Article

Motor Load Balancing with Roll Force Prediction for a Cold-Rolling Setup with Neural Networks

Sangho Lee and Youngdoo Son *

Department of Industrial and Systems Engineering, Dongguk University, Seoul 04620, Korea; 
sg218@seoul.dgu.ac.kr
* Correspondence: youngdoo@dongguk.edu

Abstract: The use of machine learning algorithms to improve productivity and quality and to maximize efficiency in the steel industry has recently become a major trend. In this paper, we propose an algorithm that automates the setup in the cold-rolling process and maximizes productivity by predicting the roll forces and motor loads with multi-layer perceptron networks in addition to balancing the motor loads to increase production speed. The proposed method first constructs multilayer perceptron models with all available information from the components, the hot-rolling process, and the cold-rolling process. Then, the cold-rolling variables related to the normal part set-up are adjusted to balance the motor loads among the rolling stands. To validate the proposed method, we used a data set with 70,533 instances of 128 types of steels with 78 variables, extracted from the actual manufacturing process. The proposed method was found to be superior to the physical prediction model currently used for setups with regard to the prediction accuracy, motor load balancing, and production speed.

Keywords: neural networks; steel manufacturing; intelligent manufacturing systems; artificial intelligence; smart factory

1. Introduction

The steel industry directly affects the national economy as it is linked to a number of upstream and downstream industries and also provides basic raw materials for major industrial sectors, including automobiles, shipbuilding, home appliances, machinery, and construction [1,2]. Over the past 100 years, the steel industry has achieved remarkable results in the technical aspects of production. Thus, the steel industry is frequently employed as an important indicator of the power of a nation [3].

Steel manufacturers have been facing severe competition among countries with mature steel industries, such as the United States, South Korea, and China, as well as from newcomers aiming for economic growth, like India [4]. In addition, strict financial margins distinguishing success and failure in the steel industry make the competition more severe and global [5]. Steel manufacturers and raw material suppliers have attempted to increase competitiveness based on economies of scale, i.e., the enlargement of suppliers and the merger and acquisition of major steel manufacturers. Therefore, there is a growing need for the development of new and innovative technologies to increase the quality of products and the efficiency of manufacturing [6].

Massive amounts of data regarding production, process design, assembly, quality control, and fault detection have been stored in database management systems and data warehouses for several decades. In modern industrial environments, however, it is critical to utilize the data effectively as well as to collect and manage data [7].

Data mining, which can extract meaningful patterns from data, has emerged as an important tool for acquiring knowledge from databases [5]. Data mining and analytics have also played important roles in discovering knowledge and making decisions in process industries, including the steel industry, for a few decades. In recent years, machine
learning techniques, such as artificial neural networks and reinforcement learning methods, have been rapidly developed to more effectively perform data analysis tasks, including information retrieval, pattern recognition, and variable prediction. Additionally, there have been many attempts to apply machine learning methods to real industrial problems due to various policies and plans to improve manufacturing processes and performance in the United States, Germany, and China [8–13].

According to these recent trends, machine learning methods have been employed for various applications in the process industry to improve the quality, productivity, and efficiency. One application is monitoring and managing the quality based on identifying, classifying, and correcting faults in advance using data generated from the manufacturing process [11,14–19]. Other applications include managing supply chains by predicting the demand of the participants in the supply chain, sharing information [20], and applying reinforcement learning to resource management problems, such as job scheduling or relay selection [21].

A large amount of data has been produced in the steel industry, similar to other process industries. There exist reports that machine learning applications, such as the prediction of the temperature of molten steel [22,23], flow deformation temperature [24], quality and mechanical properties of products [25–28], and roll forces in the hot and cold-rolling processes increase competitiveness by improving the quality and reducing redundant costs. However, the industry does not fully utilize the stored data and employs insufficient variables and instances.

Additionally, most of the existing steel manufacturing processes still utilize a complex and inflexible physical model with limited information. As the types of steel, especially high-strength steels, and their properties have become more diverse, the demand for more accurate models with high capabilities to increase the productivity and quality has significantly increased. It is also necessary to increase the production speed and the capacity utilization, which directly affect the productivity and reduce redundant costs.

In this paper, we propose an algorithm that automates setup and maximizes productivity in the cold-rolling process. First, we construct neural network models that outperform the currently used physical model in predicting roll forces and motor loads by using various input variables such as the hot-rolling, steel component, and cold-rolling information to improve the quality of the cold-rolling output. Second, we suggest a procedure that maximizes the production speed and balances the motor loads to increase the productivity and capacity utilization and to maintain the robustness of the production process.

The remainder of this paper is organized as follows. In the next section, we review the milestone results obtained from machine learning applications to industrial problems, especially those in the steel industry. In Section 3, we briefly introduce the basic concepts of steel manufacturing processes and artificial neural networks as preliminary information. Then, we describe the proposed method, which includes prediction models for roll forces and motor loads and a motor load balancing algorithm to improve the quality and increase the productivity of the cold-rolling process in Section 4. The experimental results are described in Section 5, and we present concluding remarks with further research directions in Section 6.

2. Related Work

In this section, we briefly review the milestone and recent results from machine learning applications regarding industrial problems. We also focus on applications to steel manufacturing problems, which are highly related to the proposed method.

2.1. Machine Learning Applications to Industrial Problems

Predictive machine learning models have been applied to several industrial problems, such as error detection and diagnosis [29–31], quality manufacturing [32–34], process monitoring [4,35], and manufacturing automation [36,37]. These machine learning models, which include regression methods, decision trees, support vector machines, and artificial
neural networks, have shown better performance compared with conventional methods in various situations.

Numerous studies have been conducted in the field of quality prediction and management using fault detection. In quality management processes with machine learning, monitoring is performed based on a large amount of collected data and analysis of the characteristics of products to detect, classify, and correct faults in advance. Identifying faults using machine learning helps managers avoid unnecessary and expensive additional processes during physical inspection.

Thus, many studies have been performed to increase the accuracy of fault detection. Kang [14] first constructed a neural network model to regress the characteristics related to faults and trained another neural network model using the regressed values to classify the faults. Lieber et al. [18] trained prediction models with support vector machines and decision trees after extracting features from data. Kang and Liu [19] also performed feature extraction and error detection with neural networks to predict surface defects on cold-rolled strips.

Some studies employed machine learning algorithms to solve problems other than fault detection. Carbonneau et al. [20] focused on information sharing over the supply chain. They applied two types of neural network models, multilayer perceptrons (MLPs) and recurrent neural networks, to accurately predict and share the demand of each participant in the supply chain. Another study applied reinforcement learning to resource management. Mao et al. [21] solved job scheduling and relay selection in internet communication and congestion control, which frequently occurred in computer systems but had been only solved using empirical methods.

Bagheripoor and Bisadi [38] predicted roll forces and torques in the aluminum hot-rolling process using neural network models based on four input variables: the initial temperature, thickness reduction, rolling speeds, and interface resistance coefficients. However, the models suffered from lack of data, and therefore artificial instances were created with finite element methods to train the neural networks.

These studies demonstrate that machine learning techniques, such as neural network models, can be applied to various industrial fields to optimize the process efficiency and improve the quality of products.

2.2. Machine Learning in the Steel Industry

Due to the importance of the steel industry and its competitive atmosphere, many predictive models have been constructed based on machine learning methods to improve the performance of manufacturing processes.

The productivity and quality of molten steel—an essential material in the steel manufacturing industry—are highly sensitive to temperature. Tian et al. [22] and Tian and Mao [23] proposed hybrid models that combined extreme learning machines and modified decision trees to accurately predict the temperature of molten steel. Thermal deformation, defined as a change in the shape of the final product due to temperature changes, is one of the key determinants of quality and is closely related to the flow stress. Lin et al. [24] used a neural network model to improve the prediction accuracy of flow stress.

The data for training the network consist of temperature, strain, and the logarithm of the strain rate of 42CrMo steel, which is a representative medium carbon and low-alloy steel. It is also important to predict and manage the mechanical properties of the steel, including the elastic modulus, yield point, tensile strength, and hardness. Thus, artificial neural network models have been employed to predict the mechanical properties of hot-rolled and cold-rolled steels [25,28]. Guo and Sha [26] also constructed a neural network model predicting the properties of maraging steels. In addition, Perzyk and Kochański [27] predicted the quality of ductile cast iron using neural network models.

Many studies on the roll force and motor load prediction, which are closely related to this study, exist. Larkiola et al. [39] proposed the roll force prediction model for the cold-rolling process by combining a conventional physical model and a neural network model.
The neural network model estimated the coefficients of the conventional physical model and the conventional model predicted the roll forces for four stands of the cold-rolling mill. The authors used about 6000 instances with 18 input variables, including hot-rolling information, such as alloy element, temperature, thickness, and component information, to construct the neural network model, and the resulting model showed good performance. However, the model had a limitation as the intermediate coefficients for the conventional physical model were predicted rather than the roll forces themselves.

Cho et al. [40] presented a roll force prediction and correction factor derivation model for five stands in the cold-rolling process using an MLP. The proposed model was trained using 8000 coil data points with 12 input variables, including the entry/delivery thickness and tensions for each stand, the width, outer diameter, length, roll type, and proportions of carbon, manganese, and silicon. The model reduced the prediction errors for the roll forces by 30 to 50 percent and improved the stability by combining the proposed neural network model with the existing physical model. However, the neural network model showed unstable and inferior results when it was solely used.

Son et al. [41] also predicted the coefficients for roll forces in the hot-rolling process using neural network models and employed the coefficients as the input for the physical model to reduce the computation time and improve the prediction accuracy. The resulting model predicted the roll forces of three of six stands with five input variables, namely the learning rate, strain, strain rates, carbon equivalent, and temperature for each stand. The model was trained with 13,728 instances, and both online and offline procedures were suggested for continuous learning. The results exhibited better performance over the established physical model.

Mahmoodkhani et al. [42] combined the finite element method and neural networks for the rapid prediction of the roll force in skin pass rolling of ultrahigh strength steel, including 980DP and 1180CP. This model iteratively modified the resistance coefficient, one of the input variables, to derive the optimal roll force. The neural network model was constructed as one hidden layer with four hidden nodes and trained with seven input variables: the initial strip thickness, initial strip width, thickness reduction, roll outer diameter, resistance coefficient, and entry/delivery tensions.

The prediction error was reduced to less than 10 percent, and the prediction speed was sufficient for online usage. These studies show that machine learning techniques, including neural network models can be successfully used to solve practical issues in the steel industry, including the rolling processes, and performed better than the existing systems in terms of efficiency and quality.

There have been many studies on machine learning applications to practical problems in the steel industry, including roll force prediction in the cold-rolling process, due to its importance. However, the proposed method in this paper is different from those reported in previous studies in several aspects. First, the proposed method predicts not only the roll forces but also the motor loads for each stand. Moreover, the proposed method maximizes the production speed and finds the proper motor load distribution using the motor load balancing algorithm proposed in this paper to achieve setup automation, increased productivity, and quality improvement.

Second, while previous studies mainly focused on the middle-stage coefficients or a combination of machine learning models and existing models, the proposed method predicts the final targets directly by constructing an end-to-end model, and thus the proposed model can completely replace existing models. Finally, to the best of our knowledge, this study employed the largest number of data instances and input variables. We used 70,353 reliable instances of 128 types of steel collected from the actual processes at a steel company in South Korea. The number of total input variables was 78 for all stands, and 34 to 35 variables were used for each stand, reflecting the hot-rolling and component information as well as the cold-rolling information.
3. Preliminary Information

In this section, we summarize the steel manufacturing process, especially the cold-rolling process, on which we focus, and the MLP model, briefly as preliminary information.

3.1. Steel Manufacturing Process

The steel manufacturing process can be roughly divided into four stages: iron making, steel making, continuous casting, and rolling. In the iron-making stage, iron ore and limestone are sintered and cokes are produced with coking coal. Afterward, molten iron is produced in a blast furnace by melting the sintered iron ore while burning the cokes with hot blowing air. The molten iron is changed into molten steel in the steel-making process. Pure oxygen is blown into the mixture of molten steel, scrap, and other raw materials in the blast furnace to remove impurities, such as phosphorus and sulfur. The necessary components are then added to create molten steel with the desired properties.

During the continuous casting process, the molten steel is injected into a mould, continuously cooled, and solidified by passing through a continuous casting machine. The outputs of the continuous casting process are semi-products, such as slabs, blooms, and billets. The final process of steel manufacturing is the rolling process, during which the semi-products produced in the continuous casting process are stretched or thinned with continuous force by passing them through several rotating rolls. There are two major types of rolling processes: hot-rolling and cold-rolling.

The hot-rolling process is conducted at a temperature higher than the recrystallization temperature. Cold-rolling is typically conducted after hot-rolling at room temperature. The hot-rolling process is typically carried out with the reduction ratio, which is the ratio of the reduced thickness to the entry thickness, whilst cold-rolling is typically performed with a low reduction rate. Figure 1 summarizes the steel manufacturing process. Cold-rolled steels usually have higher tensile strength and hardness and are also more accurate in size than hot-rolled steels.

![Figure 1. Overview of the steel manufacturing process.](image-url)

The demand for cold-rolled products continues to grow because they are used as basic materials for various products, including household appliances, industrial equipment, and various construction materials. Accordingly, the importance of quality and productivity improvements in the cold-rolling process has also increased. Specifically, since the cold-rolling process is the final processing step in manufacturing, it directly affects the quality and production speed of the final product. Therefore, this study provides a novel procedure that not only predicts the roll forces and motor loads in the cold-rolling process based on neural networks but also balances the motor loads and maximizes the production speed.

3.2. Cold-Rolling Process

In the cold-rolling process, a coil, the output of the hot-rolling process, is rolled to the desired thickness by passing it through a mill, usually a tandem mill in which the coil is rolled by passing it through several stands sequentially. Figure 2 shows a brief structure of a tandem mill. Sometimes each rolling stand in the tandem mill consists of several rolls, such as work rolls (WR), intermediate rolls (IMR), and back up rolls (BR).
A brief overview of the cold-rolling process is shown in Figure 2. The initial setup, or the normal part setup, is conducted before the coil enters the rolling mill. In the normal part setup, information from only one representative point, usually near the centre of the next coil to be processed, is extracted, and all stands are configured based on the setup values, including the roll force and motor loads for each stand and the speed of the coil, determined from the extracted information. However, because the head and tail parts are welded with other coils, they are not well represented by the sampled point. Thus, the setup values for each stand are changed during the rolling process only for the welded parts immediately before each stand, in a process known as the dynamic setup.

The normal part setup is generally performed using predictions based on physical models. However, these physical models have a limitation in that they only use the cold-rolled information. Therefore, they cannot reflect information, usually hot-rolling information, that is not included in the physical formula, such as the coil temperatures when the hot-rolling starts and ends, the duration of spraying after hot-rolling, and the acceleration in hot-rolling.

Changes in these variables induce large errors in prediction, which can cause problems in the product quality. In addition, the conventional system does not consider the motor load distribution for the stands in the rolling mill and, thus, does not fully utilize the available resources, which leads to a low production speed. Higher motor loads in specific rolling stands can also reduce the productivity because the rolling mill cannot work when even one stand is under maintenance.

To solve the limitations of the existing physical model, which employs limited information to predict roll forces and motor loads, we constructed MLP networks for the roll force and motor load prediction, reflecting various types of information, such as the heating furnace condition, spray time for the cooling hot-rolled coil, coil thickness, and width in cold-rolling, and component information, including carbon, manganese, and phosphorus. We also propose a motor load balancing algorithm based on the roll force and motor load prediction to maximize the utilization among the rolling stands so that the production speed is maximized and unnecessary costs are reduced.

### 3.3. Neural Network Models

Neural network models, inspired by neurons in the human brain, are some of the most extensively used and studied machine learning methods. A neural network model is composed of several nodes (or neurons) and their connections. A neuron receives input values from other neurons and transmits an output value to other neurons as in Figure 3. The output value is calculated by applying an activation function to the weighted sum of the input values and a bias as in Equation (1).

$$y = g\left(\sum w_i \times x_i + b\right)$$  \hspace{1cm} (1)

where $x_i$ is the input value received from the preceding layer neuron, $w_i$ is the corresponding weight of each connection, and $b$ is the bias.
One of the typical choices for the activation function, $g(z)$, is the rectified linear unit,

$$g(z) = \max(0, z)$$ (2)

One of the simplest neural network models is the MLP model, which is composed of several layers of neurons. Each neuron transmits the output value to all neurons in the next layer as shown in Figure 3. There are three types of layers: input, output, and hidden layers. The input layer receives the input values from the data and transmits them to the next layer, while the output layer returns the final output to the user. The layers between the input and output layers are called hidden layers. The output value is calculated as the input values passed through the layers sequentially in the forward direction, and the weights and biases in the model are trained through error back propagation [43]. MLP models are widely used for typical classification and regression tasks [44].

![Figure 3. Neurons in neural network models and a multilayer perceptron (MLP) model.](image)

Neural network models, including MLP, have strong advantages, including high performance and no special processing requirements, but they have disadvantages, such as a large number of hyperparameters, proneness to overfitting, and a large computational burden.

4. Proposed Method

Figure 4 shows an overview of the entire framework for the proposed method for cold-rolling setup. First, the roll forces for each stand in the tandem mill are predicted by the trained MLP model. Then, the output of the roll force prediction model is merged with the original input variables to be used as input for the motor load prediction model, which is also a trained MLP model. Based on the predicted motor loads, the stopping criteria are checked. If the motor loads satisfy the stopping criteria, the procedure is completed, and the final output values are returned as setup values to the control system.

If the stopping criteria are not satisfied, the input values originating from the cold-rolling, specifically the target thickness and the corresponding speeds for each stand, are modified. Afterward, the roll forces and motor loads are predicted again using the modified inputs. This variable modification and the subsequent roll force and motor load prediction procedures are repeated until one of the stopping criteria is satisfied.

We targeted a cold-rolling tandem mill with six stands. However, we constructed the roll force and motor load prediction models for only the first five stands because the final stand works with the predetermined roll forces, while the other stands focus on the accuracy of the target thickness. For each prediction model, we trained five MLP models with two hidden layers by random sampling of the training and validation data sets and created an ensemble model to find the predicted output, which is the mean of the predicted values from five MLP models.
Since the roll force prediction model in the proposed method is composed of MLP models, its computational complexities of the training and test phases are \(O(NW\epsilon)\) and \(O(MW)\), respectively, where \(N\) is the number of training data, \(W\) is the number of parameters in the network, \(\epsilon\) is the maximum number of epochs for training, and \(M\) is the number of test data.

The conventional physical model, however, has the complexity of \(O(MC)\) where \(C\) is the number of calculations in the physical formula to obtain the result of a single data point. The training complexity of the physical model cannot be calculated because the parameters are obtained from the experimental results. Both the proposed method and the conventional model have complexities that are linearly proportional to the number of instances, although \(W\) is usually several times larger than \(C\).

If we consider the entire process of the proposed method, the time complexity of the test phase becomes \(O(M(W' + b)i)\) where \(W'\) is the total number of parameters in the roll force and motor load prediction models, \(b\) is the maximum number of iterations in the balancing procedure, and \(i\) is the maximum number of repetitions of the entire proposed method. Regardless, the computation complexity is linearly proportional to the number of data points.

**4.1. Roll Force and Motor Load Prediction Models**

Figure 5 shows the detailed procedure for the roll force and motor load prediction algorithm. Unlike the conventional physical model, which only employs cold-rolling information, the proposed method employs steel component, hot-rolling, and cold-rolling information. The variables from these sources are merged and standardized. The input variables are split for the corresponding stands. For example, the input variables for each stand include all of the component and hot-rolling information as well as variables related to each stand from the cold-rolling information. These input variables are used to predict the roll forces for each stand from the constructed ensemble MLP models.

Specifically, let \(X = [x_1, x_2, \ldots, x_N]^T\) where \(x_i = [x_{i1}, x_{i2}, \ldots, x_{iD}]\), \(N\) is the number of training instances in a training batch, and \(D\) is the number of input variables. For roll force prediction models, \(D\) is 34 for the third and fourth stand models and 35 for the other stands. For motor load prediction models, a predicted roll force value of each stand is also employed as an input variable; thus, \(D\) is 35 for the third and fourth stands and 36 for the others. Then, the prediction is conducted as follows.

\[
H_1 = ReLU(W_1X + B_1) \quad (3)
\]
\[
H_2 = ReLU(W_2H_1 + B_2) \quad (4)
\]
\[
\hat{Y} = W_3H_2 + B_3 \quad (5)
\]
where $W_j$ and $B_j$ are the matrices of the weights between neurons and biases in $j$-th layer, $\hat{Y}$ are the predicted roll force or motor load values, and $ReLU(\cdot)$ is a rectified linear unit activation function, i.e., $ReLU(z) = \max(z, 0)$.

![Diagram of the roll force and motor load prediction algorithm](image)

**Figure 5.** Detailed procedure for the roll force and motor load prediction algorithm.

Each network is trained by gradient descent method error back propagation algorithm [43] with the errors $E$ as follows.

$$w_{ij}^{new} = w_{ij}^{old} - \alpha \left( \frac{\partial E}{\partial w_{ij}} \right)$$

$$\frac{\partial E}{\partial w_{ij}} = \delta_i z_j$$

where $w_{ij}$ is a weight connecting $rgw$ $i$-th neuron and $j$-th input, $\alpha$ is a step size for the gradient descent method, $z_j$ is the output value of $j$-th node in the previous layer, $\delta_j = \hat{y}_j - y_j$ for the output layer, and $\delta_j = h'(a_j) \sum_k w_{jk} \delta_k$ with an activation function $h(\cdot)$ and $a_j = \sum_j w_{ij} z_j$, $\hat{y}_j$ and $y_j$ denote the predicted and true target values, respectively. Thus, the weights in the network can be adjusted backwardly from the output layer to the first hidden layer because $\delta_i$ in the current layer depends on $\delta_j$'s in the next layer.

In this study, the mean absolute percentage error (MAPE) was employed to measure the errors since it provides intuitive results of the errors—the percentage errors of the predictive values—to the engineers.

$$MAPE = \frac{\sum |e_t / Y_t|}{n} \times 100$$

where $e_t$ is the prediction error, $Y_t$ is the corresponding actual value, and $n$ is the number of data points. In this case, $e_t = \hat{Y}_t - Y_t$.

### 4.2. Motor Load Balancing Model

Figure 6 shows an overview of the motor load balancing algorithm applied after the first prediction of roll forces and motor loads to use the motors in each stand evenly and reduce the idle time of the mill. First, we find the speed limit for each stand and compare it to the current setup values. Then, we can maximize the rolling speed without exceeding the speed limit, i.e., we calculate the ratio between the maximal speed and the current speed for every stand that has a speed limit and increase the speed of all stands with the minimal ratio among the stands.

The roll forces and motor loads are calculated again with the modified speed. At this time, the above-mentioned roll force and motor load prediction MLP models are used, respectively. After that, the stopping criteria for the motor load balancing method are checked. These criteria include the differences between the maximal and minimal motor usage among the rolling stands and the maximal number of iterations. If the stopping
criteria are not satisfied, the cold-rolling variables, such as the thickness and speed before and after the stands are modified to balance the motor loads.

Specifically, the thickness reduction ratio for the stand with the highest motor usage is reduced, and that of the stand with the lowest motor usage is increased. The other variables, such as the speed and thickness of all stands, are changed according to these modifications while maintaining a constant volumetric flow rate, defined as the flow of volume per unit time. The roll forces and motor loads are then predicted using the above-mentioned MLP models with the modified variables, and the stopping criteria are checked again. These procedures are repeated until one of the stopping criteria is satisfied.

Figure 6. Procedure of the motor load balancing algorithm.

5. Experiments
5.1. Data Description and Experimental Setup

The total number of data used to construct the roll force and motor load prediction model was 70,353 with 128 types of steel using different components. We employed 78 variables in total, including the component, hot-rolling, and cold-rolling information. These variables were assigned to the corresponding stands; thus, 35 variables were used for the roll force prediction of the first, second, and fifth stands, and 34 variables were used for the third and fourth stands. For the motor load prediction task, each roll force was added to the input variables for the corresponding stand.

We employed the MAPE, which is mentioned above, and mean squared error (MSE) to measure the accuracy of the training model. MSE, one of the most commonly used measures for the regression, was also used because it is relatively sensitive to outliers and, thus, can reflect other aspects of the prediction errors than with MAPE.

\[
MSE = \frac{\sum e_t^2}{n}
\]  

(9)

where \( e_t \) is the prediction error and \( n \) is the number of data points.

Among 70,353 instances, we assigned 30% of the instances as the test data points. We randomly selected 80% of the remaining data points as the training data points, and the rest were used for validation. The numbers of training, validation, and test data instances were 39,397, 9850, and 21,106, respectively. The separation of training and validation sets was repeated five times. The number of training epochs was controlled with the early stopping method, which increases the training speed while avoiding overfitting [45]. In this study, we set the patience as 20 with 100 burning epochs. In other words, if the validation error did not improve for 20 epochs after the first 100 epochs, the training was ended, and the best model until that point was selected.

The numbers of hidden layers and hidden nodes used for the MLP models for the roll force and motor load prediction were selected through a grid search with the goal of having the lowest validation errors. The range of each parameter for the roll force and motor load prediction model is shown in Table 1.
Table 1. The range of each parameter for the roll force and motor load prediction models.

| Parameter                        | Range                  |
|----------------------------------|------------------------|
| The number of hidden layers      | 1–4                    |
| The number of hidden nodes       | 300–1000               |
| The number of hidden nodes (Last| 10–70, 100–300          |
| hidden layer)                    |                        |
| Learning rate                    | 0.1, 0.01, 0.001, 0.0001|
| Activation Function              | Sigmoid, Hyperbolic Tangent (Tanh) Rectified Linear Unit (ReLU) |

Tables 2–4 show some of the good results (MAPE) of the grid search for roll force prediction. The best results for each stand are boldfaced.

Table 2. Experimental results with various activation functions for the roll force prediction models. (The number of hidden layers: 1. The number of hidden nodes: 700. Learning rate: 0.001.)

| STAND 1 | STAND 2 | STAND 3 | STAND 4 | STAND 5 |
|---------|---------|---------|---------|---------|
| Sigmoid | 2.23    | 2.31    | 2.27    | 2.52    | 2.30    |
| Tanh    | 2.18    | 2.22    | 2.11    | 2.31    | 2.19    |
| ReLU    | 2.01    | 1.97    | 1.91    | 2.11    | 1.92    |

Table 3. Experimental results with various learning rates for the roll force prediction models. (The number of hidden layers: 1. The number of hidden nodes: 700. Activation function: ReLU.)

| STAND 1 | STAND 2 | STAND 3 | STAND 4 | STAND 5 |
|---------|---------|---------|---------|---------|
| 0.1     | 2.18    | 2.17    | 2.21    | 2.42    | 2.26    |
| 0.01    | 2.11    | 2.10    | 2.06    | 2.24    | 2.07    |
| 0.001   | 2.01    | 1.97    | 1.91    | 2.11    | 1.92    |
| 0.0001  | 2.09    | 2.00    | 1.98    | 2.25    | 2.07    |

Table 4. Experimental results with neural network architectures for the roll force prediction models. (Learning rate: 0.001. Activation function: ReLU.)

| STAND 1  | STAND 2  | STAND 3  | STAND 4 | STAND 5 |
|----------|----------|----------|---------|---------|
| 1–300    | 2.11     | 2.07     | 2.03    | 2.20    | 2.02    |
| 1–700    | 2.01     | 1.97     | 1.91    | 2.10    | 1.92    |
| 1–800    | 2.06     | 1.99     | 1.92    | 2.09    | 1.92    |
| 1–900    | 2.05     | 2.02     | 1.97    | 2.14    | 1.94    |
| 2–300/50 | 1.99     | 1.93     | 1.86    | 2.13    | 1.91    |
| 2–700/50 | **1.89** | 1.86     | **1.76** | **2.00** | 1.84    |
| 2–800/50 | 1.92     | 1.86     | 1.79    | 2.02    | **1.83** |
| 2–900/50 | 1.93     | **1.84** | 1.82    | 2.03    | 1.83    |
| 3–700/300/50 | 1.92 | 1.86 | 1.84 | 2.02 | 1.87 |
| 3–700/700/50 | 1.97 | 1.88 | 1.86 | 2.07 | 1.89 |
| 4–700/300/300/50 | 1.94 | 1.91 | 1.78 | 2.05 | 1.83 |
| 4–700/700/300/50 | 1.95 | 1.94 | 1.81 | 2.05 | 1.89 |
| 4–700/700/700/50 | 2.04 | 1.99 | 1.91 | 2.20 | 1.97 |

Tables 5–7 show some of the good results (MAPE) in the grid search for motor load prediction.
Table 5. Experimental results with various activation functions for the motor load prediction models. (The number of hidden layers: 1. The number of hidden nodes: 700. Learning rate: 0.001.)

|        | STAND 1 | STAND 2 | STAND 3 | STAND 4 | STAND 5 |
|--------|---------|---------|---------|---------|---------|
| Sigmoid| 2.51    | 2.30    | 2.28    | 2.39    | 2.49    |
| Tanh   | 2.45    | 2.20    | 2.14    | 2.23    | 2.31    |
| ReLU   | **2.37**| **2.13**| **2.01**| **2.11**| **2.15**|

Table 6. Experimental results with various learning rates for the motor load prediction models. (The number of hidden layers: 1. The number of hidden nodes: 700. Activation function: ReLU.)

|        | STAND 1 | STAND 2 | STAND 3 | STAND 4 | STAND 5 |
|--------|---------|---------|---------|---------|---------|
| 0.1    | 2.51    | 2.35    | 2.29    | 2.31    | 2.46    |
| 0.01   | 2.42    | 2.21    | 2.17    | 2.23    | 2.30    |
| 0.001  | **2.37**| **2.13**| **2.01**| **2.11**| **2.15**|
| 0.0001 | 2.47    | 2.22    | 2.15    | 2.23    | 2.38    |

Table 7. Experimental results with neural network architectures for the motor load prediction models (Learning rate: 0.001. Activation function: ReLU.)

|        | STAND 1 | STAND 2 | STAND 3 | STAND 4 | STAND 5 |
|--------|---------|---------|---------|---------|---------|
| 1–300  | 2.41    | 2.25    | 2.11    | 2.20    | 2.29    |
| 1–700  | 2.37    | 2.13    | 2.01    | 2.11    | 2.15    |
| 1–800  | 2.30    | 2.14    | 1.91    | 2.00    | 1.92    |
| 1–900  | 2.25    | 2.01    | 1.93    | 1.96    | 2.01    |
| 2–300/50| 1.97    | 1.65    | 1.40    | 1.55    | 1.53    |
| 2–700/50| **1.92**| 1.61    | **1.34**| 1.51    | **1.45**|
| 2–800/50| 2.00    | 1.59    | 1.36    | **1.43**| 1.50    |
| 2–900/50| 2.01    | **1.56**| 1.43    | 1.47    | 1.53    |
| 3–700/300/50 | 2.05 | 1.59 | 1.39 | 1.45 | 1.55 |
| 3–700/700/50 | 2.06 | 1.62 | 1.47 | 1.52 | 1.57 |
| 4–700/300/300/50 | 2.04 | 1.62 | 1.40 | 1.56 | 1.48 |
| 4–700/700/300/50 | 2.12 | 1.70 | 1.49 | 1.58 | 1.61 |
| 4–700/700/700/50 | 2.17 | 1.72 | 1.51 | **1.63**| 1.70 |

It is essential to rapidly process the upcoming jobs to be applied to a real cold-rolling process. The deeper the layer, the more difficult it is to operate the process in a short time. Therefore, the optimal parameters were selected to achieve an appropriate speed within a range. In other words, when the prediction performances were similar, the model was selected to have low complexity. In addition, it is noteworthy that large and complex networks did not always yield good performances even in terms of the prediction accuracy as shown in Tables 4 and 7. Finally, the number of hidden layers for all MLP models was determined to be two, and the numbers of hidden nodes for each hidden layer are shown in Tables 8 and 9.

Table 8. The numbers of hidden nodes for each layer for the roll force prediction model.

|        | STAND 1 | STAND 2 | STAND 3 | STAND 4 | STAND 5 |
|--------|---------|---------|---------|---------|---------|
| LAYER 1 | 700     | 900     | 700     | 700     | 800     |
| LAYER 2 | 50      | 50      | 50      | 50      | 50      |
Table 9. The numbers of hidden nodes for each layer for the motor load prediction model.

| Layer  | STAND 1 | STAND 2 | STAND 3 | STAND 4 | STAND 5 |
|--------|---------|---------|---------|---------|---------|
| Layer 1| 700     | 900     | 700     | 800     | 700     |
| Layer 2| 50      | 50      | 50      | 50      | 50      |

We used the rectified linear unit (ReLU) function as an activation function, and the learning rate—which determines the speed of learning—was set to $10^{-4}$. The maximum number of training epochs was 30,000, and the size of the minibatch was 32.

Other parameters were included for motor load balancing. In this case, the opinions of experts considering the speed, time, and accuracy were reflected for smooth operation of the process in determining the parameters. Table 10 shows the parameters determined by the physical limit, including the maximal motor loads for each rolling stand and the speed limits. The maximal motor load values were 5.497 MW (7372 HP) for the first stands and 6.397 MW (8579 HP) for the others.

There were two speed limits, namely the entry speed for the first stand and the delivery speed for the last stand, which were 5.83 m/s (350 m/min) and 26.67 m/s (1600 m/min), respectively. Some parameters must be set by the user, such as the maximum iteration number and the maximal allowable motor load difference, which are related to the stopping criteria. In this paper, the maximum iteration number for motor load balancing and the maximal allowable motor load difference were set as 300 and 3%, respectively.

Table 10. Physical limits of the tandem mill.

| Maximal Motor Load | Speed Limit |
|-------------------|-------------|
| STAND 1           | STAND 2     |
| 5.497 MW (7372 HP)| 6.397 MW (8579 HP) |
| STAND 3           | STAND 4     |
| 6.397 MW (8579 HP)| 6.397 MW (8579 HP) |
| STAND 5           | STAND 1 Entry |
| 6.397 MW (8579 HP)| 5.83 m/s (350 m/min) |
| STAND 6 Delivery  |
| 26.67 m/s (1600 m/min) |

Finally, for comparison with the proposed method in this paper, we also constructed Support Vector Regression (SVR) and Random Forest (RF) models. These models were also selected through grid search, which aimed at the lowest validation error.

5.2. Experimental Results

First, we compared the results of the roll force and motor load prediction models to those of the conventional physical models currently used by a steel manufacturer in South Korea. Table 11 presents the notations for the combination of the model and the data set.

Tables 12 and 13 show the MAPE and MSE for the constructed MLP models for roll force prediction compared with those from the conventional physical model. Although the physical model does not require separate data sets, such as training, validation, and test sets, the MAPE and MSE were measured for the same data sets as those used in the MLP models for comparison.

For every rolling stand, the MLP prediction model had lower errors for both MAPE and MSE when compared with the conventional physics model currently used in real world manufacturing. We also applied a paired $t$-test to test the statistical significance of the differences between the errors from the MLP model and those from the physical model. For every case, the $p$-value was lower than $10^{-8}$; thus, we can conclude that the proposed MLP model had significantly better performance compared with the conventional model.
Table 11. Notations for the model–data set combinations.

| Model–Data set                                         | Notation   |
|--------------------------------------------------------|------------|
| Conventional physical model–All datasets               | P_All      |
| Conventional physical model–Training dataset           | P_Train    |
| Conventional physical model–Validation dataset         | P_Val      |
| Conventional physical model–Test dataset               | P_Test     |
| SVR model–All datasets                                 | SVR_All    |
| SVR model–Training dataset                             | SVR_Train  |
| SVR model–Validation dataset                           | SVR_Val    |
| SVR model–Test dataset                                 | SVR_Test   |
| RF model–All datasets                                  | RF_All     |
| RF model–Train dataset                                 | RF_Train   |
| RF model–Validation dataset                            | RF_Val     |
| RF model–Test dataset                                  | RF_Test    |
| MLP model without motor load balancing–All datasets    | MLP_All    |
| MLP model without motor load balancing–Training dataset| MLP_Train  |
| MLP model without motor load balancing–Validation dataset| MLP_Val |
| MLP model without motor load balancing–Test dataset    | MLP_Test   |
| MLP model with motor load balancing–All datasets       | Bal_All    |
| MLP model with motor load balancing–Training dataset   | Bal_Train  |
| MLP model with motor load balancing–Validation dataset | Bal_Val    |
| MLP model with motor load balancing–Test dataset       | Bal_Test   |

Table 12. Comparison of the roll force prediction using MAPE.

| MAPE (%) | STAND 1 | STAND 2 | STAND 3 | STAND 4 | STAND 5 |
|----------|---------|---------|---------|---------|---------|
| P_Train  | 3.88    | 3.14    | 2.93    | 3.32    | 3.35    |
| P_Val    | 3.88    | 3.15    | 2.94    | 3.35    | 3.36    |
| P_Test   | 3.89    | 3.19    | 2.98    | 3.38    | 3.41    |
| MLP_Train| 1.62    | 1.56    | 1.52    | 1.71    | 1.54    |
| MLP_Val  | 1.89    | 1.84    | 1.76    | 2.00    | 1.83    |
| MLP_Test | 1.89    | 1.85    | 1.77    | 2.00    | 1.83    |

Table 13. Comparison of the roll force prediction using MSE.

| MSE (MN²/mm² (t²/mm²)) | STAND 1 | STAND 2 | STAND 3 | STAND 4 | STAND 5 |
|-------------------------|---------|---------|---------|---------|---------|
| P_Train                 | 0.68    | 0.35    | 0.29    | 0.32    | 0.24    |
| (7014.20)               | (3686.65)| (3004.71)| (3325.48)| (2506.96)|         |
| P_Val                   | 0.68    | 0.37    | 0.30    | 0.33    | 0.25    |
| (7107.42)               | (3790.28)| (3095.01)| (3435.96)| (2580.89)|         |
| P_Test                  | 0.68    | 0.37    | 0.29    | 0.33    | 0.24    |
| (6935.30)               | (3803.58)| (3054.02)| (3389.56)| (2533.75)|         |
| MLP_Train               | 0.12    | 0.10    | 0.09    | 0.09    | 0.07    |
| (1233.94)               | (1062.18)| (921.59)| (959.56)| (687.86)|         |
| MLP_Val                 | 0.15    | 0.13    | 0.11    | 0.11    | 0.08    |
| (1527.57)               | (1316.26)| (1103.69)| (1189.67)| (846.12)|         |
| MLP_Test                | 0.15    | 0.13    | 0.11    | 0.12    | 0.09    |
| (1529.80)               | (1337.20)| (1134.85)| (1233.80)| (879.92)|         |

Tables 14 and 15 show the results obtained from the MLP and conventional physical models for the motor load prediction using MAPE and MSE, respectively. As with the roll force prediction, the proposed method showed superior results to the conventional physical model in all cases for the motor load prediction tasks. Additionally, the p-value
from the paired \( t \)-test was lower than \( 10^{-8} \) in every case; thus, the proposed MLP model was significantly better than the conventional model.

Table 14. Comparison of the motor load prediction using MAPE.

|               | STAND 1 | STAND 2 | STAND 3 | STAND 4 | STAND 5 |
|---------------|---------|---------|---------|---------|---------|
| P_Train       | 28.92   | 24.79   | 28.81   | 23.52   | 23.09   |
| P_Val         | 28.93   | 24.77   | 28.81   | 23.49   | 23.03   |
| P_Test        | 28.78   | 24.68   | 28.70   | 23.43   | 22.98   |
| MLP_Train     | 1.65    | 1.38    | 1.12    | 1.23    | 1.22    |
| MLP_Val       | 1.93    | 1.57    | 1.33    | 1.42    | 1.45    |
| MLP_Test      | 1.92    | 1.56    | 1.34    | 1.43    | 1.45    |

Table 15. Comparison of the motor load prediction using MSE.

|               | STAND 1 | STAND 2 | STAND 3 | STAND 4 | STAND 5 |
|---------------|---------|---------|---------|---------|---------|
| P_Train       | 1.95    | 3.27    | 3.93    | 3.08    | 2.61    | \((1.09 \times 10^6)\) \((1.82 \times 10^6)\) \((2.18 \times 10^6)\) \((1.71 \times 10^6)\) \((1.45 \times 10^6)\) |
| P_Val         | 1.96    | 3.28    | 3.94    | 3.08    | 2.61    | \((1.09 \times 10^6)\) \((1.82 \times 10^6)\) \((2.19 \times 10^6)\) \((1.71 \times 10^6)\) \((1.45 \times 10^6)\) |
| P_Test        | 1.94    | 3.24    | 3.91    | 3.06    | 2.60    | \((1.08 \times 10^6)\) \((1.80 \times 10^6)\) \((2.17 \times 10^6)\) \((1.70 \times 10^6)\) \((1.44 \times 10^6)\) |
| MLP_Train     | 0.02    | 0.02    | 0.01    | 0.02    | 0.02    | \((8.36 \times 10^3)\) \((1.25 \times 10^4)\) \((7.60 \times 10^3)\) \((1.03 \times 10^4)\) \((8.89 \times 10^3)\) |
| MLP_Val       | 0.02    | 0.03    | 0.02    | 0.02    | 0.02    | \((9.54 \times 10^3)\) \((1.41 \times 10^4)\) \((8.96 \times 10^3)\) \((1.19 \times 10^4)\) \((1.06 \times 10^4)\) |
| MLP_Test      | 0.02    | 0.03    | 0.02    | 0.02    | 0.02    | \((9.53 \times 10^3)\) \((1.36 \times 10^4)\) \((8.95 \times 10^3)\) \((1.18 \times 10^4)\) \((1.07 \times 10^4)\) |

The performance improvement for the motor load prediction was much larger than for the roll force prediction. The major reason is that motor load prediction is generally not considered an important factor in the cold-rolling process because it does not directly affect the quality or productivity unless the motor load exceeds the physical limit. However, in this study, accurate prediction of the motor load is also important for the subsequent motor load balancing procedure.

We also compared the proposed method with baseline of machine learning methodologies, such as SVR and RF. Tables 16 and 17 show the results obtained from MLP, SVR, and RF for roll force prediction using MAPE and MSE. Compared to SVR, the MLP model outperformed the training, validation, and test data in all stands. Compared with the RF, in the training and validation data, the RF showed good results; however, in the test data, which is the most important criterion for judging the generalized performance, the MLP model showed better results (MSE), except for stand 4.
Table 16. Comparison with a baseline of machine learning methodologies for roll force prediction using MAPE.

|                | MAPE (%) |
|----------------|----------|
|                | STAND 1  | STAND 2  | STAND 3  | STAND 4  | STAND 5  |
| SVR_Train      | 5.65     | 5.26     | 5.10     | 5.13     | 4.76     |
| SVR_Val        | 5.64     | 5.26     | 5.11     | 5.14     | 4.75     |
| SVR_Test       | 5.61     | 5.25     | 5.09     | 5.12     | 4.76     |
| RF_Train       | 1.05     | 0.97     | 0.91     | 0.95     | 0.91     |
| RF_Val         | 1.34     | 1.23     | 1.15     | 1.20     | 1.15     |
| RF_Test        | 2.25     | 2.10     | 1.95     | 2.04     | 1.96     |
| MLP_Train      | 1.62     | 1.56     | 1.52     | 1.71     | 1.54     |
| MLP_Val        | 1.89     | 1.84     | 1.76     | 2.00     | 1.83     |
| MLP_Test       | 1.89     | 1.85     | 1.77     | 2.00     | 1.83     |

Table 17. Comparison with a baseline of machine learning methodologies for roll force prediction using MSE.

|                | MSE (MN^2/mm^2 (t^2/mm^2)) |
|----------------|-----------------------------|
|                | STAND 1  | STAND 2  | STAND 3  | STAND 4  | STAND 5  |
| SVR_Train      | (11363.30) | (9326.35) | (8132.82) | (7595.32) | (5433.51) |
| SVR_Val        | (11374.18) | (9411.45) | (8181.41) | (7646.91) | (5446.32) |
| SVR_Test       | (10910.68) | (9133.65) | (7932.03) | (7454.48) | (5370.89) |
| RF_Train       | (459.50) | (358.10) | (282.85) | (264.93) | (206.84) |
| RF_Val         | (736.41) | (567.04) | (432.58) | (412.77) | (315.71) |
| RF_Test        | (2017.80) | (1586.36) | (1266.79) | (1212.96) | (930.99) |
| MLP_Train      | (1233.94) | (1062.18) | (921.59) | (959.56) | (687.86) |
| MLP_Val        | (1527.57) | (1316.26) | (1103.69) | (1189.67) | (846.12) |
| MLP_Test       | (1529.80) | (1337.20) | (1134.85) | (1233.80) | (879.92) |

Tables 18 and 19 show the results obtained from MLP, SVR, and RF for motor load prediction in MAPE and MSE, respectively. Similar to the results of the roll force prediction model, when compared to SVR, the MLP model showed superior performance in all stands for the training, validation, and test data. Compared to RF, in training and validation data, there were stands where RF showed slightly better results; however, in the test data, which are the most important criterion for judging the generalized performance, the MLP model showed better results in all stands.
Table 18. Comparison with a baseline of machine learning methodologies for motor load prediction using MAPE.

|                | STAND 1 | STAND 2 | STAND 3 | STAND 4 | STAND 5 |
|----------------|---------|---------|---------|---------|---------|
| SVR_Train      | 19.63   | 18.71   | 20.72   | 18.83   | 21.30   |
| SVR_Val        | 19.57   | 18.55   | 20.36   | 18.92   | 20.94   |
| SVR_Test       | 19.48   | 18.48   | 20.29   | 18.80   | 20.95   |
| RF_Train       | 1.74    | 1.25    | 1.26    | 1.19    | 1.39    |
| RF_Val         | 2.19    | 1.59    | 1.60    | 1.51    | 1.77    |
| RF_Test        | 3.68    | 2.70    | 2.71    | 2.55    | 3.01    |
| MLP_Train      | 1.65    | 1.38    | 1.12    | 1.23    | 1.22    |
| MLP_Val        | 1.93    | 1.57    | 1.33    | 1.42    | 1.45    |
| MLP_Test       | 1.92    | 1.56    | 1.34    | 1.43    | 1.45    |

Table 19. Comparison with a baseline of machine learning methodologies for motor load prediction using MSE.

|                | STAND 1 | STAND 2 | STAND 3 | STAND 4 | STAND 5 |
|----------------|---------|---------|---------|---------|---------|
| SVR_Train      | 0.94    | 1.85    | 2.03    | 1.96    | 2.10    |
|                | ($5.21 \times 10^5$) | ($1.03 \times 10^6$) | ($1.13 \times 10^6$) | ($1.09 \times 10^6$) | ($1.17 \times 10^6$) |
| SVR_Val        | 0.93    | 1.83    | 1.98    | 1.96    | 2.10    |
|                | ($5.18 \times 10^5$) | ($1.02 \times 10^6$) | ($1.10 \times 10^6$) | ($1.09 \times 10^6$) | ($1.17 \times 10^6$) |
| SVR_Test       | 0.92    | 1.82    | 0.01    | 1.94    | 2.03    |
|                | ($5.11 \times 10^5$) | ($1.01 \times 10^6$) | ($1.09 \times 10^6$) | ($1.08 \times 10^6$) | ($1.13 \times 10^6$) |
| RF_Train       | 0.01    | 0.01    | 0.01    | 0.02    | 0.02    |
|                | ($7.46 \times 10^3$) | ($8.40 \times 10^3$) | ($7.96 \times 10^3$) | ($8.20 \times 10^3$) | ($9.80 \times 10^3$) |
| RF_Val         | 0.02    | 0.02    | 0.02    | 0.02    | 0.03    |
|                | ($1.18 \times 10^4$) | ($1.32 \times 10^4$) | ($1.25 \times 10^4$) | ($1.29 \times 10^4$) | ($1.57 \times 10^4$) |
| RF_Test        | 0.06    | 0.07    | 0.06    | 0.06    | 0.08    |
|                | ($3.20 \times 10^4$) | ($3.85 \times 10^4$) | ($3.46 \times 10^4$) | ($3.55 \times 10^4$) | ($4.27 \times 10^4$) |
| MLP_Train      | 0.02    | 0.02    | 0.01    | 0.02    | 0.02    |
|                | ($8.36 \times 10^3$) | ($1.25 \times 10^4$) | ($7.60 \times 10^3$) | ($1.03 \times 10^4$) | ($8.89 \times 10^3$) |
| MLP_Val        | 0.02    | 0.02    | 0.02    | 0.02    | 0.02    |
|                | ($9.54 \times 10^3$) | ($1.41 \times 10^4$) | ($8.96 \times 10^3$) | ($1.19 \times 10^4$) | ($1.06 \times 10^4$) |
| MLP_Test       | 0.02    | 0.02    | 0.02    | 0.02    | 0.02    |
|                | ($9.53 \times 10^3$) | ($1.36 \times 10^4$) | ($8.95 \times 10^3$) | ($1.18 \times 10^4$) | ($1.07 \times 10^4$) |

Finally, we assessed the effect of the proposed motor load balancing algorithm on the productivity. Table 20 shows the average coil speed at each stand from the motor load balancing algorithm and that of the conventional setting, which is the given speed, similar to the input data initially used for the roll force and motor load prediction. A faster production speed directly results in higher productivity. The rolling speed for every stand increased with motor load balancing compared to the original setup speed. These speed differences were also statistically significant in all cases according to the paired t-test results with all p-values being lower than $10^{-8}$. 

Table 20. The speed comparison results.

| Unit: m/s (m/min) | STAND 1 Entry | STAND 1 Delivery | STAND 2 Delivery | STAND 3 Delivery | STAND 4 Delivery | STAND 5 Delivery | STAND 6 Delivery |
|------------------|--------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|
| MLP_Train        | 4.71 (283.01)| 5.89 (353.87)   | 5.75 (454.41)   | 7.57 (589.52)   | 9.82 (733.26)   | 12.22 (880.12)  | 14.88 (892.56)  |
| MLP_Val          | 4.71 (282.83)| 5.89 (353.69)   | 5.75 (454.22)   | 7.57 (589.24)   | 9.82 (733.06)   | 12.23 (879.97)  | 14.87 (892.38)  |
| MLP_Test         | 4.72 (283.29)| 5.90 (354.10)   | 7.58 (454.67)   | 9.83 (589.84)   | 12.23 (733.73)  | 14.39 (880.73)  | 16.01 (974.06)  |
| Bal_Train        | 5.42 (325.22)| 7.36 (441.60)   | 9.57 (574.43)   | 12.20 (731.95)  | 14.39 (863.41)  | 16.01 (960.68)  | 16.24 (974.19)  |
| Bal_Val          | 5.42 (325.13)| 7.36 (441.60)   | 9.57 (574.47)   | 12.20 (732.04)  | 14.39 (863.56)  | 16.01 (960.88)  | 16.24 (974.38)  |
| Bal_Test         | 5.42 (325.07)| 7.36 (441.63)   | 9.57 (574.36)   | 12.20 (731.82)  | 14.39 (863.25)  | 16.01 (960.69)  | 16.24 (974.11)  |

Table 21 shows the ratio of the average speed to the maximal speed from the physical limits in Table 10. There were two limit speeds, one for the entry speed of the first stand, which was 5.83 m/s (350 m/min), and the other for the delivery speed of the final stand, which was 26.67 m/s (1600 m/min). Note that the delivery speed of the final stand is directly related to the productivity. The entry speed for the first stand increased by about 12 percentage points, and the delivery speed for the last stand increased by about 5 percentage points compared to the original setup value, which did not consider the balance among the motor loads of the rolling stands. These differences were also tested using the paired t-test and were found to be statistically significant.

Table 21. Comparison of the usage rates of the speed.

| Unit: % | STAND 1 Entry Speed | STAND 6 Delivery Speed |
|---------|----------------------|------------------------|
| MLP_Train | 80.86                | 55.78                  |
| MLP_Val   | 80.81                | 55.77                  |
| MLP_Test  | 80.94                | 55.82                  |
| Bal_Train | 92.92                | 60.89                  |
| Bal_Val   | 92.89                | 60.05                  |
| Bal_Test  | 98.88                | 60.88                  |

The comparison results for the ratio of the predicted motor loads and the physical limits are presented in Table 22. The results from the motor load balancing algorithm are compared with the motor load prediction results without balancing. First, the balanced motor load usage was higher than the non-balanced usage for all stands. The statistical significance was tested with a paired t-test. Additionally, the difference between the maximal and minimal motor load usage was much lower for the balanced motor loads compared with for the non-balanced loads. The results indicated that the proposed motor load balancing algorithm was effective for increasing and balancing the motor usage as well as increasing the production speed.

Table 22. Comparison of the usage rates of the motor load.

| Unit: % | STAND 1 | STAND 2 | STAND 3 | STAND 4 | STAND 5 | MAX-MIN |
|---------|---------|---------|---------|---------|---------|---------|
| MLP_Train | 70.38   | 84.50   | 83.53   | 85.63   | 82.91   | 15.24   |
| MLP_Val   | 70.38   | 84.52   | 83.53   | 85.75   | 82.94   | 15.36   |
| MLP_Test  | 70.06   | 84.18   | 83.19   | 85.34   | 82.59   | 15.28   |
| Bal_Train | 86.03   | 86.91   | 87.06   | 88.59   | 88.78   | 2.75    |
| Bal_Val   | 86.07   | 86.94   | 87.09   | 88.68   | 88.83   | 2.76    |
| Bal_Test  | 85.96   | 86.84   | 86.97   | 88.57   | 88.74   | 2.78    |
6. Conclusions

In this paper, we proposed roll force and motor load prediction methods along with a motor load balancing algorithm for the cold-rolling process based on MLP models to improve the quality, increase the productivity, and reduce redundant costs. First, we constructed MLP models to predict the roll forces and motor loads of each rolling stand in a tandem mill to obtain accurate setup values for quality improvement. Next, we suggested and implemented a motor load balancing algorithm that maximized the production speed for productivity increases and balanced the motor usage among the rolling stands for redundant cost reductions.

The proposed method was validated using actual field data from a steel manufacturing company in South Korea and compared to the conventional physical model currently used in manufacturing. The data set contained 70,353 instances of 128 different steel types and 78 input variables in total. Based on the MAPE values, the constructed MLP model for roll force prediction showed better results than the conventional model. Specifically, the prediction errors decreased to 1% to 2% compared to error rates of 3% for the conventional physical model.

The proposed MLP model also performed better than the conventional model in predicting the motor loads for rolling stands. The proposed motor load balancing algorithm increased the rolling speeds of the stands compared to the original setup. Specifically, the delivery speed of the final stand, which is directly related to the productivity, was increased by about 10 percent.

Further, the motor load balancing algorithm reduced the gap between the maximal and minimal motor loads among stands below a preset value, which can reduce redundant costs by preventing severe overload of a specific roll. Therefore, we can conclude that the proposed method achieved quality improvement through accurate prediction of the roll forces and motor loads for the setup, increased productivity by increasing the production speed, and reduced redundant costs by balancing the motor loads among the rolling stands in a tandem mill.

This study has the following limitations. First, the number of instances for different steel types is invariant; hence, steel types with a small number of instances may have bad prediction results. Some techniques for solving unbalanced data problems, including oversampling and weighted costs, could improve the prediction performance. Second, the constructed prediction models may not work well for unseen steel types, especially for steel types whose components lie outside the domain of the training instances. This is an example of an extrapolation problem, and neural network models are somewhat weak in dealing with this task in general. Applying techniques for extrapolation, such as linear approximation from the boundary, may be helpful for the prediction of instances from unseen steel types. Finally, the criteria for determining the thickness change rate in the motor load balancing algorithm may improve the performance of the proposed method.

**Author Contributions:** Conceptualization, S.L. and Y.S.; methodology, S.L. and Y.S.; software, S.L.; validation, S.L. and Y.S.; formal analysis, S.L.; investigation, S.L.; resources, Y.S.; data curation, S.L. and Y.S.; writing—original draft preparation, S.L.; writing—review and editing, Y.S.; visualization, S.L.; supervision, Y.S.; project administration, Y.S.; funding acquisition, Y.S. Both authors have read and agreed to the published version of the manuscript.

**Funding:** This research was funded by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT: Ministry of Science and ICT) (No. 2020R1C1C1003425).

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

1. Ma, S.H.; Wen, Z.G.; Chen, J.N.; Wen, Z.C. Mode of circular economy in China’s iron and steel industry: A case study in Wu’an city. J. Clean. Prod. 2014, 64, 505–512. [CrossRef]

2. Du, T.; Shi, T.; Liu, Y.; Ye, J.B. Energy Consumption and its influencing factors of iron and steel enterprise. J. Iron. Steel Res. Int. 2013, 20, 8–13. [CrossRef]

3. Dong, L.; Zhang, H.; Fujita, T.; Ohnishi, S.; Li, H.; Fujii, M.; Dong, H. Environmental and economic gains of industrial symbiosis for Chinese iron/steel industry: Kawasaki’s experience and practice in Liuzhou and Jinan. J. Clean. Prod. 2013, 59, 226–238. [CrossRef]

4. Hsu, C.C.; Chen, M.S. Intelligent maintenance prediction system for LED wafer testing machine. J. Intell. Manuf. 2016, 27, 335–342. [CrossRef]

5. Choudhary, A.K.; Harding, J.A.; Tiwari, M.K. Data mining in manufacturing: A review based on the kind of knowledge. J. Intell. Manuf. 2009, 20, 501–521. [CrossRef]

6. Kovačić, M.; Šarler, B. Application of the genetic programming for increasing the soft annealing productivity in steel industry. Mater. Manuf. Process. 2009, 24, 369–374. [CrossRef]

7. Baur, C.; Wee, D. Manufacturing’s Next Act; McKinsey & Company: Vancouver, BC, USA, 2015; Volume 6.

8. Wuest, T.; Ergens, C.; Thoen, K.D. An approach to monitoring quality in manufacturing using supervised machine learning on product state data. J. Intell. Manuf. 2014, 25167–1180. [CrossRef]

9. Posada, J.; Toro, C.; Barandiaran, I.; Oyarzun, D.; De Amicis, R.; Pinto, E.B.; Eisept P.; Döllner, J.; Vallarino, I. Visual computing as a key enabling technology for industri 4.0 and industrial internet. IEEE Comput. Graph. Appl. 2015, 35, 26–40. [CrossRef]

10. Lu, Y. Industry 4.0: A survey on technologies, applications and open research issues. J. Ind. Inf. Integr. 2017, 6, 1–10. [CrossRef]

11. Ge, Z.; Song, Z.; Ding, S.X.; Huang, B. Data mining and analytics in the process industry: The role of machine learning. IEEE Access 2017, 5, 20590–20616. [CrossRef]

12. Huang, Q. High-precision quality inspection for screws using artificial intelligence technology. In Proceedings of the CIB W78-International Council for Research and Innovation in Building and Construction, Newcastle, UK, 18–20 September 2019.

13. Zheng, J.; Lu, C.; Hao, C.; Chen, D.; Guo, D. Improving the generalization ability of deep neural networks for cross-domain visual recognition. IEEE Trans. Cogn. Dev. Syst. 2020. [CrossRef]

14. Kang, S. Joint modeling of classification and regression for improving faulty wafer detection in semiconductor manufacturing. J. Intell. Manuf. 2020, 31, 319–326. [CrossRef]

15. Venkatusubramanian, V.; Rengaswamy, R.; Kavuri, S.N.; Yin, K. A review of process fault detection and diagnosis: Part III: Process history based methods. Comput. Chem. Eng. 2003, 27, 327–346. [CrossRef]

16. Ge, Z.; Song, Z.; Gao, F. Review of recent research on data-based process monitoring. Ind. Eng. Chem. Res. 2013, 52, 3543–3562. [CrossRef]

17. Qin, S.J. Survey on data-driven industrial process monitoring and diagnosis. Annu. Rev. Control 2012, 36, 220–234. [CrossRef]

18. Lieber, D.; Stolpe, M.; Konrad, B.; Deuse, J.; Morik, K. Quality prediction in interlinked manufacturing processes based on supervised & unsupervised machine learning. Procedia CIRP 2013, 7, 193–198.

19. Kang, G.W.; Liu, H.B. Surface defects inspection of cold rolled strips based on neural network. In Proceedings of the 2005 International Conference on Machine Learning and Cybernetics, Guangzhou, China, 18–21 August 2005; Volume 8.

20. Carbonneau, R.; Laframboise, K.; Vahidov, R. Application of machine learning techniques for supply chain demand forecasting. Eur. J. Oper. Res. 2008, 184, 1140–1154. [CrossRef]

21. Mao, H.; Alizadeh, M.; Menache, I.; Kandula, S. Resource management with deep reinforcement learning. In Proceedings of the 15th ACM Workshop on Hot Topics in Networks, Atlanta, GA, USA, 9–10 November 2016; pp. 50–56.

22. Tian, H.; Mao, Z.; Wang, Y. Hybrid modeling of molten steel temperature prediction in LF. ISIJ Int. 2008, 48, 58–62. [CrossRef]

23. Tian, H.X.; Mao, Z.Z. An ensemble ELM based on modified AdaBoost. RT algorithm for predicting the temperature of molten steel in ladle furnace. IEEE Trans. Autom. Sci. Eng. 2009, 7, 73–80. [CrossRef]

24. Lin, Y.; Zhang, J.; Zhong, J. Application of neural networks to predict the elevated temperature flow behavior of a low alloy steel. Comput. Mater. Sci. 2008, 43, 752–758. [CrossRef]

25. Ghasiari, J.; Jannesari, H.; Vatani, M. Artificial neural network predictors for mechanical properties of cold rolling products. Adv. Eng. Softw. 2012, 45, 91–99. [CrossRef]

26. Guo, Z.; Sha, W. Modelling the correlation between processing parameters and properties of maraging steels using artificial neural network. Comput. Mater. Sci. 2004, 29, 12–28. [CrossRef]

27. Perzyk, M.; Kochański, A.W. Prediction of ductile cast iron quality by artificial neural networks. J. Mater. Process. Technol. 2001, 109, 305–307. [CrossRef]

28. Ozerdem, M.S.; Kolukisa, S. Artificial Neural Network approach to predict mechanical properties of hot rolled, nonresulfurized, AISI 10xx series carbon steel bars. J. Mater. Process. Technol. 2008, 199, 437–439. [CrossRef]

29. He, S.G.; He, Z.; Wang, G.A. Online monitoring and fault identification of mean shifts in bivariate processes using decision tree learning techniques. J. Intell. Manuf. 2013, 24, 25–34. [CrossRef]

30. Li, C.; Sanchez, R.V.; Zurita, G.; Cerrada, M.; Cabrera, D.; Vásquez, R.E. Multimodal deep support vector classification with homologous features and its application to gearbox fault diagnosis. Neurocomputing 2015, 168, 119–127. [CrossRef]
31. Seera, M.; Lim, C.P.; Loo, C.K. Motor fault detection and diagnosis using a hybrid FMM-CART model with online learning. *J. Intell. Manuf.* 2016, 27, 1273–1285. [CrossRef]

32. Bai, Y.; Sun, Z.; Zeng, B.; Long, J.; Li, L.; de Oliveira, J.V.; Li, C. A comparison of dimension reduction techniques for support vector machine modeling of multi-parameter manufacturing quality prediction. *J. Intell. Manuf.* 2019, 30, 2245–2256. [CrossRef]

33. García, V.; Sánchez, J.S.; Rodríguez-Picón, L.A.; Méndez-González, L.C.; de Jesús Ochoa-Domínguez, H. Using regression models for predicting the product quality in a tubing extrusion process. *J. Intell. Manuf.* 2019, 30, 2535–2544. [CrossRef]

34. Lee, K.B.; Kim, C.O. Recurrent feature-incorporated convolutional neural network for virtual metrology of the chemical mechanical planarization process. *J. Intell. Manuf.* 2020, 31, 73–86. [CrossRef]

35. Miao, Q.; Zhang, X.; Liu, Z.; Zhang, H. Condition multi-classification and evaluation of system degradation process using an improved support vector machine. *Microelectron. Reliab.* 2017, 75, 23–232. [CrossRef]

36. Li, D.C.; Chen, W.C.; Liu, C.W.; Lin, Y.S. A non-linear quality improvement model using SVR for manufacturing TFT-LCDs. *J. Intell. Manuf.* 2012, 23, 835–844. [CrossRef]

37. Liu, L.; Tian, S.; Xue, D.; Zhang, T.; Chen, Y. Industrial feedforward control technology: A review. *J. Intell. Manuf.* 2019, 30, 2819–2833. [CrossRef]

38. Bagheripoor, M.; Bisadi, H. Application of artificial neural networks for the prediction of roll force and roll torque in hot strip rolling process. *Appl. Math. Model.* 2013, 37, 4593–4607. [CrossRef]

39. Larkiola, J.; Myllykoski, P.; Nylander, J.; Korhonen, A. Prediction of rolling force in cold rolling by using physical models and neural computing. *J. Mater. Process. Technol.* 1996, 60, 381–386. [CrossRef]

40. Cho, S.; Cho, Y.; Yoon, S. Reliable roll force prediction in cold mill using multiple neural networks. *IEEE Trans. Neural Netw.* 1997, 8, 874–882.

41. Son, J.S.; Lee, D.M.; Kim, I.S.; Choi, S.G. A study on on-line learning neural network for prediction for rolling force in hot-rolling mill. *J. Mater. Process. Technol.* 2005, 164, 1612–1617. [CrossRef]

42. Mahmoodkhani, Y.; Wells, M.A.; Song, G. Prediction of roll force in skin pass rolling using numerical and artificial neural network methods. *Ironmak. Steelmak.* 2016, 44, 281–286. [CrossRef]

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

44. Zhang, S.; Zhang, A.; Ma, Y.; Zhu, W. Intrinsic plasticity based inference acceleration for spiking multi-layer perceptron. *IEEE Access* 2019, 7, 73685–73693. [CrossRef]

45. Goodfellow, I.; Bengio, Y.; Courville, A. *Deep Learning*; MIT Press: Cambridge, UK, 2016; Volume 1.