

Industry 4.0 - Optimization of Dressing Cycle Frequency and Power Requirements of Grinding Machine using Neural Networks

Apoorv M. Satpute¹, Swaneet D. Rao², Ishan C. Mane³, Dr. Prakash M. Mainkar⁴
Department of Electronics & Telecommunication,
Maharashtra Institute of Technology,
Pune

Abstract- In today's world Industry 4.0 is an emerging trend towards automation of manufacturing processes and technologies which include internet of things, cognitive and cloud computing, artificial intelligence and machine learning. Our main objective is to develop a method which increases the efficiency and reliability of machines using machine learning.

I. INTRODUCTION

Grinding is one of the most important processes in manufacturing industries. It is an abrasive machining process deals with removing excess material from a raw material surface. This process uses grinding wheel as cutting tool to make precision grinding and produces very fine finish within required tolerance. After successive grinding, the grinding wheel surface becomes blunt, losing its sharpness. Hence a diamond is used to remove the blunt surface and expose the sharp edges underneath. This process is called as "Dressing". In present scenario, there is no active feedback given to the machine by virtue of it, the machine controller can decide the exact time for performing grinding wheel dressing and subsequently the frequency of dressing. In the absence of this closed loop feedback, machine operators set the dressing frequency manually. High dressing frequency leads to unnecessary removal of grinding wheel material, thereby reducing the life of grinding wheel. Poor dressing frequency leads to various quality issues such as chatter marks, ovality, poor surface finish (High Ra value) and high-power consumption. A correct balance must be achieved for dressing cycle (Dressing frequency) to maintain both the quality of the product and life of grinding wheel thereby increasing the reliability of final finished product and reduced grinding wheel cost.

Neural networks [9] can be used to give real time feedback to machine of dressing frequency based on past cycles. Neural networks can also forecast the worst case cycles and can be used to alert line supervisors and engineers to analyse and rectify the problem thereby reducing the chances of machine failure. This helps in reducing the losses during production and increase overall manufacturing efficiency.

II. SELECTION OF DATA

Various dynamic forces are present during operation on grinding head. As the wheel becomes blunt, changes in the forces can be observed in the form of vibration and acceleration parameters. As the grinding wheel becomes blunt more force is applied to remove the material. Increased forces lead to more power consumption by the

machine spindle. Hence the power count of variable frequency drive has a direct correlation with the forces present. Hence power count is selected as input quantity for the neural network.

III. COLLECTION OF DATA

Most of the machines in industries are controlled by programmable logic controllers (PLC). Power count is collected from Mitsubishi MELSEC Q PLC as an Open Database Connectivity (ODBC) client via Ethernet. Kepware ServerX software was used for the collection and decoding of data and is stored in SQL tabular format on MS Access. Over 5000 dressing cycle data were collected over the period of 26 days. To make the resulting neural network more generalized and applicable, cycles during a specific environment, failures, worst case scenarios are also collected.

IV. PREPARATION OF DATA

Noisy and pilot cycles were removed as they were result of human error/intervention or due to other unrelated problems. Then the data is split into three parts namely training data, test data and holding data. Using sklearn's MinMaxScaler data is normalized. TensorFlow's time-series generator was used to generate batches of data along with the output array for the validation. Also training data is shaped according to number of features, length of input data with timestamps and batch size.

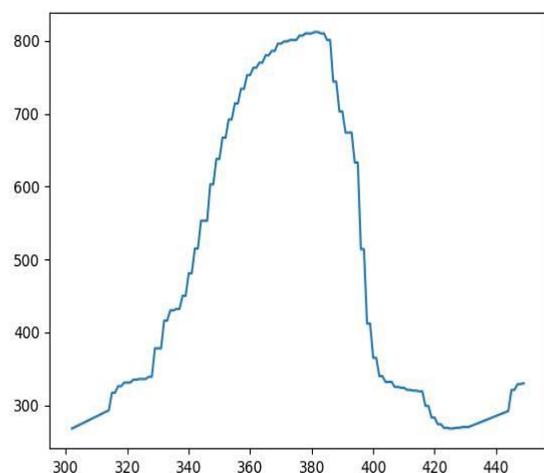


Fig.1 Power count Vs. One Cycle time

This graph shows us how the power count increases during a grinding cycle. The grinding is done gradually which can be observed graphically through the power count which denotes the power being used by the grinding head.

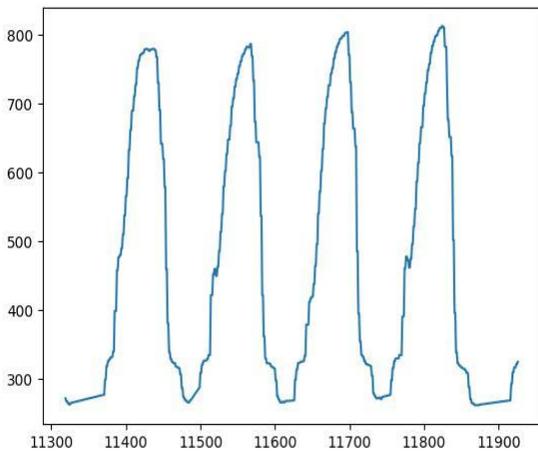


Fig. 2 Power Count vs. Grinding Cycles over time

From the above graph, it is clearly observed that as the number of grinding cycle increases, the power count also increases indicating an increase in power requirement of the grinding wheel. This is because of grinding surface becoming blunt over time, therefore grinding head requires more force and more power to make precision and accurate grinding.

V.MODEL SELECTION

As the model is going to predict the future cycle based on previous ten cycles, model must have a memory. The type of problem here is Univariate time series forecasting [5,10]. The recurrent neural network [2,5,6] gives better results for time series forecasting. But such neural networks heavily suffer from vanishing, exploding gradient [8] and take more time for training. Hence to overcome this problem we have chosen long short term memory (LSTM) architecture [1,7,11]. To learn more intricate patterns of time series a bi-directional wrapper is used on LSTM network. Bi-directional LSTM [3,4] neural network uses both past as well as future data of the target point for learning. Due to this architecture input information available to the layers are increased.

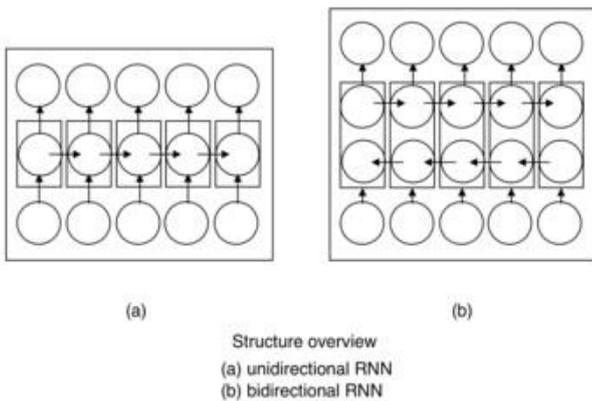


Fig. 3 Structure Overview of Recurrent Neural Network

VI.MODEL BUILDING

- Environment Variables:
 1. Programming Language : Python 3.7.1
 2. Development IDE : Anaconda 3
 3. Framework : TensorFlow
 4. Library : TensorFlow – GPU 2.0.0
 5. High level API : Keras 1.0.8
 6. Visualization : Tensorboard 2.0.0
- Layer Architecture:
 1. First Layer: This is the input layer consisting of 999 neurons, accepts the fix shape of data.
 2. Second Layer: This is a bi-directional LSTM layer consisting of 50 LSTM cells.
 3. Third Layer: This is a bi-directional LSTM layer consisting of 20 LSTM cells.
 4. Fourth Layer: This layer consists fully connected 50 neurons with a dropout probability of 0.25
 5. Fifth Layer: This layers consists fully connected 1 neuron which has linear activation
- Optimizer: Adaptive Moment Estimation (ADAM)
- Matrices: Mean square error (MSE), Root MSE
- Call backs: Early stopping, Tensorboard, checkpoints

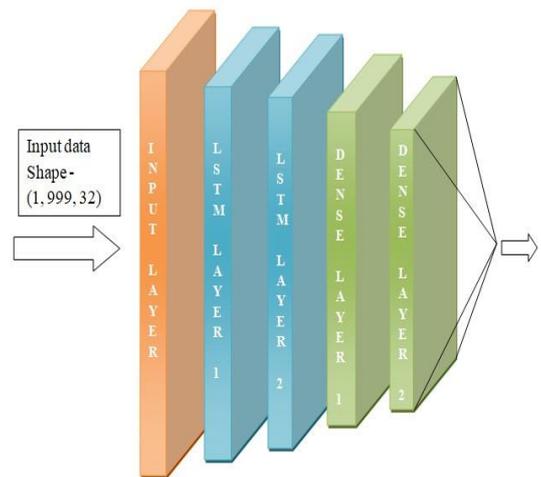


Fig. 4 Different layers in model

The above diagram shows different layers used in model generation which is used to predict the future power count cycle based on the previous ten power count cycles while grinding.

VI.MODEL TRAINING

- Number of epochs of training: 4
- Batch Size: 32
- Training Time: 3 hours, 16 Minutes
- Training Hardware: Nvidia GEFORCERTX 2060 Super
- Final training data MSE: 0.0011
- Final test data MSE: 0.0008648

```
In [10]: model.summary()

Model: "sequential_1"

Layer (type)                Output Shape                Param #
-----
bidirectional_2 (Bidirection (None, 999, 100)  208000
bidirectional_3 (Bidirection (None, 40)        19360
dense_1 (Dense)              (None, 50)                  2050
dense_2 (Dense)              (None, 1)                   51
-----
Total params: 42,261
Trainable params: 42,261
Non-trainable params: 0
```

Fig. 5 Model Summary

The above table shows us various information of the layers inside the model generated. It consists of different parameters like numbers of input, hidden, output neurons and their output shapes and sizes.

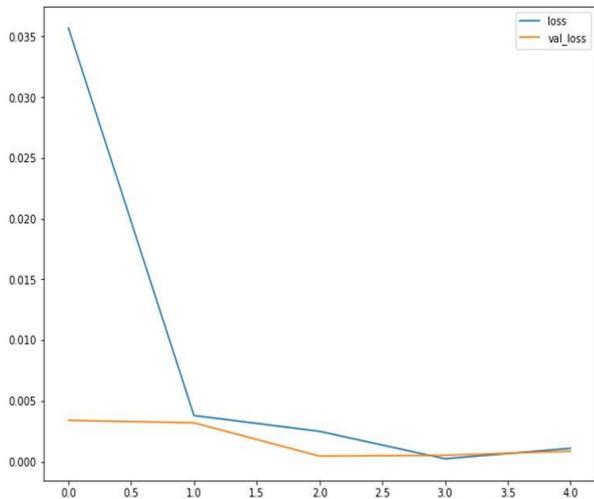


Fig. 6 Epochs vs. MSE

The above graph shows us how MSE reduces as the number of epochs increases. Error loss is plotted for both training data and test data.

VII.RESULTS

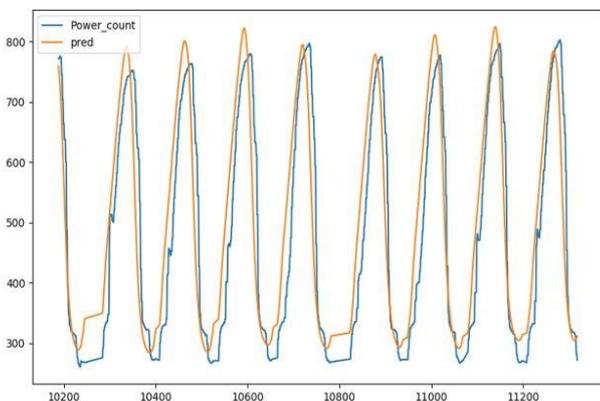


Fig. 7 Predictions on Test Data

From the above graph, it can be seen that power count predicted were very close to the actual power count which indicates high accuracy prediction by our generated model.

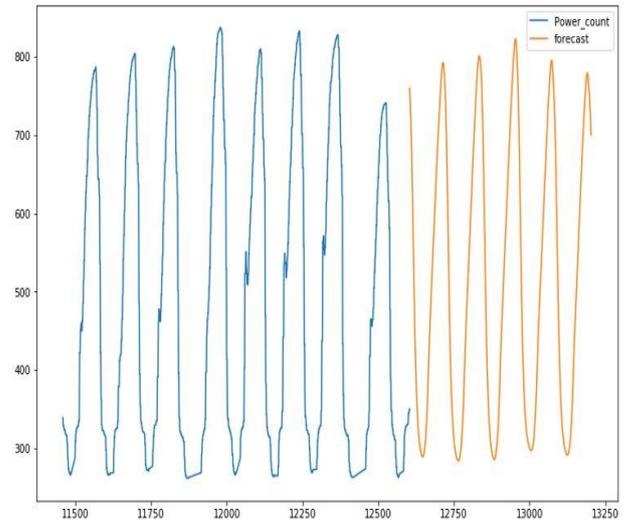


Fig. 8 Predictions on Holding Data

The above graph shows the predictions of power count on holding data. The predictions shown are the future cycles of power count predicted based on the previous ten grinding cycles. It was observed that the output predicted by our model was very accurate and very much closer to the actual real time cycles. These all graphs confirm that our model has learnt the grinding cycle nature and can predict the power count of the present as well as future power cycles accurately.

VIII.CONCLUSION

As from the results observed, future cycles of power cycles were accurately predicted. This prediction can be used as an active real time feedback to the machine where the dressing frequency can be adjusted automatically based on the power count predictions made instead of doing it manually by the factory line supervisor. This reduces the unnecessary dressing of grinding wheel and prolongs its life as well. As there is no human interference, chances of error are highly reduced resulting in highly efficient operation and very reliable finished products. This will greatly help industries to minimize their production losses and provide reliable, high quality products to their customers.

IX.FUTURE SCOPE

This kind of machine learning techniques can also be implemented on other machines other than grinding thereby increasing the scalability, repeatability of this project. We can also integrate this system with Augmented reality to visualize the failures in advance by analysing different parameters of machines real time and offering predications based on them which can be used to generate any future failure patterns based on the current scenario which will help industries take timely action and prevent any unnecessary interruptions beforehand.

ACKNOWLEDGEMENT

We would like to thank our research mentor Dr. Prakash Mainkar for his valuable guidance. We submit this paper with great humility and utmost regard.

REFERENCES

- [1] Hochreiter, S., & Schmidhuber, J. (1995). *Long short-term memory* (Tech. Rep. No. FKI-207-95). Fakultät für Informatik, Technische Universität München
- [2] D'informatique, D & N, Ese & Esent, Pr & Au, Ee & Gers, Felix & Hersch, Prof & Esident, Pr & Frasconi, Prof. (2001). Long Short-Term Memory in Recurrent Neural Networks. 10.5075/epfl-thesis-2366.
- [3] M. Schuster and K. K. Paliwal. Bidirectional recurrent neural networks. IEEE Volume: 45, Issue: 11, November 1997.
- [4] Sun, G., Chen, H., and Lee, Y. (1993). Time warping invariant neural networks
- [5] Sepp Hochreiter and Jürgen Schmidhuber. Long Short-Term memory. Neural computation, 1997
- [6] Alex Graves. Generating sequences with recurrent neural networks. CoRR, abs/1308.0850, 2013.
- [7] Klaus Greff, Rupesh Kumar Srivastava, Jan Koutník, Bas R. Steunebrink, and Jürgen Schmidhuber. LSTM: A Search Space Odyssey. CoRR, abs/1503.04069, 2015.
- [8] S. Grossberg. Recurrent neural networks. Scholarpedia, 8(2):1888, 2013. revision #138057.
- [9] Stephen Grossberg. Nonlinear neural networks: Principles, mechanisms, and architectures. Neural Networks, 1988.
- [10] Eldad Haber and Lars Ruthotto. Stable architectures for deep neural networks. Inverse Problems, 2017.
- [11] Alex Graves. Supervised sequence labelling with recurrent neural networks. PhD thesis, Technical University Munich, 2008.