International Journal of Computing and Digital Systems
ISSN (2210-142X)
Int. J. Com. Dig. Sys. 11, No.1 (Jan-2022)

http://dx.doi.org/10.12785/ijcds/110112

Tool Wear Prediction in Milling: A Comparative Analysis Based on Machine Learning and Deep Learning Approaches

Pooja V. Kamat[1,2], Anish Nargund[1], Satish Kumar[3], Shruti Patil[3] and Rekha Sugandhi[2]

[1]*Symbiosis Institute of Technology, Symbiosis International (Deemed) University, Pune, MH 412115 INDIA*
[2]*MIT School of Engineering, MIT-ADT University, Pune, MH 412201 INDIA*
[3]*Symbiosis Centre for Applied Artificial Intelligence, Symbiosis International (Deemed University), Pune, MH 412115 INDIA*

Received 26 Feb 2021, Revised 9 Aug 2021, Accepted 20 Sep 2021, Published 9 Jan 2022

Abstract: The milling machine's cutting tool is a vital asset; its breakdown results in unplanned downtime, which reduces industrial efficiency.Tool-Wear Monitoring(TWM) is one of the primary goals of the manufacturing industry due to the manifold benefits it provides, such as optimizing production efficiency, improving performance, and increasing the life of the tool. Most of the work carried out in this domain involves statistical-based techniques, which require expert domain knowledge in formulating degradation models of the tools. Data-driven machine learning and deep learning models have recently been used to analyze tool wear data and make efficient predictions about its remaining useful life. This paper presents a comparative approach to tool wear monitoring using the clustering machine learning technique of K-Nearest Neighbour (k-NN) and deep learning technique of Convolutional Neural Network (CNN) and hybrid Autoencoder-LSTM (AE-LSTM) models. The CNN and AE-LSTM techniques out-perform k-NN by achieving a higher degree of separability of around 93% and 87%, respectively, as per the ROC-AUC values. The techniques provide improved outcomes in terms of precision, recall, and f1-score, indicating that the models are more accurate at detecting false positives.

Keywords: Tool-wear; Milling; Convolutional Neural Network; k-Nearest Neighbours, Autoencoders;LSTM

1. INTRODUCTION AND OVERVIEW

The basic function of a milling machine is to generate flat surfaces in any acclimatization and surfaces with varied configurations such as radial or contoured surfaces. Such functions are carried out by slowly inserting the workpiece into the rotating edged circular cutter at moderately high speed, as indicated in Figure 1.

Rapid progress in the manufacturing domain and especially in the milling process has seen an up-trend in using high-end tools that exhibit extensive processing range and large production efficiency. However, inadequate toughness of these tools leads to tools being brittle, ultimately resulting in tool breakage or tool wear [1]. Premature tool failures are often expensive to repair and eventually lead to damage to the workpiece and probable harm to the machine and the personnel operating it [2]. Figure 2 shows an example of tool wear during the milling process.The abrasive motion of apertures and debris from a built-up edge, among other things, causes flank wear. This wear mainly occurs at low speed. Abrasion and metal diffusion at the tool face trigger crater wear when machining ductile materials and mostly occurs at high speed. It occurs at the corner of the tool and occurs due to flank wear and crater wear.

There is a dire need for a research-based solution that detects early tool wear with a lesser incidence of missed failure alerts. Approximately 40percent of production costs can be saved in case of timely detection of tool failure [3]. Less scientific experiments have been conducted on tool wear in the milling process since it is one of the most complex processes owing to the near-surface interaction between the tool and the workpiece. Figure 3 depicts the various approaches used in tool wear monitoring- knowledge-based model approach, data-driven based model approach, and physics-based model approach. Data-driven approaches are preferred over the other two due to the higher prediction accuracy despite moderate domain expertise [4].

This paper shows a comparative study of the prediction of tool wear between using the machine learning algorithm using of the k-Nearest Neighbour(k-NN) and the extensive deep learning algorithms of Convolutional Neural Network (CNN) and Autoencoder-LSTM(AE-LSTM). The remainder of the paper is structured in the following manner. Section II describes some of the recent work carried out in this domain. Section III explains in detail the methodology and models used in this study. Section IV describes the findings and discussions of the implemented algorithms, followed by conclusion and potential future scope.

E-mail address: pooja.kamat@sitpune.edu.in, nargund.anish@sitpune.edu.in, satish.kumar@sitpune.edu.in, shruti.patil@sitpune.edu.in, and rekha.sugandhi@mituniversity.edu.in

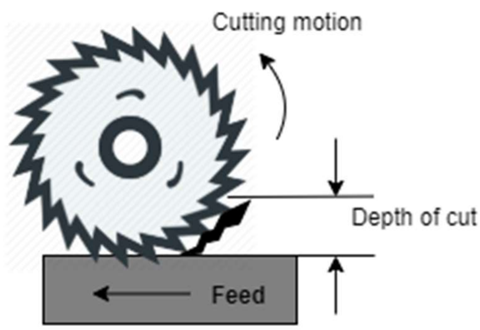


Figure 1. Schematic diagram of conventional milling operation

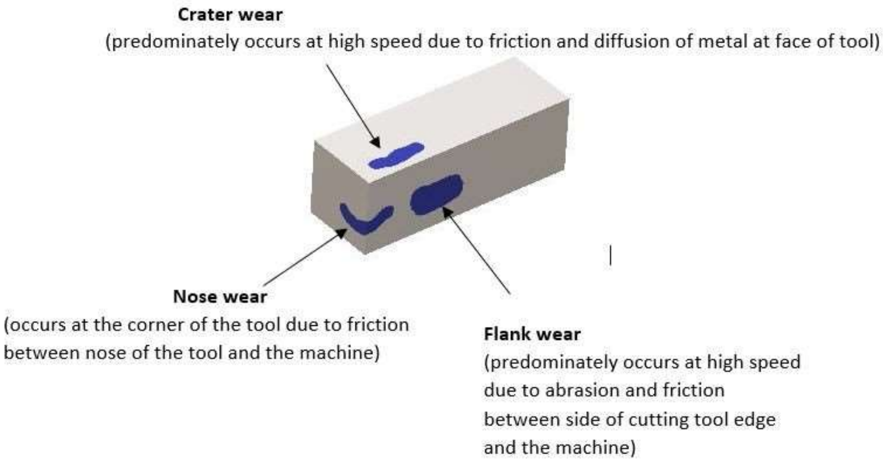


Figure 2. Tool wear and its types

2. RECENT WORK

Manufacturers have a common goal of improving production at a reduced cost. Replacing a dull tool at proper intervals ensured continued production at optimal maintenance of tool-cutting machining processes. Hence the demand for tool wear monitoring (TWM) has increased manifold, and effective tool wear predictive techniques have been hugely popular. Some of the objectives of TWM are as follows:

- Early detection of tool wear
- Regular monitoring of machining accuracy for enabling corrective actions
- Prevention of brittle tool breakage

Due to the benefitsit offers, Tool wear monitoring approaches have undergone revolutionary changes in the past few years, starting from physical model-driven monitoring to recent machine learning and deep learning-based data-driven monitoring. Progress in Tool Wear Monitoring (TWM) is depicted in Figure 4. Physical model-based tool wear monitoring representation of machine degradation in the form of mathematical models. However, this technique requires expert knowledge and a precise understanding of the degradation process, which is often challenging. Data-driven tool wear monitoring involves smart monitoring of the tool using sensors and effective decision-making using machine learning and deep learning techniques. Machine learning techniques are slightly tedious over deep learning models as extensive dimensionality reduction, and feature extraction processes are required.

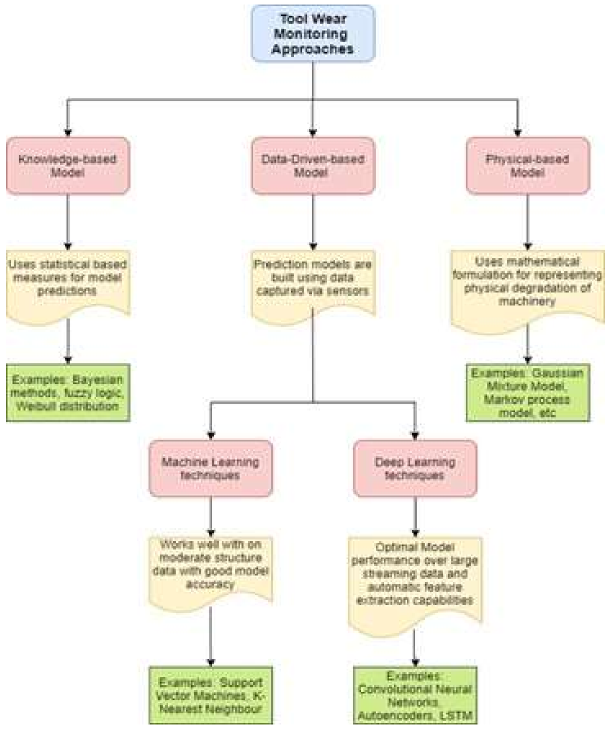


Figure 3. Tool wear monitoring approaches

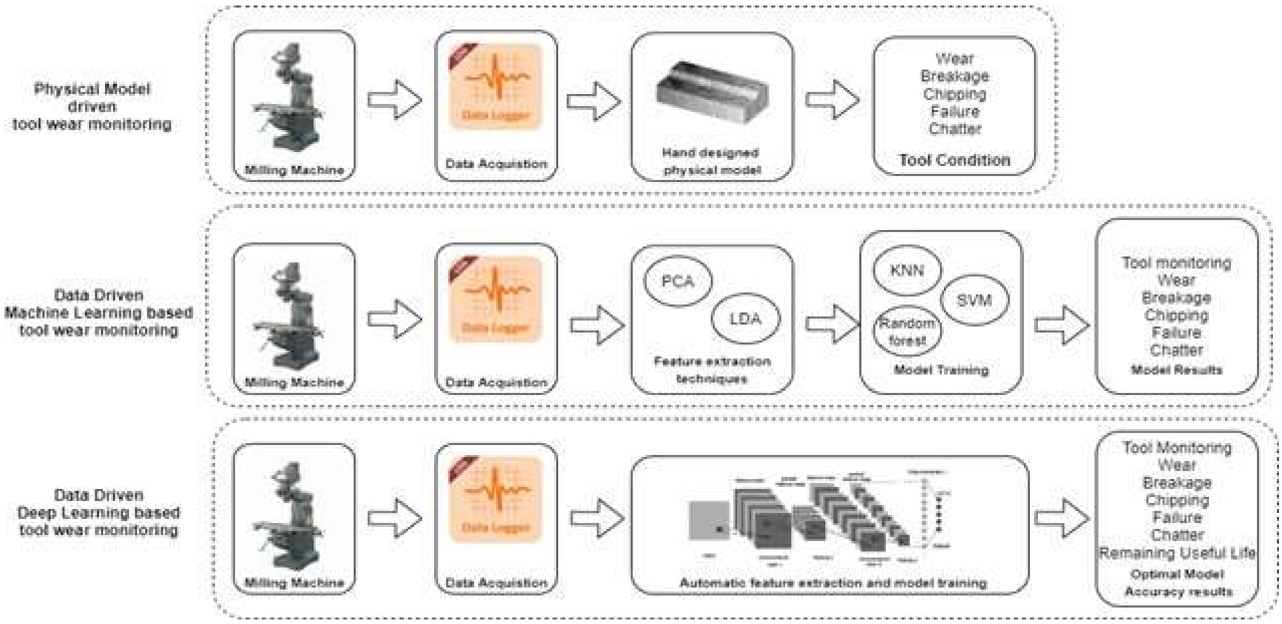


Figure 4. Progress in tool wear monitoring framework over recent years

Anomaly detection in the milling process has a lot of potential benefits. One of the most common causes of failures in milling is tool wear, wherein without accurate set thresholds, it can lead to usage of worn tools resulting in the poor surface quality of the end product and wastage of resources and time. In such a scenario, an effective anomaly detection technique can alert in advance about carrying out maintenance of the tool or even replacement of the worn tool. The most critical task here is identifying the tool's wear threshold from a permissible to a non-permissible limit [5]. Many researchers have approached this problem in the past using various sensing modalities such as vibration [6], force [7], data fusion [8], acoustics [9], and motor current [10]. Recent work involves using acoustic emission sensors in monitoring tool wear as the frequency range of acoustic sensors is higher than the vibration and background noise [11]. Tool wear monitoring can be categorized as sensor-based and sensor-less systems [12]. Usually, tool life was predicted using Taylor's standard tool life equation. Some manual techniques, such as using the tool maker's microscope, are often used to assess tool wear [13]. The sensor-based approach is divided into direct and indirect methods [14]. The data collected using the sensor-based approach need to be analyzed using machine learning or deep learning approaches due to the volume of the data; hence analysis using statistical techniques would be difficult. Machine learning algorithms have been popular for fault data pattern recognition in tool wear monitoring in the past. They include algorithms such as Support Vector Machine (SVM) [15], Artificial Neural Network (ANN) [16], Bayesian networks [17], hidden Markov models [18], decision tree networks [19], and k-Nearest Neighbor (k-NN) technique [20]. However, most of these data-driven techniques lack the ability to process unstructured data and hence cannot be easily generalized. Further, the large amount of multivariate data generated in manufacturing processes combined with the high correlation and high dimensional characteristics often require automatic feature extraction and representation capabilities [21,22]. Deep learning algorithms that take into account the temporal correlations of manufacturing data can achieve these capabilities.In Milling cutting, Speed, Feed, and Depth of cut are the major three input parameters. Many researchers had studied the effect of these parameters by measuring the cutting forces, tool wear, and surface roughness using traditional approaches. Few researchers have also worked on the indirect method and found several experimental limits in correlating the data collected with tool wear by using different computational algorithms [23]. Ren et al. [24] used a Fuzzy approach for cutting force data in Tool condition monitoring. But found the model was not effective in hesitating approximation errors. Artificial neural networks were used to predict the tool wear using the images of the tool [25]. Chungchoo and Saini [26] used a fuzzy neural network (FNN) algorithm to estimate the average flank and crater wear using the acoustic emission and cutting forces. All the data collected using sensors are majorly time series data in which anomaly detection is one of the major concerns. Detection of anomalies in time series data is not a new issue [27]. Kamat, P & Sugandhi, R point out the advantages of having an effective anomaly detection technique in place has many advantages such as reduction in unplanned downtimes of the machine, optimum utilization of resources for carrying out predictive maintenance, saving of financial expenditure on unplanned scheduling of maintenance activities, etc. [28]. Hu et al. proposed a fault detection model in industrial equipment using the hybrid Boltzmann machine algorithm with a multi-grained scanning forest technique [29]. Madhusudana et al. implemented the J48 decision tree algorithm for fault classification of healthy and faulty conditions of face milling tools and achieved an accuracy of around 81% [30].

Table 1 describes some of the majorly used AI algorithms in recent papers for tool wear monitoring with the input parameters provided to the algorithms, features captured, sensors used, output parameters monitored during the manufacturing processes. Most of the work highlight the use of sound,force and vibration sensor to monitor tool wear. Data captured was in the form of vibration, audio files and images of wear. Neural Networks such as Convolutional Nueral Networks and Autoencoders exhibited good performance.

3. METHODOLOGY

Figure 5 depicts the methodology used in this paper. In milling operation, tools have many cutting edges, and it's very difficult to measure the flank wear of the tool. The dataset used in this paper is based on the machining experiments which were carried on the wax material (Size: 2"x 2"x1.5") at the University of Michigan smart lab [35]. Certain pre-processing techniques were applied to the dataset. Further, the processed data was trained on three separate techniques Convolutional Neural Network(CNN), Autoencoder-Long Short Term Memory(AE-LSTM), and k-Nearest Neighbor (k-NN). The hyperparameters and thresholds were tuned to improve the accuracy of the models. In the end, a comparative analysis of all the three techniques wrt accuracy, precision, recall, and f1-score were made.

A. Data Acquisition & Preprocessing:

Figure 6 shows the data distribution histogram of the experiments.In total, 18 experiments were carried out, 8 with unworn tools and 10 with worn tools. The S-shape shown in Figure 7 was produced on the wax for every experiment. There are 47 features in the dataset. The major parameters are coordinates, speed, and acceleration in x,y,z coordinates, power, and current input and output parameters of each motor. The values of each experiment were stored in comma-separated values, CSV files, separately. The data pre-processing is mainly involved in data reduction, where all the data from the normal experiments is used, butthe faulty data available is about 30% (total worn data is 13311). The distribution for training data can be seen in Figure 8, where normal (÷75%) and faulty (÷25%) data used is in the ratio of 3:1, respectively. The data is thenappendedintoCSV files in the said ratio and then normalized

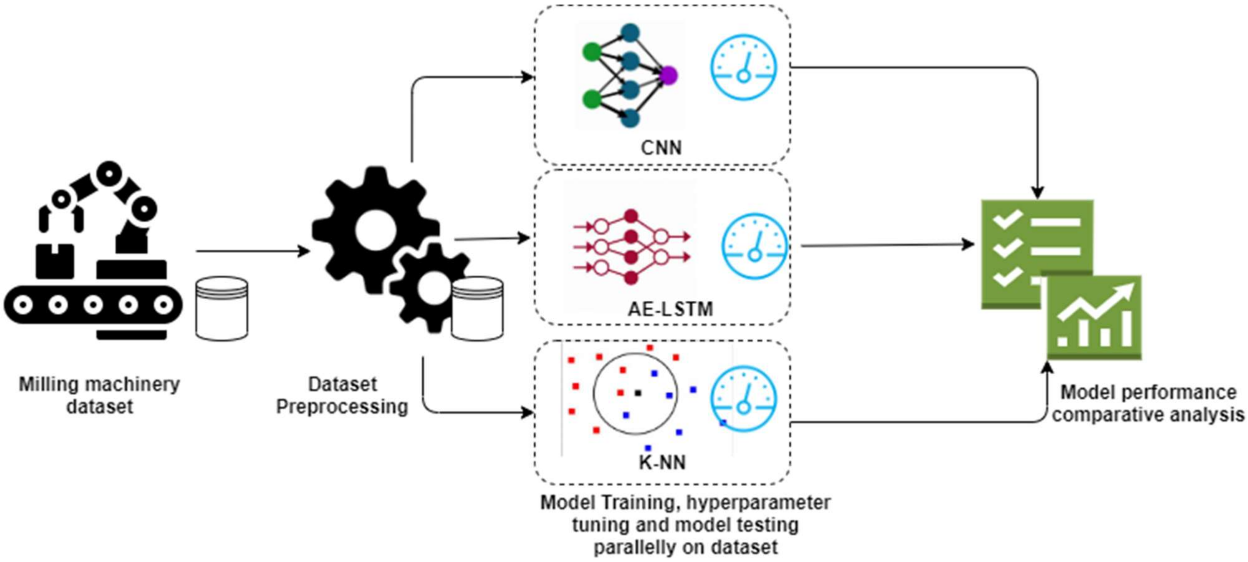


Figure 5. System Methodology

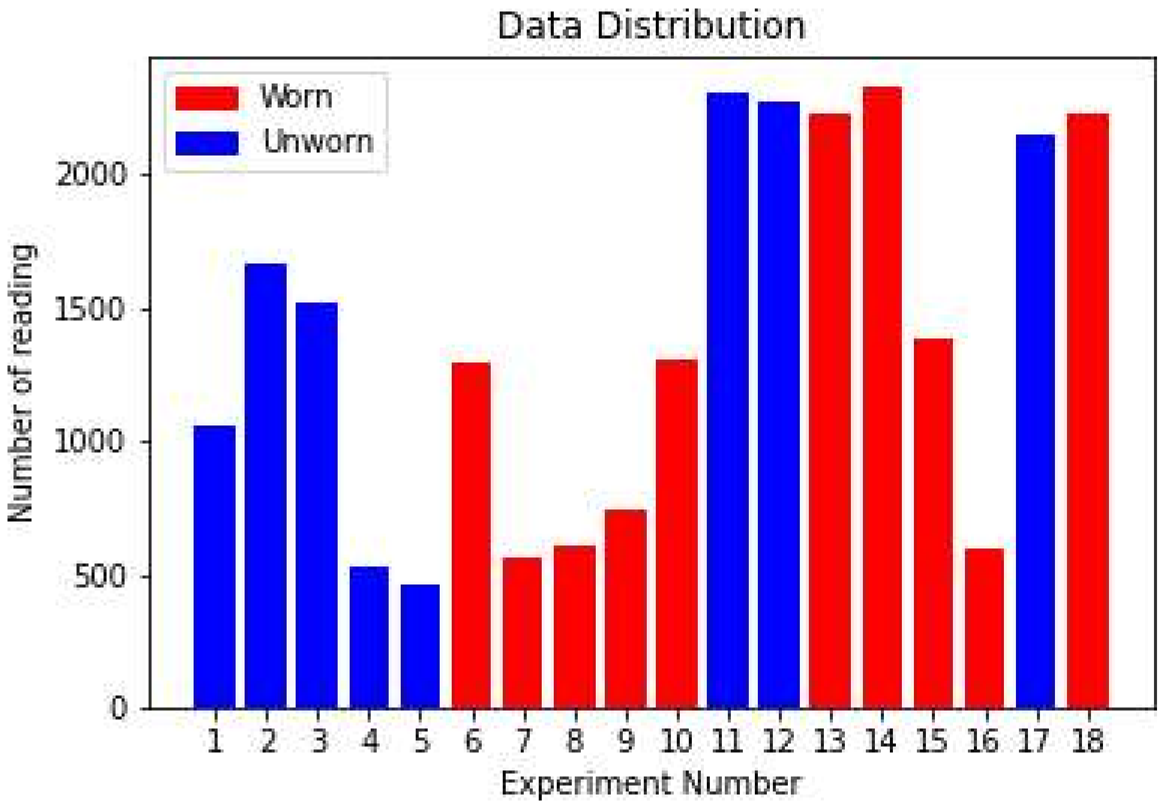


Figure 6. Data Distribution graph of wornunworn experiments

TABLE I. MAJOR AI TECHNIQUES USED IN TOOL WEAR MONITORING

Algorith	Input Parameters	Features captured	Sensors Used	Output Parameters	Application /Machinery/	Process Contribution/Limitation of the paper
ANN [31]	depth-of-cut,feed speed	Images (flank wear)	Force/torque sensors	Flank wear	Drilling-Aerospace assembly	Fractal and time-domain analysis improved the robustness of the ANN model by providing an overall RMSE(root mean squared error value) equal to 0.00032 and an average RMSE equal to 0.00113. A condition-based monitoring strategy instead of a time-based can improve the accuracy further.
CNN [32]	depth-of-cut, feed, speed	Audio(sound)	Microphone	Tool wear	Milling (vertical milling center)	CNN was able to decode and classify audio signals effectively with an accuracy as high as 99.5%. The accuracy of the model can be further probed on data collected in noisy environments.
Autoencoder[33	depth-of-cut, speed	Spindle current signals	LT 108-S7 closed-loop Hall current sensor	Tool wear	CNC milling machine	The stacked sparse autoencoder algorithm achieved 98.79% accuracy. Multi-parameter fusion of variable speed was not considered during this study
SVM [34]	depth of cut, feed, speed	Vibrations and sound	Piezotronics PCB accelerometer	minor flank face images	Micro milling	The technique achieved 97.54% accuracy with the help of the Recursive Feature Elimination (RFE) technique. However, the approach was a supervised one.

using a Scalar function which helps scale down the values to reduce computation with large values enabling us to reduce computation and training time. All three algorithms are flexible and robust; hence no further pre-processing techniques were required.

B. Model Construct:

1) Conventional Neural Network (CNN)

Convolutional Neural Networks came as a new and innovative method to draw higher and deeper features from the data. Figure 9 depicts the architecture of a one-dimensional CNN technique. CNN was developed for classifying 2D data where it accepts pixels of an image to learn the features. CNN works the same way irrespective of the dimension. The distinguishable factors are the structure of the data fed to the network and how the filters or kernels move across the data to extract and learn the characteristics of the data. So 1D CNN extracts features from the sequence data and maps the internal features of the sequence. They have proved to be effective for drawing out attributes from fixed-length sequences or time-series data regardless of the feature's location to be extracted from the entire data [36]. The filters slide vertically and horizontally over the grid to consider every pixel for convolution to draw out features. The resulting matrix is the summation of the product of each filter-tile pair which gives the output matrix. Here the 2D network uses small filter sizes to extract features. On the other hand, the 1D network allows large filter sizes like 7 or 9. But the difference is that in a 3x3 filter for 2D CNN, the number of feature vectors is 9, whereas in 1D CNN, a window size of 3 will contain only 3 feature vectors. Hence it is easy to afford large kernel sizes. For down-sampling, the convolved features to save processing time, like 2D CNN even 1D CNN, use pooling layers that help

Figure 7. S-Shaped Wax Artifact

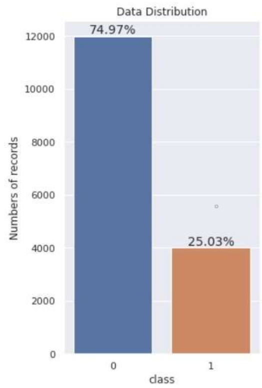


Figure 8. Training Data Distribution

reduce the dimensions of the feature map while retaining the most crucial feature details. Generally, max pooling is used for this purpose. When it comes to 1D CNN, they require simple array operations where 1D array replaces 2D matrix (for 2D networks) both kernels and feature maps, making the computational complexity low compared to 2D CNN. They have relatively less intricate architecture than 2D CNN, which allows them to derive valid and germane features from 1D data. Hence with a less condensed model architecture, the computational time is less, and there is no additional hardware required for training 1D networks. Considering two signals f and g with n denoting the index, the 1D spatial convolution the formula is given by equation (1):

$$(f * g)[n] = \sum_{m=-\infty}^{\infty} f[m] * g[n-m] \quad (1)$$

The fully connected layers process raw data and 'learn to extract' characteristics used for categorization. These layers have all nodes from the preceding layer connected to all the nodes in the next layer. They are responsible for defining the characteristics extracted through the convolutions. The final fully connected layer consists of an activation function, which, for each classification mark to be expected by the model, gives a probability value of 0 to 1. Both element extraction and categorization are melded into one procedure that can be streamlined to increase the classification rate. This gives asignificant edge to 1D CNNs, which can bring low computational intricacy. The main activity with a critical expense is an order of 1D convolutions, which are direct weighted aggregates of two 1D arrays [37]. In this study, 'Sequential (),' a Functional API, is used for building the 1D CNN model layer-by-layer. Since it is simple to describe models where layers are linked to more than just previous and next layers, the functional API allows for far more scalable models to be created.Linking layers to any other layer can be done with ease. It also helps to construct complex networks.

2) k-Nearest Neighbors algorithm (k-NN)

The k-nearest neighbor algorithm is a simple supervised machine learning algorithm. It is computationally cheap and takes very little time to train compared to CNNs. It is widely used for classification problems, but it can also be used in predictive regression problems. The 'k's in the k-NN algorithm represent the number of neighbors taken into consideration to take a vote. The value of k influences the performance of the algorithm and the accuracy as it helps make boundaries to differentiate between classes. Usually, the algorithm is run several times with varying values of k. Then the value of k, which has the least number of

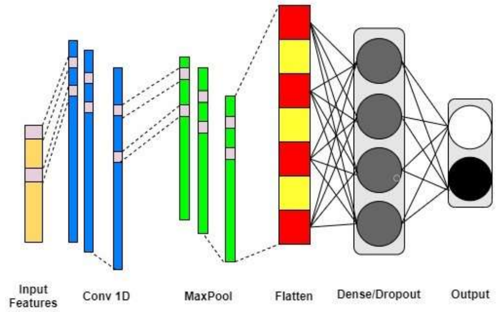


Figure 9. A General 1D CNN Architecture

wrong classifications, is chosen. The right K must be able to predict the data that it has not seen before. Typically, the prediction is unstable when the value of k is 1. As the value increases, the prediction becomes stable, but after a certain value, it again starts destabilizing. At this point, the value of k is pushed too far. For a tie-breaking vote for labels, the value of K chosen is an odd integer. The value of k considered in this study is 5. k-NN relies on labelled data and works on the assumption that similar things exist in close proximity. It keeps track of all the cases and categorizes new ones using the similarity metric. It is also called case-based reasoning, example-based reasoning; instance-based learning; memory-based learning; lazy learning. It functions to predict classification labels for fresh incoming unlabelled observations by memorizing observations within a classified data set. It makes predictions based on how similar training observations are to the test set. The more similar the values, the better are the classification accuracy. The distance calculated between two points is usually the Euclidean distance. It assigns the point to the class among its k nearest neighbors. Considering two points p and q, the formula for calculating the distance in n dimensions is given by equation (2):

$$d(p,q) = d(q,p) = \sqrt{[(q_1 - p_1)]^2 + [(q_2 - p_2)]^2 + (q_n - p_n)]^2}$$

$$= \sqrt{\sum_{i=1}^{n} [(q_i - p_i)]^2)} \;\ldots\ldots\ldots\ldots\ldots (2)$$

Figure 10 depicts the generic flowchart of the K-nearest neighbor algorithm. The algorithm starts with the initialization of the "k" number of clusters. Next, the distance between the test sample and training samples is computed, and the distances are sorted.A new k is initialized based on minimum distance, and outliers in data are observed. One of the major drawbacks of the k-NN algorithm is that each K neighbor is equally essential. Naturally, the closer the unknown point is to the neighbor, the greater the probability that it will belong to that category. [38]. k-NN needs the data to be noise-free or with significantly less noise. The data has to be labelled for it memorizes the feature to classify new observations. Also, with more number features, it becomes hard for the algorithm to classify correctly. Hence the most relevant features are to be preserved, which must have distinguishable subgroups. Also, k-NN is usually used to train small data sets as the computation time increases drastically with larger data sets. For this study the model is built by instantiating k nearest neighbour objects. This model is used for training and classification on the training and test set, respectively. The weight parameter of the model used was uniform, where all points in each neighborhood are equally weighted. The metric used was 'Minkowski,' which was equivalent to the standard Euclidean metric.

3) Autoencoder-LSTM Model

A typical Autoencoder-LSTM framework is depicted in Figure 11. An autoencoder is an unsupervised neural network that uses the encoding-decoding mechanism to learn from compressed representations of the input and reconstruct the input at the output layer side. The reconstruction of the input occurs at the midpoint. The model trains itself automatically on the features learned during the encoding process; hence it is also an automatic feature extraction model. LSTM-Autoencoder model is a hybrid variant of the basic recurrent neural network setup for prediction over sequential data. For a given sequential data, the encoder-decoder LSTM is modeled to read the input sequence, encode the sequence, decode it and reconstruct it. The reconstruction loss is calculated at every step, and loss above a certain threshold can predict outliers in the sequences. This makes the Autoencoder-LSTM model an effective technique for detecting anomalies such as the prediction of tool wear. Equation (3) depicts the mathematical formulae to calculate the anomaly score in autoencoders. For input X, the objective function is to find weight vectors for the encoder and decoder to minimize the reconstruction error.

$$\boldsymbol{\phi} : X \rightarrow h$$

$$\boldsymbol{\varphi} : h \rightarrow X'$$

$$h = \boldsymbol{\sigma}(Wx + b)$$

$$\boldsymbol{\phi}, \boldsymbol{\varphi} = argmin \parallel X - (\boldsymbol{\varphi} \circ \boldsymbol{\phi})X \parallel^2$$

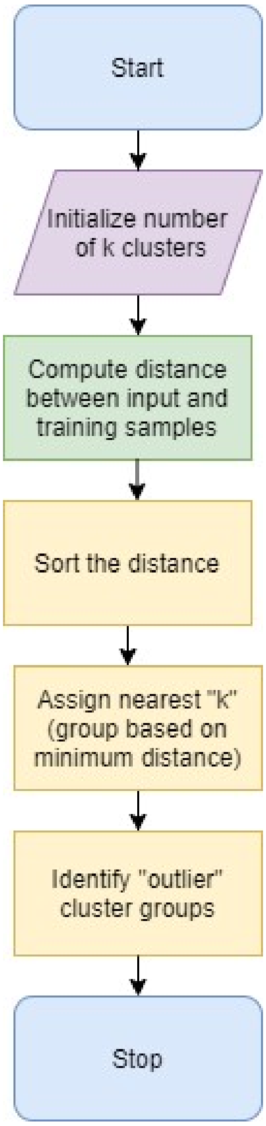


Figure 10. Steps for KNN Implementation

$$Anomalyscore = f(X' - X) \quad (3)$$

where
h= latent variable/representation

$$\boldsymbol{\sigma} = activation\ functionsuchassigmoid/RELU$$

W=Weight matrix, b= bias vector

4. RESULTS AND DISCUSSION

The training graphs of CNN and AE-LSTM are depicted in Figure 12. A good fit is described as a training and validation loss that gradually decreases to a stable point with a small difference between the two final loss values. The graphs indicate that both the models fit well on the training and the testing data.

To validate the efficiency of all three models, a test set is utilized, which has 2995 normal cases and 1000 faulty cases. The confusion matrix is computed to define the output of the models on the test data. The true values of this test data are known to determine the performance of the algorithm. Most output metrics like accuracy, precision, etc., are calculated from the confusion matrix. From the matrix, we get:

- True Positive (TP): observation is classified as normal when it indeed is normal
- False Negative (FN): observation is classified as faulty when it indeed is normal

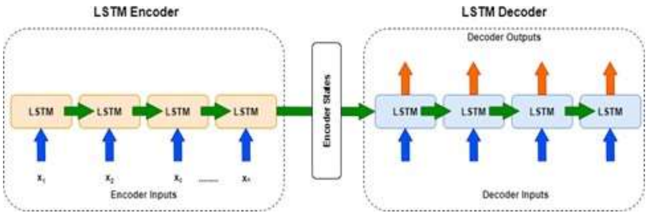
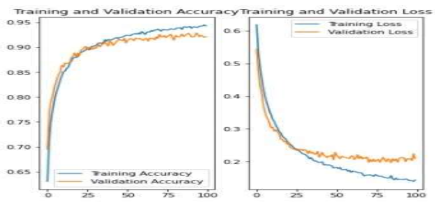
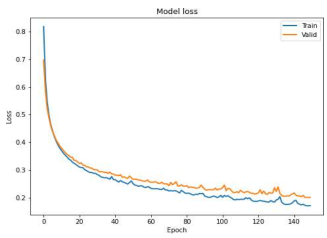


Figure 11. Autoencoder-LSTM framework



(a) CNN

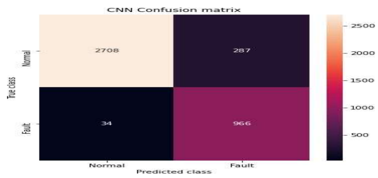


(b) AE-LSTM

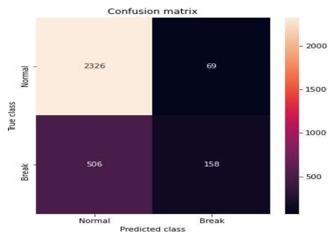
Figure 12. Training and Validation Loss of Deep Learning Techniques

- True Negative (TN): observation is classified as faulty when it indeed is faulty
- False Positive (FP):observation is classified as normal when it indeed is faulty

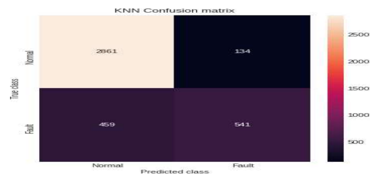
The accuracy can be calculated by adding the number of true positives and negatives and dividing this sum by the total samples in the test set. Hence the accuracy values can be computed for CNN, AE-LSTM, and k-NN from their confusion matrices shown in Figure13 (a), (b),and(c), respectively. CNN achieved good performance by identifying the correct values of true positives and true negatives and achieving an accuracy of 93%. The AE-LSTM model had a slightly lower accuracy of 87% compared to CNN, and lastly, the k-NN model achieved an accuracy of 74%. AE-LSTM had the least number of false positives (FP) compared to k-NN and CNN because of the LSTM layer, capable of learning long-term dependencies and retaining better characteristics. But CNN had the least number of false negatives (FN) among the three techniques used in this study. Accessing the models on the accuracy values



(a) CNN



(b) LSTM-AE

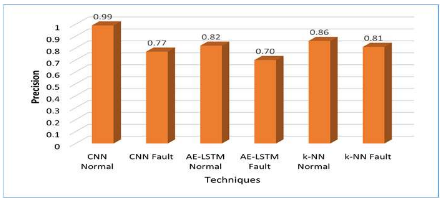


(c) k-NN

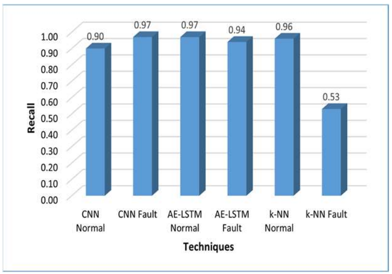
Figure 13. Confusion Matrixfor (a) CNN, (b)AE-LSTM and (c) k-NN techniques

alone doesn't yield hard facts to base a strong conclusion. Therefore further analysis was carried out.

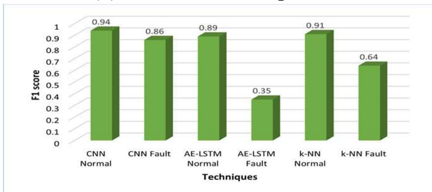
For further evaluation of the performance of the models, classification reports are generated for each algorithm. In certain cases of imbalance dataset, some extra parameters such as precision, recall, and f1-score are used to analyze the performance of a model. Precision can be defined as the relevant predictions among all the classified values. Precision is important when getting the prediction wrong is costlier than the cost of getting the right prediction. The following formula can give precision:



(a) Precision score comparison

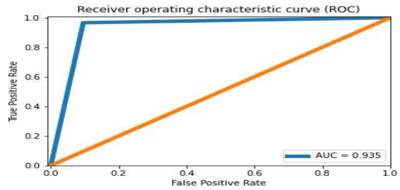


(b)Recall score comparison

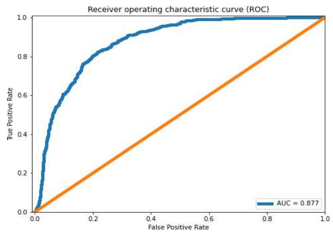


(c) F1-score comparison

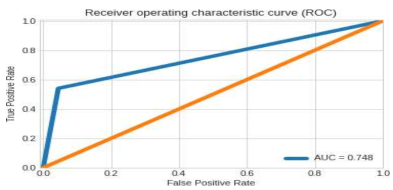
Figure 14. Performance comparison over normal and faulty data of (a) CNN, (b)AE-LSTM and (c)k-NN techniques techniques



(a) CNN



(b)LSTM-AE



(c) k-NN

Figure 15. ROC-AUC curves for (a) CNN, (b)AE-LSTM and (c) k-NN techniques

$$Precision = \frac{TruePositives}{TruePositives + FalsePositives} \quad (3)$$

Next is Recall, which can be defined as the ratio of the model's predicted correctly to the true labels. The recall is crucial if we do not want to miss any prediction at the cost of a wrong prediction. Recall can be formulated as follows:

$$Recall = \frac{TruePositives}{TruePositives + TruePositives + FalseNegatives} \quad (4)$$

Finally, F1- score is the harmonic average of Precision and Recall.

$$F1score = \frac{2X(PrecisionXRecall)}{Precision + Recall} \quad (5)$$

Tables 2,3, and 4 show the precision, recall, and F1 scores of the techniques used. Also,Figure 14 presents a performance evaluation and a comparative visualization in terms of classification scores of all the three models in the case of normal (unworn tool) and fault (worn tool) data. If the precision is high, then the model was able to predict more true positives. This relates to a low false positive rate.

Figure 14(a) validates that the precision values for Normal and fault samples are highest for CNN. In Figure 14(b), the recall measures for fault are compared where CNN and AE-LSTM out-perform the k-NN by a significant margin as k-NN is easily susceptible to noise and insensitive to small changes. Since the F1 scores are a weighted average of precision and recall, CNN has a better score than k-NN and AE-LSTM and intuitively helps access the models better. Figure 14(c) shows the comparison of F1 scores.

To visualize the performance of the classification, the Area Under the Curve (AUC) – Receiver Operating Characteristic (ROC) curve is used. It tells us the capability of the model to distinguish between the classes. Figure 15 depicts the ROC curve for all three techniques.The ROC curve is obtained by plotting the TP rate or the sensitivity as a function of the FP rate. It shows the competence of the model at different classification thresholds.An AUC value close to one signifies that it is a good model and is capable enough to classify the observations. So higher the AUC better is the model performance. As shown in Figure 15(a), the CNN techniquehas a good AUC value of 0.935. The AE-LSTM AUC curve shown in Figure 15(b) is more gradual

TABLE II. CLASSIFICATION SCORES FOR CNN

CNN	Precision	Recall	F1-Score	Support
Normal (0)	0.99	0.90	0.94	2995
Fault (1)	0.77	0.97	0.86	1000
Macro Average	0.88	0.94	0.90	3995
Weighted Average	0.93	0.92	0.92	3995

TABLE III. CLASSIFICATION SCORES FOR AE-LSTM

AE-LSTM	Precision	Recall	F1-Score	Support
Normal (0)	0.82	0.97	0.89	2395
Fault (1)	0.70	0.24	0.35	664
Macro Average	0.76	0.60	0.62	3059
Weighted Average	0.79	0.81	0.77	3059

TABLE IV. CLASSIFICATION SCORES FOR K-NN

k-NN	Precision	Recall	F1-Score	Support
Normal (0)	0.86	0.96	0.91	2995
Fault (1)	0.81	0.53	0.64	1000
Macro Average	0.83	0.74	0.77	3995
Weighted Average	0.85	0.85	0.84	3995

than CNN but attains a good cure compared to k-NN in Figure 15(c). The curves signify that the CNN model has a better True positive rate and can distinguish samples better than the other models used in this study. Considering all the proofs, CNN has proved to have a high tolerance for noise, draw out better characteristics, and better overall execution in this research. The required internal structure can be discovered and extracted to automatically produce deep data characteristics using convolution and pooling operations [39].

5. CONCLUSION

The authors carried performance analysis of the CNN, AE-LSTM, and k-NN techniques on the Smart Michigan Lab milling dataset. CNN and AE-LSTM are deep learning techniques as they contain more than one hidden layer. k-NN is a machine learning technique with shallow structure. The evaluation was done based onthe confusion matrix, and accuracy was calculated. For further comparison, precision, recall, and f1-score of all the three techniques on healthy and faulty data were also calculated. ROC-AUC curves for all three techniques were also plotted. The following conclusions can be derived:

1) The accuracy of the CNN model was the highest at 93%, followed by AE-LSTM at 87% and k-NN at 74%.
2) AE-LSTM had the least number of false positives (69) compared to the other two techniques. However, CNN had the least number of false negatives (34).
3) The classification measures of precision, recall, and f1-score on normal and faulty data proved that CNN performed better than AE-LSTM and k-NN techniques.

This paper presents tool wear monitoring in milling using the basic variant of CNN and AE-LSTM techniques. CNN model proved to have better accuracy in identifying true positives, and the AE-LSTM model works well with sequential time-series data. The authors conclude from this study that deep learning approaches have better capabilities of fault classification over time series data due to their automatic feature extraction capabilities. Also, the nodes in the hidden layers of deep learning techniques can auto-adjust their weights for better prediction accuracy depending on the feature importance. Therefore, in the future, the authors propose applying the ensemble Convolutional Autoencoder-LSTM approach to improve the model's accuracy further and predict the life expectancy of the tool as ensemble models mostly produce better predictions to a single model. Also, the models can be reused as part of transfer learning for monitoring tool wear in case of other manufacturing processes such as drilling and turning. Using some pre-trained models and fine-tuning them for the problem at hand, Transfer Learning (TL) algorithms can help boost model efficiency.This eliminates the need to train the machine learning algorithm from the ground up, saves massive computations, and reduces the amount of training data needed.

REFERENCES

1 S. Sun, X. Hu, W. Cai, and J. Zhong, ”Tool breakage detection of milling cutter insert based on SVM,” *IFAC-PapersOnLine*, 2019.

[0][1]S. Cho, S. Asfour, A. Onar, and N. Kaundinya, “Tool breakage detection using support vector machine learning in a milling process,”*Int. J. Mach. Tools Manuf.*, 2005.

[2]M. A. Elbestawi, T. A. Papazafiriou, and R. X. Du, “In-process monitoring of tool wear in milling using cutting force signature,” *Int. J. Mach. Tools Manuf*, 1991

[3]Sayyad, S., Kumar, S., Bongale, A., Bongale, A., Patil, S. (2021). Estimating Remaining Useful Life in Machines Using Artificial Intelligence: A Scoping Review. *ilosophy and Practice* 1-26.

[4]C. Cooper, J. Zhang, R. X. Gao, P. Wang, and I. Ragai, “Anomaly detection in milling tools using acoustic signals and generative adversarial networks,”*a Manuf.*,2020.

[5]B. Cuka and D. W. Kim, “Fuzzy logic based tool condition monitoring for end-milling,” *omput. Integr. Manuf.* 2017.

[6]M. Nouri, B. K. Fussell, B. L. Ziniti, and E. Linder, “Real-time tool wear monitoring in milling using a cutting condition independent method,”*Mach. Tools Manuf.*,2015.

[7]N. Ghosh et al., “Estimation of tool wear during CNC milling using neural network-based sensor fusion,”*Mech. Syst.Signal Process*,2007.

[8]T. Roney et al., “Acoustic signal analysis for prediction of flank wear during conventional milling,” in *ASME International Mechanical Engineering Congress and Exposition, Proceedings (IMECE)*, 2018.

[9]A. H. Ammouri and R. F. Hamade, “Current rise criterion: A process-independent method for tool-condition monitoring and prognostics,” *Int. J. Adv. Manuf. Technol.*, 2014.

[10]A. G. Rehorn, J. Jiang, and P. E. Orban, “State-of-the-art methods and results in tool condition monitoring: A review,” *International Journal of Advanced Manufacturing Technology.* 2005.

[11]Z. B. Patel and S. Muthuswamy, “A Machine Learning Scheme for Tool Wear Monitoring and Replacement in IoT-Enabled Smart Manufacturing,” in *Lecture Notes in Mechanical Engineering*, 2020.

[12]S. Kumar, M. Nagraj, R. Suresh, A. Bongale and N. Khedkar, ”The Effect of Holding Time of Deep Cryogenic-Treated AISI D3 Cutting Tool on Machinability of Low Carbon Steels Using Taguchi’s Technique”, *Journal of Advanced Manufacturing Systems, vol. 19, no. 02, pp. 215-233*, 2020.

[13]M. Pratama, E. Dimla, C. Y. Lai, and E. Lughofer, “Metacognitive learning approach for online tool condition monitoring,” *J. Intell. Manuf.*, 2019.

[14]S. Cho, S. Asfour, A. Onar, and N. Kaundinya, “Tool breakage detection using support vector machine learning in a milling process,” *Int. J. Mach. Tools Manuf.*, 2005.

[15]E. Kuljanic, G. Totis, and M. Sortino, “Development of an intelligent multisensor chatter detection system in milling,” *Mech. Syst. Signal Process.*, 2009.

[16]D. A. Tobon-Mejia, K. Medjaher, and N. Zerhouni, “CNC machine tools wear diagnostic and prognostic by using dynamic Bayesian networks,” *Mech. Syst. Signal Process.*, 2012.

[17]O. Geramifard, J. X. Xu, J. H. Zhou, and X. Li, “A physically segmented hidden markov model approach for continuous tool condition monitoring: Diagnostics and prognostics,” *IEEE Trans. Ind. Informatics*, 2012.

[18]N. Gangadhar, H. Kumar, S. Narendranath, and V. Sugumaran, “Fault Diagnosis of Single Point Cutting Tool through Vibration Signal Using Decision Tree Algorithm,” *Procedia Mater. Sci.*, 2014.

[19]P. Junior, D. M. D’Addona, P. Aguiar, and R. Teti, “Dressing tool condition monitoring through impedance-based sensors: Part 2—neural networks and K-nearest neighbor classifier approach,” *Sensors (Switzerland)*, 2018.

[20]G. Mart ́ınez-Arellano, G. Terrazas, and S. Ratchev, “Tool wear classification using time series imaging and deep learning,” *Int. J. Adv. Manuf. Technol.*, 2019.

[21]T. Wuest, D. Weimer, C. Irgens, and K. D. Thoben, “Machine learning in manufacturing: Advantages, challenges, and applications,” *Prod. Manuf. Res.*, 2016

[22]M. Yuan and M. Wang, “A feature selection method based on an improved fruit fly optimization algorithm in the process of numerical control milling,” *Adv. Mech. Eng.*, 2018.

[23]Q. Rena, M. Balazinskia, L. Barona, and K. Jemielniakb, “TSK fuzzy modeling for tool wear condition in turning processes: An experimental study,” *Eng. Appl. Artif. Intell.*, 2011.

[24]T. Mikołajczyk, K. Nowicki, A. Bustillo, and D. Y. Pimenov, “Predicting tool life in turning operations using neural networks and image processing,” *Mech. Syst. Signal Process.*, 2018.

[25]C. Chungchoo and D. Saini, “On-line tool wear estimation in CNC turning operations using fuzzy neural network model,”*Int. J. Mach. Tools Manuf.*, 2002.

[26]P. Jinka and B. Schwartz, *Anomaly detection for monitoring: A statistical approach to time series anomaly detection..* O’Reilly Media, 2015.

[27]P. Kamat and R. Sugandhi, “Anomaly detection for predictive maintenance in industry 4.0-A survey,” in *E3S Web of Conferences*, 2020.

[28]G. Hu, H. Li, Y. Xia, and L. Luo, “A deep Boltzmann machine and multi-grained scanning forest ensemble collaborative method and its application to industrial fault diagnosis,” *Comput. Ind.*, 2018.

[29]C. K. Madhusudana, H. Kumar, and S. Narendranath, “Fault Diagnosis of Face Milling Tool using Decision Tree and Sound Signal,” in*Materials Today: Proceedings*, 2018.

[30] Segreto, T., Caggiano, A., Nele, L. (2020). Smart Multi-Sensor Monitoring in Drilling of CFRP/CFRP Composite Material Stacks for Aerospace Assembly Applications.*Applied Sciences*, 10(3), 758.

[31] Cooper, C., Wang, P., Zhang, J., Gao, R. X., Roney, T., Ragai, I., Shaffer, D. (2020). Convolutional neural network-based tool condition

monitoring in vertical milling operations using acoustic signals. *Procedia Manufacturing*, 49, 105-111.

[32] Ou, J., Li, H., Huang, G., Zhou, Q. (2020). A Novel Order Analysis and Stacked Sparse Auto-Encoder Feature Learning Method for Milling Tool Wear Condition Monitoring. *Sensors*, 20(10), 2878.

[33] Gomes, M. C., Brito, L. C., da Silva, M. B., Duarte, M. A. V. Tool wear monitoring in micromilling using Support Vector Machine with vibration and sound sensors.*Precision Engineering*, 67, 137-151.

[34] CNC Mill Tool wear-Kaggle. https: //www.kaggle.com/shasun/tool-wear-detection-in-cnc-mill (Accessed on July 2020)

[35]D. F. Hesser and B. Markert, "Tool wear monitoring of a retrofitted CNC milling machine using artificial neural networks," *Manuf. Lett.*, 2019.

[36]G. Terrazas, G. Mart ́ınez-Arellano, P. Benardos, and S. Ratchev, "Online Tool Wear Classification during Dry Machining Using Real Time Cutting Force Measurements and a CNN Approach," *J. Manuf. Mater. Process.*, 2018.

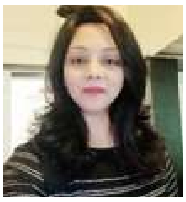
[37]L. Wang, L. Khan, and B. Thuraisingham, "An e ffective evidence theory based K-nearest neighbor (KNN) classification," in *Proceedings - 2008 IEEE/WIC/ACM International Conference on Web Intelligence*, WI 2008, 2008.

[38]B. Zhao, H. Lu, S. Chen, J. Liu, and D. Wu, "Convolutional neural networks for time series classification," *J. Syst. Eng. Electron.*, 2017.

Prof. Pooja Kamat works as an Assistant Professor in the Computer Science Engineering and Information Technology department at Symbiosis Institute of Technology, Symbiosis International (Deemed University), Pune, Maharashtra, India. She has completed her M.Tech from Mumbai University and is currently pursuing Ph.D. in the domain of Predictive Maintenance. She has a teaching experience of 12 years and has guided many UG and PG students in the domain of Artificial Intelligence and Machine Learning. Her research interests include Predictive Analytics and its application in the domain of manufacturing and healthcare. She has authored many international/national journal and conferences publications. According to Google Scholar, she has 100 + citations, with an H-index of 5 and an i10- index of 5.

Mr. Anish Nargund is an alumnus of B.Tech (Electronics & Telecommunication Engineering -2016 batch) of Symbiosis Institute of Technology, Symbiosis International (Deemed University), Pune India. He worked on developing a mobile solution to detect cardiovascular diseases using Machine Learning as part of his B.Tech thesis. At present, he is working as an ETL tool developer at Innovation Consulting. His research interests lie in NLP, Image Processing, Industry 4.0, ERP platform and Data Analysis.

Dr. Satish Kumar is Assistant Professor in the Department of Mechanical Engineering at Symbiosis Institute of Technology, Symbiosis International (Deemed University), Pune India. He did his Master degree (M.Tech) and Doctoral degree (PhD) in 2013 and 2020 from K Visvesvaraya Technological University, Belgaum, Karnataka, India. He has 8 years of experience in teaching, research and industries. His area of research interests include Smart Manufacturing, Condition Monitoring, Composites, Cryogenic Treatment, Additive manufacturing and Hard materials machining. He has authored more than 24 international/national journal and conferences publications. According to Google Scholar, he has 120 + citations, with an H-index of 7 and an i10- index of 4.



Dr Shruti Patil has been an industry professional in the past, currently associated with Symbiosis Institute of Technology as a professor and with SCAAI as a research associate,PuneMaharastra. She has completed her M.Tech in Computer Science and Ph.D in the domain of Data Privacy from Pune University. She has 3 years of industry experience and 10 years of academic experience. She has expertise in applying innovative technology solutions to real world problems. Her research areas include applied artificial intelligence, natural language processing, acoustic AI, adversarial machine learning, data privacy, digital twin applications, GANS, multimodal data analysis. She is currently working in the application domains of healthcare, sentiment analysis, emotion detection and machine simulation via which she is also guiding several UG, PG and PhD students as a domain expert. She has published 30+ research articles in reputed international conferences and Scopus/ web of science indexed journals, books with 70+ citations .

Dr. Rekha Sugandhi, Ph.D (Computer Science Engineering) is the Head and Professor in the Information Technology department at MIT School of Engineering, MIT-ADT University, Maharashtra, India. She has a vast teaching experience of 21 years. Her subject and research interests include Natural Language Processing, Computational Linguistics, Computer Graphics and Artificial Intelligence. She is a research supervisor for UG, PG and PhD doctoral students at MIT-ADT University.