which MPF has been used for assigning class labels manually. As can be observed from Figure 4(a), the MPF of faulty spectrograms near the class separation boundary will have closer resemblance to healthy ones than that of the faulty sets towards the end of the progression. This

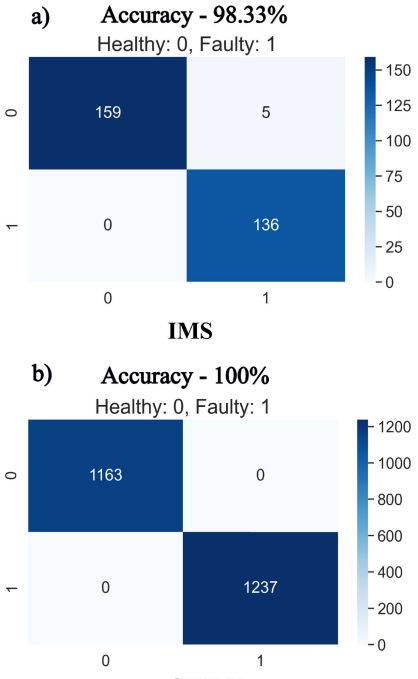

FIGURE 7. Confusion matrix showing results obtained for binary classification performed on a) IMS dataset, and b) CWRU dataset.

phenomenon is reflected in the performance of the model for samples lying near class boundary which have low confidence for class separation.

2) CWRU

CASE I

Here both train and test samples were taken from the data subset A (Table 3) for binary classification ('Healthy', and 'Faulty'). The faulty class represents spectrograms with fault

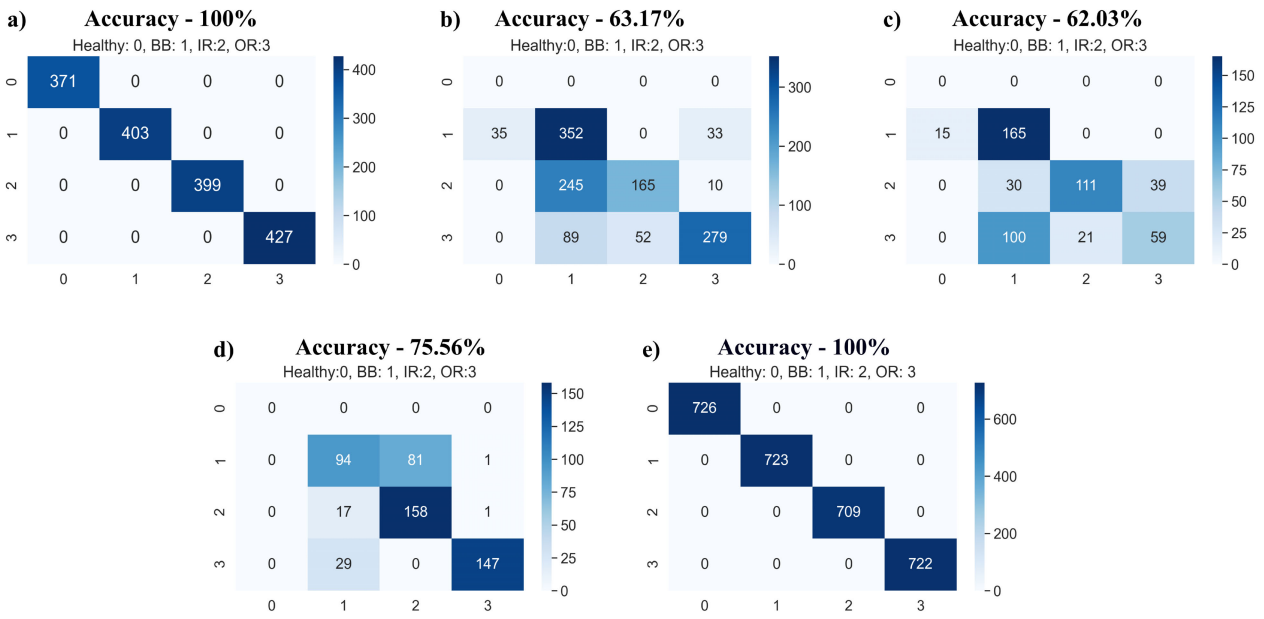

FIGURE 8. Model performance example for various combinations of subset of CWRU dataset, as described in section IV-B: a) case II, b) case III.A, c) case III.B, d) case III.C, e) case IV. The model has perfect accuracy (100 % when presented with 'seen' data in test i.e. data obtained from all sensors is used for training.

of any type - BB, IR or OR at the DE location, captured by the sensor at the DE location, whereas healthy signifies spectrograms of motor at normal running condition, all data being sampled at a rate of 12 kHz. A total of 12 000 such spectrograms were generated from the time-series data files using a sliding window in time, and were then split randomly in the ratio of 80:20 for train and test respectively resulting in 9600 spectrograms for training and 2400 for testing. Five such trials were performed and a perfect test accuracy was achieved in all of them. Figure 7(b) shows the confusion matrix of predictions for this case. Since, all classes were predicted correctly all the time, the F_1 scores for all classes in this case is 1.

CASE II

Here again both train and test samples were taken from the data subset A (Table 3), but used for multi-class classification into four classes: 'Healthy', 'BB' (Ball bearing fault), 'IR' (Inner raceway fault) and 'OR' (Outer raceway fault). A total of 8000 such spectrograms having 2000 spectrograms from each class were generated from the time-series data. Spectrograms from each class were split randomly in the ratio of 80:20 for train and test respectively, resulting in a total of 6400 spectrograms for training and 1600 for testing. In this case too, the model was able to predict all the test cases correctly in all five trials resulting in an F_1 score of 1 for all classes. Figure 8(a) shows the confusion matrix of predictions for one such case.

CASE III

In the following three sub-cases multi-class classification was performed, wherein the model was trained using data subset A but tested with data from sensors at different locations and sampling rates. This was done to check the generalization and adaptation capability of the model. Without any specific interventions, the performance of CNNs is expected to be poor when faced with unseen data (i.e. not used for training).

a: CASE III.A

In this case, the spectrograms from the data subset A were chosen for training, while spectrograms picked from data subset B (Table 3) were used for testing. This corresponds to training using data from the sensor placed at DE, while testing based on data obtained from a different location - FE. The model was trained for multi-class classification with same class notations as that of Case 2. A total of 6400 spectrograms were used for training the model while 1260 spectrograms (420 for each of the three fault classes - BB, IR and OR) were used for testing. The obtained test accuracy was in the range of 49 %-65 %. Figure 8(b) shows an example of the confusion matrix of predictions for one of the five trials for this case. The accuracy for this test case is 63.17 % with the class-wise F_1 scores as Healthy: 0.96, BB: 0.63, IR: 0.51, OR: 0.75. The class-wise F_1 scores for all five trials vary in the range of 0.84-0.96 for Healthy, 0.19-0.63 for BB, 0.48-0.61 for IR and 0.67-0.75 for OR.

b: CASE III.B

In this case, the spectrograms for training were taken from the data subset A while for testing, spectrograms were picked from data subset C (Table 3) corresponding to the sensor placed at a different location - BE in this case. The model was trained for multi-class classification with same class notations as that of Case 2. A total of 6400 spectrograms were

used for training the model while 540 spectrograms (180 for each of the three fault classes - BB, IR and OR) were used for testing. The obtained test accuracy was in the range of 50 %-64 %. Figure 8(c) shows an example of the confusion matrix of predictions for one of the trials for this case. The accuracy for this test case is 62.03 % with class-wise F_1 scores: Healthy: 0.98, BB: 0.69, IR: 0.71, OR: 0.42. The class-wise F_1 scores for all five trials vary in the range of 0.97-0.98 for Healthy, 0.68-0.69 for BB, 0.69-0.71 for IR and 0.42-0.48 for OR.

c: CASE III.C

In this case, the spectrograms for training were taken from the data subset A while for testing, spectrograms were picked from data subset D (Table 3) corresponding to data captured at a different sampling rate - 48 kHz. The test data were down-sampled by a factor of 4 before generation of spectrograms. The model was trained for multi-class classification with same class notations as that of Case 2. A total of 6400 spectrograms were used for training the model while 528 spectrograms (176 for each of the three fault classes - BB, IR and OR) were used for testing. The obtained test accuracy was in the range of 72 %-76 %. Figure 8(d) shows an example of the confusion matrix of predictions for one of the trials for this case. The accuracy for this test case is 75.56 % with class-wise F_1 scores: Healthy: 1, BB: 0.60, IR: 0.76, OR: 0.90. The class-wise F_1 scores for all five trials are 1 for Healthy, and range from 0.60-0.71 for BB, 0.53-0.76 for IR and 0.903-0.904 for OR.

Thus, when the model is presented with data obtained from a sensor not included in the training set, the classification accuracy significantly reduces. This drop is higher for a change of sensor location from the drive end (set A) to the fan end (set B) of the motor (case III.A) than to the motor base (set C) (case III.B). The fan end is physically located farther from the drive end as compared to the motor base (Figure 2). When sensor data from same location used for training, but recorded with a different sampling rate is presented (case III.C), the accuracy drops significantly due to presence of additional information in spectrograms obtained with larger sampling rate. The drop in accuracy in these three cases is fundamentally attributed to inability of the model to approximate data from unseen distribution. This is overcome by including portions of data from sensors at fan end and base of the motor into the training set. The model is able to approximate all test cases correctly, as presented below.

CASE IV

Here, the unseen test data sets from cases III.A, III.B and III.C above were also included in training. Thus both the train and test samples were taken from the data subset E (Table 3) for multi-class classification with same class notations as that of Case 2. A total of 14 400 such spectrograms having 3600 spectrograms from each class were generated from the time series data and were split randomly in the ratio of 80:20 for train and test respectively. It was observed that the model was able to predict all the test cases correctly when trained with the entire dataset as expected. Figure 8(e) shows the confusion matrix of predictions for this case. Here again, F_1 scores for all classes in all trials are consistently equal to 1.

TABLE 5. Comparison of run-time per inference (testing) on embedded platform (Raspberry Pi - RPi) versus desktop workstation.

Case	Run time on workstation	Run time on RPi
IMS	151.2 ms	1.21 s
I	153.8 ms	982.5 ms
II	142.2 ms	924.3 ms
III.A	148.1 ms	1.003 s
III.B	151.5 ms	981.5 ms
III.C	149.9 ms	997.0 ms
IV	151.9 ms	1.002 s

V. EDGE IMPLEMENTATION ON RASPBERRY PI

The trained model was implemented on a Raspberry Pi 4 Model B (RPi) single board computer to evaluate the feasibility of implementation on an embedded platform with 8 GB RAM and 64-bit quad-core Cortex-A72 (ARM v8) processor. The time taken per inference for all test cases of CWRU and IMS datasets are tabulated in Table 5. A comparison for (trained) model run-time on a standard workstation with 16 GB RAM and an Intel(R) core i7-8700 processor is presented. This workstation is used to perform model training in all cases. Note that the time taken per inference on RPi is approximately 1 s, while that on a desktop computer is roughly 0.15 s. In a typical IIoT setup, a machine might be diagnosed once every few hours or days, depending on machine criticality. While edge-computing on RPi requires 6.75× higher test time than cloud-based or workstation-based inference, the time scales required for the computation are insignificant compared to the frequency of taking such measurements and remedial action as necessary. Using a cost-effective embedded platform such as RPi provides additional advantages such as scalability (multiple installations at several locations in a plant), and on-the-spot diagnosis, that is immune to problems associated with loss of cloud connectivity and privacy threats.

VI. CONCLUSION

Bearing in mind the requirement of an edge implementable model, the CNN model presented in this work realizes a light-weight yet efficient edge-compatible machine health state classifier. The model has been extensively tested on the labeled Case Western Reserve University (CWRU) bearing dataset as well as the unlabeled University of Cincinnati Center for Intelligent Maintenance Systems (IMS) bearing dataset. The trained model provides impressive performance (98-100 % accuracy) for learnt distributions of various data combinations presented in this work, when tested

on 'seen' data. This study also validates the use of mean peak frequency of spectrograms as a useful machine health state labeling parameter. To achieve high-performance bearing fault diagnosis under varying sensor locations, load conditions and sampling rates, extensive training can be performed offline and the trained model can then be used for on-edge diagnosis. We observe that the sensor location and sampling rate are key parameters that influence the temporal and spectral scales of a vibration spectrogram and thus it is advisable to train the model extensively on all possible combinations of sensor placements and data acquisition rates for a robust classification accuracy. The work presented in this paper could be used to realize a cost-effective CBM solution. The feature engineering, model design and embedded implementation presented here are applicable across several domains for data-driven deep learning applications involving supervised classification.

Our future work will focus on model validation on additional datasets currently under development in our laboratory. Further work is needed to improve the generalization capability of the model without compromising the requirement of edge implementation. This requires optimization on both algorithmic as well as architectural front, and we aim to explore methods such as fully convolutional networks [43], [44], transfer learning techniques for domain adaptation [45], [46], and layer optimization techniques such as depth separable convolution [47], [48]. Generation of spectrograms as input features from time-series signals consumes a significant percentage of the processing time and thus exploration of alternate features is another area we seek to explore. Moreover, the availability of sufficiently large volume of labeled data collected for all possible faults is a practically relevant challenge. Therefore, another direction we aim to pursue in future work is the development of edge-compatible unsupervised machine health state classifiers.

ACKNOWLEDGMENT

The authors would like to thank Prof. V. Rajbabu at IIT Bombay for insightful discussions on CNNs. Indrani Mukherjee would like to thank Analog Devices India Pvt. Ltd. for supporting her M.Tech. fellowship.

REFERENCES

[1] R. B. Randall, *Vibration-Based Condition Monitoring: Industrial, Aerospace and Automotive Applications*. Hoboken, NJ, USA: Wiley, 2011.

[2] P. Li, Y. Jiang, and J. Xiang, "Experimental investigation for fault diagnosis based on a hybrid approach using wavelet packet and support vector classification," *Sci. World J.*, vol. 2014, Feb. 2014, Art. no. 145807.

[3] J. B. Ali, B. Chebel-Morello, L. Saidi, S. Malinowski, and F. Fnaiech, "Accurate bearing remaining useful life prediction based on weibull distribution and artificial neural network," *Mech. Syst. Signal Process.*, vols. 56–57, pp. 150–172, May 2015.

[4] Z. Chen, X. Zeng, W. Li, and G. Liao, "Machine fault classification using deep belief network," in *Proc. IEEE Int. Instrum. Meas. Technol. Conf.*, May 2016, pp. 1–6.

[5] Z. Zhuang, H. Lv, J. Xu, Z. Huang, and W. Qin, "A deep learning method for bearing fault diagnosis through stacked residual dilated convolutions," *Appl. Sci.*, vol. 9, no. 9, p. 1823, May 2019.

[6] S. Li, G. Liu, X. Tang, J. Lu, and J. Hu, "An ensemble deep convolutional neural network model with improved D-S evidence fusion for bearing fault diagnosis," *Sensors*, vol. 17, no. 8, p. 1729, Jul. 2017.

[7] E. Davies, "Deep-learning networks," in *Computer Vision*, 5th ed., E. Davies, Ed. New York, NY, USA: Academic, 2018, ch. 15, pp. 453–493. [Online]. Available: https://www.sciencedirect.com/science/article/pii/B9780128092842000150

[8] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, vol. 25, Dec. 2012, pp. 1097–1105.

[9] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.

[10] P. Cao, S. Zhang, and J. Tang, "Preprocessing-free gear fault diagnosis using small datasets with deep convolutional neural network-based transfer learning," *IEEE Access*, vol. 6, pp. 26241–26253, 2018.

[11] S. S. Udmale, S. K. Singh, R. Singh, and A. K. Sangaiah, "Multi-fault bearing classification using sensors and ConvNet-based transfer learning approach," *IEEE Sensors J.*, vol. 20, no. 3, pp. 1433–1444, Feb. 2020.

[12] Z. He, H. Shao, X. Zhong, and X. Zhao, "Ensemble transfer CNNs driven by multi-channel signals for fault diagnosis of rotating machinery cross working conditions," *Knowl.-Based Syst.*, vol. 207, Nov. 2020, Art. no. 106396.

[13] Q. Zhang, M. Zhang, T. Chen, Z. Sun, Y. Ma, and B. Yu, "Recent advances in convolutional neural network acceleration," *Neurocomputing*, vol. 323, pp. 37–51, Jan. 2019.

[14] M. P. Véstias, "A survey of convolutional neural networks on edge with reconfigurable computing," *Algorithms*, vol. 12, no. 8, p. 154, Jul. 2019.

[15] J. Misra and I. Saha, "Artificial neural networks in hardware: A survey of two decades of progress," *Neurocomputing*, vol. 74, nos. 1–3, pp. 239–255, Dec. 2010.

[16] Y. Umuroglu, N. J. Fraser, G. Gambardella, M. Blott, P. Leong, M. Jahre, and K. Vissers, "FINN: A framework for fast, scalable binarized neural network inference," in *Proc. 2017 ACM/SIGDA Int. Symp. Field-Program. Gate Arrays*, 2017, pp. 65–74.

[17] C. Zhang, P. Li, G. Sun, Y. Guan, B. Xiao, and J. Cong, "Optimizing FPGA-based accelerator design for deep convolutional neural networks," in *Proc. 2015 ACM/SIGDA Int. Symp. Field-Program. Gate Arrays*, 2015, pp. 161–170.

[18] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer, "SqueezeNet: AlexNet-level accuracy with 50 x fewer parameters and <0.5 MB model size," 2016, *arXiv:1602.07360*. [Online]. Available: http://arxiv.org/abs/1602.07360

[19] R. Hadidi, J. Cao, Y. Xie, B. Asgari, T. Krishna, and H. Kim, "Characterizing the deployment of deep neural networks on commercial edge devices," in *Proc. IEEE Int. Symp. Workload Characterization (IISWC)*, Nov. 2019, pp. 35–48.

[20] W. A. Smith and R. B. Randall, "Rolling element bearing diagnostics using the Case Western Reserve University data: A benchmark study," *Mech. Syst. Signal Process.*, vol. 64, pp. 100–131, Dec. 2015.

[21] Y. Xie and T. Zhang, "Fault diagnosis for rotating machinery based on convolutional neural network and empirical mode decomposition," *Shock Vibrat.*, vol. 2017, Aug. 2017, Art. no. 3084197.

[22] L. A. Pinedo-Sánchez, D. A. Mercado-Ravell, and C. A. Carballo-Monsivais, "Vibration analysis in bearings for failure prevention using CNN," *J. Brazilian Soc. Mech. Sci. Eng.*, vol. 42, no. 12, pp. 1–17, Nov. 2020.

[23] O. R. Seryasat, M. A. Shoorehdeli, F. Honarvar, and A. Rahmani, "Multi-fault diagnosis of ball bearing based on features extracted from time-domain and multi-class support vector machine (MSVM)," in *Proc. IEEE Int. Conf. Syst., Man Cybern.*, Oct. 2010, pp. 4300–4303.

[24] E. D. Valle, S. Ceri, F. V. Harmelen, and D. Fensel, "It's a streaming world! Reasoning upon rapidly changing information," *IEEE Intell. Syst.*, vol. 24, no. 6, pp. 83–89, Nov. 2009.

[25] G. R. Garcia, G. Michau, M. Ducoffe, J. S. Gupta, and O. Fink, "Temporal signals to images: Monitoring the condition of industrial assets with deep learning image processing algorithms," *Proc. Inst. Mech. Eng., O, J. Risk Rel.*, Feb. 2021, Art. no. 1748006X21994446, doi: 10.1177/1748006X21994446.

[26] J. Wang, Z. Mo, H. Zhang, and Q. Miao, "A deep learning method for bearing fault diagnosis based on time-frequency image," *IEEE Access*, vol. 7, pp. 42373–42383, 2019.

[27] B. Boashash, *Time-Frequency Signal Analysis and Processing: A Comprehensive Reference*. New York, NY, USA: Academic, 2015.

[28] R. Yan and R. X. Gao, "Multi-scale enveloping spectrogram for vibration analysis in bearing defect diagnosis," *Tribol. Int.*, vol. 42, no. 2, pp. 293–302, Feb. 2009.
[29] Y. Lu, R. Xie, and S. Y. Liang, "CEEMD-assisted kernel support vector machines for bearing diagnosis," *Int. J. Adv. Manuf. Technol.*, vol. 106, nos. 7–8, pp. 3063–3070, Feb. 2020.
[30] 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.
[31] S. Ma, W. Liu, W. Cai, Z. Shang, and G. Liu, "Lightweight deep residual CNN for fault diagnosis of rotating machinery based on depthwise separable convolutions," *IEEE Access*, vol. 7, pp. 57023–57036, 2019.
[32] M. Xia, T. Li, L. Xu, L. Liu, and C. W. de Silva, "Fault diagnosis for rotating machinery using multiple sensors and convolutional neural networks," *IEEE/ASME Trans. Mechatronics*, vol. 23, no. 1, pp. 101–110, Feb. 2018.
[33] J. Zhang, Y. Sun, L. Guo, H. Gao, X. Hong, and H. Song, "A new bearing fault diagnosis method based on modified convolutional neural networks," *Chin. J. Aeronaut.*, vol. 33, no. 2, pp. 439–447, Feb. 2020.
[34] L. Yuan, D. Lian, X. Kang, Y. Chen, and K. Zhai, "Rolling bearing fault diagnosis based on convolutional neural network and support vector machine," *IEEE Access*, vol. 8, pp. 137395–137406, 2020.
[35] L. Wen, X. Li, L. Gao, and Y. Zhang, "A new convolutional neural network-based data-driven fault diagnosis method," *IEEE Trans. Ind. Electron.*, vol. 65, no. 7, pp. 5990–5998, Jul. 2018.
[36] V. Singh and N. K. Verma, "Intelligent condition based monitoring techniques for bearing fault diagnosis," *IEEE Sensors J.*, early access, Sep. 4, 2020, doi: 10.1109/JSEN.2020.3021918.
[37] *Case Western Reserve University Bearing Data Center*. Accessed: Sep. 2020. [Online]. Available: https://csegroups.case.edu/ bearingdatacenter/home
[38] J. Lee, H. Qiu, G. Yu, J. Lin, and Rexnord Technical Services, "Bearing data set," IMS, Univ. Cincinnati, NASA Ames Prognostics Data Repository, NASA Ames Res. Center, Moffett Field, CA, USA, 2007. [Online]. Available: https://ti.arc.nasa.gov/tech/dash/groups/pcoe/prognostic-data-repository/
[39] S. Lu, G. Qian, Q. He, F. Liu, Y. Liu, and Q. Wang, "*In situ* motor fault diagnosis using enhanced convolutional neural network in an embedded system," *IEEE Sensors J.*, vol. 20, no. 15, pp. 8287–8296, Aug. 2020.
[40] J. Lee, J. Ni, D. Djurdjanovic, H. Qiu, and H. Liao, "Intelligent prognostics tools and e-maintenance," *Comput. Ind.*, vol. 57, no. 6, pp. 476–489, Aug. 2006.
[41] H. Qiu, J. Lee, J. Lin, and G. Yu, "Wavelet filter-based weak signature detection method and its application on rolling element bearing prognostics," *J. Sound Vibrat.*, vol. 289, nos. 4–5, pp. 1066–1090, Feb. 2006.
[42] 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.
[43] L. Rosafalco, A. Manzoni, S. Mariani, and A. Corigliano, "Fully convolutional networks for structural health monitoring through multivariate time series classification," *Adv. Model. Simul. Eng. Sci.*, vol. 7, no. 1, pp. 1–31, Dec. 2020.
[44] W. Zhang, F. Zhang, W. Chen, Y. Jiang, and D. Song, "Fault state recognition of rolling bearing based fully convolutional network," *Comput. Sci. Eng.*, vol. 21, no. 5, pp. 55–63, Sep. 2019.
[45] C. Cheng, B. Zhou, G. Ma, D. Wu, and Y. Yuan, "Wasserstein distance based deep adversarial transfer learning for intelligent fault diagnosis," 2019, *arXiv:1903.06753*. [Online]. Available: http://arxiv.org/abs/1903.06753
[46] F. Shen, C. Chen, R. Yan, and R. X. Gao, "Bearing fault diagnosis based on SVD feature extraction and transfer learning classification," in *Proc. Prognostics Syst. Health Manage. Conf. (PHM)*, Oct. 2015, pp. 1–6.
[47] W. Lu, B. Liang, Y. Cheng, D. Meng, J. Yang, and T. Zhang, "Deep model based domain adaptation for fault diagnosis," *IEEE Trans. Ind. Electron.*, vol. 64, no. 3, pp. 2296–2305, Mar. 2017.
[48] S. Shao, S. McAleer, R. Yan, and P. Baldi, "Highly accurate machine fault diagnosis using deep transfer learning," *IEEE Trans. Ind. Informat.*, vol. 15, no. 4, pp. 2446–2455, Apr. 2019.

INDRANI MUKHERJEE received the B.Tech. degree in electronics and communication engineering from the Institute of Engineering and Management, Kolkata, India, in 2016. She is currently pursuing the Master of Technology (M.Tech.) degree with the Department Electrical Engineering, IIT Bombay, with a focus on electronic systems. She is currently working as a Research Assistant with the Applied Integrated Micro Systems (AIMS) Laboratory, IIT Bombay. Prior to joining IIT Bombay, she worked as an Assistant System Engineer with Tata Consultancy Services, Kolkata, from 2016 to 2018. Her research interests include digital image processing, signal acquisition and compression, machine prognostics, and embedded systems.

SIDDHARTH TALLUR received the B.Tech. degree in electrical engineering from IIT Bombay, Mumbai, India, in 2008, and the M.S. and Ph.D. degrees in electrical and computer engineering from Cornell University, Ithaca, NY, USA, in 2011 and 2013, respectively. From 2013 to 2016, he worked with Analog Devices Inc., Wilmington, MA, USA, in roles spanning MEMS products and applications engineering. In November 2016, he joined the Department of Electrical Engineering, IIT Bombay, where he is currently working as an Associate Professor. He is also the Faculty-in-Charge with the Wadhwani Electronics Laboratory (WEL), IIT Bombay. He holds a Visiting Appointment with the International College of Semiconductor Technology (ICST), National Yang Ming Chiao Tung University (NYCU), Hsinchu, Taiwan. His current research interests include MEMS and photonics, high resolution and low cost physical and biosensors, and high speed instrumentation and embedded systems for sensing applications.