



DEFECT CLASSIFICATION OF RADIUS SHAPING IN THE TIRE CURING PROCESS USING FINE-TUNED DEEP NEURAL NETWORK

Zendi Iklima¹, Bugi Nur Rohman², Rahmat Muwardi³, Zody Arifiansyah⁴

^{1,2,3}Department of Electrical Engineering, Faculty of Engineering, Mercu Buana University

⁴Department of Information Technology, Faculty of Computer Science, Esa Unggul University

Abstract

The curing process or vulcanization process is the final stage of the tire manufacturing process, where the properties of the tire compound change from rubber-plastic material to become elastic by forming cross-links in its molecular structure. Green tire is formed in the curing process which is placed on the bottom mold and the inside of the green tire surrounds the bladder (former of the inner tire). The top mold will close to carry out the next curing process. In the process of closing the mold there is a shaping process which is the process of forming a green tire which is placed on the bladder and is given a proportional pressure. Seventy percent of product defects are caused by improper or abnormal radius shaping results. This paper proposed abnormal detection of radius shaping in curing process using fine-tuned Deep Neural Network. Several DNN models have been examined to analyze an optimized DNN model for abnormal detection of radius shaping in the curing process. The fine-tuned DNN architecture has been exported for the curing system. The DNN was trained with a training accuracy of 97.88%, validation accuracy of 95%, testing accuracy of 100%, and a loss of 4.93%. The model also performs with precision, recall, and f1-score equals to 100%.

This is an open access article under the [CC BY-NC](#) license



Keywords:

Abnormal Detection;
Radius Shaping; Fined Tuned;
Deep Neural Network;

Article History:

Received: May 2, 2019

Revised: May 29, 2019

Accepted: June 2, 2019

Published: June 2, 2019

Corresponding Author:

Zendi Iklima

Electrical Engineering

Department, Universitas Mercu Buana, Indonesia

Email:

zendi.iklima@mercubuana.ac.id

INTRODUCTION

Tire curing is the last stage of the tire manufacturing process. Green tires produced from the assembly process are then fed into the curing area for vulcanization. The curing process is vulcanization with high temperature and pressure without polymer (rubber) with carbon black and sulfur with the help of chemical compounds to obtain the required characteristics so that it becomes a quality tire product. In the green tire vulcanization process, there are still defects or products that can reduce product quality [1]. There are six factors that affect product defects, namely: open mold, blond tread, inner pass, crack bead, mold-bead, and under cure. Under Cure defects in green tire production have a percentage of 40%. This

is caused by bladder conditions, overload, oil pump drop, pneumatic, and others [2]. This study focuses on the classification of under-cured product defects caused by bladder conditions, which are divided into 2 categories: crown bare (CB) defect and cracked inner liner (CIL) defect. A crown bare occurs when the radius or pressure is too high while the delay is too low. Cracked inner liner defect occurs because the radius or pressure is too low while the delay is too high.

Several artificial intelligence algorithms are used to reduce the number of manufacturing defects, especially in the tire curing process. Tobacco curing control technology has a higher labor intensity but also cannot be adjusted based on the conditions of

different tobacco leaf batches, which reduces the tobacco quality. Tobacco quality can be improved after curing and labor intensity can be reduced by predicting the state of the tobacco curing process, predicting the tobacco curing state accurately, and making timely adjustments to the curing process. The State Prediction Fusion Model (SPFM) method has been implemented as a control system for the tobacco curing process. The accuracy obtained in the SPFM training process is 97.4%, indicating that this model is very good at controlling the tobacco curing process [3].

TobaccoNet was developed based on the convolutional neural network (CNN) architecture. This model can classify labels for dry-bulb temperature and wet-bulb temperature with an accuracy of 99.65% and 96.83%. Promising results show that TobaccoNet is effective and reliable for modeling intelligent mass tobacco curing processes. The effect of different CNN structures on the prediction accuracy of dry-bulb temperature and wet-bulb temperature was analyzed from the perspective of computational complexity and prediction performance. The proposed sequential CNN structure is more suitable for analyzing bulk tobacco curing [4].

Physics-informed neural network (PINN) to simulate the thermochemical evolution of composite materials in autoclave curing equipment. The PINN framework for modeling exothermic heat transfer in composite tool systems undergoing a full cure cycle is presented. A proposed sequential approach to intermittent PINN training is presented which overcomes the instability in PINN training. In the training process, the network parameters are constrained by the introduction of a physics-based loss function. Transfer learning demonstrates improvement in PINN training and demonstrates its extension to a surrogate modeling setting by including heat transfer coefficients as input parameters [5].

Based on some of the literature above, this study develops a classification system for production defects in the curing bang process using radius shaping parameters, namely defect crown bare (CB) and defect cracked inner liner (CIL). The proposed model is an optimized deep neural network (DNN) model by configuring the network architecture based on the neural network hyperparameters. Two DNN models are combined to classify the radius shaping parameter configuration error in the tire curing process.

METHOD

Material

Figure 1 shows the sensor installation on a tire curing machine. An ultrasonic sensor is installed to read the radius when the green tire curing process is carried out. This radius value will be aligned with the pressure value, the actual radius value, and the delay value. Data on the curing machine for green tires cannot be retrieved digitally, so the installation of ultrasonic sensors helps to obtain configuration values on the curing machine which can be converted into trainable parameters.

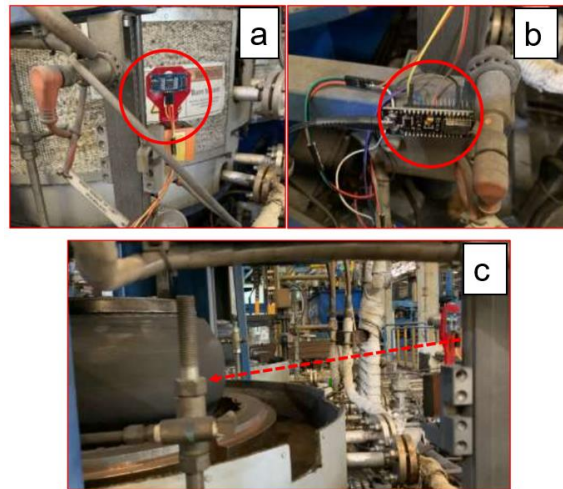


Figure 1. Sensor Installation in Tire Curing Machine

The readings of the radius, pressure, and delay values can be converted as shown in Figure 2 and Figure 3.

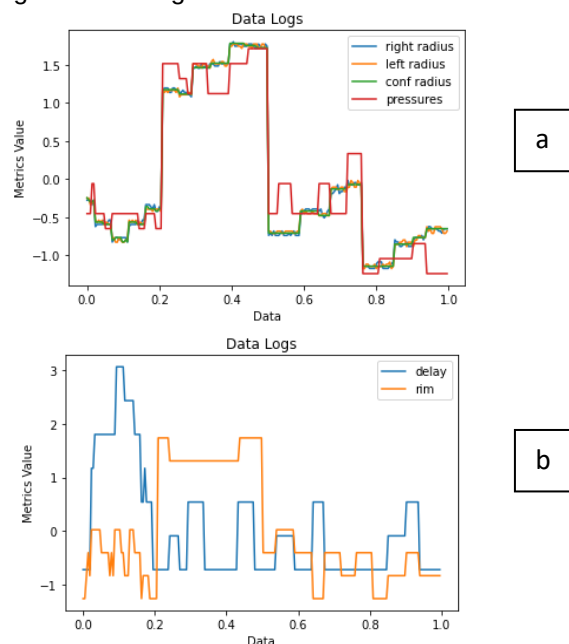


Figure 2. Data Logs of (a) Radius and Pressure, (b) Delay and Rim

Figure 3 shows the retrieval of data logs that have data labels, namely, CB, OK/NORMAL, and CIL.

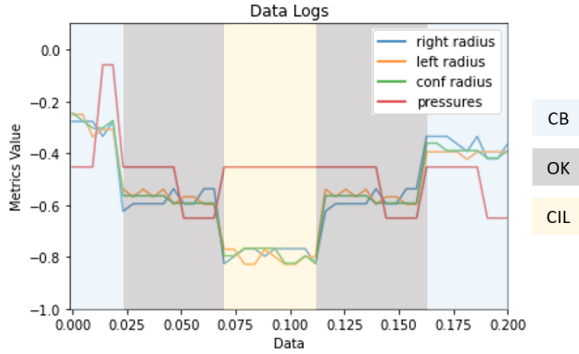


Figure 3. Data Logs and Labels

When the radius or pressure is too large and the delay is too short, a crown bare (CB) ensues. The inner liner cracks (CIL) when the radius or pressure is too low and the delay is too long.

Methods

Backpropagation is a supervised learning approach for artificial neural networks that uses gradient descent. The approach computes the gradient of the error function with respect to the neural network's weights, given an artificial neural network and an error function. It is a multilayer feedforward neural network extension of the delta rule for perceptron.

Based on Figure 2, our datasets are consisting of input-output pairs of size N is denoted $X = \{(\vec{x}_1, \vec{y}_1), \dots, (\vec{x}_N, \vec{y}_N)\}$. While the error function in classic backpropagation is the mean squared error, denoted as [6][7],

$$E(X, \theta) = \frac{1}{2N} \sum_{i=1}^N (\hat{y}_i - y_i)^2 \quad (1)$$

Where $\hat{y}_i$ denoted as the computed output of the network on input $\vec{x}$, y_i denoted as the target value for input-output pair $(\vec{x}_i, \vec{y}_i)$. the partial derivative of the error function E with respect to a weight in the hidden layers w_{ij}^k for $1 \leq k < m$ is [6][7][8],

$$\frac{\partial E}{\partial w_{ij}^k} = g'(a_j^k) o_i^{k-1} \sum_{l=1}^{r^{k+1}} w_{jl}^{k+1} \delta_l^{k+1} = \delta_j^k o_i^{k-1} \quad (2)$$

The error δ_j^k at layer k depend on the error δ_l^{k+1} at the next layer $k+1$. The errors flow

backward is to computed the first error terms based on the computed output $\hat{y} = g_o(a_l^m)$ and the target output y . The error terms for the previous layer are computed by performing a product sum (weighted by w_{jl}^{k+1}) of the error terms for the next layer and scaling it by $g'(a_j^k)$, repeated until the input layer is reached. Therefore, the partial derivatives for each input-output pairs denoted as [6][7],

$$\frac{\partial E(X, \theta)}{\partial w_{ij}^k} = \frac{1}{N} \sum_{d=1}^N \frac{\partial}{\partial w_{ij}^k} \left(\frac{1}{2} (\hat{y}_d - y_d)^2 \right) = \frac{1}{N} \sum_{d=1}^N \frac{\partial E_d}{\partial w_{ij}^k} \quad (3)$$

Update the weights according to the learning rate α and total gradient

$$\frac{\partial E(X, \theta)}{\partial w_{ij}^k} \text{ formulated as [6][7],}$$

$$\Delta w_{ij}^k = -\alpha \frac{\partial E(X, \theta)}{\partial w_{ij}^k} \quad (4)$$

The DNN Architecture is required to be designed to represent the curing machine data logs shown in Figure 2. This paper proposed a two-DNN architecture that includes both the predictor and classifier models. The predictor model consists of 2 inputs (x_1, x_2) that represent a pair of ultrasonic sensors. The predictor model generates predicted values ($y_1, \dots, y_4$) such as pressure values, actual radius values, delay values, and rim values. Based on Figure 3, the classifier generates the label of radius shaping defects ($z_1, \dots, z_3$) namely crown bare (CB), cracked inner liner (CIL), and Normal. Figure 4 represents the proposed DNN architecture.

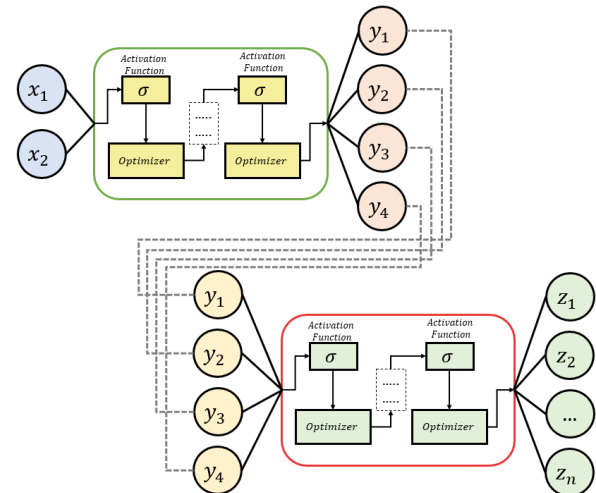


Figure 4. The Proposed DNN Architecture

Table 1 shows the DNN architecture hyperparameters [8][9].

Table 1. Network Hyperparameters	
Hyperparameters	Configurations
Num of Layers	[4, 8, 16, 32, 64]
Batch Sizes	[4, 8, 16, 32, 64]
Num of Neurons	[4, 8, 16, 32, 64]
Activation Functions	Sigmoid, LeakyReLU
Optimizers	Adam [11]
Loss Function	Cross-Entropy [12]
Learning rate	0.0001

The hyperparameters designed as the parameter to build DNN architecture in which to find the optimum values of number of layers, number of hidden layers, activation functions, optimizer, and the learning rate [13]. Table 2 shows several DNN models which constructed by num of layers (nL), num of neurons (nN), batch size (bS), and activation function.

Table 2. DNN Models

Model No	Layer Config (nL/nN/bS)	Activation Function
1	4 / 8 / 4	Sigmoid
2	4 / 8 / 4	LeakyReLU
3	4 / 16 / 4	LeakyReLU
4	8 / 16 / 4	LeakyReLU
5	8 / 32 / 4	LeakyReLU
6	16 / 32 / 4	LeakyReLU
7	8 / 32 / 8	LeakyReLU
8	8 / 32 / 16	LeakyReLU
9	8 / 32 / 32	LeakyReLU
10	8 / 32 / 64	LeakyReLU
11	8 / 32 / 128	LeakyReLU
12	8 / 64 / 8	LeakyReLU

RESULTS AND DISCUSSION

The DNN models were performed using Intel(R) Core i5-6300HQ CPU@2.30GHz, 16GB RAM, and Nvidia GeForce GTX 960M 4GB VRAM [11]. Table 3 compares the performance of the DNN models in terms of training accuracy, validation accuracy, training accuracy, training loss, validation loss, and execution time.

Table 3. DNN Models Performance

Model No	Acc (Tr/Val/Ts) %	Loss (Tr/Val) %	Exec. Time (s)
1	72.13 / 63.36 / 77.27	62.25 / 71.30	20.5
2	93.78 / 90.63 / 95.45	15.98 / 20.50	20.2
3	95.07 / 93.22 / 95.45	11.62 / 16.33	21
4	97.58 / 95.03 / 100	6.21 / 13.78	23.5
5	97.88 / 95.00 / 100	4.93 / 20.63	28.4
6	94.54 / 92.63 / 95.45	10.56 / 21.43	37.4
7	95.68 / 93.68 / 95.45	9.11 / 19.71	17.7
8	95.52 / 91.01 / 95.45	9.89 / 26.51	11.4
9	95.70 / 92.53 / 95.45	10.42 / 23.93	8.54
10	91.05 / 87.37 / 95.45	19.59 / 28.99	7.04
11	91.55 / 88.00 / 95.45	20.07 / 26.11	6.46
12	97.07 / 96.85 / 100	6.44 / 7.16	18.2

Table 3 shows the proposed model was trained using Adam optimize in 200 epochs. Model 4, 5, and 6 have good model performance, resulting in the training accuracy above 97%. The trained model needs to be validated to show that it can classify well given the distribution of different datasets. Figure 5 shows a summary of the proposed model training process.

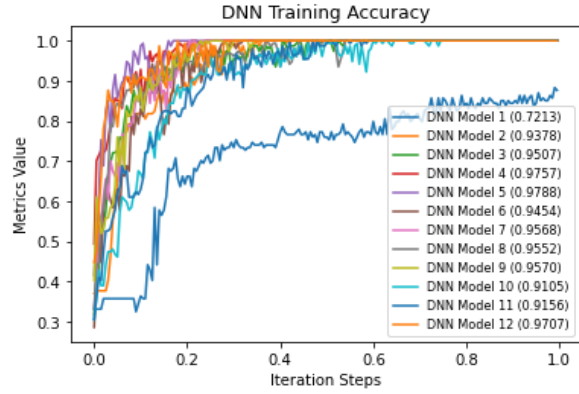


Figure 5. DNN Models Training Performance

Figure 6, shows the DNN Model (5) with the layer configuration num of layers (nL=8), num of neurons (nN=32), batch size (bS=4), and Activation Function ("LeakyReLU").

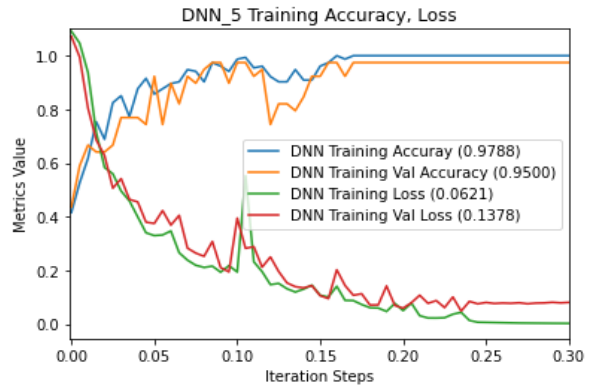


Figure 6. DNN Models (5) Training Performance

This DNN model shows a fit representation of training and validation values, which means the trained DNN model has a good data distribution for training and validation. Moreover, this DNN model is capable of predicting and classifying the defects of the radius shaping process in data test distribution which shown in Figure 7.

In order to represent the performance of particular DNN models, this paper assigned the precision, recall, and f1-score which formulated as [14][15],

$$recall(TPR) = \frac{(TP)}{(TP + FN)} \quad (6)$$

$$\text{precision} = \frac{(TP)}{(TP + FP)} \quad (7)$$

$$f1\text{-score} = 2 \frac{\text{recall} \times \text{precision}}{\text{recall} + \text{precision}} \quad (8)$$

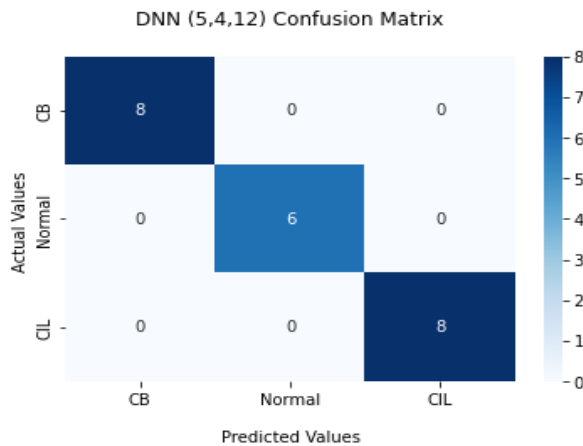


Figure 7. DNN Models (5, 4, and 12) Confusion Matrix

Figure 7 represents the confusion matrix in DNN Model (5) which produce precision, recall, and f1-score equals to 100%.

CONCLUSION

This research suggests utilizing a fine-tuned Deep Neural Network to identify aberrant radius shaping during the curing process. Several DNN models were investigated in order to develop an optimum DNN model for defects detection of radius shaping during the curing process. The fine-tuned DNN architecture has been exported for use in the curing system. The DNN was trained with the training accuracy of 97.88%, the validation accuracy of 95%, the testing accuracy of 100%, and the loss of 4.93%. The model also has the precision, recall, and f1-score of 100%.

ACKNOWLEDGMENT

The author expresses gratitude to the Electrical Engineering Department and Research Center of Mercu Buana University, Jakarta, Indonesia. Under the support, the authors have been completed this research.

REFERENCES

- [1] T. Yuniarto and C. Rangkuti, "Analisa Jenis Insulasi Terhadap Pipa Penghantar Uap Panas Untuk Mesin Pemasak Ban Motor Dengan Kapasitas 2000 Ban Per Hari Pada Pt. Xyz Di Tangerang," *Pros. Semin. Nas. Cendekiawan*, no. 1, pp. 1–23, 2019.
- [2] J. Hardono, H. Pratama, and A. Friyatna, "Analisis Cacat Produk Green Tyre dengan Pendekatan Seven Tools," *J. INTECH Tek. Ind. Univ. Serang Raya*, vol. 5, no. 1, p. 1, 2019, doi: 10.30656/intech.v5i1.1462.
- [3] Y. Wang and L. Qin, "Research on state prediction method of tobacco curing process based on model fusion," *J. Ambient Intell. Humaniz. Comput.*, no. April, 2021, doi: 10.1007/s12652-021-03129-5.
- [4] J. Wu and S. X. Yang, "Modeling of the Bulk Tobacco Flue-Curing Process Using a Deep Learning-Based Method," *IEEE Access*, vol. 9, pp. 140424–140436, 2021, doi: 10.1109/ACCESS.2021.3119544.
- [5] S. Amini Niaki, E. Haghighat, T. Campbell, A. Poursartip, and R. Vaziri, "Physics-informed neural network for modelling the thermochemical curing process of composite-tool systems during manufacture," *Comput. Methods Appl. Mech. Eng.*, vol. 384, p. 113959, 2021, doi: 10.1016/j.cma.2021.113959.
- [6] M. A. Nielsen, "Neural Networks and Deep Learning." Determination Press, 2015, Accessed: Jan. 18, 2022. [Online]. Available: <http://neuralnetworksanddeeplearning.com>.
- [7] John McGonagle, George Shaikouski, Christopher Williams, Andrew Hsu, Jimin Khim, and Aaron Miller, "Backpropagation | Brilliant Math & Science," Jan. 20, 2020. <https://brilliant.org/wiki/backpropagation/> (accessed Jan. 20, 2022).
- [8] Z. Iklima, M. I. Muthahhar, A. Khan, and A. Zody, "Self-Learning of Delta Robot Using Inverse Kinematics and Artificial Neural Network," vol. 25, no. 3, pp. 237–244, 2021.
- [9] Z. Zhang, "Improved Adam Optimizer for Deep Neural Networks," *2018 IEEE/ACM 26th Int. Symp. Qual. Serv. IWQoS 2018*, pp. 1–2, 2019, doi: 10.1109/IWQoS.2018.8624183.
- [10] S. Bock, J. Goppold, and M. Weiß, "An improvement of the convergence proof of the ADAM-Optimizer," pp. 1–5, 2018, [Online]. Available: <http://arxiv.org/abs/1804.10587>.
- [11] I. Zendi, "A Microservices-based for Distributed Deep Neural Network of Delta Robot Control System," *2020 IEEE Int. Conf. Commun. Networks Satell. Comnetsat 2020 - Proc.*, pp. 218–221, 2020, doi: 10.1109/Comnetsat50391.2020.9328936.
- [12] R. Almodfer, S. Xiong, M. Mudhsh, and P.

- Duan, "Enhancing AlexNet for Arabic Handwritten words Recognition Using Incremental Dropout," 2017, doi: 10.1109/ICTAI.2017.00106.
- [13] S. Teerapittayanon, B. McDanel, and H. T. Kung, "Distributed Deep Neural Networks over the Cloud, the Edge and End Devices," *Proc. - Int. Conf. Distrib. Comput. Syst.*, pp. 328–339, 2017, doi: 10.1109/ICDCS.2017.226.
- [14] A. Abbas, M. M. Abdelsamea, and M. M. Gaber, "Classification of COVID-19 in chest X-ray images using DeTraC deep convolutional neural network," *Appl. Intell.*, vol. 51, no. 2, pp. 854–864, 2021, doi: 10.1007/s10489-020-01829-7.
- [15] M. Loey, G. Manogaran, M. Hamed, N. Taha, N. Eldeen, and M. Khalifa, "A hybrid deep transfer learning model with machine learning methods for face mask detection in the era of the COVID-19 pandemic," *Measurement*, vol. 167, no. May 2020, p. 108288, 2021, doi: 10.1016/j.measurement.2020.108288.