




## Article

# Tool Wear Monitoring Using Improved Dragonfly Optimization Algorithm and Deep Belief Network

Leo Gertrude David <sup>1</sup>, Raj Kumar Patra <sup>2</sup>, Przemysław Falkowski-Gilski <sup>3,\*</sup>,  
Parameshchari Bidare Divakarachari <sup>4,\*</sup> and Lourdusamy Jegan Antony Marcinlin <sup>5</sup>

<sup>1</sup> Department of Visual Communication, Kumaraguru College of Liberal Arts and Science, Coimbatore 641035, India

<sup>2</sup> Department of Computer Science and Engineering, CMR Technical Campus, Hyderabad 501401, India

<sup>3</sup> Faculty of Electronics, Telecommunications and Informatics, Gdansk University of Technology, Narutowicza 11/12, 80-233 Gdansk, Poland

<sup>4</sup> Department of Electronics and Communication Engineering Nitte Meenakshi Institute of Technology, Bangalore 560064, India

<sup>5</sup> Department of ECE, Sathyabama Institute of Science and Technology, Chennai 600119, India

\* Correspondence: przemyslaw.falkowski@eti.pg.edu.pl (P.F.-G.); paramesh@nmit.ac.in (P.B.D.)

**Abstract:** In recent decades, tool wear monitoring has played a crucial role in the improvement of industrial production quality and efficiency. In the machining process, it is important to predict both tool cost and life, and to reduce the equipment downtime. The conventional methods need enormous quantities of human resources and expert skills to achieve precise tool wear information. To automatically identify the tool wear types, deep learning models are extensively used in the existing studies. In this manuscript, a new model is proposed for the effective classification of both serviceable and worn cutting edges. Initially, a dataset is chosen for experimental analysis that includes 254 images of edge profile cutting heads; then, circular Hough transform, canny edge detector, and standard Hough transform are used to segment 577 cutting edge images, where 276 images are disposable and 301 are functional. Furthermore, feature extraction is carried out on the segmented images utilizing Local Binary Pattern (LBPs) and Speeded up Robust Features (SURF), Harris Corner Detection (HCD), Histogram of Oriented Gradients (HOG), and Grey-Level Co-occurrence Matrix (GLCM) feature descriptors for extracting the texture feature vectors. Next, the dimension of the extracted features is reduced by an Improved Dragonfly Optimization Algorithm (IDOA) that lowers the computational complexity and running time of the Deep Belief Network (DBN), while classifying the serviceable and worn cutting edges. The experimental evaluations showed that the IDOA-DBN model attained 98.83% accuracy on the patch configuration of full edge division, which is superior to the existing deep learning models.

**Keywords:** canny edge detector; deep belief network; dragonfly optimization algorithm; image processing; local binary pattern; tool wear monitoring



**Citation:** David, L.G.; Patra, R.K.; Falkowski-Gilski, P.; Divakarachari, P.B.; Antony Marcinlin, L.J. Tool Wear Monitoring Using Improved Dragonfly Optimization Algorithm and Deep Belief Network. *Appl. Sci.* **2022**, *12*, 8130. <https://doi.org/10.3390/app12168130>

Academic Editor: Fabio La Foresta

Received: 26 June 2022

Accepted: 8 August 2022

Published: 14 August 2022

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

Tool wear is the result of cutting temperature, cutting force, and mechanical friction in the milling process of Computer Numerical Control (CNC) machines [1,2]. Tool wear reduces the workpiece quality and increases the workpiece surface roughness [3]; serious tool wear causes chatter, fracturing, and chipping, which damages both the machine tools and the workpiece, and can also lead to serious processing accidents [4]. Therefore, appropriate monitoring and classification mechanisms could help to decrease the loss caused by tool wear, and to obtain better surface quality [5]. By modifying the structure of solid materials, the machining process is the most significant industrial method for producing semi-finished and final outputs. Cutting parameters are typically utilized to eliminate chips in the substance [6]. However, even though advanced technologies

in cutting tool compositions and milling machine control technologies have resulted in extended mechanical properties, the instruments eventually wear down with time [7]. Tool wear invariably has a direct impact on the surface polish and predicted probability of the completed workpiece, resulting in failures [8]. Tool removal and unplanned shutdowns because of worn tools could even result in machine breakdown and product inconsistencies, and eventually in financial loss. As a result, it is critical to avoid unscheduled downtime throughout operation, which has a significant effect on organizational performance [9]. Tool condition monitoring methodologies have been adopted for nearly two decades to minimize this, and they have proved themselves as a basic requirement of modern machine tools [10].

Coated tools are more widely utilized in numerous machining operations than uncoated tools because they have a higher tool performance and increased processability [11]. The remaining thickness of coated tools throughout manufacturing operations might be beneficial for the technician to monitor when using worn tools [12]. To determine the depth of a coating layer, aggressive tools such as microhardness monitoring and metallography would be used, although they are costly, time-consuming, and difficult to utilize [13]. The tool wear monitoring is categorized into direct approaches and indirect approaches based on the hardness of the machining process [14]. The direct approaches based on the measurement of flank wear consists of electrical resistance, radioactivity, and vision inspection [15]. The direct approaches are not conducive to the practical applications, due to the limitation of accessing similar levels of illumination, cutting fluid, and the presence of chips, and high requirements must be reached to measure the environment [16]. Therefore, indirect approaches are effective in monitoring tool wear based on image analysis. The computer vision approach directly measures tool wear, which helps in achieving high levels of reliability and precision [17,18]. Nonetheless, the bulk of tool condition measurement techniques reported in currently published studies have utilized complex, non-production-ready sensing technologies, including force dynamometers [19]. Due to the diversity of signals that must be acquired from the operating equipment, even with sophisticated data-gathering hardware, collecting high-quality datasets to train the machine is challenging and time-consuming [20]. Cutting tools are typically used to decrease cutting temperature and friction during machining processes to increase tool life and surface quality. Water-oil emulsions, with emulsifiers and additives to stop corrosion and bacterial development, make up the majority of cutting fluids. Poisonous compounds are present in the emulsions, machine oils, and the heavy metals that are combined with the fluids during the machining process. In order to lessen the impact on the environment, cutting fluid application must be reduced and hazardous components must be eliminated. In this manuscript, a new Improved Dragonfly Optimization Algorithm – Deep Belief Network (IDOA-DBN) model is proposed for effective tool wear monitoring; the major contributions of this study are listed below:

- After image collection, cutting edge detection is accomplished using the circular Hough transform technique, canny edge detector, and standard Hough transform.
- Subsequently, feature extraction is performed utilizing Speeded up Robust Features (SURF) and Local Binary Pattern (LBP) feature descriptors to extract the texture feature vectors; then, IDOA is proposed to reduce the dimensions of the extracted texture feature vectors, which helps in improving the system complexity and running time of the classifier.
- Lastly, the selected discriminative feature vectors are fed into the DBN to classify serviceable and worn cutting edges. The IDOA-DBN model's effectiveness is evaluated by means of F-score, recall, precision, and accuracy.

This manuscript is structured as follows: recent papers on the topic of tool wear monitoring are reviewed in Section 2. Theoretical descriptions and results of the experimental analysis of the IDOA-DBN model are provided in Sections 3 and 4, respectively. The conclusion of the manuscript is presented in Section 5.



## 2. Related Works

H. Oo [21] combined Multiple Linear Regression (MLR) and Random Forest Classifier (RFC) techniques to evaluate grinding capacity and detect the wear conditions of robotic belt grinding. Here, five distinct belts exhibiting varied tool wear conditions were used in the proposed simulation process, and 300 observations of grinding belt surface wear were used as training and testing parameters for the belt condition classifier. The fact that the data were fixed was considered one of the limitations. G. Li [22] combined time–frequency, frequency, and time domain feature extraction techniques to extract feature vectors from the vibration and force signals of the CNC machines. Next, a Gradient Boosting Decision Tree (GBDT) approach was applied for selecting the optimal feature vectors, and then classification was carried out using a hybrid method. Here, Prognostics and Health Management (PHM) challenge 2010 dataset was utilized to confirm the projected system. In this case, not every trait was automatically linked to tool wear. In addition, M.T. García-Ordás [23] segmented the cutting edges from the images of edge profile cutting heads, and then a local binary pattern descriptor was used to extract feature values from the segmented wear patch regions. Next, the Support Vector Machine (SVM) classifier was used to classify the cutting edges as serviceable or worn, and it was discovered that the overlapping between these patches began at the bottom of the main edge. G.D. Simon and R. Deivanathan [24] conducted a descriptive statistical evaluation to extract feature vectors from drilling-induced vibration signals, after which, the K-star classification method was used for classifying the tool conditions: bad tool at high speed, bad tool at low speed, good tool at high speed, and good tool at low speed. However, because of its wear resistance properties, the tool proved challenging to cut.

In reference [25], M. Marani utilized a Long Short-Term Memory (LSTM) network for tool flank wear prediction that helped in obtaining better machined surface with low manufacturing cost. Using a validation set, the Root Mean Square Error (RMSE) for the LSTM networks with hidden layers was determined. The findings revealed that the most efficient LSTM contains two layers and eight hidden layers, as well as a lower RMSE. Nonetheless, the prediction behavior of the two-layer LSTM network surpassed the three-layer model. In reference [26], X. Liu utilized raw signals as network inputs for monitoring the tool wear of a high-speed CNC machine. After acquiring the raw sensor signals, feature extraction was accomplished using a parallel residual network for extracting multiscale local feature vectors. Furthermore, a stacked bidirectional LSTM network was utilized to obtain the time series feature vectors related to the tool wear properties. The suggested LSTM had maximum convergence speed, but, at the same time, it exhibited high training loss. In reference [27], Z. Huang developed a Deep Convolutional Neural Network (DCNN) model for tool wear monitoring based on multidomain feature fusion. However, when used as a predictor for machine vibration, the DCNN algorithm would have been unable to detect tool wear in actual environments. X.C. Cao [28] combined a CNN model and Derived Wavelet Frames (DWF) for tool wear state prediction utilizing machine spindle vibration signals. The reconstituted subsignals were then layered into a 2-D signal vector to replicate the design of a 2-D CNN, while simultaneously maintaining additional information. Unfortunately, this raised the network's complex nature, making it even harder to optimize and increase the likelihood of overfitting.

X. Liao [29] combined an SVM classifier and Genetic Algorithm (GA) to predict tool wear state. Initially, wavelet packet decomposition techniques, frequency domain, and time domain statistics were used to extract cutting force signal. Finally, the SVM classifier with Grey Wolf Optimizer (GWO) was developed to obtain the state recognition results. On the training dataset, the performance accuracy was indeed the lowest. To increase tool wear prediction accuracy, X. Wu [30] used Bidirectional LSTM (BiLSTM) and Singular Value Decomposition (SVD) methods. The measuring input was received from the cutting force output. The raw cutting force data were then recreated using the Hankle matrix, and then, the signal features were extracted using the SVD of the regenerated matrix. This technique properly detects the tool's wear phase; however, it is unable to estimate the tool's

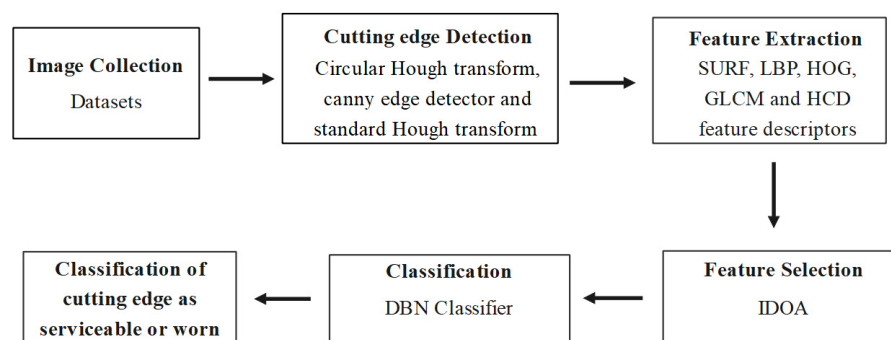


wear rating. In [31], S. Laddada integrated an improved extreme learning machine and complex continuous wavelet transform to investigate the condition of cutting tools, and to predict their remaining lifetime. The suggested technique entails combining the Complex Continuous Wavelet Transform (CCWT) and the Improved Extreme Learning Machine (IELM). Moreover, since it was premised on the use of past data, its reliability is low.

By reviewing the abovementioned existing studies, it is clear that deep-learning-based classification techniques perform substantial data reductions, which may cause information loss. Additionally, with such techniques it is difficult to develop feature descriptors for precise tool wear estimation, due to the variations in cutting force. Therefore, a novel feature optimization method based DBN is proposed in this manuscript, which is free from three major concerns: overfitting, training, and testing efficiency.

### 3. Methodology

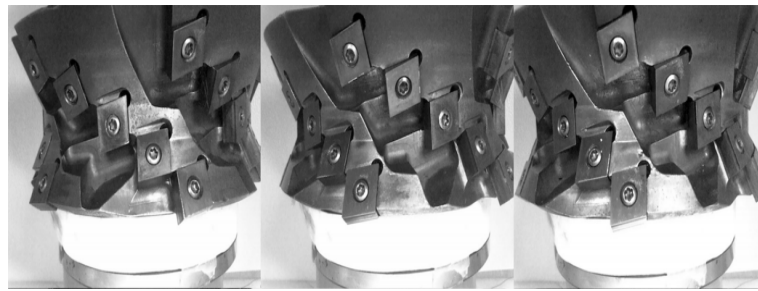
Deep Belief Networks (DBN) are used to establish tool wear prediction models due to their superior learning speeds and fast rates of convergence to the optimal outcomes, regardless of whether the sample data are small or large. This improves the modeling accuracy and efficiency of the tool wear monitoring system. DBN parameters are typically manually adjusted, which results in low predictive accuracy and efficiency. The Improved Dragonfly Optimization Algorithm (IDOA) is then designed to optimize the computational complexity and running time of the DBN. Therefore, the combination of IDOA and DBN provides better results when compared to other approaches, which lack efficiency and accuracy, especially when the sample becomes very large, and exhibit prolonged computing times. Thus, this research proposes that the combination of IDOA and DBN is suitable for tool wear prediction. In terms of tool wear classification, the proposed IDOA-DBN model includes five stages: image collection (dataset); cutting edge detection (circular Hough transform, canny edge detection, and standard Hough transform); feature extraction (SURF and LBP feature descriptors); feature optimization (IDOA); classification (DBN). The workflow of the IDOA-DBN model is specified in Figure 1.



**Figure 1.** Workflow of the IDOA-DBN model.

#### 3.1. Image Collection

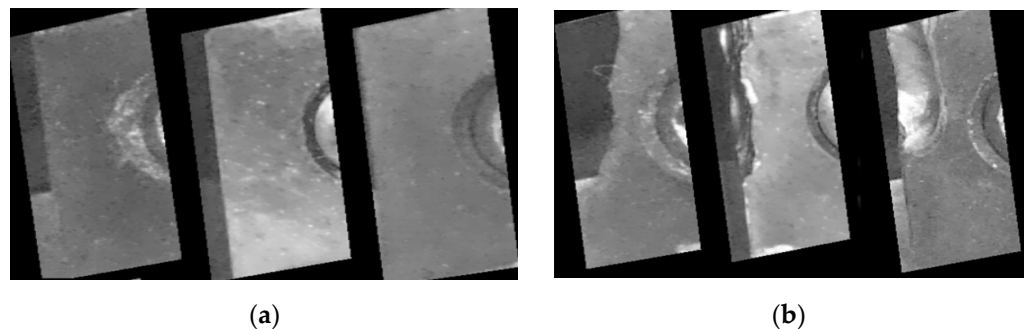
In this manuscript, a dataset with 254 cutting edge images of an edge-profile-milling machine is utilized for investigating the effectiveness of the proposed model. In this study, the tool head is cylindrical in shape and comprises 30 inserts, which are arranged in six groups of five inserts. In this dataset, a Dalsa Genie m1280 1/3 camera with Azure-2514 mm lens is used for capturing the images, with a pixel resolution of  $2592 \times 1944$  and focal length of 25 mm. Three Light-Emitting Diode (LED) bar lights are used to achieve independent illumination conditions in the environment [23]. The sample-acquired images are presented in Figure 2.



**Figure 2.** Acquired sample images from cutting edge dataset.

### 3.2. Cutting Edge Detection

After image acquisition, the circular Hough transform technique is used for detecting the screws located in the inserts, and the canny edge detector is employed for detecting the edges of the inserts [32,33]. The definition of the image is expanded to include ellipsoids, circles, and expressions with powers of three and higher. Irises are localized using the circular Hough transform; we also suggest computing the initial derivatives of the brightness of the image and thresholding the resulting values to generate the points of the parametric form. Furthermore, the standard Hough transform technique is utilized for detecting the vertical lines, and finally, the cutting edge of the insert is extracted from the acquired images [34]. Among 577 cutting edge images, the 276 images are labeled as worn edges, and 301 images are labeled as serviceable edges. The cutting edges of six cutting tools are presented in Figure 3.



**Figure 3.** Detected portion of cutting edges: (a) serviceable edges and (b) edges with high wear.

### 3.3. Feature Extraction

After collecting the images, a patch-based technique is applied for validating the insert wear level, which completely relies on dividing the cutting edge images into wear patches with dissimilar orientations, shapes, and sizes. Then, the wear patches are further categorized into serviceable wear or disposable wear. The different patch configurations for the cutting edges are full edge division, homogeneous grid division, small edge division, half edge division, and two-band division. After dividing the cutting edge images, feature extraction is carried out utilizing SURF and LBP feature descriptors.

#### 3.3.1. Speeded Up Robust Features (SURF)

The SURF feature descriptor is a robust and fast algorithm for comparison, local, and similarity invariant representation of the images [35]. The SURF feature descriptor completely relies on its faster computation of operators utilizing box filters. While the Hessian matrix determinant is maximum, the SURF feature descriptor detects the blob-like



structure. Let us consider the point  $x = (x, y)$  in the acquired image  $I$ ; the Hessian matrix  $H(x, \sigma)$  with scale  $\sigma$  is determined at  $x$  using Equation (1):

$$H(x, \sigma) = \begin{bmatrix} L_{xx}(x, \sigma) & L_{xy}(x, \sigma) \\ L_{xy}(x, \sigma) & L_{yy}(x, \sigma) \end{bmatrix} \tag{1}$$

where  $\frac{\partial^2}{\partial x^2}$  indicates Gaussian second-order derivation,  $L_{xx}(x, \sigma)$ ,  $L_{yy}(x, \sigma)$ , and  $L_{xy}(x, \sigma)$  states convolution of  $\frac{\partial^2}{\partial x^2}$  at point. In the SURF feature descriptor, the scale spaces are categorized into octaves to recognize the interest points at different scales, where each octave includes a series of intervals. The convolutional window scale with the parameters octave ( $o$ ) and interval ( $i$ ), is specified in Equation (2), and the relationship between scale  $\sigma$  and window size is mathematically stated in Equation (3):

$$L = 3 \times (i \times 2^o + 1) \tag{2}$$

$$L = \sigma \times 9/1.2 \tag{3}$$

Furthermore, the SURF key points are calculated utilizing Equation (4):

$$DoH(x, L) = \max \left( \sum_{k_x=i-1}^{i+1} \sum_{k_y=x-2^o}^{x+2^o} \sum_{k_y=y-2^o}^{y+2^o} DoH(k_x, k_y, o, k_i) \right) \geq \lambda \tag{4}$$

where  $DoH$  indicates determinant of the Hessian matrix and  $\lambda$  denotes positive threshold value. If the Hessian matrix trace value is higher than zero, a bright blob with scale  $L = 3 \times (i \times 2^o + 1)$  is detected at  $(x, y)$ . Around 1288 feature vectors are extracted from the acquired images using the SURF feature descriptor.

### 3.3.2. Local Binary Pattern (LBP)

LBP is a texture feature descriptor that extracts pixel-wise information from the acquired images. The LBP feature descriptor considers the number of neighbors ( $p$ ) in each pixel ( $c$ ) within the radius ( $r$ ). If the grey-level value of  $p$  is higher than or equal to  $c$ , the value of one is assigned, or zero otherwise. By summing up the values to the power of two, the LBP of every pixel is calculated using Equation (5) [36]:

$$LBP_{p,r} = \sum_{p=0}^{p-1} s(g_p - g_c)2^p, s(x) = \begin{cases} 1 & \text{if } x \geq 0 \\ 0 & \text{if } x < 0 \end{cases} \tag{5}$$

where  $p$  indicates the number of neighbors,  $g_p$  represents the value of the  $p^{th}$  neighbor, and  $g_c$  denotes the center pixel value; the LBP feature descriptor extracts 3470 texture feature vectors. Using the feature-level fusion method, the SURF and LBP feature vectors are integrated; thus, the model becomes multidimensional, increasing its complexity.

### 3.3.3. Histogram of Oriented Gradients (HOG)

HOG is a shape identifier that determines the layout and structure of a local character [37]. This HOG describes how the intensity gradient spreads across the region, which is used to determine the character. The HOG is often based on the gradient direction accumulating via the pixels of a small spatial region, such as a single cell. The HOG descriptor in each cell collects the gradient direction's local 1-D histogram. This process is performed by constructing the local histogram over a large spatial region, typically a block of cells, and using the data to normalized all of the block's cells. The detection window is placed over the overlapping grid in this case. Equation (6) represents the pixel's horizontal and vertical gradient:

$$\begin{cases} G_p = I'(p + 1, q) - I'(p - 1, q) \\ G_q = I'(p, q + 1) - I'(p, q - 1) \end{cases} \tag{6}$$



The magnitude and orientation of HOG gradients are expressed in Equations (7) and (8), respectively:

$$G(p, q) = \sqrt{G_p^2 + G_q^2} \quad (7)$$

$$\theta(p, q) = \operatorname{arctan}\left(\frac{G_q}{G_p}\right) \quad (8)$$

The weights of the gradient magnitudes that are in the direction of interest are combined to generate the cell's histogram vector. Furthermore, the HOG's output vector is made up of normalized cells from many detection window elements and frames.

### 3.3.4. Grey-Level Co-Occurrence Matrix (GLCM)

The GLCM is a powerful feature descriptor that evaluates the spatial link between two pixels to assess the textural qualities of an image. Because the spacing and orientation between pixels are altered, the simultaneous presence of pixel pairs can be determined. There are 14 types of features presented in the GLCM. Here, some of the textural features are derived [38]. Successful description is based on the textural information gained from the GLCM.

The instantaneous occurrence probability of two pixels in the GLCM is stated as  $P(i, j, \delta, \theta)$ . It contains the pixel with, respectively, grey-level  $i$  and  $j$  as  $f(x, y)$  and  $(x + \Delta x, y + \Delta y)$ ;  $\theta$  is defined as declination, and  $\delta$  is stated as distance. The statistical formulation is stated in Equation (9):

$$P(i, j, \delta, \theta) = \left\{ \begin{array}{l} [(x, y), (x + \Delta x, y + \Delta y)] | f(x, y) = i, \\ (x, y), \quad f(x + \Delta x, y + \Delta y) = j; x = 0, 1, \dots, N_x - 1; \\ y = 0, 1, \dots, N_y - 1 \end{array} \right\} \quad (9)$$

where  $i, j = 0, 1, \dots, L - 1$ ; image pixel coordinates are represented as  $x$  and  $y$ ; the grey-level image is stated as  $L$ ; numbers of columns and rows are stated as  $N_x$  and  $N_y$ . As a result, the appropriate feature selection is performed with IDOA, which decreases the model's operating performance and computational efficiency.

### 3.3.5. Harris Corner Detection (HCD)

The Harris corner point's features are utilized to separate the background and foreground. Rotating the window in every orientation along a corner point should produce a significant shift in luminance. The point intensity of Harris's point in the fingerprint foreground areas is much greater than in the background regions. The Harris method, originally proposed by Harris et al., is a variation or extension of the Moravec corner detection algorithm that extracts corner points using grey disparity between images [39]. Assuming that a greyscale image  $I$  and window  $w(x, y)$  are considered to be removed (through movements of  $u$  in the  $x$  direction, and  $v$  in  $y$  direction), the intensity variation is calculated as Equation (10):

$$E(u, v) = \sum_{x, y} w(x, y) [I(x + u, y + v) - I(x - y)]^2 \quad (10)$$

where  $I(x, y)$  and  $I(x + u, y + v)$  are intensities at position  $(x, y)$  and at moving window  $x + u, y + v$ , correspondingly. Because windows with corners have been explored with a wide range of intensity, we decided to focus on them. As a result, the component from Equation (10) is rewritten as Equation (11):

$$\sum_{x, y} [I(x + u, y + v) - I(x - y)]^2 \quad (11)$$

Equation (12) is attained by Taylor extension:

$$E(u, v) \approx \sum_{x,y} [I(x, y) + uI_x + vI_y - I(x, y)]^2 \quad (12)$$

Escalating Equation (12) and eliminating the appropriate products leads to Equation (13):

$$E(u, v) \approx \sum_{x,y} u^2 I_x^2 + 2uvI_x I_y + v^2 I_y^2 \quad (13)$$

By evaluating Equation (13) in matrix form, we attain Equation (14):

$$E(u, v) \approx [u \ v] \begin{pmatrix} \sum_{x,y} w(x, y) I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{pmatrix} \begin{bmatrix} u \\ v \end{bmatrix} \quad (14)$$

Alternatively, consider  $M = \sum_{x,y} w(x, y) \begin{pmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{pmatrix}$ .

In this instance, Equation (14) is revised and presented as Equation (15):

$$E(u, v) \approx [u \ v] M \begin{bmatrix} u \\ v \end{bmatrix} \quad (15)$$

The eigenvalues of matrix  $M$  are stated as  $\lambda_1$  and  $\lambda_2$ , which creates a rotationally invariant description. At this time, three points need to be considered:

- As soon as an eigenvalue ( $\lambda_1$  or  $\lambda_2$ ) is high and significantly superior, an edge occurs.
- Pixel point is stated to be in a flat area when  $\lambda_1 \approx \lambda_2$  and its values are low.
- Pixel point is stated to be in a corner when  $\lambda_1 \approx \lambda_2$  and its values are high.
- A score can be determined for every window to evaluate whether it is likely to represent a corner, as shown in Equation (16):

$$R = \det(M) - k(\text{trace}(M))^2 \quad (16)$$

where  $\det(M) = \lambda_1 \lambda_2$ ,  $\text{trace}(M) = \lambda_1 + \lambda_2$ , and constant  $k$  is fixed in the range of 0.04 to 0.06. A window is obtained through score  $R$  that is greater than a positive threshold rate, i.e., considered to consume a corner.

### 3.4. Feature Optimization

After feature extraction, the IDOA is proposed for discriminative feature selection, where IDOA is a metaheuristic optimization algorithm that mimics the dynamic and static behaviors of dragonflies. The feature selection process is considered a problem of global combinatorial optimization, which seeks to reduce the quantity of features and redundant, noisy, and redundant data, while producing a uniform level of classification accuracy. As a result, this research introduces the Improved Dragonfly Optimization Technique (IDOA), an optimization algorithm that performs better. The discrete search space consists of all feasible arrangements of attributes chosen from the dataset. It might be possible to list every potential subset of characteristics, given the limited number of features. The improved dragonfly employs more group knowledge to influence its own behavior, ensuring that the group is diverse, and creating a balance between the stages of exploration and exploitation to increase the algorithm's search efficiency. This feature selection technique is typically quicker and more efficient, and minimizes overfitting, which eliminates redundant and noisy data to find a subset of relevant features using the strength of the IDOA to improve classification outcomes. The two major phases in the IDOA are exploitation and exploration, which are modeled statically or dynamically, either by avoiding the enemy or searching for food [40]. Usually, the swarms have three behaviors: cohesion, alignment, and separation. Furthermore, two additional behaviors are added to these three behaviors in the IDOA: avoiding the enemy and moving towards food. The purpose of including these two



behaviors in the IDOA is to increase the survival time of the swarm. In this algorithm, two vectors are considered: the position and step for updating the position of dragonflies in a search space. The step vector is also considered as the speed that determines the direction of dragonflies. After the step vector calculation, the position vector is updated.

In the IDOA, the coefficients (cohesion, alignment, separation, food factor, inertia coefficient, enemy factor, and iteration number) are enabled to perform exploitative and exploratory behaviors. The cohesion coefficient is high and the alignment coefficient is low in the exploitation process; conversely, the cohesion coefficient is low and the alignment coefficient is high in the exploration process. In the conventional DOA, the Levy flight mechanism is used to enhance the probabilistic behavior, randomness, and the discovery of artificial dragonflies. Hence, the Levy flight mechanism improves the DOA efficacy to a certain extent. However, the step size control is contrary to the nature of the Levy flight mechanism. The agents have to move outside the search space, if a long step is considered. To overcome these issues, the Brownian motion ( $P_g$ ), is considered in the IDOA for enhancing the probabilistic behavior, randomness, and the discovery of dragonflies. The Brownian motion ( $P_g$ ) is mathematically determined in Equations (17) and (18):

$$P_g = \frac{1}{s\sqrt{2\pi}} \exp\left(-\frac{(\text{dimension} - \text{agents})^2}{2s^2}\right) \quad (17)$$

$$s = \sqrt{\frac{m_t}{m_s}}, \text{ and } m_s = 100 \times m_t \quad (18)$$

where  $m_t = 0.01$  indicates the motion time of an agent, and  $m_s$  specifies the number of sudden motions. The parameter settings of the IDOA are as follows: the number of search agents is five, the search domain is [0–1], the dimension is equal to the extracted feature vectors, and the number of iteration is 20. The proposed IDOA selects 3476 feature vectors, which are used as input values in the DBN for classification.

### 3.5. Classification

After the selection of discriminative feature vectors, tool wear batch classification is carried out using DBN. The DBN is one of the effective deep learning models that consists of a number of Restricted Boltzmann Machines (RBMs) for data classification [41]. The learned activation unit of the first RBM is the input for succeeding RBMs in the stacks. In addition, the DBN is an undirected graphical technique, where the visible variables are linked to the hidden units using undirected weights. However, the DBNs are constrained; there is no connection within the visible and hidden variables. The probability distribution,  $p_d$ , of visible variables ( $m$ ), hidden units ( $n$ ) and energy function ( $E(m, n; \theta)$ ) is mathematically depicted in Equation (19):

$$-\log p_d(m, n) \alpha E(m, n; \theta) = - \sum_{i=1}^{|V|} \sum_{j=1}^{|Q|} w_{ij} m_i n_j - \sum_{i=1}^{|V|} b_i m_i - \sum_{j=1}^{|Q|} a_j n_j \quad (19)$$

where  $\theta = (w, b, a)$  indicates parameter set,  $b_i$  and  $a_j$  denote bias,  $w_{ij}$  represents the symmetric weight between the visible variables ( $m$ ), and  $\alpha$  represents learning rate. In the DBN model, the numbers of hidden and visible layers are considered as  $|Q|$  and  $|V|$ . The conditional probability distribution of visible variables ( $m$ ) and hidden units ( $n$ ) is defined in the Equations (20) and (21):

$$p_d(n_j | m; \theta) = \text{sigm} \sum_{i=1}^{|V|} w_{ij} m_i + a_j \quad (20)$$

$$p_d(m_i | n; \theta) = \text{sigm} \sum_{j=1}^{|Q|} w_{ij} n_j + b_j \quad (21)$$

where  $sigm(M) = \left(\frac{1}{1+e^{-m}}\right)$  represents sigmoid activation function, and the parameter  $\theta$  represents learned utilizing contrastive divergence. In the DBN classifier, the parameter  $\theta$  is obtained utilizing RBM, and it is defined by  $p_d(n|\theta)$  and  $p_d(m|n, \theta)$ . The probability of creating a visible variable is stated in Equation (22):

$$p_d(m) = \sum_n p_d(n|\theta)p_d(m|n, \theta) \tag{22}$$

The term  $p_d(m|n, \theta)$  is maintained after determining  $\theta$  from an RBM; then,  $p_d(n|\theta)$  is exchanged using consecutive RBMs that treat the previous RBM hidden layer as a visible value. The parameter settings of the DBN are as follows: the transfer function is sigmoid function, the dropout rate is 0.1, the batch size is 0.5, the learning rate is 0.01, maximum iteration is 100, and initial and final momentum are 0.5 and 0.9. Figure 4 shows the architecture of the IDOA-DBN process.

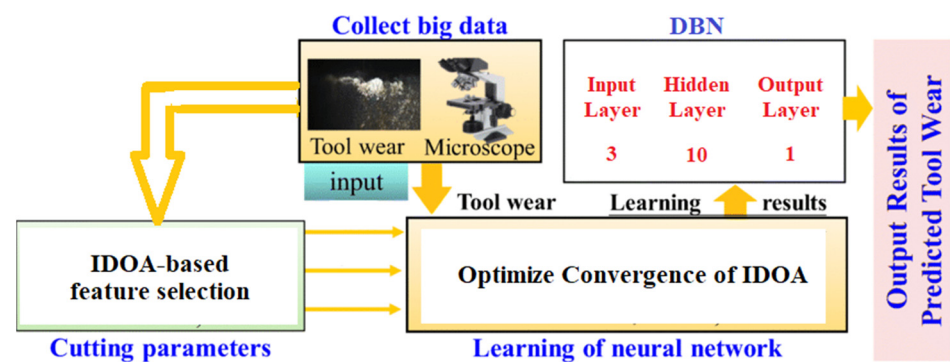


Figure 4. Architecture of the IDOA-DBN process.

#### 4. Experimental Results

In the present research, the cutting edges were divided into a number of subregions by the proposed method (wear patches, or WP). Each WP was defined using textural descriptors, and were categorized as worn or serviceable using a Deep Belief Network. Lastly, the number of WPs designated as worn determines whether a cutting edge is serviceable or worn. This study separated each training set image into patches that were later identified as being in a worn or serviceable zone. The objective of this research was to develop a classification model that evaluates individual patches and, based on its predictions, renders judgement on the degree of tool wear. Therefore, manual division prevents potentially flawed patch extractions that can result in the creation of suboptimal classifiers. After manually extracting the patches, the images were left with 896 patches, 466 of which were serviceable and 430 of which were worn. The proposed IDOA-DBN model’s effectiveness was validated using the MATLAB 2020a software tool on a system configuration with 16 GB random access memory, Linux operating system, and a 4 TB hard disk. The IDOA-DBN model’s efficacy was analyzed by implementing many classification techniques and feature optimization algorithms, and tested using performance metrics such as F-score, recall, precision, and accuracy. In this manuscript, the confusion matrix was used to calculate the performance metrics. Here, the serviceable class was a negative class and the worn class was a positive class. The confusion matrix is clearly depicted in Table 1.

Table 1. Confusion matrix.

Classes	Prediction Class	
	Serviceable	Worn
Serviceable	TN	FP
Worn	FN	TP

Precision, recall, accuracy, and F-score were the metrics used in this research to evaluate the methodology. The confusion matrix depicted in Figure 4 was created by allocating the worn class as the positive class, and the serviceable class as the negative class. The performance metric, the F-score, was determined as the harmonic mean of recall and precision, and it was estimated using Equation (23). Similarly, the recall was defined as the fraction of worn inserts, and it was computed utilizing Equation (24). In the tool wear batch classification, the performance metric, recall, played a vital role, since the cost of misclassifying a serviceable cutting edge was lower than the cost of misclassifying a worn cutting edge:

$$F - score = \frac{2TP}{FP + 2TP + FN} \times 100 \quad (23)$$

$$Recall = \frac{TP}{TP + FN} \times 100 \quad (24)$$

Similarly, the performance metric, precision, was defined as the fraction of inserts classified as worn that were actually worn. The accuracy was determined by the successful prediction for the total number of samples, where  $TP$ ,  $TN$ ,  $FP$ , and  $FN$  were defined as true positive, true negative, false positive, and false negative. Figure 5 shows the confusion matrix. The mathematical expressions of precision and accuracy are defined in Equations (25) and (26):

$$Precision = \frac{TP}{TP + FP} \times 100 \quad (25)$$

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \times 100 \quad (26)$$

**Confusion Matrix**

<b>Output Class</b>	<b>Serviceable</b>	273 47.3%	4 0.7%	98.6% 1.4%
	<b>Worn</b>	3 0.5%	297 51.5%	99.0% 1.0%
	<b>Over all</b>	98.9% 1.1%	98.7% 1.3%	98.8% 1.2%
		<b>Serviceable</b>	<b>Worn</b>	<b>Over all</b>
		<b>Target Class</b>		

**Figure 5.** Confusion matrix.

#### 4.1. Quantitative Evaluation

The proposed IDOA-DBN model's effectiveness was validated utilizing dissimilar feature optimization techniques, such as Grasshopper Optimization Algorithm (GOA), genetic algorithm, Particle Swarm Optimization (PSO) algorithm, DOA, and IDOA, by means of precision, accuracy, F-score, and recall. In this study, the performance analysis was conducted for different patch configurations, such as Full Edge Division (FED), Homogeneous Grid Division (HGD), Small Edge Division (SED), Half Edge Division (HED), and



Two-Band Division (TBD), respectively. By viewing Table 2, the IDOA with DBN classifier obtained efficient performance in tool wear classification compared to other existing optimizers, such as GOA, genetic algorithm, PSO, and DOA.

**Table 2.** Experimental results of various feature optimization techniques with the IDOA.

Patch Configuration	Optimizer	F-Score (%)	Recall (%)	Precision (%)	Accuracy (%)
FED	GOA	80.90	78.90	80.87	79.08
	Genetic	82.34	86.30	85.60	83.49
	PSO	87.90	88.56	88.70	88.60
	DOA	90.80	92.30	93.94	94.59
	IDOA	95.30	94.53	96.70	98.83
HGD	GOA	75.39	67.69	70.70	80.88
	Genetic	77.78	72.30	74.50	81.20
	PSO	85.49	75.60	78.90	85.60
	DOA	86.06	83.20	83.50	88.88
	IDOA	86.20	87.90	88.88	90.87
SED	GOA	77.50	70.90	68.90	87.30
	Genetic	75.40	78.78	80.82	83.40
	PSO	85.06	83.40	84.50	86.67
	DOA	89.80	90.76	89.88	91.20
	IDOA	93.20	92.03	94.50	96.78
HED	GOA	78.49	76.40	80	84.50
	Genetic	77	78.90	83.42	85.55
	PSO	87.74	88.36	82.20	89.28
	DOA	89.76	87.30	89.87	92.20
	IDOA	90.45	93.20	94.56	95.60
TBD	GOA	77.65	78.43	70.60	78.09
	Genetic	78.70	82.49	74.05	80.93
	PSO	84.50	85.60	85.60	86.70
	DOA	90.10	89.40	92.30	91.29
	IDOA	92.03	94.57	95.55	94.56

The IDOA with DBN classifier achieved an F-score of 95.30%, recall of 94.53%, precision of 96.70%, and accuracy of 98.83% for FED patch configuration. Regarding HGD patch configuration, the IDOA with DBN classifier attained an F-score of 86.20%, recall of 87.90%, precision of 88.88%, and accuracy of 90.87%. For the SED, HED, and TBD patch configurations, the proposed IDOA with DBN achieved 93.20%, 90.45%, and 92.03% F-scores; 92.03%, 93.20%, and 94.57% recall values; 94.50%, 94.56%, and 95.55% precision values; and 96.78%, 95.60%, and 94.56% classification accuracy.

AlexNet and ResNet models are capable of preserving the useful information in the network without overtrain or loss the features, while some of the recent CNN architecture pooling layers tend to lose useful features by overtraining the features. Furthermore, AlexNet and ResNet are suitable to handle large-scale data analysis. Therefore, in this study, these two models were preferred for the comparative analysis. In Table 3, it can be seen that the performance evaluation was conducted using various classification techniques, such as AlexNet, Autoencoder, ResNet-14, ResNet-18, and DBN, on five different patch configurations, namely FED, HGD, SED, HED, and TBD, by means of F-score, recall, precision, and accuracy. By viewing Table 3, the combination of DBN classifier with IDOA obtained maximum performance in tool wear classification as compared to other classifiers, such as AlexNet, Autoencoder, ResNet-14, and ResNet-18. Compared to other classification techniques, the DBN has two major advantages: reductions in both the over-smoothing problem and the training data fragmentation problem.

**Table 3.** Experimental results of various classification techniques with DBN.

Patch Configuration	Classifier	F-Score (%)	Recall (%)	Precision (%)	Accuracy (%)
FED	AlexNet	83.95	80.96	85.85	84.58
	Autoencoder	86.77	87.60	87.67	87.40
	ResNet-14	88.90	89.96	89.70	88.68
	ResNet-18	93.86	90.80	94.96	95.58
	DBN	95.30	94.53	96.70	98.83
HGD	AlexNet	78.50	77.60	76.70	83.90
	Autoencoder	79.78	78.90	77.57	84.20
	ResNet-14	84.40	79.93	79.98	85.50
	ResNet-18	85.48	84.78	86.70	87.87
	DBN	86.20	87.90	88.88	90.87
SED	AlexNet	76.70	74.90	76.98	88.30
	Autoencoder	78.90	76.78	82.80	89.40
	ResNet-14	86.06	85.40	88.50	89.69
	ResNet-18	88.80	88.70	93.84	92.22
	DBN	93.20	92.03	94.50	96.78
HED	AlexNet	79.40	77.78	80.90	85.50
	Autoencoder	81.20	78.98	85.49	86.75
	ResNet-14	88.78	84.86	86.26	89.29
	ResNet-18	88.76	89.30	90.88	93.20
	DBN	90.45	93.20	94.56	95.60
TBD	AlexNet	80.65	79.40	79.67	80.49
	Autoencoder	82.70	84.40	84.75	80.99
	ResNet-14	86.60	86.78	87.68	88.90
	ResNet-18	91.70	88.89	92.34	94.11
	DBN	92.03	94.57	95.55	94.56

#### 4.2. Comparative Evaluation

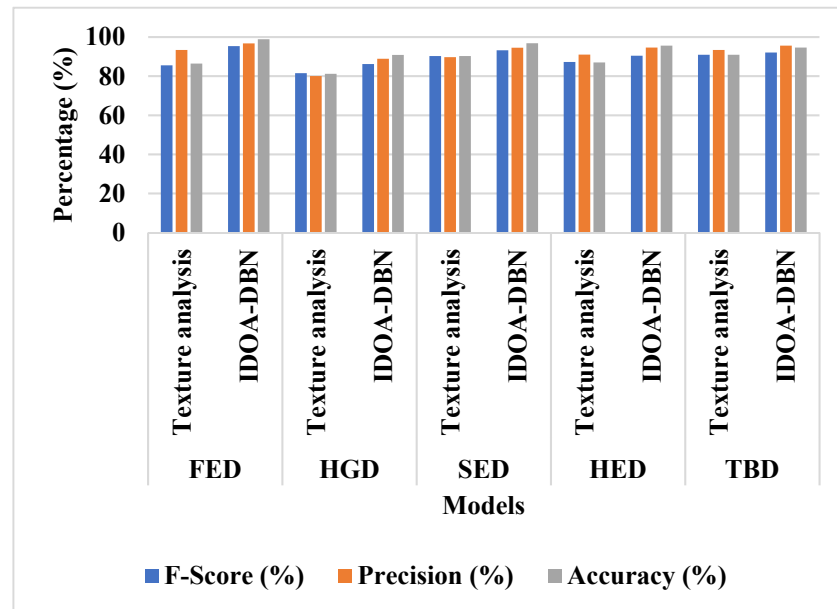
The comparative investigation between the proposed IDOA-DBN model and the existing image texture analysis model [23] is presented in Table 4. M.T. García-Ordás, et al. [23] used circular Hough transform technique, canny edge detector, and standard Hough transform technique for segmenting the cutting edges of acquired images. Furthermore, the texture feature extraction was accomplished using conventional, completed, and adaptive LBP descriptors. The extracted feature vectors were fed into the SVM classifier for classifying the cutting edges as serviceable or worn. The experimental evaluation showed that the proposed IDOA-DBN model attained efficient performance in tool wear batch classification related to the existing image texture analysis model [23] on five different patch configurations. The feature optimization utilizing IDOA is an integral part of this manuscript that significantly reduces the computational complexity and running time of the DBN by selecting the discriminative feature vectors. The IDOA-DBN model took 43 s to process the whole dataset, which is better than other deep learning models, and the complexity of the proposed model is linear when selecting the discriminative feature vectors. Table 4 shows the comparison results of various patch configurations. Figure 6 shows the graphical presentation of the various patch configurations.

The researchers tuned the network model's design and hyperparameters for the signal matrix data, leading to a high recognition accuracy. Furthermore, the suggested tool wear state technique's optimized neural network with double neurons is incompatible with all of these datasets. Based on the design structure provided in the literature, calculations were performed to discover a suitable system model, and the obtained accuracy results are presented in Table 5. The suggested IDOA-DBN approach delivers superior identification accuracy (98.83%) and a more compact network structure than existing DWF-CNN [28] techniques, which achieved 98.7%. Figure 7 shows the graphical illustration of accuracy with existing DWF-CNN [28] methods.

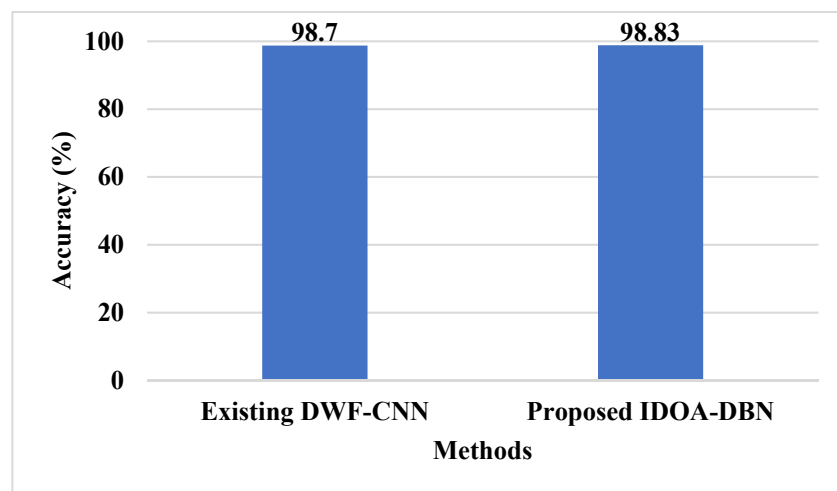


**Table 4.** Results of various patch configurations.

Patch Configurations	Model	F-Score (%)	Precision (%)	Accuracy (%)
FED	Texture analysis [23]	85.50	93.40	86.40
	IDOA-DBN	95.30	96.70	98.83
HGD	Texture analysis [23]	81.50	80	81.20
	IDOA-DBN	86.20	88.88	90.87
SED	Texture analysis [23]	90.30	89.70	90.30
	IDOA-DBN	93.20	94.50	96.78
HED	Texture analysis [23]	87.20	91	87
	IDOA-DBN	90.45	94.56	95.60
TBD	Texture analysis [23]	90.90	93.40	90.90
	IDOA-DBN	92.03	95.55	94.56



**Figure 6.** Graphical presentation of the various patch configurations with existing method.



**Figure 7.** Graphical illustration of accuracy with existing method.

**Table 5.** Comparative analysis of accuracy with existing method.

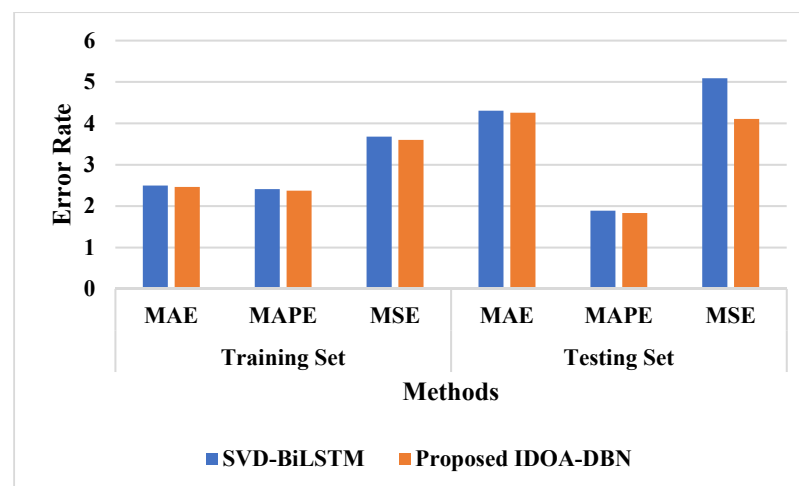
Method	Accuracy (%)
Existing DWF-CNN [28]	98.7
Proposed IDOA-DBN	98.83

The prediction impacts of BiLSTM methods based on SVD features and time domain features were evaluated to demonstrate the effectiveness of IDOA-DBN features in tool wear detection. SVD-BiLSTM [30] was utilized for experimental studies to further validate the functionality of the IDOA-DBN tool wear prediction system. Because all learning methods have the same architecture, only the type of network layers needed to be changed; the performance parameters were maintained, and all design variables were SVD features. Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE), and Mean Square Error (MSE) values were computed to generate time domain features of the same scale before the SVD features. Table 6 lists the outcomes of the prediction results in terms of the training and testing sets. The proposed IDOA-DBN model was more suitable in the error assessment of the training set, and computation was easier. Therefore, IDOA-DBN produced better training outcomes in the training set in the same time frame. Despite this, existing SVD-BiLSTM [30] lacks cell state and has inadequate long-term memory capacity, which is evident in the test set error values produced by SVD-BiLSTM [30].

**Table 6.** Comparative analysis of prediction results in training and testing set.

Method	Training Set			Testing Set		
	MAE	MAPE	MSE	MAE	MAPE	MSE
SVD-BiLSTM [30]	2.4948	2.4112	3.6782	4.3044	1.8896	5.0873
Proposed IDOA-DBN	2.4625	2.3729	3.5988	4.2553	1.8309	4.1032

In this study, IDOA-DBN was proposed and considered to be the appropriate design in all data and test sets, because the IDOA-DBN can memorize all data features. Consequently, in both the training set and the test set, IDOA-DBN is superior to SVD-BiLSTM [30]. The model's input data were constructed using a time step of five, which means that the cutting edge signals for each of the previous five sampling periods were matched with the tool wear value for the 5th sampling period. The first 265 sets of samples were used as the model training sample out of 315 total measured data, which were divided into 311 sets of sample data. The remaining 46 groupings of samples were employed as a model test sample. Figure 8 shows the graphical illustration of the prediction results.

**Figure 8.** Graphical illustration of prediction results with existing method.

## 5. Conclusions

In this manuscript, an IDOA-DBN model is implemented as an effective tool wear monitoring technique. Since Deep Belief Network (DBN) performs better than other models in terms of learning speed, and quickly converges to the best results, regardless of the size of the sample data, it was utilized to create tool wear prediction models. This increased the tool wear monitoring system's modeling efficiency and accuracy. Due to the manual adjustment of DBN parameters, prediction accuracy and efficiency were generally low. In order to reduce the computational complexity and operating time of the DBN at this point, the Improved Dragonfly Optimization Algorithm (IDOA) was proposed. The inclusion of the IDOA in the proposed model significantly diminished the computational complexity and running time of the classifier. As indicated in the resulting phase, the proposed IDOA-DBN model obtained superior performance values for tool wear monitoring compared to the existing deep learning models, such as AlexNet, Autoencoder, ResNet-14, and ResNet-18, on different patch configurations. The IDOA-DBN model attained a maximum classification accuracy of 98.83% on the patch configuration of full edge division. As a future extension, a hybrid feature selection algorithm should be included in the proposed model to further enhance the tool wear monitoring technique.

**Author Contributions:** The paper investigation, resources, data curation, writing—original draft preparation, writing—review and editing, and visualization were performed by L.G.D. The paper conceptualization, software, were conducted by R.K.P. and L.J.A.M. The validation and formal analysis, methodology, supervision, project administration, and funding acquisition of the version to be published were conducted by P.F.-G. and P.B.D. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Not applicable.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Hesser, D.F.; Markert, B. Tool wear monitoring of a retrofitted CNC milling machine using artificial neural networks. *Manuf. Lett.* **2019**, *19*, 1–4. [[CrossRef](#)]
2. Guo, J.; Li, A.; Zhang, R. Tool condition monitoring in milling process using multifractal detrended fluctuation analysis and support vector machine. *Int. J. Adv. Manuf. Technol.* **2020**, *110*, 1445–1456. [[CrossRef](#)]
3. Bergs, T.; Holst, C.; Gupta, P.; Augspurger, T. Digital image processing with deep learning for automated cutting tool wear detection. *Procedia Manuf.* **2020**, *48*, 947–958. [[CrossRef](#)]
4. Wu, X.; Liu, Y.; Zhou, X.X.; Mou, A. Automatic identification of tool wear based on convolutional neural network in face milling process. *Sensors* **2019**, *19*, 3817. [[CrossRef](#)] [[PubMed](#)]
5. Martínez-Arellano, G.; Terrazas, G.; Ratchev, S. Tool wear classification using time series imaging and deep learning. *Int. J. Adv. Manuf. Technol.* **2019**, *104*, 3647–3662. [[CrossRef](#)]
6. Peng, R.; Pang, H.; Jiang, H.; Hu, Y. Study of tool wear monitoring using machine vision. *Autom. Control Comput. Sci.* **2020**, *54*, 259–270. [[CrossRef](#)]
7. Qiao, H.; Wang, T.; Wang, P. A tool wear monitoring and prediction system based on multiscale deep learning models and fog computing. *Int. J. Adv. Manuf. Technol.* **2020**, *108*, 2367–2384. [[CrossRef](#)]
8. Stavropoulos, P.; Papacharalampopoulos, A.; Souflas, T. Indirect online tool wear monitoring and model-based identification of process-related signal. *Adv. Mech. Eng.* **2020**, *12*, 1687814020919209. [[CrossRef](#)]
9. Liu, T.; Zhu, K.; Wang, G. Micro-milling tool wear monitoring under variable cutting parameters and runout using fast cutting force coefficient identification method. *Int. J. Adv. Manuf. Technol.* **2020**, *111*, 3175–3188. [[CrossRef](#)]
10. Chen, N.; Hao, B.; Guo, Y.; Li, L.; Aqib Khan, M.; He, N. Research on tool wear monitoring in drilling process based on APSO-LS-SVM approach. *Int. J. Adv. Manuf. Technol.* **2020**, *108*, 2091–2101. [[CrossRef](#)]
11. Liu, T.; Zhu, K. A switching hidden semi-Markov model for degradation process and its application to time-varying tool wear monitoring. *IEEE Trans. Ind. Inform.* **2020**, *17*, 2621–2631. [[CrossRef](#)]

12. Marani, M.; Zeinali, M.; Kouam, J.; Songmene, V.; Mechefske, C.K. Prediction of cutting tool wear during a turning process using artificial intelligence techniques. *Int. J. Adv. Manuf. Technol.* **2020**, *111*, 505–515. [[CrossRef](#)]
13. Jamshidi, M.; Rimpault, X.; Balazinski, M.; Chatelain, J.F. Fractal analysis implementation for tool wear monitoring based on cutting force signals during CFRP/titanium stack machining. *Int. J. Adv. Manuf. Technol.* **2020**, *106*, 3859–3868. [[CrossRef](#)]
14. Zhang, X.; Han, C.; Luo, M.; Zhang, D. Tool wear monitoring for complex part milling based on deep learning. *Appl. Sci.* **2020**, *10*, 6916. [[CrossRef](#)]
15. García-Ordás, M.T.; Alegre-Gutiérrez, E.; González-Castro, V.; Alaiz-Rodríguez, R. Combining shape and contour features to improve tool wear monitoring in milling processes. *Int. J. Prod. Res.* **2018**, *56*, 3901–3913. [[CrossRef](#)]
16. Dou, J.; Xu, C.; Jiao, S.; Li, B.; Zhang, J.; Xu, X. An unsupervised online monitoring method for tool wear using a sparse auto-encoder. *Int. J. Adv. Manuf. Technol.* **2020**, *106*, 2493–2507. [[CrossRef](#)]
17. Antić, A.; Popović, B.; Krstanović, L.; Obradović, R.; Milošević, M. Novel texture-based descriptors for tool wear condition monitoring. *Mech. Syst. Signal Process.* **2018**, *98*, 1–15. [[CrossRef](#)]
18. Zhang, X.; Pan, T.; Ma, A.; Zhao, W. High efficiency orientated milling parameter optimization with tool wear monitoring in roughing operation. *Mech. Syst. Signal Process.* **2022**, *165*, 108394. [[CrossRef](#)]
19. Ong, P.; Lee, W.K.; Lau, R.J.H. Tool condition monitoring in CNC end milling using wavelet neural network based on machine vision. *Int. J. Adv. Manuf. Technol.* **2019**, *104*, 1369–1379. [[CrossRef](#)]
20. Kuntoğlu, M.; Aslan, A.; Yurievich Pimenov, D.; Ali Usca, Ü.; Salur, E.; Kumar Gupta, M.; Mikolajczyk, T.; Giasin, K.; Kapłonek, W.; Sharma, S. A review of indirect tool condition monitoring systems and decision-making methods in turning: Critical analysis and trends. *Sensors* **2020**, *21*, 108. [[CrossRef](#)]
21. Oo, H.; Wang, W.; Liu, Z. Tool wear monitoring system in belt grinding based on image-processing techniques. *Int. J. Adv. Manuf. Technol.* **2020**, *111*, 2215–2229. [[CrossRef](#)]
22. Li, G.; Wang, Y.; He, J.; Hao, Q.; Yang, H.; Wei, J. Tool wear state recognition based on gradient boosting decision tree and hybrid classification RBM. *Int. J. Adv. Manuf. Technol.* **2020**, *110*, 511–522. [[CrossRef](#)]
23. García-Ordás, M.T.; Alegre-Gutiérrez, E.; Alaiz-Rodríguez, R.; González-Castro, V. Tool wear monitoring using an online, automatic and low cost system based on local texture. *Mech. Syst. Signal Process.* **2018**, *112*, 98–112. [[CrossRef](#)]
24. Simon, G.D.; Deivanathan, R. Early detection of drilling tool wear by vibration data acquisition and classification. *Manuf. Lett.* **2019**, *21*, 60–65. [[CrossRef](#)]
25. Marani, M.; Zeinali, M.; Songmene, V.; Mechefske, C.K. Tool wear prediction in high-speed turning of a steel alloy using long short-term memory modelling. *Measurement* **2021**, *177*, 109329. [[CrossRef](#)]
26. Liu, X.; Liu, S.; Li, X.; Zhang, B.; Yue, C.; Liang, S.Y. Intelligent tool wear monitoring based on parallel residual and stacked bidirectional long short-term memory network. *J. Manuf. Syst.* **2021**, *60*, 608–619. [[CrossRef](#)]
27. Huang, Z.; Zhu, J.; Lei, J.; Li, X.; Tian, F. Tool wear predicting based on multi-domain feature fusion by deep convolutional neural network in milling operations. *J. Intell. Manuf.* **2020**, *31*, 953–966. [[CrossRef](#)]
28. Cao, X.C.; Chen, B.Q.; Yao, B.; He, W.P. Combining translation-invariant wavelet frames and convolutional neural network for intelligent tool wear state identification. *Comput. Ind.* **2019**, *106*, 71–84. [[CrossRef](#)]
29. Liao, X.; Zhou, G.; Zhang, Z.; Lu, J.; Ma, J. Tool wear state recognition based on GWO-SVM with feature selection of genetic algorithm. *Int. J. Adv. Manuf. Technol.* **2019**, *104*, 1051–1063. [[CrossRef](#)]
30. Wu, X.; Li, J.; Jin, Y.; Zheng, S. Modeling and analysis of tool wear prediction based on SVD and BiLSTM. *Int. J. Adv. Manuf. Technol.* **2020**, *106*, 4391–4399. [[CrossRef](#)]
31. Laddada, S.; Si-Chaib, M.O.; Benkedjough, T.; Draï, R. Tool wear condition monitoring based on wavelet transform and improved extreme learning machine. *Proc. Inst. Mech. Eng. Part C J. Mech. Eng. Sci.* **2020**, *234*, 1057–1068. [[CrossRef](#)]
32. Okokpujie, K.; Noma-Osaghae, E.; John, S.; Ajulibe, A. An improved iris segmentation technique using circular Hough transform. In *IT Convergence and Security*; Springer: Singapore, 2018; pp. 203–211.
33. Yang, Y.; Zhao, X.; Huang, M.; Wang, X.; Zhu, Q. Multispectral image-based germination detection of potato by using supervised multiple threshold segmentation model and Canny edge detector. *Comput. Electron. Agric.* **2021**, *182*, 106041. [[CrossRef](#)]
34. Jothi, A.; Jayaram, S.; Dubey, A.K. Intra-ocular lens defect detection using generalized Hough transform. In *Proceedings of the 6th International Conference on Reliability, Infocom Technologies and Optimization, Trends and Future Directions, ICRITO, Noida, India, 20–22 September 2017*; pp. 177–181.
35. Yuk, E.H.; Park, S.H.; Park, C.S.; Baek, J.G. Feature-learning-based printed circuit board inspection via speeded-up robust features and random forest. *Appl. Sci.* **2018**, *8*, 932. [[CrossRef](#)]
36. Islam, M.A.; Yousuf, M.S.I.; Billah, M.M. Automatic plant detection using HOG and LBP features with SVM. *Int. J. Comput. (IJC)* **2019**, *33*, 26–38.
37. Zhou, W.; Gao, S.; Zhang, L.; Lou, X. Histogram of oriented gradients feature extraction from raw Bayer pattern images. *IEEE Trans. Circuits Syst. II Express Briefs* **2020**, *67*, 946–950. [[CrossRef](#)]
38. Wang, J.S.; Ren, X.D. GLCM based extraction of flame image texture features and KPCA-GLVQ recognition method for rotary kiln combustion working conditions. *Int. J. Autom. Comput.* **2014**, *11*, 72–77. [[CrossRef](#)]
39. Bakheet, S.; Al-Hamadi, A.; Youssef, R. A fingerprint-based verification framework using Harris and SURF feature detection algorithms. *Appl. Sci.* **2022**, *12*, 2028. [[CrossRef](#)]

40. Mirjalili, S. Dragonfly algorithm: A new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems. *Neural Comput. Appl.* **2016**, *27*, 1053–1073. [[CrossRef](#)]
41. Yang, Y.; Zheng, K.; Wu, C.; Niu, X.; Yang, Y. Building an effective intrusion detection system using the modified density peak clustering algorithm and deep belief networks. *Appl. Sci.* **2019**, *9*, 238. [[CrossRef](#)]