

A Stacked Autoencoder Neural Network based Automated Feature Extraction Method for Anomaly detection in On-line Condition Monitoring

Mohendra Roy

*School of Electrical and Electronic Engineering
Nanyang Technological University
Singapore 639798
mohendra.roy@ntu.edu.sg*

Sumon Kumar Bose

*School of Electrical and Electronic Engineering
Nanyang Technological University
Singapore 639798
Bose0003@e.ntu.edu.sg*

Bapi Kar

*School of Electrical and Electronic Engineering
Nanyang Technological University
Singapore 639798
bapik@ntu.edu.sg*

Pradeep Kumar Gopalakrishnan

*School of Electrical and Electronic Engineering
Nanyang Technological University
Singapore 639798
pradeepgk@ntu.edu.sg*

Arindam Basu*

*School of Electrical and Electronic Engineering
Nanyang Technological University
Singapore 639798
arindam.basu@ntu.edu.sg*

Abstract—Condition monitoring is one of the routine tasks in all major process industries. The mechanical parts such as a motor, gear, bearing are the major components of a process industry and any fault in them may cause a total shutdown of the whole process, which may result in serious losses. Therefore it is very crucial to predict any approaching defects before its occurrence. Several methods exist for this purpose and many research are being carried out for better and efficient models. However, most of them are based on the processing of raw sensor signals, which is tedious and expensive. Recently, there has been an increase in the feature based condition monitoring, where only the useful features are extracted from the raw signals and interpreted for the prediction of the fault. Most of these are handcrafted features, where these are manually obtained based on the nature of the raw data. This of course requires the prior knowledge of the nature of data and related processes. This limits the feature extraction process. However, recent development in the autoencoder based feature extraction method provides an alternative to the traditional handcrafted approaches; however, they have mostly been confined in the area of image and audio processing. In this work, we have developed an automated feature extraction method for on-line condition monitoring based on the stack of the traditional autoencoder and an on-line sequential extreme learning machine (OSELM) network. The performance of this method is comparable to that of the traditional feature extraction approaches. The method can achieve 100% detection accuracy for determining the bearing health states of NASA

bearing dataset. The simple design of this method is promising for the easy hardware implementation of Internet of Things (IoT) based prognostics solutions.

Index Terms—Condition Monitoring, Feature Extraction, Autoencoder, OSELM, Handcrafted Feature, Neural Network, predictive maintenance.

I. INTRODUCTION

Condition Monitoring (CM) of machine health and maintenance are parts of the routine job in all major process industries. This is also an integral part of the evaluation of risk and asset management [1]. Specially, the on-line monitoring and diagnosis of machine health help in assessing the impact of aging parts on the reliability of the overall system. Further, this helps in several important aspects such as improving system safety, reducing processing and operation time, increasing system availability, maintenance planning, alerting the crew about the impending failure to take proactive measures and so on [2]. Recently, there has been an increase in on-line machine health monitoring research aiming for predictive maintenance and to extend equipments service life, as well as to develop on-board Integrated Systems Health Management (ISHM). This has been fuelled by the recent development in the machine learning research [3]–[7]. Many of them are possible due to the increase in the computation capability, efficient models as well as the availability of dataset. To further accelerate this type of research, the National Aeronautics and Space Administration

This work was conducted within the Delta-NTU Corporate Lab for Cyber-Physical Systems with funding support from Delta Electronics Inc. and the National Research Foundation (NRF) Singapore under the Corp Lab@University Scheme.

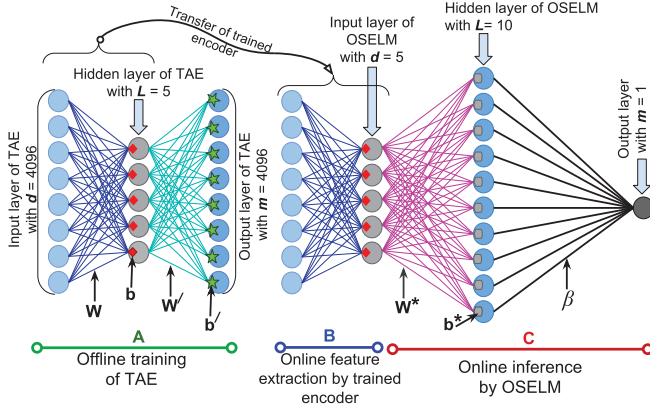


Fig. 1. Graphical illustration of the proposed stack arrangement of neural networks for the on-line feature extraction and condition monitoring. This stack arrangement consists of two types of neural networks. The first type is a traditional autoencoder (TAE) having the same number of nodes in input and output layers, i.e., of 4096 neurons and a single hidden layer with five neurons (see network A). The second type of network (network C) is an on-line sequential extreme learning machine (OSELM), having an input layers with 5 neurons, hidden layers with 10 neurons, and a single output node. In our proposed method, the network A was first trained in off-line mode using the training dataset with smoothed input data to learn to get the compressed representation of the data. This trained network (network B) was then used (only the encoder part of network A) for the on-line extraction of the features from the test dataset with raw features. This extracted features from the trained encoder (i.e. output of network B) were then fed into the classifier network C (input layer of OSELM), and used for on-line learning. After the convergence of network parameters in C, this trained model is then used for inference of the remaining samples.

(NASA) in conjunction with various industry and academia has open sourced several datasets related to machine health [8].

Many research groups have successfully developed various prognostics models based on these datasets [9]. Most of these datasets are generated by acquiring the raw sensor signals. This increases the dimension of the dataset and may also contain unwanted interference and noise. The dimensionality of the data has a direct impact on the accuracy as well as training time of the artificial neural network based models, and it turns out to be more crucial for the power critical applications like edge computing and Internet of things (IoT) based diagnosis solutions [10]. A popular approach to solving this kind of problem is to reduce the dimension of the raw input signals, which streamlines the training and utilization times, data visualization, data understanding, reduce the memory requirement, and energy required for computation etc. Several methods are there for achieving this [11]. Most commonly used time domain methods are, (a) root mean square (RMS) [12], (b) Kurtosis [13], [14], (c) Skewness [15], [16], (d) Crest factor [15], [16], (e) Peak-to-Peak [17], [18], and so on. All these are handcrafted approaches and require domain knowledge combined with “trial and error” strategies, which are tedious and time consuming.

Recent progress in autoencoder based feature extraction method has provided an alternative means for feature extraction [19], [20]. This has not only helped in solving the curse

of dimensionality [21], but also capable of providing more discriminating features as compared to the traditional feature engineering approaches [22]. Moreover, the best features for that task are directly “learned” from the data avoiding adhoc trial and error strategies. In some cases, it outperforms the traditional handcrafted features [23]. A considerable amount of progress has been made in the area of automated image and speech feature extraction [24]. As per our knowledge, no such effort has been made towards machine condition monitoring.

In this paper, we demonstrate a stack of the traditional autoencoder (TAE) and an On-line Sequential Extreme Learning Machine (OSELM) for automated feature extraction and condition monitoring of bearing health. The objective of this work is to develop an automated feature extraction method followed by an on-line condition monitoring system, which is comparable to the traditional feature extraction approaches. The detail of this method and results are described in the following sections.

II. METHODS

A. Stack of artificial neural networks

For the automated feature extraction, we have used the traditional autoencoder [25]–[27], which was trained to find the compress representation of the input data. The hidden layer output from this trained autoencoder was then used as an input to the boundary type OSELM [28] (see Fig. 1) for the inference of the bearing health.

The traditional autoencoder is an artificial neural network that can be trained to produce a replica of its input to its output [29], each having d nodes. The reduced code layer (also known as hidden layer) h , having L nodes, captures the compressed representation of the input. This compressed representation can be used as extracted features. An autoencoder consists of two parts; the encoder part takes the input $X \in \mathbb{R}^d$ and maps it to the hidden layer $h \in \mathbb{R}^L$, where

$$h = \sigma(WX + b) \quad (1)$$

Here σ , W and b denote the activation function of the hidden (latent) layer h , input weight matrix, and bias vector respectively. The decoder part maps the latent representation $h \in \mathbb{R}^L$ to the output $\hat{X} \in \mathbb{R}^m$ (for autoencoder $m = d$). Here $\hat{X} = \gamma(W'h + b')$, where \hat{X} is the reconstruction of the input and γ , W' , b' are the activation function, output weight matrix, and bias vector respectively.

In this work, we have used a single hidden layer autoencoder with $d = 4096$, $L = 5$ and $m = 4096$. We have used the Rectified Linear Units (ReLU) as an activation function in both encoder and decoder parts [30], [31] and the weights were optimized using the Adam optimizer [32].

The network was first trained in off-line mode using the training dataset for one epoch, keeping out the test dataset for the on-line feature extraction. The trained system (only the encoder part) was then used for the on-line feature extraction for the raw samples (see Fig. 1). Here the samples from the test dataset were fed into the input layer of the autoencoder (see network B in Fig. 1) in sequence to compute the hidden

layer output (the hidden layer output of network B is the same to the input of network C in Fig. 1). In each instance, these hidden layer outputs were then fed to the input layer of the OSELM (see the input of network C in Fig. 1) for the on-line training of OSELM and subsequently for the inference, i.e., condition monitoring.

The OSELM engine employs an on-line version of Extreme Learning Machine (ELM), which is known for its fast and straightforward learning method [28], [33]. Typically an ELM based network consists of a single hidden layer feed forward network, where the input weights (W^*) and biases (b^*) are generated randomly from a continuous probability distribution and are kept intact without updating. The hidden layer outputs are computed as in equation 1. Only the output weights of the network are learned, in a single step resulting in faster convergence.

In batch mode, the output weight matrix β is determined as:

$$\beta = H^T \left(\frac{I}{C} + HH^T \right)^{-1} Y \quad (2)$$

where H , I , Y and C are the hidden layer output, identity matrix, target output and a hyper-parameter respectively. For the on-line version of ELM, β is updated sequentially as follows:

$$\beta_n = \beta_{n-1} + P_n^{-1} (Y_n - H_n \beta_{n-1}) H_n^T \quad (3)$$

where $P_n = (P_{n-1} + H_n H_n^T)$ and for the first batch of samples, i.e., for $n = 1$, $P_{n-1} = P_0 = \left(\frac{I}{C} + H_0 H_0^T \right)$ [34].

Here we have used the OSELM as one class classifier for anomaly detection, in boundary mode [35], [36]. This means that the network was trained for only one particular class, with only healthy samples from the machine assuming health condition has not degraded at the beginning of the lifetime. The output of the hidden layer was mapped to only one output node (for example Y is mapped to a value of 1 for healthy samples). As a result, it required less number of multiplication and accumulation (MAC) operation compared to that of the reconstruction mode where the number of output neurons (m) is the same as input dimension d . The input layer of the OSELM engine has the same dimension as that of the hidden layer of the traditional autoencoder (see Fig. 1), i.e., $d = 5$. The hidden layer of OSELM engine is having a dimension of $L = 10$ and a single output node, i.e., $m = 1$ (see network C in Fig. 1). We have trained this network in an on-line fashion with features that were extracted by the traditional autoencoder (i.e., the trained encoder part of the autoencoder, see Fig. 1). The initial output weight of OSELM engine (β_0) was calculated using the first 10 samples in a batch. Then the β was updated in on-line manner for each sample until it converged. Thus the network was trained by the on-line extracted features. Since the bearings are considered to be in healthy state at the earlier stage of their life, therefore the network was trained only for the healthy class. The converged network was then used for the on-line inference for rest of the samples. In this mode, if the samples were from the unhealthy state of the bearing, then

the output of the network deviates from its trained class value of $Y = 1$. Any deviation from this value that surpasses a well defined threshold is considered as anomaly or a faulty state of the machine being monitored.

Since the algorithm converges very fast [37], therefore we can train it using a few samples (average convergence length is of 582 samples). The convergence criteria is based on how small the change of β has become during the update (or training), i.e., $\Delta\beta$ should be very small for a fair number of consecutive samples. In this work, we adopted $\% \Delta\beta < Tc$ for a consecutive ten samples as a convergence criteria. Here $Tc = 0.1\%$ is used as the termination value. The average convergence length for the dataset with auto extracted features is 582, which is about 5 times less to that of the average sample length of the dataset (the average sample length of the NASA dataset is 3154). For the same dataset, the average convergence length is 481 with handcrafted features. This shows that for the both type of feature extraction, the network get converged within the first $(1/5)^{th}$ of the total samples of the dataset. Thus OSELM is suitable for the on-line training and subsequently for inference using the same dataset. From application point of view, a different model has to be learned for each machine since the placement of sensors and type of fault of each machine may be unique. Hence, it is obvious if healthy models are learned for each machine separately in an online manner on the deployed hardware. The choice of ELM as the classifier is motivated by both its fast convergence as well as availability of power efficient hardware [38], [39] for deployment.

B. Dataset Preparation for on-line feature extraction

In this work, we used NASA Bearing dataset [8], provided by the Center for Intelligent Maintenance Systems (IMS), University of Cincinnati. There are three types of datasets, each of them consisting of vibration signals from four different bearings. The dataset 1 contains vibration signals for all the four bearings which were acquired from the accelerometers for both X and Y axis. For the dataset 2 and 3, the vibration signals for all the four bearings were obtained only from the accelerometers along the X-axis. These vibration signals were recorded for the duration of 1 second, at an interval of 10 minutes, with a sampling rate of $20KHz$. The description of each bearing dataset and their health status are summarized in table I.

The raw signals from the accelerometers carry significant amount of noise, which were reduced by taking the average of every consecutive five raw data samples. This averaging helps in eliminating the high-frequency noise as well as reduces the number of input features in every sample from 20480 to $20480/5 = 4096$. This help in reducing the number of computations and also yields more stable weight model during the training phase. We have used these filtered signals for rest of this work, i.e., for the training of the network as well as for the evaluation of bearing health. For this purpose, we have arranged the whole NASA bearing dataset into two parts, as training and testing dataset using leave one out strategy [40].

TABLE I
DETAILED DESCRIPTION OF NASA BEARING DATASET [8]

Dataset	Bearing	No. of Samples	No. of raw features in each sample	Condition
Dataset1	Bearing1	2156	20480	No defect
	Bearing2	2156	20480	No defect
	Bearing3	2156	20480	Defect in inner race
	Bearing4	2156	20480	Roller element defect
Dataset2	Bearing1	984	20480	Outer race failure
	Bearing2	984	20480	No defect
	Bearing3	984	20480	No defect
	Bearing4	984	20480	No defect
Dataset3	Bearing1	6324	20480	No defect
	Bearing2	6324	20480	No defect
	Bearing3	6324	20480	Outer race failure
	Bearing4	6324	20480	No defect

Since the NASA bearing dataset has total 12 bearings (see table I), therefore we have arranged 12 sets of datasets. Each set contains 11 bearing datasets for training and the left out one for the testing. For example, to evaluate the condition of the bearing1 of dataset1, we have treated the Dataset1_Bearing1 as test dataset (i.e. for on-line testing) and rest of the remaining 11 bearing datasets as training dataset (i.e. in off-line training mode).

C. Preparation of dataset with handcrafted features

For the comparison of our proposed method, we have performed the same on-line inference using five handcrafted features that extracted from the raw signals. We have used the following five conventional time-domain methods to create the handcrafted features: RMS, kurtosis, skewness, crest factor and peak-to-peak. Thus we have the dataset of 12 bearings with five handcrafted features, to be applied directly to the OSELM engine represented by network *C*, as highlighted in Fig. 1. The method of training and inference of the OSELM engine remains the same as described in the previous section.

III. RESULTS AND DISCUSSION

To check the reliability of the proposed method, we obtained the maximum amount of deviation of OSELM output to that of the healthy state of the same bearing, for the entire lifetime of the bearing (i.e. length of the dataset). Fig. 2 (a) shows the maximum deviations for each of the bearing, using automatically extracted features from the autoencoder. From the figure

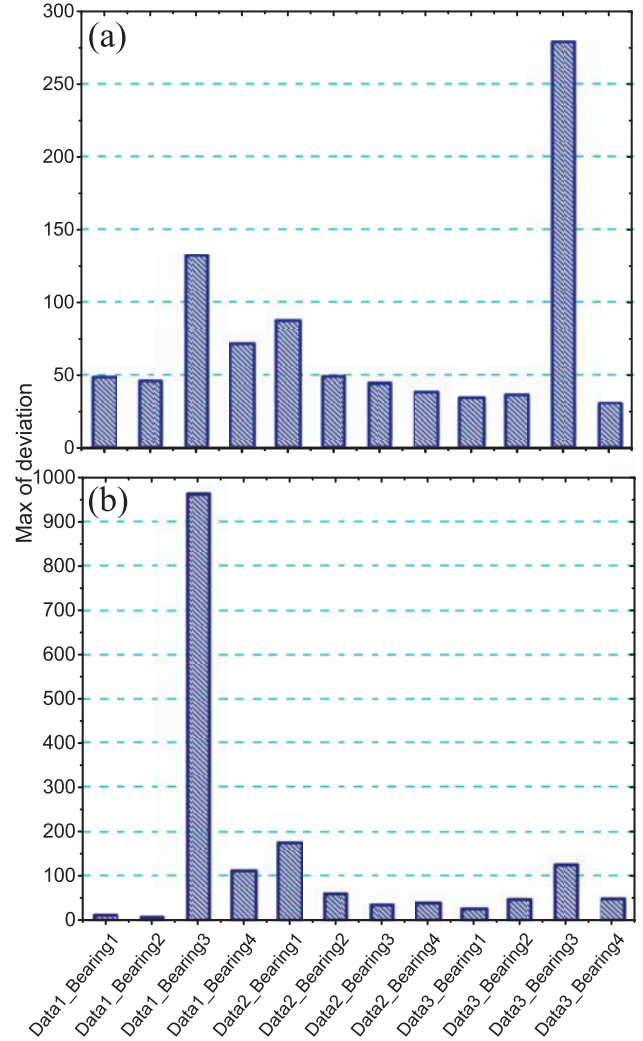


Fig. 2. Condition monitoring result of NASA bearing dataset using both the proposed feature extraction method and traditional handcrafted approach. Maximum deviation from expected value is shown for each test bearing data for (a) automatically extracted features and (b) handcrafted features. In both figures the faulty Data1_Bearing3, Data1_Bearing4, Data2_Bearing1, and Data3_Bearing3 are having more deviation compared to the rest of the good bearings, which correlate well with the dataset description as in table I.

we can see that the Dataset1_Bearing3, Dataset1_Bearing4, Dataset2_Bearing1, and Dataset3_Bearing3 are having comparatively large deviation than the rest of the bearings. This is because all these four datasets are from faulty bearings according to the ground truth (see the dataset description in table I). The results from the handcrafted features also follow a similar behavior (see Fig. 2(b)). These results also show that the proposed method is able to indicate the health status of the faulty bearings.

However, we want to develop an automated process that spontaneously indicate the condition of the bearing health. This requires a scheme to identify an appropriate threshold value. Therefore we have implemented an adaptive thresh-

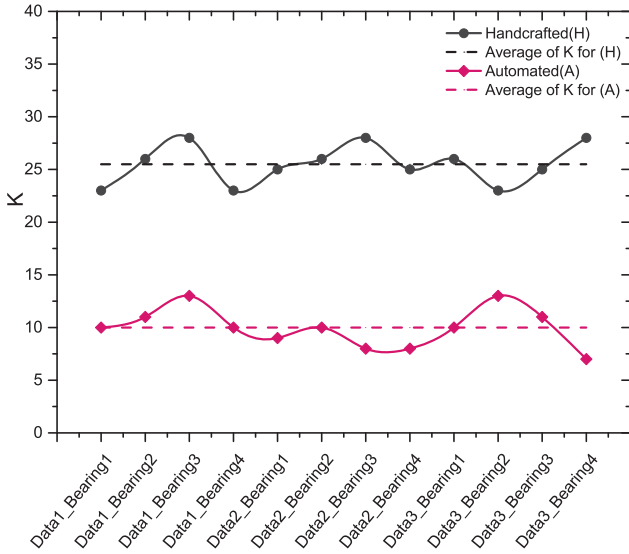


Fig. 3. Evaluation of K for determining the adaptive threshold for each of the test dataset. Here the plot in black is for the dataset with traditional handcrafted features and the plot in magenta is for the dataset with the automatically extracted features. In both cases the K value was evaluated using the leave one out strategy.

olding method that is based on the average and the standard deviation of the all the output deviations of OSELM engine at the time of on-line training. These deviations are a measure of the amount of noise in the data. The threshold T is given by:

$$T = K(\mu_t + \sigma_t) \quad (4)$$

Here μ_t is the average of all the output responses from the OSELM (in terms of output deviation) to the input training samples, when the bearing is healthy during its early life, and σ_t is the standard deviation of all these responses(i.e. the differences). We used K as an hyper-parameter which we aim to adjust. In the inference mode, for any bearing dataset if the output deviation of the OSELM was higher than the corresponding threshold, then it indicates an unhealthy state of the bearing.

Since the threshold also depends on the hyper-parameter K , we need to optimize the value of K for better accuracy in detecting the bearing health. For that purpose, we used a leave-one-out based cross-validation strategy. In other words, we have used the same training dataset that was used for the training of the autoencoder to find an optimal K value and used it for determining the bearing state of the test dataset. After finishing the training of the autoencoder, we used the same training dataset containing the 11 bearing dataset. For each bearing dataset, we extracted their compressed features (using the trained autoencoder) and used it for the on-line training of the OSELM. During the training of OSELM engine, we evaluated the parameters μ_t , σ_t and then found the value of K which determined the true state of the corresponding bearing dataset. We selected the K value that yields the

maximum accuracy in determining the bearing health state of all the 11 bearings in the training dataset.

After finding the optimum value of K , we then used it for the actual test dataset, i.e., which was left out as a test dataset and evaluated its performance by determining the true state of the bearing health as per as the ground truth (as described in the table I). In all cases, the value of K chosen from training data was able to successfully classify both the healthy as well as the faulty bearing during testing. Optimal K values, that helps in detecting the health status of each test dataset correctly, are shown in Fig. 3. Similar K values corresponding to the handcrafted features are also shown in black line in Fig. 3.

We further validate our findings by evaluating the percentage of accuracy in determining the bearing health state of all the 12 bearings against a range of K values in inference mode of OSELM engine. Here the percentage of accuracy implies the accuracy in detecting the health of all the bearings in the context of ground truth as described in table I. The experimental results in Fig. 4 presents the variation of K and the percentage of accuracy for the auto-extracted features. This plot shows that the accuracy reaches 100% for $K \approx 10$. This means that the predictions for all the bearings using $K \approx 10$ is analogous to the ground truth. Similar observation for handcrafted features shows that 100% accuracy can be achieved for $K \approx 25$. These values roughly match the average values of K obtained from the training dataset as described earlier (see Fig. 3).

Although the respective K values for the automated method and the handcrafted method are different, both the approaches can predict the health of all the bearings with 100% accuracy. From these results, it is evident that the automatically extracted features are as useful as handcrafted features in detecting the bearing health condition. Since the proposed automated feature extraction method does not require any prior knowledge of the dataset and it requires to train only once for a particular type of dataset, therefore the proposed method is more feasible for applying to different machines.

It is to be noted that the above results are valid for a particular seed value used for generating the random input weights and bias values in OSELM engine. Subsequently, we performed the same exercise for five epochs with different seed values, in order to validate the robustness of the proposed method. This shows an average accuracy of 98.4% with a standard deviation of 3.2%. Needless to say that the value of K for each seed was adjusted using the same optimization procedure as described above.

IV. CONCLUSION

In this paper, we demonstrate an automated feature extraction method that can extract a compressed representation of the raw signals, which can be utilized for the on-line condition monitoring of machine health. The performance of this method is similar to that of the handcrafted (manual) feature extraction approaches. Once the training of the autoencoder is performed in off-line mode, it is ready for automatic extraction of the

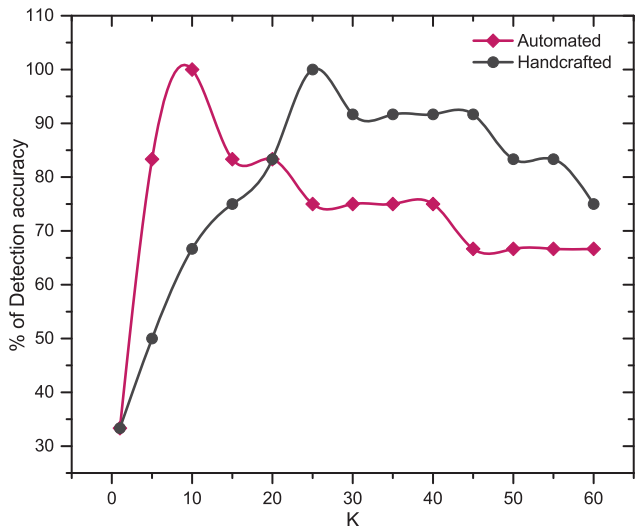


Fig. 4. K Vs. % of detection accuracy in determining the bearing health state of all the bearings of NASA bearing dataset as per as the ground truth as described in table I. The plot in black is for the dataset with handcrafted features and the plot in magenta is for the dataset with automatically extracted features. This result validate the result from Fig. 3, which shows that the for handcrafted features the optimum value of $K \approx 25$ and for automated features it is about 10. Again it shows that both the methods are capable in detecting bearing states with 100% accuracy.

features from raw data using the trained hyper-parameters. Given that, the proposed approach can be realized in hardware for in-situ feature extraction and subsequent classification of the data as desirable in low power Edge-computing platforms for IoT applications.

In future, we intend to study the performance of various classifiers using the same automatically extracted features and also an ELM based ensemble engine.

ACKNOWLEDGMENT

We would like to thank the reviewers for their valuable comments for the improvement of this paper.

REFERENCES

- [1] M. Leva, G. Baldissoni, R. Caso, M. Demichela, L. Lawlor, and B. McAleer, "Cost benefit evaluation of maintenance options for aging equipment using monetised risk values: a practical application," *Procedia Manufacturing*, vol. 19, pp. 119 – 126, 2018, proceedings of the 6th International Conference in Through-life Engineering Services, University of Bremen, 7th and 8th November 2017. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S2351978918300179>
- [2] H. M. Elattar, H. K. Elminir, and A. M. Riad, "Prognostics: a literature review," *Complex & Intelligent Systems*, vol. 2, no. 2, pp. 125–154, Jun 2016. [Online]. Available: <https://doi.org/10.1007/s40747-016-0019-3>
- [3] D. Martínez-Rego, O. Fontenla-Romero, B. Pérez-Sánchez, and A. Alonso-Betanzos, "Fault prognosis of mechanical components using on-line learning neural networks," in *Artificial Neural Networks – ICANN 2010*, K. Diamantaras, W. Duch, and L. S. Iliadis, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 60–66.
- [4] C. Zhang, G. S. Hong, H. Xu, K. C. Tan, J. H. Zhou, H. L. Chan, and H. Li, "A data-driven prognostics framework for tool remaining useful life estimation in tool condition monitoring," in *2017 22nd IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, Sept 2017, pp. 1–8.
- [5] T. Fan and W. Zhao, "Ensemble of model-based and data-driven prognostic approaches for reliability prediction," in *2017 Prognostics and System Health Management Conference (PHM-Harbin)*, July 2017, pp. 1–6.
- [6] N.-H. Kim, D. An, and J.-H. Choi, *Data-Driven Prognostics*. Cham: Springer International Publishing, 2017, pp. 179–241. [Online]. Available: https://doi.org/10.1007/978-3-319-44742-1_5
- [7] A. Nuhic, T. Terzimehic, T. Soczka-Guth, M. Buchholz, and K. Dietmayer, "Health diagnosis and remaining useful life prognostics of lithium-ion batteries using data-driven methods," *Journal of Power Sources*, vol. 239, pp. 680 – 688, 2013. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0378775312018605>
- [8] NASA Prognostics Data Repository, <https://ti.arc.nasa.gov/tech/dash/groups/pcoc/prognostic-data-repository/>.
- [9] P. Nectoux, R. Gouriveau, K. Medjaher, E. Ramasso, B. Chebel-Morello, N. Zerhouni, and C. Varner, "PRONOSTIA : An experimental platform for bearings accelerated degradation tests." in *IEEE International Conference on Prognostics and Health Management, PHM'12.*, vol. sur CD ROM. Denver, Colorado, United States: IEEE Catalog Number : CPF12PHM-CDR, Jun. 2012, pp. 1–8. [Online]. Available: <https://hal.archives-ouvertes.fr/hal-00719503>
- [10] X. Sun and N. Ansari, "Edgeiot: Mobile edge computing for the internet of things," *IEEE Communications Magazine*, vol. 54, no. 12, pp. 22–29, December 2016.
- [11] I. Guyon and A. Elisseeff, *Feature Extraction, Foundations and Applications*. [Online]. Available: http://link.springer.com/10.1007/978-3-540-35488-8_1
- [12] J. Igba, K. Alemzadeh, C. Durugbo, and E. T. Eiriksson, "Analysing rms and peak values of vibration signals for condition monitoring of wind turbine gearboxes," *Renewable Energy*, vol. 91, pp. 90 – 106, 2016. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0960148116300064>
- [13] J. Tian, C. Morillo, M. H. Azarian, and M. Pecht, "Motor bearing fault detection using spectral kurtosis-based feature extraction coupled with k-nearest neighbor distance analysis," *IEEE Transactions on Industrial Electronics*, vol. 63, no. 3, pp. 1793–1803, March 2016.
- [14] X. Niu, L. Zhu, and H. Ding, "New statistical moments for the detection of defects in rolling element bearings," *The International Journal of Advanced Manufacturing Technology*, vol. 26, no. 11, pp. 1268–1274, Nov 2005. [Online]. Available: <https://doi.org/10.1007/s00170-004-2109-4>
- [15] K. Medjaher, F. Camci, and N. Zerhouni, "Feature extraction and evaluation for Health Assessment and Failure prognostics." *Proceedings of First European Conference of the Prognostics and Health Management Society, PHM-E'12.*, pp. 1–6, 2012. [Online]. Available: <http://hal.archives-ouvertes.fr/hal-00718330/>
- [16] W. Caesarendra and T. Tjahjowidodo, "A Review of Feature Extraction Methods in Vibration-Based Condition Monitoring and Its Application for Degradation Trend Estimation of Low-Speed Slew Bearing," *Machines*, vol. 5, no. 4, p. 21, 2017. [Online]. Available: <http://www.mdpi.com/2075-1702/5/4/21>
- [17] J. Liu and E. Zio, "Prediction of peak values in time series data for prognostics of critical components in nuclear power plants," *IFAC-PapersOnLine*, vol. 49, no. 28, pp. 174 – 178, 2016, 3rd IFAC Workshop on Advanced Maintenance Engineering, Services and Technology AMEST 2016. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S2405896316324545>
- [18] V. Nistane and S. Harsha, "Failure evaluation of ball bearing for prognostics," *Procedia Technology*, vol. 23, pp. 179 – 186, 2016, 3rd International Conference on Innovations in Automation and Mechatronics Engineering 2016, ICIAME 2016 05-06 February, 2016. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S2212017316300160>
- [19] J. Masci, U. Meier, D. Cireşan, and J. Schmidhuber, "Stacked convolutional auto-encoders for hierarchical feature extraction," in *Artificial Neural Networks and Machine Learning – ICANN 2011*, T. Honkela, W. Duch, M. Girolami, and S. Kaski, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 52–59.
- [20] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006. [Online]. Available: <http://science.sciencemag.org/content/313/5786/504>
- [21] P. Indyk and R. Motwani, "Approximate nearest neighbors: Towards removing the curse of dimensionality," in *Proceedings of the Thirtieth Annual ACM Symposium on Theory of Computing*, ser. STOC '98. New York, NY, USA: ACM, 1998, pp. 604–613. [Online]. Available: <http://doi.acm.org/10.1145/276698.276876>

- [22] M. Yousefi-Azar, V. Varadharajan, L. Hamey, and U. Tupakula, "Autoencoder-based feature learning for cyber security applications," in *2017 International Joint Conference on Neural Networks (IJCNN)*, May 2017, pp. 3854–3861.
- [23] G. Antipov, S.-A. Berrani, N. Ruchaud, and J.-L. Dugelay, "Learned vs. hand-crafted features for pedestrian gender recognition," in *Proceedings of the 23rd ACM International Conference on Multimedia*, ser. MM '15. New York, NY, USA: ACM, 2015, pp. 1263–1266. [Online]. Available: <http://doi.acm.org.ezlibproxy1.ntu.edu.sg/10.1145/2733373.2806332>
- [24] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 8, pp. 1798–1828, Aug 2013.
- [25] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Parallel distributed processing: Explorations in the microstructure of cognition, vol. 1," D. E. Rumelhart, J. L. McClelland, and C. PDP Research Group, Eds. Cambridge, MA, USA: MIT Press, 1986, ch. Learning Internal Representations by Error Propagation, pp. 318–362. [Online]. Available: <http://dl.acm.org/citation.cfm?id=104279.104293>
- [26] J. Xu, L. Xiang, Q. Liu, H. Gilmore, J. Wu, J. Tang, and A. Madabhushi, "Stacked sparse autoencoder (ssae) for nuclei detection on breast cancer histopathology images," *IEEE Transactions on Medical Imaging*, vol. 35, no. 1, pp. 119–130, Jan 2016.
- [27] J. Deng, Z. Zhang, E. Marchi, and B. Schuller, "Sparse autoencoder-based feature transfer learning for speech emotion recognition," in *2013 Humaine Association Conference on Affective Computing and Intelligent Interaction*, Sept 2013, pp. 511–516.
- [28] N. y. Liang, G. b. Huang, P. Saratchandran, and N. Sundararajan, "A fast and accurate online sequential learning algorithm for feedforward networks," *IEEE Transactions on Neural Networks*, vol. 17, no. 6, pp. 1411–1423, Nov 2006.
- [29] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016, <http://www.deeplearningbook.org>.
- [30] A. F. Agarap, "Deep learning using rectified linear units (relu)," *CoRR*, vol. abs/1803.08375, 2018. [Online]. Available: <http://arxiv.org/abs/1803.08375>
- [31] K. Eckle and J. Schmidt-Hieber, "A comparison of deep networks with ReLU activation function and linear spline-type methods," 2018. [Online]. Available: <http://arxiv.org/abs/1804.02253>
- [32] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *CoRR*, vol. abs/1412.6980, 2014. [Online]. Available: <http://arxiv.org/abs/1412.6980>
- [33] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, "Extreme learning machine: Theory and applications," *Neurocomputing*, vol. 70, no. 1, pp. 489 – 501, 2006, neural Networks. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0925231206000385>
- [34] R. P. Singh, N. Dabas, V. Chaudhary, and Nagendra, "Online sequential extreme learning machine for watermarking," in *Proceedings of ELM-2014 Volume 2*, J. Cao, K. Mao, E. Cambria, Z. Man, and K.-A. Toh, Eds. Cham: Springer International Publishing, 2015, pp. 115–124.
- [35] C. Gautam and A. Tiwari, "On the construction of extreme learning machine for one class classifier," in *Proceedings of ELM-2015 Volume 1*, J. Cao, K. Mao, J. Wu, and A. Lendasse, Eds. Cham: Springer International Publishing, 2016, pp. 447–461.
- [36] C. Gautam, A. Tiwari, and Q. Leng, "On the construction of extreme learning machine for online and offline one-class classificationan expanded toolbox," *Neurocomputing*, vol. 261, pp. 126 – 143, 2017, advances in Extreme Learning Machines (ELM 2015). [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0925231217302096>
- [37] J. Tang, C. Deng, G. B. Huang, and J. Hou, "A fast learning algorithm for multi-layer extreme learning machine," in *2014 IEEE International Conference on Image Processing (ICIP)*, Oct 2014, pp. 175–178.
- [38] E. Yao and A. Basu, "Vlsi extreme learning machine: A design space exploration," *IEEE Trans. on VLSI*, vol. 25, pp. 60 – 74, 2017.
- [39] A. Patil, S. Shen, E. Yao, and A. Basu, "Hardware architecture for large parallel array of random feature extractors applied to image recognition," *Neurocomputing*, vol. 261, pp. 193–203, 2017.
- [40] H. Cheng, D. J. Garrick, and R. L. Fernando, "Efficient strategies for leave-one-out cross validation for genomic best linear unbiased prediction," *Journal of Animal Science and Biotechnology*, vol. 8, no. 1, p. 38, May 2017. [Online]. Available: <https://doi.org/10.1186/s40104-017-0164-6>