Abstract: Kerf width is one of the most important quality items in cutting of thin metallic sheets. The aim of this study was to develop a convolutional neural network (CNN) model for analysis and prediction of kerf width in laser cutting of thin non-oriented electrical steel sheets. Three input process parameters were considered, namely, laser power, cutting speed, and pulse frequency, while one output parameter, kerf width, was evaluated. In total, 40 sets of experimental data were obtained for development of the CNN model, including 36 sets for training with $k$-fold cross-validation and four sets for testing. Compared with a deep neural network (DNN) model and an extreme learning machine (ELM) model, the developed CNN model had the lowest mean absolute percentage error (MAPE) of 4.76% for the final test dataset in predicting kerf width. This indicates that the proposed CNN model is an appropriate model for kerf width prediction in laser cutting of thin non-oriented electrical steel sheets.

Keywords: laser cutting; kerf width; convolutional neural network; non-oriented electrical steel

1. Introduction

Non-oriented electrical steels are produced from Fe–Si or Fe–Si–Al alloys and used as the core material in electrical machinery [1]. Generally, the stator and rotor of electric motors are formed by lamination of non-oriented electrical steel sheets with a thickness of 0.1 mm to 1 mm. Such parts are usually stamped in a cost-effective way for mass production with limited precision. However, expensive fixtures and tools seem to be a drawback of stamping for low-volume production or rapid prototyping [2]. Laser cutting is an alternative to stamping, which could provide the availability to minimize the cost for small quantity production [2]. In addition, the plastic and elastic stresses induced by mechanical cutting could result in deterioration of the magnetic properties of electrical steel [3] and efficiency of the core [4–6]. Shear deformation at the cutting edge is typically found in conventional mechanical cutting processes, which might have a detrimental effect on the core performance in electrical machinery [7,8]. The magnetic field and flux density of electrical steels are affected by residual stress [9,10]. For laser cutting, no remarkable shear deformation at the cutting edge is found [11].

There are several process parameters influencing the kerf quality of laser cutting, including laser power, pulse frequency, and cutting speed. High-quality kerf of non-oriented electrical steel sheet is achievable with proper design of laser process parameters. Therefore, an investigation into the prediction of kerf width using various combinations of laser cutting process parameters is essential. Several methods have been proposed for kerf width prediction in laser cutting of metals [12–17]. Mathematical models [12–14] have been widely used to assess kerf quality for laser cutting processes. Statistical analysis has been conducted to study the effect of process parameters on the laser cutting quality [13,14]. Recently, the artificial intelligence (AI) technique has attracted more attention in effectively predicting the kerf quality of laser cutting [15,17]. Although there are a variety of AI
methods for prediction of cutting quality in laser cutting [15–17], studies for the prediction of kerf width in laser cutting of non-oriented electrical steel are limited.

The convolutional neural network (CNN) model is mainly used for image analysis in different fields, e.g., remote sensing [18,19] and biometrics [20,21]. The convolutional approach of CNN is its major advantage. The CNN obtains major features automatically in a learning process such that the manual feature extraction step is neglected. However, CNN is also interesting for other applications that do not deal with image analysis, such as one-dimensional data analysis. To improve the classification performance of patient-specific electrocardiograms (ECGs), a CNN model was developed to combine two major aspects, namely, feature extraction and classification [22]. Feature extraction can be used to classify data in a fast and accurate manner [22]. Moreover, a simple CNN-based method was applied for the classification of vibrational spectroscopic data [23]. It was demonstrated that the CNN could be applied to select important spectral regions [23]. This suggests that the CNN was less dependent on preprocessing for vibrational spectroscopic data classification and achieved excellent performance on raw data [23]. Another CNN method was built up for chemometric data regression [24]. The standard CNN architecture was modified for adaptation to one-dimensional data analysis [24]. The results yielded by CNN were very promising with great accuracy, suggesting that the CNN model was able to achieve feature extraction for regression of one-dimensional data [22–24]. In this regard, CNN was employed in this study for kerf width prediction in the laser cutting of electrical steel sheets.

The objective of this study was to develop an effective CNN model for prediction of kerf width in the laser cutting of thin metallic sheets. In particular, experiments were performed using a pulsed laser for cutting 0.1 mm thick non-oriented electrical steel sheets. After cutting, the output quality of kerf width was measured using a charge-coupled device (CCD) camera. The three input parameters considered were laser power, pulse frequency, and cutting speed. The experimental data were used to train and test the CNN model through \( k \)-fold cross-validation. The performance of the kerf width prediction by the developed CNN model was also compared with that of other AI-based methods, namely, deep neural network (DNN) and extreme learning machine (ELM).

2. Experiment and Model
2.1. Experimental Setup

The experiments were carried out using a 20 W ytterbium pulsed fiber laser with a wavelength of 1064 nm (YLP-1-100-20, IPG Photonics Co., Oxford, MA, USA). As shown in Figure 1, the laser system consisted of scanning mirrors and a focusing lens attached to a computed numerically controlled (CNC) machine. They moved together through a moveable mechanism controlled by the CNC under various cutting speeds from 0.1 to 0.5 mm/s. Laser power \((P)\), cutting speed \((v)\), and laser pulsed frequency \((f)\) were selected as the input process parameters. The values of process parameters considered are listed in Table 1.

Table 1. Laser cutting parameters.

| Parameter               | Value      |
|-------------------------|------------|
| Wavelength (nm)         | 1064       |
| Laser power (W)         | 9.4–14.07  |
| Pulse frequency (kHz)   | 20–60      |
| Pulse width (ns)        | 100        |
| Focal length (mm)       | 127        |
| Spot diameter (µm)      | 40         |
| Cutting speed (mm/s)    | 0.1–0.5    |

A schematic of the laser cutting of a 20 mm long straight slit on a commercial thin non-oriented electrical steel sheet is shown in Figure 2. The dimensions of the workpiece
were 20 mm in width, 40 mm in length, and 0.1 mm in thickness (ST-100, Nikkin Denji Kogyyo Co., Ltd., Saitama, Japan). After completion of the cutting process, the kerf width of each sample was measured using a computer vision system consisting of a light source, a CCD camera (BFS-U3-51S5M-C, VS Technology Co., Tokyo, Japan) with a CMOS sensor (IMX250, Sony Co., Tokyo, Japan), and a lens (VS-TCH3-60, VS Technology Co., Tokyo, Japan). As shown in Figure 2, the kerf width was determined by taking the average of the measurements in the segment AB.

![Figure 1. Schematic of the laser cutting of a non-oriented electrical steel sheet.](image1)

![Figure 2. Schematic of specimen size and laser cutting path (left); optical micrograph of kerf (right).](image2)

### 2.2. Convolutional Neural Network Model

The CNN structure for kerf width prediction is shown in Figure 3. The first part of the CNN model was for feature extraction. The final, fully connected layer was added for regression analysis. The developed CNN model contained seven layers including the input layer, one convolutional layer, one flattened layer, three fully connected hidden layers, and the output layer. There were three input nodes, namely, laser power, pulse frequency, and cutting speed, while the output node was the kerf width. In total, 40 sets of experimental data (Table 2) were obtained for training and testing of the developed CNN model, including 36 sets for training and four sets for testing. Note that dataset Nos. 37–40 in Table 2 were used for final testing.

In this study, the convolutional operation processed the input data using 64 convolution filters and outputted the feature map. Each convolutional filter extracted certain...
patterns from the input data. The mathematical expression of the convolutional operation in layer $l$ is defined below [25].

$$y_{l}^{i+1}(j) = f\left(K_{l}^{i} \ast x_{l}(j) + b_{l}^{i}\right),$$

where $K_{l}^{i}$ and $b_{l}^{i}$ denote the weight matrix and bias vector of the $i$-th filter in the $l$-th layer, and $x_{l}(j)$ represents the input of the $l$-th layer. The asterisk denotes the convolution operator, and $y_{l}^{i+1}(j)$ is the output of the convolutional operation.

![Figure 3. Structure of CNN model for kerf width prediction.](image)

**Table 2.** Experimental dataset.

| No | Laser Power (W) | Pulse Frequency (kHz) | Cutting Speed (mm/s) | Kerf Width in Experiment (µm) |
|----|-----------------|-----------------------|---------------------|-------------------------------|
| 1  | 9.40            | 20                    | 0.1                 | 139.7                         |
| 2  | 11.67           | 20                    | 0.1                 | 149.5                         |
| 3  | 14.07           | 20                    | 0.1                 | 161.7                         |
| 4  | 9.40            | 40                    | 0.1                 | 23.2                          |
| 5  | 11.67           | 40                    | 0.1                 | 27.8                          |
| 6  | 11.67           | 60                    | 0.1                 | 18.8                          |
| 7  | 14.07           | 60                    | 0.1                 | 175.4                         |
| 8  | 11.67           | 20                    | 0.2                 | 34.6                          |
| 9  | 14.07           | 20                    | 0.2                 | 35.7                          |
| 10 | 11.67           | 40                    | 0.2                 | 548.3                         |
| 11 | 14.07           | 40                    | 0.2                 | 278.3                         |
| 12 | 11.67           | 60                    | 0.2                 | 486.6                         |
| 13 | 14.07           | 60                    | 0.2                 | 662.3                         |
| 14 | 9.40            | 20                    | 0.3                 | 85.5                          |
| 15 | 14.07           | 20                    | 0.3                 | 100.9                         |
| 16 | 9.40            | 40                    | 0.3                 | 44.8                          |
| 17 | 11.67           | 40                    | 0.3                 | 30.0                          |
| 18 | 14.07           | 40                    | 0.3                 | 40.4                          |
| 19 | 14.07           | 60                    | 0.3                 | 103.5                         |
| 20 | 9.40            | 20                    | 0.4                 | 65.7                          |
| 21 | 11.67           | 20                    | 0.4                 | 144.4                         |
| 22 | 14.07           | 20                    | 0.4                 | 106.0                         |
| 23 | 9.40            | 40                    | 0.4                 | 67.4                          |
| 24 | 14.07           | 40                    | 0.4                 | 494.1                         |
| 25 | 11.67           | 60                    | 0.4                 | 649.0                         |
| 26 | 14.07           | 60                    | 0.4                 | 675.0                         |
As shown in Figure 3, the input layer had \( L \times 3 \) inputs, where \( L = 36 \) was the number of datasets used for training. The depth of the convolutional layer had 64 filters. Each of the 64 filters was applied to the windows in the input layer to generate \( L \times 1 \) hidden features via convolution operation. The CNN convolutional layer was designed to improve accuracy and speed up training [26]. Thereafter, the rectified linear unit (ReLU) was applied to increase nonlinearity in the CNN [25,26], which can be expressed as follows [25]:

\[
a_{i}^{l+1} = f\left(y_{i}^{l+1}(j)\right) = \max\{0, y_{i}^{l+1}(j)\},
\]

where \( a_{i}^{l+1} \) denotes the activation function of \( y_{i}^{l+1}(j) \). The extracted features from the flattened layer were merged into one single vector which was used as an input for the first of the three fully connected layers. After flattening, the extracted features passed through the first fully connected layer with 64 nodes. The fully connected layers are similar to hidden layers in a traditional artificial neural network (ANN). The three fully connected layers had 64, 32, and 16 nodes, respectively. The output of each node in the fully connected layers can be mathematically expressed as follows [25],

\[
d^{l} = f\left(w^{l} \times d^{l-1} + b^{l}\right),
\]

where \( w \) and \( b \) represent weight and bias, respectively. ReLU was used in all fully connected layers as a typical nonlinear activation function for the regression analysis [26].

A backpropagation algorithm was used as a backward pass by adjusting the weights assigned to the nodes of all layers in order to minimize the error between the predicted and experimental values [27]. The adaptive moment estimation (ADAM) algorithm [26] was used as a key element of the backpropagation algorithm. Firstly, the backpropagation algorithm computed the gradient [27], and then the ADAM algorithm was applied to find the gradient descent to measure the error function slope. The weights were updated using Equation (4).

\[
w_{ij}^{t+1} = w_{ij}^{t} - \eta \left( \frac{\partial E}{\partial w_{ij}} \right),
\]

where \( i \) is the node number of the previous layer, \( j \) is the node number of the next layer, \( w_{ij}^{t} \) is the weight at time \( t \), \( w_{ij}^{t+1} \) is the updated weight at time \( t + 1 \), \( \eta \) is the learning rate, and \( E \) is the error function.
2.3. Deep Neural Network Model

Unlike the CNN model, the DNN model makes no assumption about the input, and it tends to perform worse for feature extraction. It consists of one input layer, one or several hidden layers, and one output layer, as shown in Figure 4. Layer \( k \) transforms the activity of the previous layer \( (h_{k-1}) \) into the current layer \( (h_k) \) by multiplying it with a weight matrix \( W_k \), adding a bias vector \( b_k \), and applying a nonlinear activation function \( f \), as expressed below [28].

\[
    h_k = f(W_k \times h_{k-1} + b_k). 
\]  

\[ 
    [29] 
\]

![Diagram of DNN model for kerf width prediction.](image)

**Figure 4.** Structure of DNN model for kerf width prediction.

As shown in Figure 4, the structure of the DNN employed in this study contained five layers, namely, the input layer, three hidden layers, and the output layer. Similar input nodes were used, namely, laser power, pulse frequency, and cutting speed. The hidden layers had 64, 32, and 16 nodes, respectively. The activation function used for the DNN was a ReLU function. The output node was the kerf width. The ADAM algorithm was used as the training algorithm for a maximum of 300 iterations.

2.4. Extreme Learning Machine Model

The concept of an ELM model was proposed by Huang et al. [29]. It is a single-hidden-layer feedforward neural network. The input weights and biases are randomly assigned without any iterative computation, and the output weights are calculated by the Moore–Penrose generalized inverse matrix [29].

The structure of the ELM model used in this study is shown in Figure 5. It had a three-layer architecture. Considering a set of \( N \) samples \( (x_i, t_i) \in \mathbb{R}^n \times \mathbb{R}^m \), where \( x_i = [x_{i1}, x_{i2}, \ldots, x_{in}]^T \in \mathbb{R}^n \) represents the input data and \( t_i = [t_{i1}, t_{i2}, \ldots, t_{im}]^T \in \mathbb{R}^m \) represents the output data, the output of the network can be expressed as follows [29]:

\[
    o_j = \sum_{i=1}^{L} \beta_i G_i(x_j) = \sum_{i=1}^{L} \beta_i G(w_i \cdot x_j + b_i), \quad j = 1, \ldots, N, 
\]  

where \( L \) is the number of hidden nodes, \( w_i = [w_{i1}, w_{i2}, \ldots, w_{in}]^T \) and \( b_i \) denote the learning parameters of the \( i \)-th hidden node, \( \beta_i = [\beta_{i1}, \beta_{i2}, \ldots, \beta_{im}]^T \) is the weight vector connecting the \( i \)-th hidden node and the output nodes, and \( G(x) \) is the activation function. The \( j \)-th
output vector of the ELM model can be expressed as \( o_j = [o_{j1}, o_{j2}, \ldots , o_{jm}]^T \). Equation (6) can be further compactly written in a matrix form as follows [29]:

\[
H\beta = T,
\]

where \( H \) is the hidden layer output matrix [29].

\[
H(w_1, \ldots , L, b_1, \ldots , b_L, x_1, \ldots , x_N) = \begin{bmatrix}
G(w_1 \cdot x_1 + b_1) & \cdots & G(w_1 \cdot x_N + b_1) \\
\vdots & \ddots & \vdots \\
G(w_L \cdot x_1 + b_1) & \cdots & G(w_L \cdot x_N + b_L)
\end{bmatrix}_{N \times L},
\]

\[
\beta = \begin{bmatrix}
\beta_1^T \\
\vdots \\
\beta_L^T
\end{bmatrix}_{L \times m},
\]

\[
T = \begin{bmatrix}
t_1^T \\
\vdots \\
t_N^T
\end{bmatrix}_{N \times m}.
\]

![Figure 5. Structure of ELM model for kerf width prediction.](image)

The output weights are constructed using the following equation [29]:

\[
\beta = H^T T,
\]

where \( H^\dagger \) is the Moore–Penrose generalized inverse of the hidden layer output matrix \( H \).

For the ELM model employed in this study, the minimum error occurred at a number of 34 hidden nodes. ReLU was also used as the activation function for hidden nodes, as shown in Figure 5.

A high-level network application programming interface (API) library (Keras 2.2.4, Google LLC, Mountain View, CA, USA) using an interpreted language (Python 3.7, Python Software Foundation, Wilmington, DE, USA) was employed to build the CNN model, DNN model, and ELM model.

3. Results and Discussion

3.1. Optimal CNN Model through k-Fold Cross-Validation

To evaluate the performance of the CNN model developed, the mean absolute percentage error (MAPE) was utilized as a statistical criterion, which can be expressed as follows:

\[
MAPE = \frac{100}{N} \sum_{i=1}^{N} \left| \frac{\hat{y}_i - y_i}{y_i} \right|,
\]
where $N$ is the number of samples, and $y_i$ and $\hat{y}_i$ represent the $i$-th experimental measurement and CNN prediction, respectively. One of the most common techniques for model evaluation is $k$-fold cross-validation [30], which can be used to evaluate a CNN model more fully and accurately with a small dataset. The main idea behind cross-validation is that each observation in the dataset has the opportunity to be tested. The key configuration parameter for $k$-fold cross-validation is $k$, which defines the number of folds to split a given dataset. Figure 6 shows an example of a $k$-fold cross-validation procedure with $k = 10$. In order to choose an appropriate $k$ value, results from each fold are averaged to produce a single estimation. Firstly, the training dataset is divided roughly into equal parts. Then, for each $m = 1, 2, 3, \ldots, k$, $MAPE_m$ is computed for the $m$-th fold, as shown in Figure 6. The average MAPE for the selected $k$ folds is then calculated as follows:

$$MAPE = \frac{1}{k} \sum_{m=1}^{k} MAPE_m. \quad (12)$$

![Schematic of k-fold cross-validation procedure (k = 10).](image)

After that, the process is repeated for various numbers of $k$, and the best $k$ is that with the smallest MAPE. In this study, $k = 5, 6, 7, 8, 9, 10$ were selected for comparison in determining the optimal value. The resulting MAPE values in Figure 7 reveal that, for the given CNN model and datasets, the MAPE of the given $k$ numbers was higher than 7.9% except for $k = 8$ and 10. The findings also suggest that $k = 10$ was slightly better than $k = 8$ in terms of a more accurate prediction. Therefore, $k = 10$ was selected as the optimal number of folds in determining the CNN model.

### 3.2. Comparison with Other ANN Models

A random forest algorithm was applied to explore the importance of the selected input parameters in correlation with the output [31]. As shown in Figure 8, among the given three inputs, cutting speed was the most important factor affecting the kerf width. The relative importance of laser power, pulse frequency, and cutting speed on kerf width was 0.15, 0.37, and 0.48, respectively.

To verify the effectiveness of the proposed CNN model, the DNN model and the ELM model described above were used for comparison. Figure 9 shows the comparison of experiment and prediction of kerf width for the same final test dataset using the given three ANN models. The prediction accuracy of the given three models was evaluated using Equation (11). The MAPE of these three models for predicting the kerf width of the same final test dataset is listed in Table 3. As shown in Table 3, the CNN model exhibited an
average MAPE of 4.76% in kerf width prediction, which was lower in comparison with 15.24% for the ELM model and 16.93% for the DNN model. This proves that the structure of the CNN model developed in this study significantly improved the feature learning performance compared to the DNN and ELM models, which had a limited performance when using original data without feature extraction.

Figure 7. Average MAPE for various $k$ values.

3.2. Comparison with Other ANN Models

A random forest algorithm was applied to explore the importance of the selected input parameters in correlation with the output [31]. As shown in Figure 8, among the given three inputs, cutting speed was the most important factor affecting the kerf width. The relative importance of laser power, pulse frequency, and cutting speed on kerf width was 0.15, 0.37, and 0.48, respectively.

To verify the effectiveness of the proposed CNN model, the DNN model and the ELM model described above were used for comparison. Figure 9 shows the comparison of experiment and prediction of kerf width for the same final test dataset using the given three ANN models. The prediction accuracy of the given three models was evaluated using Equation (11). The MAPE of these three models for predicting the kerf width of the same final test dataset is listed in Table 3. As shown in Table 3, the CNN model exhibited an average MAPE of 4.76% in kerf width prediction, which was lower in comparison with 15.24% for the ELM model and 16.93% for the DNN model. This proves that the structure of the CNN model developed in this study significantly improved the feature learning performance compared to the DNN and ELM models, which had a limited performance when using original data without feature extraction.

Figure 8. Variable importance analysis results using a random forest algorithm.

| $P$ (W) | $f$ (kHz) | $v$ (mm/s) | Kerf Width in Experiment (µm) | Predicted Kerf Width (µm) ELM | Predicted Kerf Width (µm) DNN | Predicted Kerf Width (µm) CNN | MAPE (%) ELM | MAPE (%) DNN | MAPE (%) CNN |
|--------|-----------|------------|-----------------------------|-----------------|---------------------|---------------------|-------------|-------------|-------------|
| 9.40   | 40        | 0.5        | 98.8                        | 93.0            | 92.7                | 99.0                | 5.82        | 6.17        | 0.20        |
| 11.67  | 40        | 0.4        | 64.2                        | 84.4            | 49.1                | 74.6                | 14.47       | 23.52       | 16.20       |
| 11.67  | 60        | 0.3        | 623.9                       | 714.1           | 564.0               | 626.0               | 9.60        | 9.47        | 0.34        |
| 14.07  | 20        | 0.5        | 74.2                        | 81.0            | 95.3                | 72.5                | 9.24        | 28.44       | 2.29        |
|        |           |            |                             |                 |                     |                     | Average     | 15.24       | 16.93       | 4.76        |

Table 3. Comparison of prediction results of kerf width for the same test dataset using various ANN models.
Figure 9. Comparison of experiment and prediction of kerf width for the same final test dataset using (a) ELM model, (b) DNN model, and (c) CNN model.
4. Conclusions

In summary, a CNN model was developed to predict the kerf width in the laser cutting of non-oriented electrical steel sheet. It is evident that the CNN structure was suitable for kerf width prediction with high accuracy. The conclusions drawn from the results were as follows:

1. The $k$-fold cross-validation method was employed to improve the generalization ability of the developed CNN model. Moreover, for $k = 10$, the average MAPE of the validation dataset had the lowest value of 7.51% compared to other $k$ values.

2. In comparison with other ANN methods such as DNN and ELM, the results clearly indicated that the CNN approach developed in this study exhibited improved performance and achieved the highest prediction accuracy of 4.76% in terms of MAPE for the same final test dataset. Therefore, the developed CNN with $k$-fold cross-validation is effective for kerf width prediction of the given laser cutting process.

3. The relative importance of the given three input parameters was analyzed using a random forest algorithm. It was found that the most important variable for kerf width is the cutting speed, followed by the pulse frequency and the laser power.

Author Contributions: Conceptualization, D.-T.N. and C.-K.L.; methodology, C.-K.L., J.-R.H. and P.-C.T.; investigation, D.-T.N.; resources, C.-K.L., J.-R.H. and P.-C.T.; data curation, D.-T.N. and C.-K.L.; writing—original draft preparation, D.-T.N.; writing—review and editing, C.-K.L.; project administration, C.-K.L.; funding acquisition, C.-K.L. All authors have read and agreed to the published version of the manuscript.

Funding: This work was funded by the Ministry of Science and Technology (Taiwan) under contract numbers MOST 109-2221-E-008-027 and MOST 110-2221-E-008-089.

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. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript or in the decision to publish the results.

References

1. Oda, Y.; Kohno, M.; Honda, A. Recent development of non-oriented electrical steel sheet for automobile electrical devices. *J. Magn. Magn. Mater.* 2008, 320, 2430–2435. [CrossRef]
2. Brettschneider, C. High precision laser cutting of electrical steel. *PhotonicViews* 2019, 16, 51–53.
3. Park, J.S.; Park, J.T. Effect of Stress Relief Annealing Temperature and Atmosphere on the Microstructure and Magnetic Properties of Non-oriented Electrical Steels. In Proceedings of the 2016 6th International Electric Drives Production Conference (EDPC), Nuremberg, Germany, 30 November–1 December 2016; pp. 288–292.
4. Rygal, R.; Moses, A.; Derebasi, N.; Schneider, J.; Schoppa, A. Influence of cutting stress on magnetic field and flux density distribution in non-oriented electrical steels. *J. Magn. Magn. Mater.* 2000, 215, 687–689. [CrossRef]
5. Schoppa, A.; Schneider, J.; Wuppermann, C.D. Influence of the manufacturing process on the magnetic properties of non-oriented electrical steels. *J. Magn. Magn. Mater.* 2000, 215, 74–78. [CrossRef]
6. Senda, K.; Ishida, M.; Nakasu, Y.; Yagi, M. Influence of shearing process on domain structure and magnetic properties of non-oriented electrical steel. *J. Magn. Magn. Mater.* 2006, 304, e513–e515. [CrossRef]
7. Fujisaki, K.; Hirayama, R.; Kawachi, T.; Satou, S.; Kaidou, C.; Yabumoto, M.; Kubota, T. Motor core iron loss analysis evaluating shrink fitting and stamping by finite-element method. *IEEE Trans. Magn.* 2007, 43, 1950–1954. [CrossRef]
8. Salvador, L. Influence of Cutting Process on Magnetic Properties of Electrical Steel. Master’s Thesis, Aalto University, Helsinki, Finland, 2016.
9. Loisos, G.; Moses, A.J. Effect of mechanical and Nd: YAG laser cutting on magnetic flux distribution near the cut edge of non-oriented steels. *J. Mater. Process. Technol.* 2005, 161, 151–155. [CrossRef]
10. Siebert, R.; Schneider, J.; Beyer, E. Laser cutting and mechanical cutting of electrical steels and its effect on the magnetic properties. *IEEE Trans. Magn.* 2014, 50, 1–4. [CrossRef]
11. Emura, M.; Landgraf, F.J.G.; Ross, W.; Barreta, J.R. The influence of cutting technique on the magnetic properties of electrical steels. *J. Magn. Magn. Mater.* 2003, 254, 358–360. [CrossRef]
12. Dubey, A.K.; Yadava, V. Multi-objective optimization of laser beam cutting process. *Opt. Laser Technol.* 2008, 40, 562–570. [CrossRef]

13. Yilbas, B.S. Laser cutting quality assessment and thermal efficiency analysis. *J. Mater. Process. Technol.* 2004, 155, 2106–2115. [CrossRef]

14. Hasçalık, A.; Ay, M. CO₂ laser cut quality of Inconel 718 nickel–based superalloy. *Opt. Laser Technol.* 2013, 48, 554–564. [CrossRef]

15. Hossain, A.; Hossain, A.; Nukman, Y.; Hassan, M.A.; Harizam, M.Z.; Sifullah, A.M.; Parandoush, P. A fuzzy logic-based prediction model for kerf width in laser beam machining. * Mater. Manuf. Process.* 2016, 31, 679–684. [CrossRef]

16. Pandey, A.K.; Dubey, A.K. Fuzzy expert system for prediction of kerf qualities in pulsed laser cutting of titanium alloy sheet. *Mach. Sci. Technol.* 2013, 17, 545–574. [CrossRef]

17. Klancnik, S.; Begić-Hajdarević, D.; Paulić, M.; Ficko, M.; Cekić, A.; Husic, M.C. Prediction of laser cut quality for Tungsten alloy using the neural network method. *Stroj. Vestn.—J. Mech. Eng.* 2015, 61, 714–720. [CrossRef]

18. Ghamisi, P.; Chen, Y.; Zhu, X.X. A self-improving convolutional neural network for the classification of hyperspectral data. *IEEE Geosci. Remote Sens. Lett.* 2016, 13, 1537–1541. [CrossRef]

19. Cheng, G.; Zhou, P.; Han, J. Learning rotation-invariant convolutional neural networks for object detection in VHR optical remote sensing images. *IEEE Trans. Geosci. Remote Sens.* 2016, 54, 7405–7415. [CrossRef]

20. Nogueira, R.F.; de Alencar Loto, R.; Machado, R.C. Fingerprint liveness detection using convolutional neural networks. *IEEE Trans. Inf. Forensics Secur.* 2016, 11, 1206–1213. [CrossRef]

21. Labati, R.D.; Munoz, E.; Piuri, V.; Sassi, R.; Scotti, F. Deep-ECG: Convolutional neural networks for ECG biometric recognition. *Pattern Recognit. Lett.* 2019, 126, 78–85. [CrossRef]

22. Kiranyaz, S.; Ince, T.; Gabbouj, M. Real-time patient-specific ECG classification by 1-D convolutional neural networks. *IEEE Trans. Biomed. Eng.* 2015, 63, 664–675. [CrossRef] [PubMed]

23. Acquarelli, J.; van Laarhoven, T.; Gerretzen, J.; Tran, T.N.; Buydens, L.M.; Marchiori, E. Convolutional neural networks for vibrational spectroscopic data analysis. *Anal. Chim. Acta* 2017, 954, 22–31. [CrossRef]

24. Malek, S.; Melgani, F.; Bazi, Y. One-dimensional convolutional neural networks for spectroscopic signal regression. *J. Chemom.* 2018, 32, e2977. [CrossRef]

25. Liu, X.; Zhou, Q.; Zhao, J.; Shen, H.; Xiong, X. Fault diagnosis of rotating machinery under noisy environment conditions based on a 1-D convolutional autoencoder and 1-D convolutional neural network. *Sensors* 2019, 19, 972. [CrossRef] [PubMed]

26. Geron, A. *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*, 2nd ed.; O’Reilly Media: Sebastopol, CA, USA, 2019.

27. Rumelhart, D.E.; Hinton, G.E.; Williams, R.J. Learning representations by back-propagating errors. *Nature* 1986, 323, 533–536. [CrossRef]

28. Goodfellow, I.; Bengio, Y.; Courville, A. *Deep Learning*, 1st ed.; MIT Press: Cambridge, MA, USA, 2016.

29. Huang, G.B.; Zhu, Q.Y.; Siew, C.K. Extreme learning machine: Theory and applications. *Neurocomputing* 2006, 70, 489–501. [CrossRef]

30. Picard, R.R.; Cook, R.D. Cross-validation of regression models. *J. Am. Stat. Assoc.* 1984, 79, 575–583. [CrossRef]

31. Breiman, L. Random forests. *Mach. Learn.* 2001, 45, 5–32. [CrossRef]