

Comparison of machine learning algorithms for the automatic programming of computer numerical control machine

Neelima Sharma^a, V.K. Chawla^{a*} and N. Ram^a

^aIndira Gandhi Delhi Technical University for Women, India

CHRONICLE

ABSTRACT

Article history:

Received: May 18, 2018
Received in revised format: July 22, 2019
Accepted: September 3, 2019
Available online: September 11, 2019

Keywords:

Artificial Intelligence
CNC Programming
Machine Learning Algorithms
Deep Belief Network
Restricted Boltzmann Machine
Support Vector Machine

The computer numerical control (CNC) machines are chiefly used for the production of jobs with high accuracy and high speed. The CNC machining centers perform the machining operations according to the given program instructions which are commonly programmed by a CNC programmer. In this paper, a procedure to develop an automatic CNC program for machining of different types of holes by using different machine learning algorithms is developed. The machine learning algorithms namely support vector machine (SVM) and restricted boltzmann machine algorithm (RBM) with deep belief network (DBN) are used for the automatic development of CNC machining programs of different types of holes. Initially, the position and other parameters of machining operations are identified and thereafter the CNC machining program is developed by using the MATLAB application. The automatically developed CNC programs are tested on a CNC simulator. It is found that the application of RBM machine learning algorithm with DBN outperforms the SVM machine learning algorithm for the development of automatic CNC machining program for the machining of blind holes, through holes, counterbores and countersink operations.

© 2020 by the authors; licensee Growing Science, Canada.

1. Introduction

In the present era of technology, the infusion of machine learning algorithms into different machine tools has emerged as an innovative way to minimize the human involvement for different machining operations and also to improve accuracy and productivity in various machining operations. The use of machine learning algorithms into the machine tools promotes the development of artificial intelligence into the different machines by experienced-based learning (Sadjadi & Makui, 2002; Sadrabadi & Sadjadi, 2009; Moghaddam et al., 2012; Angra et al., 2018; Balic et al., 2006; Chanda et al., 2018; Deb et al., 2006; Chawla et al., 2017; Chawla et al., 2018a, 2018b, 2018c, 2018d, 2019a, 2019b, 2019c, 2019d, 2019e). Commonly the machine learning algorithm is provided with some data sets or machining programs which are used for training of the algorithm (Warwick, 2013; Russell & Norvig, 2016). After having sufficient

* Corresponding author.

E-mail address: vivekchawla@igdtuw.ac.in (V.K. Chawla)

training with some machining program, the algorithm uses its training and evolves to develop new machining program itself for different machining scenarios. Therefore, the use of machine learning algorithms for real-time machining operations can yield considerable reduction in human involvement and also increase accuracy and productivity in the machining operations. The CNC machining centers are advance machine tools which are commonly used for highly precise, accurate and fast machining operations. The CNC machining centers are capable of machining undercuts, angular holes, machining of angular surfaces and can also perform machining operations in any axial direction within the accuracy of 1 micron. These features of CNC machining centers offer high versatility, productivity, precision, and accuracy in complex machining operations and make them a very popular choice among the users. In order to use CNC machine tools for machining operations, machine programming is carried out on the basis of the job specifications and required jobs dimensions within the tolerances. Commonly the CNC programming is carried out by a CNC programmer which uses its skills, knowledge, and experience for the development of machining programs. The machine learning algorithms namely support vector machines (SVM) and RBM along with DBN can also be used for the automatic development of CNC machine programs. Initially, the machine learning algorithms are trained with sets of machining programs and tested for their accuracy and effectiveness, Thereafter the algorithms, by their self evolving and learning capabilities observed to develop machining programs by their own for different machining requirements of jobs automatically. Therefore, in this paper we demonstrate the use of machine learning algorithms namely support vector machines (SVM) and RBM along with DBN for the automatic development of CNC machining programs for different hole operations namely for the machining of blind holes, through holes, counterbores and countersink operations in a job also portrayed in figure1 and anatomy of counterbore holes and countersink holes is shown in Fig. 2 and Fig. 3 respectively.

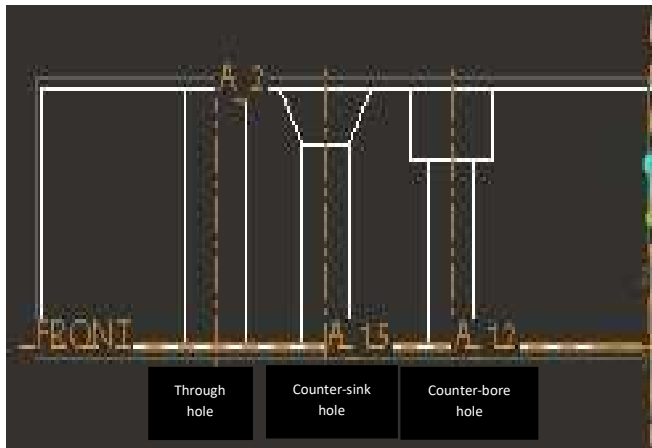


Fig.1. Drawing of different types of holes

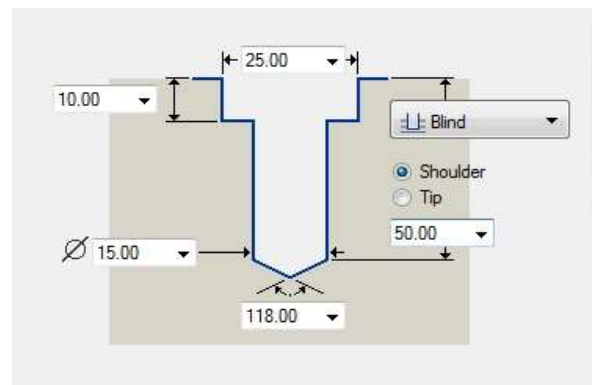


Fig.2. Anatomy of a counter-bore hole

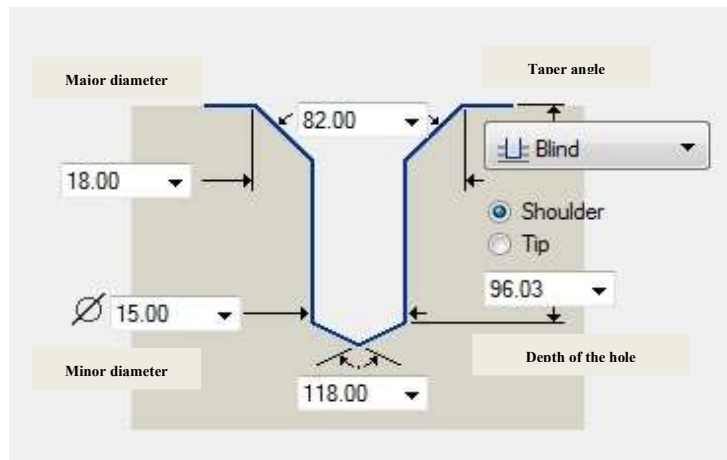


Fig.3. Anatomy of a counter-sink hole

2 Literature review

In the literature, the research for the development of automatic programs for CNC machine is continuously evolving. The application of different machine learning algorithms for the automatic learning and development of CNC programs is for different machining operations on different jobs is also observed. The artificial intelligence methods were used by the Preiss and Kaplansky (1985) for the automatic development of machining codes for a 3 axis milling machine tool. Authors used computer-aided drawing, cutting tool parameters, and type of material as an input for the development of the program. The programming is carried out on the Fortran language. The developed program found initially identified the required machine tool movements and thereafter, offers the part program. In order to search an optimum tool path for machining of holes in a job Onwubolu and Clerc (2004) applied a particle swarm optimization (PSO) algorithm. The authors considered traveling salesman problem and applied it into drilling operation so as to find an optimum path for machining of holes. The genetic algorithm (GA) for the development of an automatic turning program was developed by Balic et al. (2006). The algorithm initially identifies the dimensional specifications in the job and then it classifies the required machining specifications according to the required dimensional specifications. The GA also applies functions such as reproduction - crossover – mutation etc. to find optimum sequences for the cutting tools. A back-propagation based neural network is proposed by Deb et al. (2006) for the turning of jobs by computer-aided process planning. Similarly, a dimensional featured based computer aided process planning by using the fuzzy logic and neural network for the jobs with prismatic specifications are developed by Amaitik and Kiliç (2007). A GA based RSM approach is used by Kilickap et al. (2011) for optimizing the hole drilling parameters. Authors performed mathematical modeling for drilled hole surface finish by application of RSM. The drill operation parameters namely cutting speed, feed, and job specifications are given as input to the algorithm and model observed to yield good results. Tsagaris et al. (2012) reduced tool movement time in a CNC machine tool by using a GA based intelligent computer-aided design system for the machining of jobs. Authors used visual C++ to identify the various dimensional features from a DFX file and supplied as an input to the algorithm for optimizing the tool path movements. A numerical control program by using a feature extraction program is proposed by Rao et al. (2012).

Different hole drilling parameters namely cutting speed, drill diameter, job thickness, feed rate torque, etc. are used by Tiwari (2013) for calculation of optimum parameters by using a multi-objective genetic algorithm. The authors analyzed the interaction effect of different parameters by using the analysis of variance. Authors also performed a sensitivity analysis to find a change in output by carrying out changes in the weights assigned in the genetic algorithm. Authors observed that the accuracy of results to be highly dependent on the calculation and assignment of weights in the algorithm. The automatic development of CNC program for milling machining centers is carried out by the Klančnik et al. (2013). The dimensional specifications of the job are extracted by the algorithm and on the basis of extracted information, the algorithm develops milling machining center codes. Similarly, a particle swarm optimization algorithm (PSO) is used by Chitsaart et al. (2013) to develop an automatic CNC machining program for milling machining centers. Authors used MATLAB for the development of the extraction of features, their conversion, calculation, and development of automatic machining programs for the milling operations. Klančnik et al. (2016) proposed milling machine center programming by using the NSGAI algorithm. A model for CNC machine learning on the basis of mechanical contact and binary information on the basis of cutting tool physical contact is proposed by Balavignesh et al. (2017). Authors proposed a model on the basis of tool contact by generating a pulse of one during contact and pulse of zero if no contact is observed. In the literature, the artificial intelligence algorithms are found to be used for development and optimization of automatic CNC machining programs but use of machine learning algorithms namely restricted boltzmann machine (RBM) algorithm and support vector machines (SVM) is not observed in the literature. Therefore, in light of the aforesaid research gap in this paper, we propose the use of SVM and RBM algorithms for the development of automatic CNC machine programs for machining of holes and also compared their results.

3. Problem description

In this paper, the automatic machining program is developed by using RBM method and SVM method. The machine learning algorithms will learn from the one-time program which will be used as input for the training of the algorithms so as to develop CNC machining program automatically for any new machining operations. The objective is to propose the application of machine learning algorithms namely RBM method and SVM method to find the hole position, hole dimensions and optimum tool movement for the machining of holes in the job. Thereafter, results yield of SVM and RBM are compared with others.

3.1 Assumptions

1. The center point of the job is also its zero points.
2. The absolute programming system is used.
3. All dimensions of the job are in millimeters unit.
4. The feed rate, cutting speed, cutting tool material, job material, and other machining parameters are known.
5. No time loss is considered in a job setting and tool setting.
6. The Fanuc based programming system is used.
7. Job dimension has a dimensional range of 50X50X5 mm to 2000X2000X200 mm.
8. Cutting tool diameter has a diameter range of 2 mm to 50 mm.
9. The CNC machine tool spindle speed has a range of 0 to 4000 rpm.

3.2 General procedure

General steps for the development of automatic CNC program by using machine learning algorithms:

1. Development of a CAD file according to the job dimensions.
2. Input CAD file to image processing algorithm so as to read and extract the required information from the CAD file.
3. Training and learning of machine learning algorithms
4. Use machine learning algorithms (RBM and SVM) to search the exact position and dimensions of holes in the job.
5. Calculation by algorithm and development of CNC program.
6. Simulation runs of CNC program by using the CNC simulator.

4. Methodology

4.1 Restricted boltzmann machine algorithm (RBM) with Deep belief network (DBN)

The restricted boltzmann machine algorithm (RBM) is considered as a graphical model having an energy function which takes the negative energy to convert the graphical model into the probability distribution. A visible layer (S) is used for defining the probability distribution in the algorithm. Thereafter, the latent variables in the form of the hidden layer (T) are used to define the energy function (E_n) also written as

$$E_n(x, y) = -y^T Gx - d^T x - b^T y \quad (1)$$

$$- \sum_j \sum_k G_{j,k} y_j x_k - \sum_k d_k x_k - \sum_j b_j y_j \quad (2)$$

where,

x = binary unit of the visible layer (S)

y = binary unit of the hidden layer (T)

d = biases in the visible layer (S)

\mathbf{b} = biases in the hidden layer (T)

$$p(x, y) = \exp(-E_n(x, y)) / Q \quad (3)$$

$$p(y | x) = \prod_j p(y_j | x) \quad (4)$$

$$p(y_j = 1 | x) = \frac{1}{1 + \exp(-(b_j + G_j x))} \quad (5)$$

$$\lambda(b_j + G_j x) \quad (6)$$

$$p(x | y) = \prod_k p(x_k | y) \quad (7)$$

$$\lambda(d_k + y^T G_k) \quad (8)$$

$$p(x_k = 1 | y) = \frac{1}{1 + \exp(-(d_k + y^T G_k))} \quad (9)$$

The probability distribution over x and y is given by as

$$p(x) = \sum_{y \in \{0,1\}^Y} p(x, y) = \sum_{y \in \{0,1\}^Y} \exp(E_n(x, y)) / Q \quad (10)$$

$$= \exp\left(d^T x + \sum_{j=1}^Y \log(1 + \exp(b_j + G_j x))\right) \quad (11)$$

$$= \exp(-F(x)) / Q \quad (12)$$

The stochastic gradient descent is given by equation (13)

$$\frac{\partial -\log p(x^{(t)})}{\partial \theta} = E_y \left[\frac{\partial E(x^{(t)}, y)}{\partial \theta} \Big| x^{(t)} \right] - E_{x,y} \left[\frac{\partial E(x, y)}{\partial \theta} \right] \quad (13)$$

4.2 Restricted boltzmann machines training

The training of restricted boltzmann machines is performed in two steps namely Gibbs sampling and contrastive divergence step. Both steps are discussed below.

- 1. Gibbs sampling:** Gibbs sampling is carried out as the first step of training. An input vector V uses the $p(H|V)$ for prediction of the hidden values for H . After yield of the hidden values the $p(V|H)$ is used for the prediction of the new input values for input vector V by using Eq. (14) and (15). Thereafter, the process is repeated for the c number of iterations for obtaining the values for the input vector V .

$$p(V_i = 1 | h) = \frac{1}{1 + e^{-(f_i + G_i H_j)}} = \theta(f_i + \sum H_j G_{ij}) \quad (14)$$

$$p(H_i = 1 | V) = \frac{1}{1 + e^{-(b_i + G_i V_j)}} = \sigma(b_j + \sum V_j G_{ij}) \quad (15)$$

- 2. Contrastive divergence step:** The revised weight matrix (G) is obtained in the second step, i.e. from the contrastive divergence step Agarwalla et al. (2016). In this step, the probabilities for the hidden values of H_0 and H_k are calculated by using V_0 and V_k from equation number (16) and (17).

$$p(V_i = 1 | H) = \frac{1}{1 + e^{-(f_i + G_i H_j)}} = \theta(f_i + \sum H_j G_{ij}) \quad (16)$$

Revised weight equation:

$$\Delta G = V_0 \otimes P(H_0 | V_0) - V_K \otimes P(H_K | V_K) - V_K \quad (17)$$

The new values of weights are computed with the gradient from Eq. (18) :

$$G_{new} = G_{old} + \Delta G \quad (18)$$

The RBM algorithm can identify and extract specific data features. The RBM algorithm used with DBN can remove the gradient issues effectively. This reduces the training time of RBM significantly. The graphic process units are specialized tools which are used in deep learning so as to reduce the time required in the training process of the machine learning algorithms. The use of DBN with RBM offers an advantage that every RBM layer considers and learns all input data at a time and thereafter, it works as one unit. The DBN works effectively by using the small data sets which are further used in for training and learning of the real-time applications. The application of the RBM algorithm, when used with the DBN, found to be highly precise, accurate and effective in comparison to the other methods. The DBN training flow chart with RBM and the Automated CNC programming by using RBM with DBN is shown in figure 4 and figure 5 respectively.

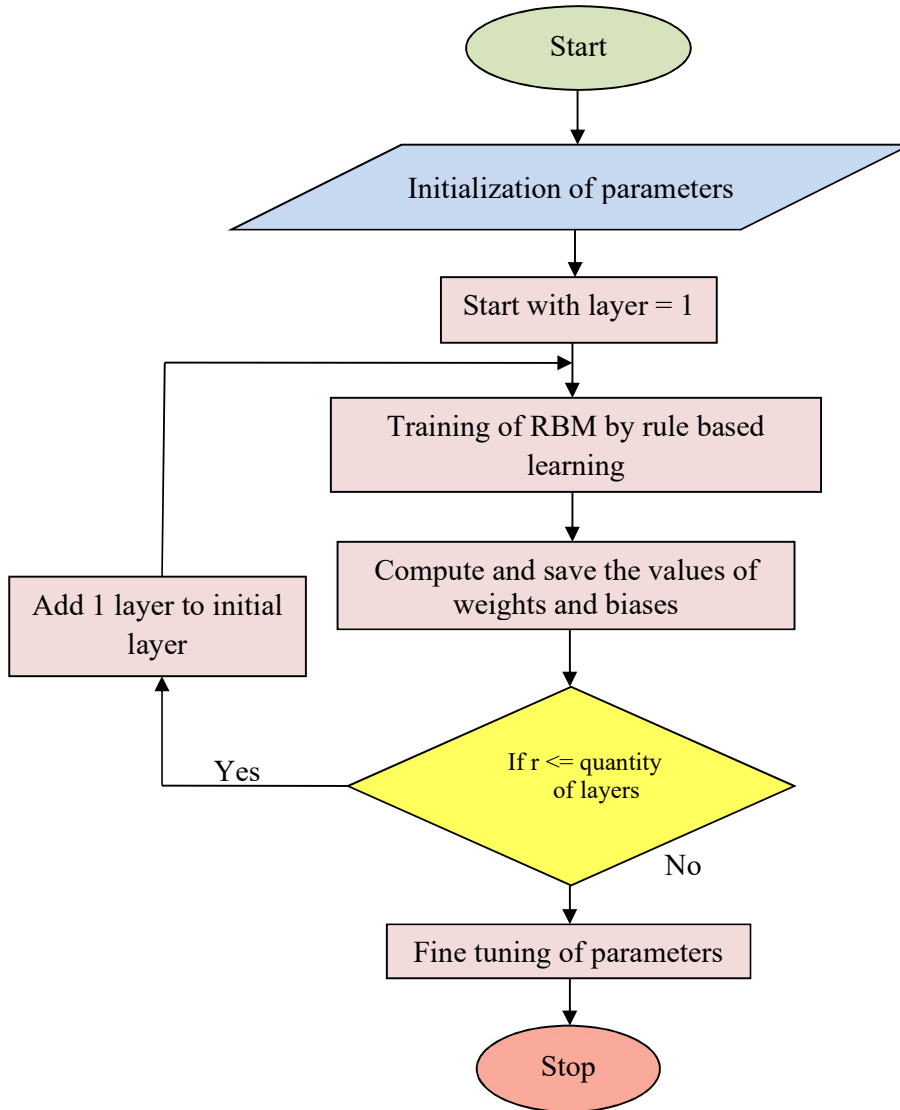


Fig. 4. The DBN training flow chart with RBM

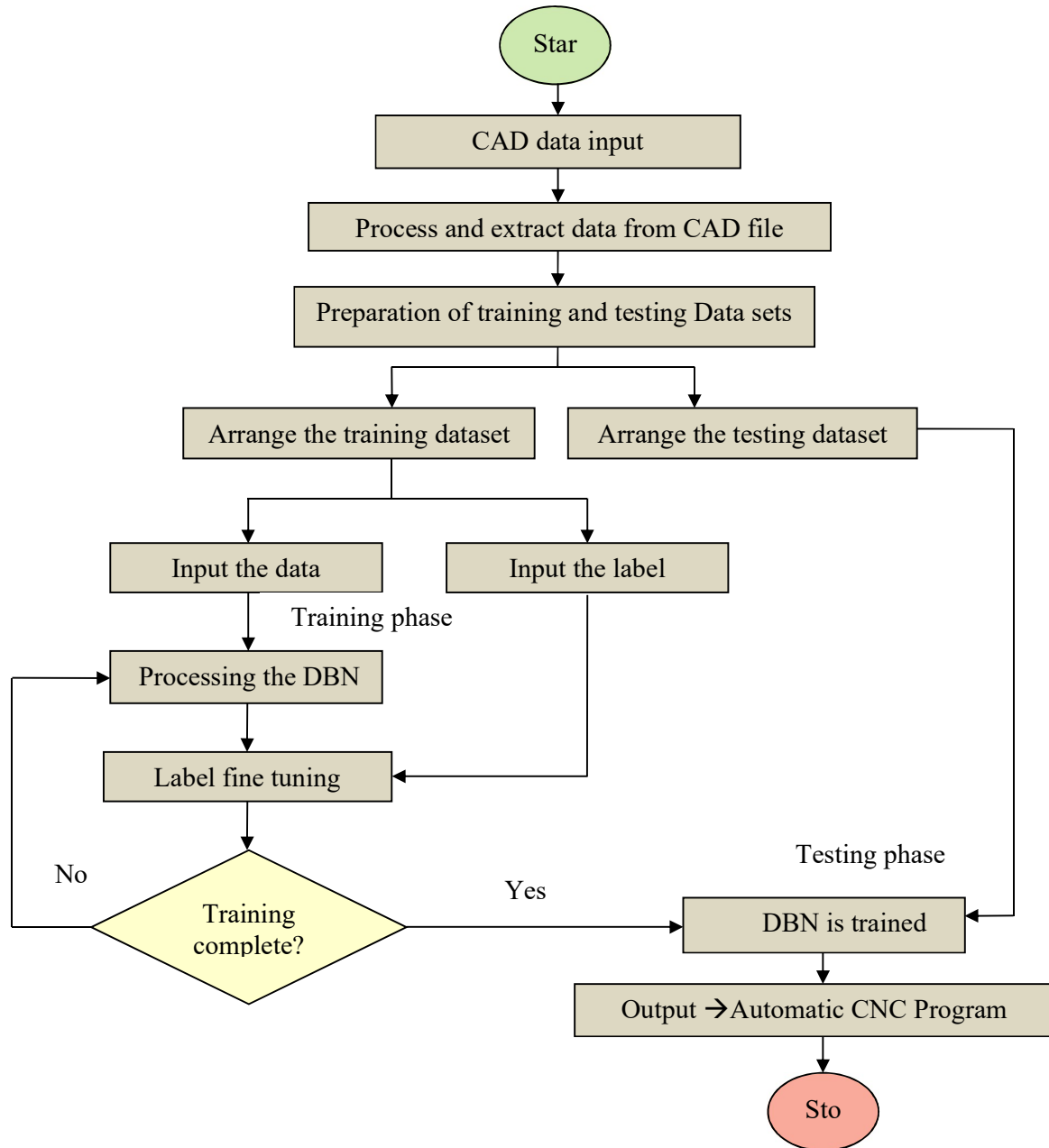


Fig. 5. Automated CNC programming by using RBM with DBN

4.3 Support vector machines

The Support vector machine (SVM) which can be defined as a supervised machine learning models with learning algorithms which can be used for both classification analysis and regression analysis. The Supervised learning can be explained as learning from the examples; training set of data is given which is considered as an example for the defined classes. The classification finds the explanation of each defined class and formulated explanation is used to forecast the class of earlier hidden objects. On the other hand, unsupervised learning can be explained is as learning from examination and invention. In unsupervised learning, a training set of the classes is absent. The system examines the defined set of data to study the similar budding out of the subsets of the data. The Support vector machines are core machine learning technique and are increasingly become useful in data mining. SVM has been applied to various tasks such as text classification, object recognition and so on. It is observed that SVM typically used in sorting

problems. Support vector machines majorly concentrate only on the points that are the most difficult to tell apart, whereas other classifiers pay attention to all of the points. The Support vector machine algorithm evolved as one of the significant machine learning algorithms. In the learning process, the unit layer provides a simpler and effective learning algorithm. However, the unit layer works efficiently only for linear decision input space. On the other hand, more than one layer of learning is highly expressive and can also work with non-linear functions. However, the learning and training in multilayer are observed to be difficult in comparison to the unit layer learning. This difficulty is overcome by the support vector machine (SVM) algorithm also known as kernel machines algorithm. This applies an efficient learning and training algorithm and can also work with highly complex and nonlinear functions etc. The SVMs search for an optimum linear separator with the largest margin in between. The Kernel function in the SVM algorithm can be of linear, polynomial, radial basis (RBF) and sigmoid type.

The SVM implementation steps and equations are given below:

1. The SVM algorithm will generate, classify and select the hyper-planes so as to make separate classes of them.
2. The classification and selection of a specific hyper-plane are based on the maximum distance of margin, between margin line and the nearest data points. The margin distance, which separates the classes of data-sets must be maximized. The classes of data sets also referred to as support vectors, due to their support in the determination of the decision boundary. The optimum or maximal hyper-plane separates the classes effectively and ensures high accuracy in classifications of the data sets.
3. In case the data sets are not separated linearly at the input stage, then some work on data or data conversion is carried out to convert data sets from initial input space to the high dimensional space. For this, a kernel function can be used so as to convert original space to higher dimensional feature space. From high dimensional space, the linear classification and separation of data become easy and fitting of margin or decision line can be carried out conveniently so as to segregate the data-sets and predict them with the required accuracy.

The norm of support vector a is calculated by using the Euclidean formula

$$a = (a_1, a_2, \dots, a_n) \quad (19)$$

$$\|a\| = \sqrt{a_1^2 + a_2^2 + \dots + a_n^2} \quad (20)$$

The direction of the support vector $a = (a_1, a_2)$ is written as v

$$v = \left(\frac{a_1}{\|a\|}, \frac{a_2}{\|a\|} \right) \quad (21)$$

The direction of a vector can be written in terms of angles also, as mentioned.

$$v = (\cos(\alpha), \cos(\beta)) \quad (22)$$

Hyperplane equation for linear separable data line function is given by $m = a.n + c$

$$a.n_1 - n_2 + c = 0 \quad (23)$$

For $n = (n_1, n_2)$ and $v = (a, -1)$

$$v.n + c = 0 \quad (24)$$

The Eq. (24) can be used for any number of dimensions. The Hypothesis function hp is given by

$$hp(n_i) = \begin{cases} +1 \rightarrow \text{if } v.n + c \geq 0 \\ -1 \rightarrow \text{if } v.n + c \leq 0 \end{cases} \quad (25)$$

In case the training data points are found to be positioned above of the separating hyper-plane then those will be classified as +1, and if the points are observed below separating hyper-plane then those will be classified as -1. The main work in the SVM algorithm is to identify and separate the optimum hyper-plane out of the available hyper-planes, and further classify the data on the basis of the data accuracy. Find the value of v and c of the optimum hyper-plane.

$$v - \sum_{i=1}^x \lambda_i m_i n_i = 0 \quad (26)$$

where λ is the Lagrangian multiplier

$$v = \sum_{i=1}^x \lambda_i m_i n_i \tag{27}$$

$$m_i(a.n^* + c) - 1 = 0 \tag{28}$$

Multiplying by m_i on R.H.S. and L.H.S.

$$c = m_i - a.n^* \tag{29}$$

where w is the number of support vectors

$$c = \frac{1}{w} \sum_{i=0}^w (m_i - a.n) \tag{30}$$

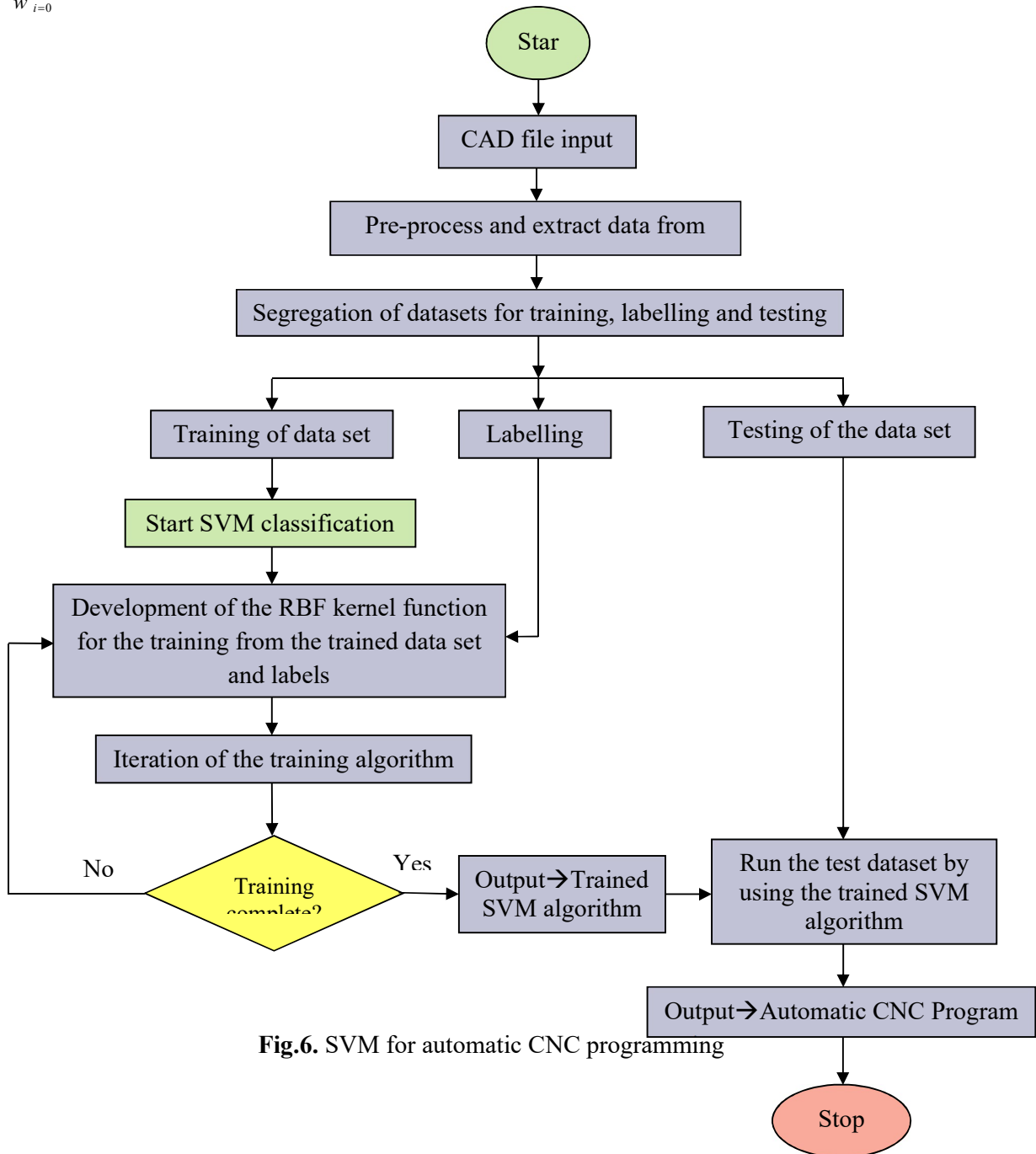


Fig.6. SVM for automatic CNC programming

5. Experiment and Results

The CAD model for machining of counter-bore and counter-sink holes is portrayed in Fig. 7 and Fig., 8 respectively. The algorithm search for dimension and position of holes that to be machined and yield an automatic program for CNC machining on the job. In order to yield the automatic program for CNC machining, the algorithm is simulated on the MATLAB application.

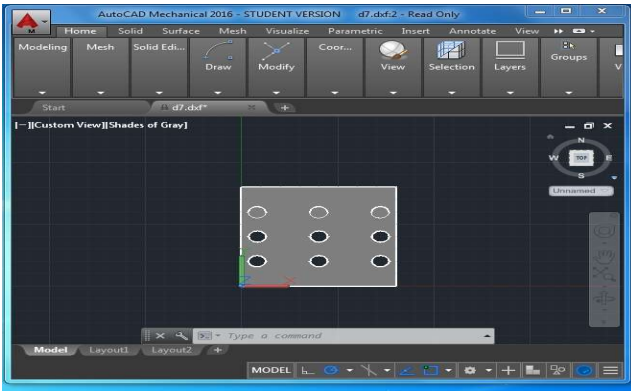


Fig.7.CAD model (holes)

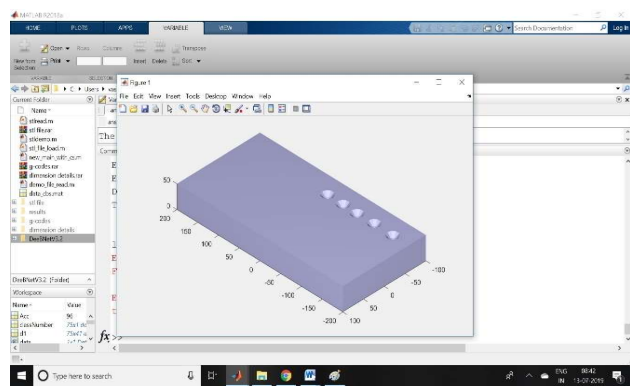


Fig.8.CAD model (countersinking)

This CAD file in .dxf extension is imported wherein features of the CAD model are recognized by the iteration of the algorithm in the MATLAB application. The DXF files are then trained with DBN in MATLAB application thereafter, the CNC machining program is developed automatically for the CAD file.

```

Command Window
Loading training dxf files.....!!!!!!
files loaded successfully..!

Loding training goode files.....!!!!
files loaded successfully..!!!!

DBN training started.....!!!!!!

***** RBM 1 ***** 60-50
epoch number:1 performance:0.0692796 remained RBM training time:7.3017
epoch number:2 performance:0.0617985 remained RBM training time:1.01126
epoch number:3 performance:0.053936 remained RBM training time:0.719631
epoch number:4 performance:0.0447646 remained RBM training time:0.862995
epoch number:5 performance:0.0362947 remained RBM training time:0.790196
epoch number:6 performance:0.0286657 remained RBM training time:0.82219
epoch number:7 performance:0.0235181 remained RBM training time:0.967801
epoch number:8 performance:0.021001 remained RBM training time:0.826299
epoch number:9 performance:0.0198305 remained RBM training time:1.12404
epoch number:10 performance:0.0192191 remained RBM training time:0.793692
    
```

Fig.9. Screenshots of execution of code in MATLAB (holes)

```

Command Window
epoch number:39 performance:0.28 remained RBM training time:0.276602
epoch number:40 performance:0.296667 remained RBM training time:0.20589
epoch number:41 performance:0.24 remained RBM training time:0.416992
epoch number:42 performance:0.29 remained RBM training time:0.227721
epoch number:43 performance:0.303333 remained RBM training time:0.216675
epoch number:44 performance:0.3 remained RBM training time:0.160576
epoch number:45 performance:0.286667 remained RBM training time:0.107119
epoch number:46 performance:0.283333 remained RBM training time:0.131124
epoch number:47 performance:0.286667 remained RBM training time:0.0604929
epoch number:48 performance:0.296667 remained RBM training time:0.0545601
epoch number:49 performance:0.273333 remained RBM training time:0.0196481
epoch number:50 performance:0.286667 remained RBM training time:0
Elapsed time is 5.423477 seconds.
Elapsed time is 6.905130 seconds.
DBN training completed..!!!!
Training Accuracy is
99.9667
    
```

Fig.10.Screenshots of execution of code in MATLAB (counter-boring)

```

Current Folder: ans
The variable ans does not exist.

Command Window
epoch number:43 performance:0.8 remained RBM training time:0.0123036
epoch number:44 performance:0.8 remained RBM training time:0.0166785
epoch number:45 performance:0.8 remained RBM training time:0.0528751
epoch number:46 performance:0.8 remained RBM training time:0.0442228
epoch number:47 performance:0.8 remained RBM training time:0.0113491
epoch number:48 performance:0.8 remained RBM training time:0.0214913
epoch number:49 performance:0.8 remained RBM training time:0.00317319
epoch number:50 performance:0.8 remained RBM training time:0
Elapsed time is 1.306761 seconds.
Elapsed time is 11.059697 seconds.
DBN training completed..!!!!
Training Accuracy is
96
    
```

Fig.11. Screenshots of execution of code in MATLAB (counter-sinking)

```

Current Folder: Command Window
G21 G94 G91
G54
G28 X0. Y0. Z0.
M06 T1
M03 S1200
G90
G43 H02
G00 X0. Y0. Z100.
G73 G99 X20. Z-30. Q3 R2. F50.
X50.
X80.
X110.
G98 X140.
G73 G99 X20. Z-30. Q3 R2. F50.
X50.
X80.
    
```

Fig.12. Screenshots of Automatic program for CNC machine

The performance and estimated time required for the automatic development of CNC machining program in the MATLAB application are shown in Fig. 9, Fig. 10 and Fig. 11. The training epoch number is kept for 200 iterations. The simulation experiments were carried out on an Intel(R) Core (TM) i5 processor. The yield of an automatic program for the CNC machine is also shown in Fig. 12.

5.1 Comparison between the output results of RBM and SVM

A comparison of the results for automatic programming of CNC machine for machining of work-piece by using the machine learning algorithms namely SVM and RBM with DBN has been carried out. In order to yield an automatic program for the CNC machine on work-piece by SVM algorithm, the selection of decision boundary is observed to be one of the significant criteria. The Margin is also an important parameter which separates and classify the two types of datasets with the required accuracy. The kernel function used in SVM also plays a significant role in the performance of SVM. The SVM has been observed to be less effective if numbers of features are more than the number of samples and then it lacks in the appropriate fitting of the kernel function. The application of SVM is observed to be less effective for the large data sets as it takes more time for training of algorithm. The RBM algorithm with deep belief architecture observed to yield better results in comparison the SVM for a yield of automatic programming for the CNC machine centers. The RBM algorithm used with DBN algorithm offers high accuracy in the training of algorithm in comparison to the SVM algorithm. However, the SVM algorithm found to be more suitable for the development of training and learning of small data-sets. On the other hand, the RBM with DBN offers good accuracy and better classification in the training of the large data-sets. The RBM with DBN is also capable in the fine-tuning that leading to significantly better results in the classification. The comparison between SVM and RBM is also mentioned in Table 1.

Table1

Comparison of the SVM and RBM

S. No.	Restricted Boltzmann Machines (RBM)	Support Vector Machines (SVM)
1	RBM is an unsupervised learning model	SVM is a supervised learning model
2	The RBM minimize the loss function by using back-propagation for classification.	The SVM maximize the marginal width for classification.
3	RBM can be fine-tuned by using the back-propagation.	SVM can't be fine-tuned
4	RBM is most suitable for large data sets.	SVM is suitable for small data sets.
5	The weights are used for classification	Partial derivatives and the Lagrangian multiplier are used for sampling the data.

From the results, it is found that the RBM with DBN yields better results for the development of automatic programming of CNC machine in comparison to the SVM algorithm. The training and learning of RBM with DBN is carried out with 96.96% of accuracy. On the other hand, the training and learning of SVM are carried out with 89.44 % of accuracy and 10.56% of error. The results are also mentioned in Table 2.

Table2

Comparison of the SVM and deep learning results for machining of holes in terms of accuracy and error

S. No.	Methodology	Accuracy (%)	Error (%)
1	Deep Learning	96.9667	3.033
2	Support Vector Machine	89.442	10.558

6. Conclusion

In this work, comparison of machine learning algorithms namely SVM and RBM with DBN for the development of the automatic program for CNC machining centers is carried out. Different operations on jobs were considered while applying the proposed methods were applied for the development of an

automatic program for CNC machines. The only CAD file was given as input and output was yield in the form of a CNC program. The results were validated by comparing it with data extracted from the CAD model and run the auto-generated CNC program file of the respective CAD drawing file on CNC TRAIN simulation software. The use of RBM with DBN was found to be more efficient in comparison to the SVM. The future research work can be extended to use different machine learning algorithms for the development of automatic CNC programs for different machining centers such as vertical machining centers or horizontal machining centers and compare their performances under different experimental conditions. The effectiveness of machine learning algorithms can also be gauged in the context of industry 4.0.

Acknowledgement

The authors would like to thank the anonymous referees for constructive comments on earlier version of this paper.

References

- Agarwalla, N., Panda, D., & Modi, M. K. (2016). Deep learning using restricted boltzmann machines. *International Journal of Computational Intelligence and Information Security*, 7(3), 1552-1556.
- Amaitik, S. M., & Kiliç, S. E. (2007). An intelligent process planning system for prismatic parts using STEP features. *The International Journal of Advanced Manufacturing Technology*, 31(9-10), 978-993.
- Angra, S., Chanda, A., & Chawla, V. (2018). Comparison and evaluation of job selection dispatching rules for integrated scheduling of multi-load automatic guided vehicles serving in variable-sized flexible manufacturing system layouts: A simulation study. *Management Science Letters*, 8(4), 187-200.
- Balavignesh G, Kumaresan P, Kavitha.B.R, Ramya Govindaraj, Venkatesan.S (2017). Computer numerical control machine based on machine learning. *International Journal of Pure and Applied Mathematics*, 116(24), 521-527.
- Balic.J., Kovacic. M., & Vaupotic.B., (2006). Intelligent programming of CNC turning operations using a genetic algorithm. *Journal of intelligent manufacturing*, 7(3), 331-340.
- Chanda, A., Angra, S., & Chawla, V. (2018). A modified memetic particle swarm optimization algorithm for sustainable multi-objective scheduling of automatic guided vehicles in a flexible manufacturing system. *International Journal of Computer-Aided Manufacturing*, 4(1), 33-47.
- Chawla, V. K., Chanda, A. K., & Angra, S. (2017). Evaluation of Dispatching Rules for Integrated Scheduling of AGVs in FMS. *In National Conference on Recent Advances in Mechanical Engineering*.
- Chawla, V. K., Chanda, A. K., Angra, S. & Rani, S., (2018a). Simultaneous dispatching and scheduling of multi-load AGVs in FMS-A simulation study. *Materials Today: Proceedings*, 5(11), 25358-25367.
- Chawla, V. K., Chanda, A. K., & Angra, S. (2018b). Multi-load AGVs scheduling by application of modified memetic particle swarm optimization algorithm. *Journal of the Brazilian Society of Mechanical Sciences and Engineering*, 40(9), 436.
- Chawla, V., Chanda, A., & Angra, S. (2018c). Sustainable multi-objective scheduling for automatic guided vehicle and flexible manufacturing system by a grey wolf optimization algorithm. *International Journal of Data and Network Science*, 2(1), 27-40.
- Chawla, V., Chanda, A., Angra, S., & Chawla, G. (2018d). The sustainable project management: A review and future possibilities. *Journal of Project Management*, 3(3), 157-170.
- Chawla, V. K., Chanda, A. K., & Angra, S. (2019a). A clonal selection algorithm for minimizing distance travel and back tracking of automatic guided vehicles in flexible manufacturing system. *Journal of The Institution of Engineers (India): Series C*, 100(3), 401-410.

- Chawla, V., Chanda, A., & Angra, S. (2019b). The scheduling of automatic guided vehicles for the workload balancing and travel time minimization in the flexible manufacturing system by the nature-inspired algorithm. *Journal of Project Management*, 4(1), 19-30.
- Chawla, V., Chanda, A., & Angra, S. (2019c). Material handling robots fleet size optimization by a heuristic. *Journal of Project Management*, 4(3), 177-184.
- Chawla, V. K., Chanda, A. K., & Angra, S. (2019d). Simultaneous workload balancing and travel time minimization of automatic guided vehicles. In *Journal of Physics: Conference Series*, 1240(1), 012001. IOP Publishing.
- Chawla, V. K., Chanda, A. K., Angra, S., & Rani, S. (2019e). Effect of nature-inspired algorithms and hybrid dispatching rules on the performance of automatic guided vehicles in the flexible manufacturing system. *Journal of the Brazilian Society of Mechanical Sciences and Engineering*, 41(10), 391.
- Chitsaart.C., Rianmora, S., Rattana-Areeyagon, M., & Namjaiprasert, W. (2013). Automatic generating CNC-code for milling machine. *International Journal of Mechanical, Aerospace, Industrial and Mechatronics Engineering*, 7, 1519-1525.
- Deb, S., Ghosh, K., & Paul, S. (2006). A neural network-based methodology for machining operations selection in computer-aided process planning for rotationally symmetrical parts. *Journal of Intelligent Manufacturing*, 17(5), 557-569.
- Kilickap, E., Huseyinoglu, M., & Yardimeden, A. (2011). Optimization of drilling parameters on surface roughness in the drilling of AISI 1045 using response surface methodology and genetic algorithm. *The International Journal of Advanced Manufacturing Technology*, 52(1-4), 79-88.
- Klancnik.S., Brezocnik, M., Balic, J., & Karabegovic.I. (2013). Programming of CNC milling machines using particle swarm optimization. *Materials and Manufacturing Processes*, 28(7), 811-815.
- Klancnik, S., Brezocnik, M., & Balic, J. (2016). Intelligent CAD/CAM system for programming of CNC machine tools. *International Journal of Simulation & Modelling*, 15(1), 109-120.
- Moghaddam, B. F., Ruiz, R., & Sadjadi, S. J. (2012). Vehicle routing problem with uncertain demands: An advanced particle swarm algorithm. *Computers & Industrial Engineering*, 62(1), 306-317.
- Onwubolu, G. C., & Clerc, M. (2004). The optimal path for automated drilling operations by a new heuristic approach using particle swarm optimization. *International Journal of Production Research*, 42(3), 473-491.
- Preiss, K., & Kaplansky, E. (1985). Automated part programming for CNC milling by artificial intelligence techniques. *Journal of Manufacturing Systems*, 4(1), 51-63.
- Rao, S. S., Satyanarayana, B., & Sarcar, M. M. M. (2012). Automated generation of NC part programs for turned parts based on 2-D drawing image files. *International Journal of Production Research*, 50(12), 3470-3485.
- Russell, S. J., & Norvig, P. (2016). *Artificial intelligence: a modern approach*. Malaysia; Pearson Education Limited.
- Sadjadi, S. J., & Makui, A. (2002). An algorithm to compute the complexity of a static production planning (RESEARCH NOTE). *International Journal of Engineering-Transactions A: Basics*, 16(1), 57-60.
- Sadrabadi, M. R., & Sadjadi, S. J. (2009). A new approach to solving multiple objective programming problems. *International Journal of Industrial Engineering & Production Research*, 20(1), 41-51.
- Tiwari, R. K. (2013), Multi-objective optimization of drilling process variables using genetic algorithm for precision drilling operation. *International Journal of Engineering Research and Development*, 6(12), 43-59.
- Tsagaris, A., Sagris, D., & Mansour, G. (2012). Intelligent CAD-based system for CNC machine controlling by genetic algorithms. *2012 IEEE 16th International Conference on Intelligent Engineering Systems (INES)* (pp. 235-239).
- Warwick, K. (2013). *Artificial intelligence: the basics*. Routledge.



© 2020 by the authors; licensee Growing Science, Canada. This is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC-BY) license (<http://creativecommons.org/licenses/by/4.0/>).