Journal of
Manufacturing and Materials Processing

MDPI

Article

Multi-Task Learning for Predictive Modeling and Optimization of Manufacturing Processes: A Stretch-Reducing Mill Application

Laura Antonini * and Claudio Giardini

Department of Management, Information and Production Engineering, University of Bergamo, Via Pasubio 7B, 24044 Dalmine, BG, Italy; claudio.giardini@unibg.it
* Correspondence: laura.antonini@unibg.it

Abstract

Stretch-reducing mills are complex multi-stand systems in which the final tube thickness depends on the interaction between inter-stand tensions, angular speeds, material deformation, and process conditions. In this study, a multi-task neural network is proposed to model and optimize the operation of a six-stand stretch-reducing mill. A structured pipeline was used to generate a feasible dataset to train and test the model. The proposed architecture employs the inter-stand tensions in input, utilizes a shared network to get the general relationships between variables, and predicts process status, tube thickness, angular speed, and the neutral radius of each stand with dedicated task-specific heads. The model achieved high performance across all the tasks on training, validation, and testing sets. The final tube thickness predictions get a very low error, showing the ability of the proposed network to gather hidden relationships throughout the process. The trained model was integrated with the differential evolution optimization algorithm to obtain the feasible inter-stand distribution and stand angular speeds to achieve the desired final tube thickness while respecting the feasibility constraint set. The results demonstrate that the proposed framework can provide support for the decision-making process optimization, representing an effective tool for intelligent modeling in stretch-reducing mill monitoring and control.

Keywords: stretch-reducing mill; multi-task neural network; process optimization; intelligent process control

Check for updates

Academic Editor: Fengchao Liu

Received: 18 August 2026
Revised: 24 September 2026
Accepted: 5 October 2026
Published: 8 October 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

1. Introduction

Multi-stand milling processes represent a critical stage in the production of tubes and pipes. the stretch-reducing mill (SRM) is a manufacturing process that progressively deforms the tube through a sequence of rolling stands, where the inter-stand axial tension is controlled by adjusting the relative speed of the stands to obtain the desired geometry [1,2]. The relationships between mechanical, thermal, and material variables are complex and strongly nonlinear, making the modeling and optimization of this process particularly challenging [3]. One of the most important variables is the inter-stand tension, since the axial tensile stress distribution affects the tube wall thickness throughout the process and, therefore, the final pipe quality [4]. These forces influence the entire process, and, if they are not properly controlled, they can damage or even break the tube [5]. The stretch-reducing mill is characterized by complex interdependencies among process parameters,

which can easily lead to unstable situations [6]. The most frequent ones are slippage and drawing: when the tube is moving slower than the maximum tangent speed of the stand, there is a slippage situation; when the tube is faster, a drawing operating condition occurs [5]. Ensuring that these tensions remain within acceptable limits while achieving the target geometry is the main challenge in the tube manufacturing industry, with a direct impact on product quality and process efficiency [5]. Traditionally, rolling mill process models have been developed using a physics-based approach, combining analytical process descriptions, empirical corrections, and computational models to study and optimize rolling operations [3,7]. However, even though these models provide interpretable and consistent predictions, they require detailed knowledge of properties and parameters that are difficult to obtain in practice, and their accuracy can be affected by the operating conditions [8]. These limitations have pushed a growing interest in data-driven approaches [9,10]. For this reason, neural-network models have been investigated as alternative data-driven tools for predicting rolling force, torque, and temperature [11].

The artificial neural network has been applied to predict rolling force, thickness, and other process variables across a range of milling conditions, demonstrating the ability of these models to capture the nonlinear relationships from operational data without explicit physical models [12,13]. More recent works have combined predictors with optimization algorithms, showing that neural networks can be used as surrogate models. In addition, they can be easily integrated into optimization loops to define the optimal operating conditions to obtain the desired final tube geometry [14–16]. Despite these progresses, most of the approaches perform the predictions of individual process variables and parameters as independent and separate tasks, using one network per output quantity predicted [17]. These separate approaches does not consider the physical interdependencies that exist between process variables and parameters, limiting predictive accuracy and efficiency.

In a previous study [17], separate artificial neural networks were used to predict the process parameters of a six-stand SRM. The present work extends that approach by replacing the independent models with a single multi-task neural network (MTNN) composed of shared layers and task-specific heads to exploit the physical relationships among the different process outputs within a unified model.

Multi-task learning (MTL) offers a principled framework to address these problems [18]. With an MTL approach, there is only a single model composed by a shared network layer to learn the general interdependencies across tasks, and task-specific heads to perform specific tasks predictions [19–22]. However, when different prediction tasks are optimized, the weighting of the task-specific losses becomes a critical aspect, especially when classification and regression objectives are combined [23]. Compared to single-task models, this approach can improve generalization, reducing overfitting through regularization, and decreasing memory requirements [24,25]. In the deep learning field, the MTL architecture is widely adopted and has demonstrated high performance across different industrial fields [19]. Talking about the stretch-reducing milling process, the problem of finding the optimal inter-stand tensions is a global and multidimensional problem. In practice, the main parameters that can be manually set and controlled in the process are the angular speeds of the rolling stands, which can be derived from the optimal tension's distribution throughout the process [26]. The objective function used for tension optimization includes discontinuous penalty terms, such as a penalty for stands in non-acceptable process states, which make gradient-based methods unreliable. Differential evolution algorithms [27] are well suited to these problems: they work without any gradient information, maintain a population of candidate solutions to avoid a local minimum, and they have been shown to converge on continuous nonlinear optimization cases across different industrial applications [14].

From the reviewed literature, three main limitations can be identified. Physics-based SRM models provide physically consistent descriptions of the process, but they generally require detailed information that may be difficult to obtain in industrial practice. Then, data-driven models developed for rolling are mainly focused on the prediction of individual quantities, and they often treat each output as an independent modeling problem. This limits their ability to exploit the physical dependencies among parameters. Optimization-oriented studies are rarely formulated to simultaneously provide multi-output SRM prediction and feasible operating conditions. Therefore, a clear gap remains in the development of an integrated SRM modeling framework able to learn the coupled process behavior, predict multiple physically related outputs at the same time, and support constrained optimization of the inter-stand tension profile and corresponding angular speeds.

In this study, an integrated framework combining a multi-task neural network and an optimization algorithm, differential evolution, to simultaneously make predictions and process optimization of a multi-stand stretch-reducing mill is presented. The MTNN is trained on data generated by a structured pipeline, and it starts from the inter-stand tensions to predict process status, tube thickness, angular speeds, and neutral radius for each stand. Then, the trained network is used as a surrogate model within the differential evolution optimization. The aim is to find the best tension profile, and therefore the angular speeds to set for the process, to obtain the desired final tube thickness while respecting feasibility constraints through a penalty on non-acceptable process states and a penalty on excessive differences between consecutive tensions, while extrapolation risk is controlled by restricting the optimization variables to the 5th–95th percentile range of the generated dataset. The proposed approach enables data-driven identification of feasible operating conditions using data generated by a simulation model and provides predictions of all the relevant process variables together with the optimal process configuration.

2. Materials and Methods

In this work, data were generated through a structured pipeline made in MatLab®R2024b and then analyzed by a multi-task neural network (MTNN) coupled with an optimization algorithm in Python 3.12.13. The timing measurements were obtained in Google Colab, version 0.0.1a2, using the CPU runtime.

2.1. Data Generation and Processing

In this study, the dataset was entirely generated by a simulation pipeline representing the operating logic and physical constraints of the SRM.

The starting points are all the process information, such as the number of stands of the milling process and their diameter, the working parameters (temperature, creep behavior, and others), and the initial tube diameter. The neutral radius is determined as the radial position at which the tube velocity equals the tangential velocity of the corresponding stand. The process status is determined from the relative velocity between the tube and the stand. Slippage occurs when the tube velocity exceeds the maximum tangential velocity of the stand, whereas drawing occurs when the tube velocity is lower than the stand velocity. The simulated states are classified as acceptable, slippage, or drawing and are used as the three classes of the MTNN status task.

To generate feasible inter-stand tensions combination, a custom software module in MatLab®R2024b was employed. In this work, these values indicate the corresponding material stresses, expressed in N/mm^2. To elaborate, the module specified the minimum, maximum, and step size of tensions (σ_{min}, σ_{max}, and $\Delta\sigma$) and iteratively searched across the whole stress range. In addition, consecutive stands could not have tension values that differ excessively to avoid tube damage and ruptures. The generated tension combinations were

first filtered according to the bounds and physically meaningful tension-trend constraints. Once the tension combinations were obtained, their values were used to compute the process status, entering and exiting tube thickness, angular speed (ω) and neutral radius of each stand. Special attention is given to the angular speeds, because they are the only parameters that can be manually set by the operator in the process. Even if in this pipeline the data are derived, this approach allowed identifying limit situations by analyzing the behavior of the milling process. Finally, all the data extracted were formatted for the neural network training. Figure 1 shows the data generation workflow. In this study, the term "reference" is used to define the thickness values calculated by the simulator.

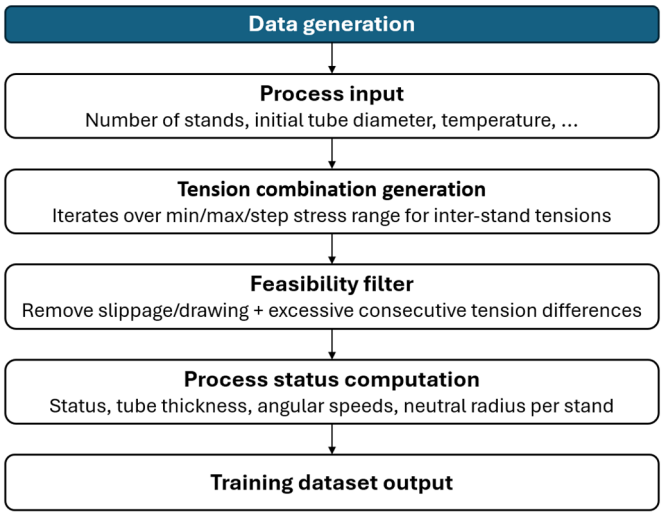

Figure 1. Data generation workflow. The "Process input" box indicates additional process-input variables not explicitly listed in the schematic.

2.2. Multi-Task Neural Networks

To capture the hidden and complex relationships between variables in the stretch-reducing milling process, a multi-task neural network was developed [22,28]. The architecture follows a shared-trunk, multi-head design. The architectural choices were defined according to both the physical structure of the SRM process and the different nature of the prediction tasks. A common shared network received the inter-stand tensions as input called T, which contains $I - 1$ values for a process of I stands. Three fully connected layers were included, with the hyperbolic tangent (Tanh) activation functions. The Tanh activation function was selected because all input and output regression variables were standardized before training. This shared network was able to capture the overall relationships between the tensions and the SRM behavior, exploiting the physical interdependencies. Since process status, angular speeds, thickness, and the neutral radius are not independent, a shared feature extractor can learn hidden relationships better than using separate and independent models. Then, four different specialized heads were defined, one for each prediction needed: status (s), angular speed (ω), thickness (t), and neutral radius (rn). Each head started from the 32-dimensional shared output, defined as a compromise between model complexity and predictive capability, and it was composed of one hidden layer with the Tanh activation, followed by a linear output layer. The number of output nodes corresponded to the number of forecast values for each task: number of values (n) multiplied by the number of classes (m) for classification, and number of values (n) for the regressions.

The common layers allow the network to learn a shared representation of the global deformation behavior, while the task-specific heads preserve the possibility of learning different parameters. This is particularly relevant because the outputs are physically related but not identical. Therefore, the adopted architecture was selected as a compact

configuration able to provide stable convergence and accurate predictions across all the considered tasks.

The MTNN was simultaneously addressing classification for status prediction, and regression problems, for angular speed, thickness, and neutral radius. The loss formulation was also defined according to the nature of the outputs. Accordingly, two different loss functions were employed: cross-entropy loss for the classification task and mean squared error (MSE) for the regressions [29]. The total training objectives was defined as the sum of the following loss components:

$$\mathcal{L}_{total} = \mathcal{L}_s + \mathcal{L}_\omega + \mathcal{L}_t + \mathcal{L}_{rn} \quad (1)$$

The status values constant to zero throughout the process were excluded to the calculation of the loss, together with the first angular speed as it represents the reference value and it is equal to one.

The dataset was split following the rule of 70% for training, 15% for validation, and 15% for testing. All input and output variables were standardized using StandardScaler, and fitted exclusively on the training subset, to prevent numerically dominant variables from affecting the loss function excessively [30]. The fitted scalers were then applied without refitting to the validation and test subsets. The network was optimized by using the Adam algorithm [31], with a learning rate scheduler to improve convergence stability [32]. To evaluate the network performances, the confusion matrix for the status, and the root mean squared error (RMSE) and R^2 correlation coefficient for the regressions were computed. Figure 2 reports the network scheme.

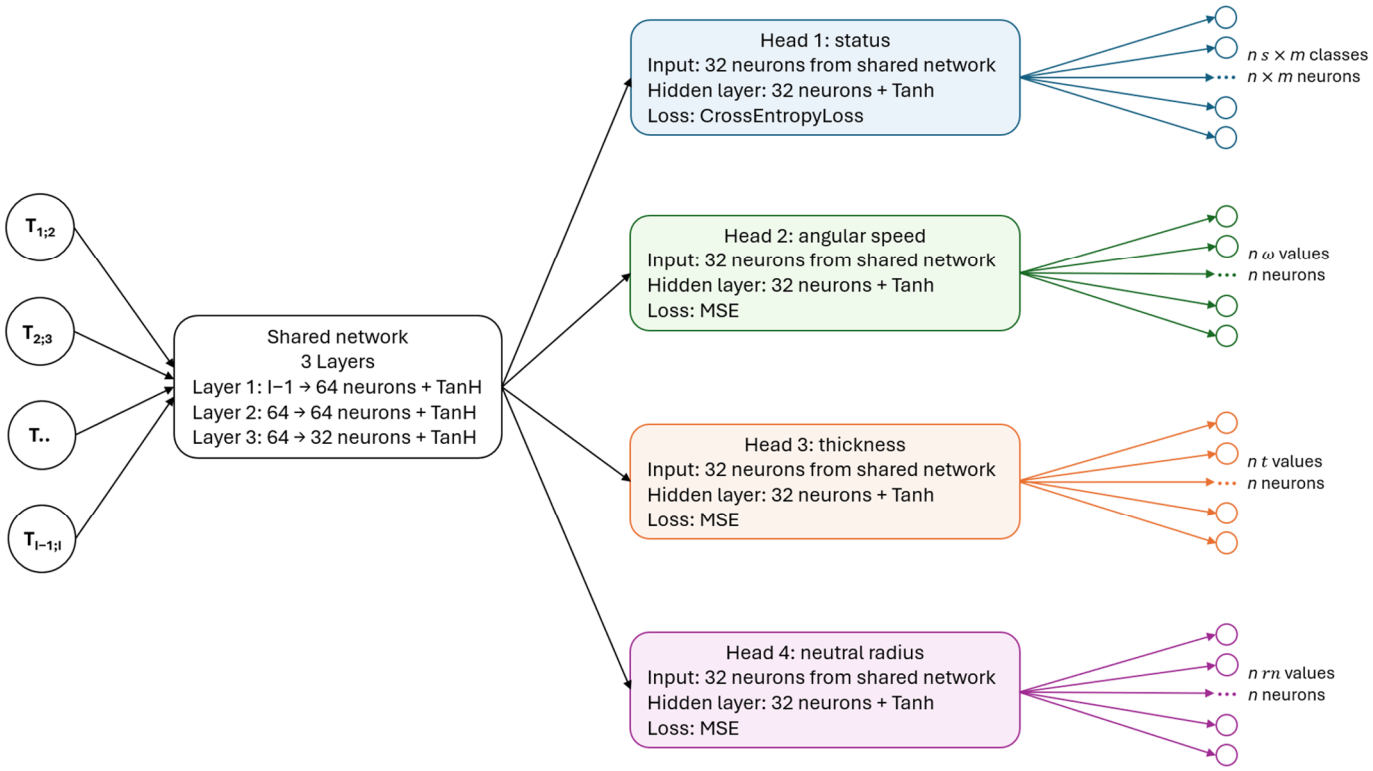

Figure 2. Multi-task neural network scheme. Ellipses indicate omitted intermediate inputs and output neurons; only representative nodes are shown for clarity.

2.3. Optimization Algorithm

Since the practical aim of the work is to define the best angular speed combination that the operator should set on the process to obtain the desired final tube thickness, the

network was coupled with an optimization algorithm. Given the target thickness, the algorithm searched for the best inter-stand tensions combination such that the MTNN's predicted output matched the target as closely as possible.

The optimization was performed using the differential evolution algorithm [27], implemented through the SciPy library [33,34]. This algorithm is a population-based global optimization method that maintains a set of candidate solutions and iteratively refines them by combining existing solutions through vector difference operations. The differential evolution does not require an initial guess because it can perform multidimensional space exploration, making it well suited to problems where the relationships between variables are complex and nonlinear. Each possible solution was assessed by the MTNN, which predicts statuses, omegas, final thicknesses, and neutral radii. The optimizer then evaluates an objective function and updates the population accordingly.

The chosen objective function corresponded to the squared error between the predicted and target final thickness, penalizing unacceptable and unrealistic situations. Specifically, the objective function contains three penalty-related terms: the process-status penalty, the regularization term, and the consecutive-tension-difference penalty. First, a penalty was added whenever the predicted status differed from the acceptable one. This term was selected because solutions leading to slippage or drawing conditions cannot be considered feasible from an operational point of view, even when the predicted final thickness is close to the target value. To control extrapolation risk, the optimization bounds were defined according to the 5th and 95th percentiles of the generated dataset, restricting the tensions to this range to prevent limit milling conditions. Another penalty was introduced to avoid excessively significant differences between consecutive tensions that could cause the rupture of the tube during the process. For this purpose, the maximum accepted difference value ($\Delta\sigma$) between consecutive tensions was defined, representing material stresses and expressed in N/mm^2. Therefore, considering I the number of stands of the process, $t_I^{predicted}$ the thickness predicted by the MTNN, t_I^{target} the final desired thickness, λ_{status}, λ_{reg}, and λ_Δ the scalar penalty weights, and P_{status}, P_{reg}, and $P_{\Delta\sigma}$ the penalties, the objective functions resulted:

$$f(T) = \left| t_I^{predicted} - t_I^{target} \right| + \lambda_{status} P_{status} + \lambda_{reg} P_{reg} + \lambda_\Delta P_{\Delta\sigma} \quad (2)$$

The maximum allowed difference between consecutive tensions was set to $\Delta\sigma_{\max} = 5$, and the convergence tolerance to 10^{-6}. The range and tension-difference penalties were used as regularization terms to avoid unacceptable tension profiles while still allowing the optimizer to minimize the final-thickness error. Since five inter-stand tensions are optimized to achieve a single final thickness target, the inverse problem is generally non-unique. Different tension profiles may therefore produce comparable final thickness values. The differential evolution algorithm identifies one feasible solution according to the defined objective-function, bounds, and penalty terms.

Once the optimization was finished, the algorithm returns the tension profile that minimizes the objective function, and the full set of predictions (process status, angular speed, tube thickness along the stands, and neutral radius). In particular, the angular speed omega represents the primary operational outcome, as it is the only parameter that can be set and controlled by the operator to achieve the target thickness. Figure 3 reports the optimization flow.

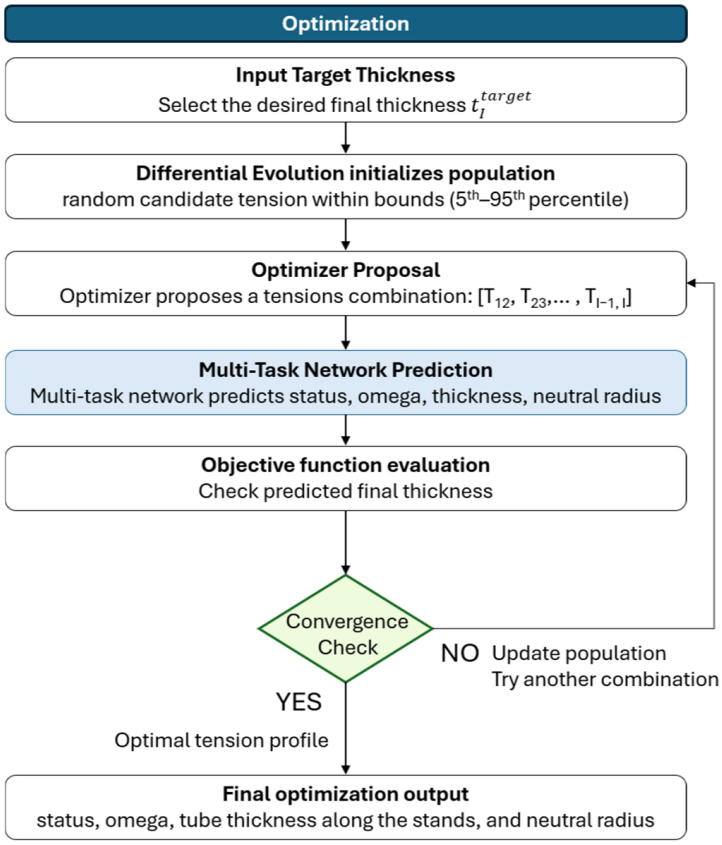

Figure 3. Optimization flow.

3. Results and Discussion

This section presents the results obtained applying the methodology explained above to a six-stand stretch-reducing mill. Following the case study presentation, both the MTNN and the optimization outcomes are reported and discussed in terms of accuracy, physical consistency, and capability of the optimized solution to provide feasible operating conditions for the stretch-reducing mill.

3.1. Case Study Presentation

The case study considers a six-stand stretch-reducing mill. The five input variables correspond to the inter-stand tensions $T_{1;2}$, $T_{2;3}$, $T_{3;4}$, $T_{4;5}$, and $T_{5;6}$ (Figure 4). The first-stand angular speed was used as the reference value and was set equal to one, while the angular speeds of the remaining stands were calculated from the process conditions. By applying the data generation pipeline reported in Section 2, the dataset to work with the neural network was created. The tension profiles were generated within the prescribed minimum and maximum values and were filtered to retain physically meaningful distributions, including the required tension trend and the maximum difference between consecutive tensions. The resulting profiles were then evaluated by the simulation pipeline and used to construct the dataset for the MTNN. The final dataset contained 2591 tension profiles, each corresponding to one simulated operating condition of the six-stand SRM. Each profile contains five inter-stand tension values and the corresponding simulated outputs, including process status, angular speeds, tube thicknesses, and neutral radii.

Table 1 reports the data specifications related to a six-stand SRM process. In detail, the angular speeds of the first stand are always set to one and the others are adjusted accordingly. As a result, the number of output neurons of each head of the MTNN were: 12 neurons for the status head, five neurons for the angular speed, and six neurons for the

thickness and the neutral radius. Each task-specific path therefore spans five linear layers in total (three shared plus two within the head), and the complete network comprises eleven linear layers.

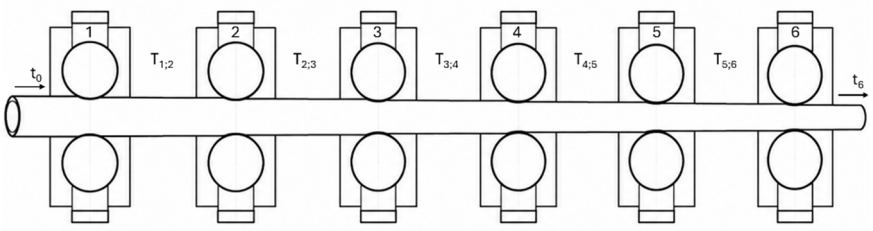

Figure 4. Six-stand stretch-reducing mill scheme.

Table 1. Variable specifications.

Variable Group	Columns	N of Variables	Role in the Model	Task Type	Predicted by the Network	Loss Function	Notes
Inter-stand tensions	$T_{1;2}$, $T_{2;3}$, $T_{3;4}$, $T_{4;5}$, $T_{5;6}$	5	Input	-	No	-	Main input variables describing the stress distribution between consecutive stands
Process status	$status_1$, $status_2$	2	Output excluded from loss	Classification	No	-	Constant values equal to 0; excluded because they do not provide useful learning information
Process status	$status_3$, …, $status_6$	4	Output	Classification	Yes	Cross-entropy	Classes: 0 = acceptable, 1 = slippage, 2 = drawing
Angular speeds	$omega_1$	1	Output excluded from loss	Regression	No	-	Reference angular speed, constant and equal to 1
Angular speeds	$omega_2$, …, $omega_6$	5	Output	Regression	Yes	MSE	Predicted angular speeds of stands 2–6
Tube thickness	$thickness_1$, …, $thickness_6$	6	Output	Regression	Yes	MSE	Predicted tube thickness evolution along the six stands
Neutral radius	rn_1, …, rn_6	6	Output	Regression	Yes	MSE	Predicted neutral radius for each stand

For the optimization reported in this study, the weights of the objective-function terms were initially set to $\lambda_{\text{status}} = 10.0$, $\lambda_{\text{reg}} = 0.001$, and $\lambda_{\Delta} = 1000.0$. These weights were selected empirically according to the hierarchy of the optimization objectives, considering the final-thickness error the primary objective. The objective function minimized the squared error between the predicted and target final thickness, and added a penalty proportional to the number of stands in a non-acceptable process state, a regularization term based on the normalized distance of the tensions from the center of their allowed ranges, and a penalty for exceeding the maximum difference between consecutive tensions. The tension bounds were defined using the 5th and 95th percentiles of the generated dataset. For the six-stand mill, the resulting bounds were $T_{12} \in [0, 10]$, $T_{23} \in [0, 10]$, $T_{34} \in [2.5, 30]$, $T_{45} \in [0, 10]$, and $T_{56} \in [0, 10]$. The maximum allowed difference between consecutive tensions was set to $\Delta\sigma_{\max} = 5$. Differential evolution was run with a maximum of 1000 iterations, a population-size multiplier of 20, a convergence tolerance of 10^{-6}, and a random seed. The mutation and recombination parameters corresponded to the SciPy default values.

3.2. Analysis of the Results and Discussion

Using the data generated through the structured pipeline explained above, the MTNN was trained, validated, and tested. Starting with the training and validation phases, the losses were analyzed. As can be seen from Figure 5, the loss curves present a rapid decrease in the first epochs: starting from high values of 3.5 for training and 2.7 for validation, the losses reach values lower than 0.5 in the first 20–30 epochs. Then, both the loss curves stabilize on values close to zero, showing that the model is stable and performs consistently on validation data sampled from the same simulated operating domain as the training data. Since the split was random and the data were generated on a structured parameter grid, this result should be interpreted as evidence of interpolation capability.

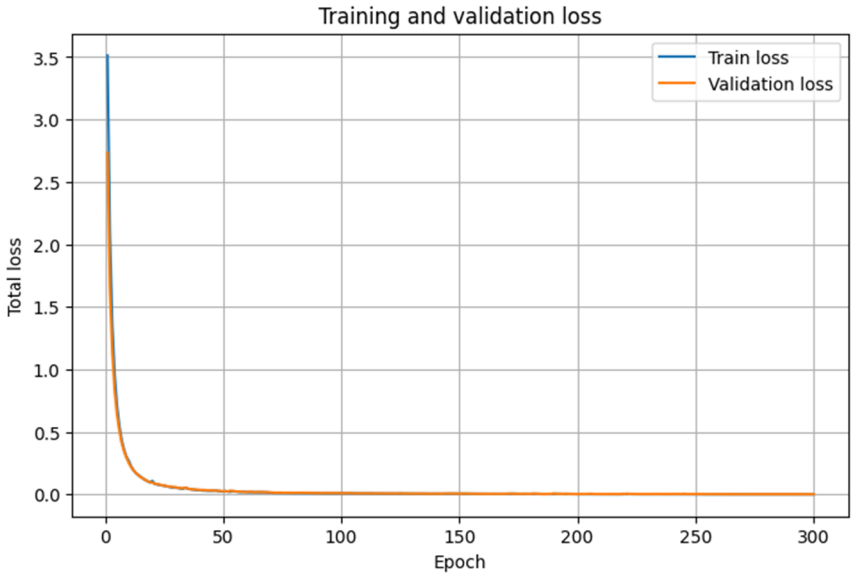

Figure 5. Training and validation loss.

The best validation loss registered is equal to 0.00197 around epoch 289, while the training total is 0.001552. Both the training and validation loss curves are overlapping throughout the whole training, showing that the MTNN does not exhibit evident overfitting.

The first task-specific head of the MTNN was used to predict the process status as acceptable, slippage, or drawing milling conditions. Looking at the global confusion matrix, Figure 6, reporting the predictions on stands 3, 4, 5 and 6 (status 1 and 2 are excluded because they are constant values for all simulated operating conditions), it is possible to notice that the model correctly classified the three process conditions with no misclassifications among the classes. Since the test set contained all three process conditions, the result shows that the model can distinguish and identify the operating conditions within the simulated domain represented in the random test split. Even if the global performance on classification is perfect on the random test split, validation of more challenging splits would be useful to evaluate the robustness using less represented milling situations.

The prediction of the final tube thickness is particularly critical since this variable is the main driver for process control. In this study, the reference thickness values are the outputs of the SRM simulation used to generate the dataset. Analyzing the thickness predictions made by the dedicated task-specific head, it is evident that the MTNN was able to predict the tube thickness with a high level of accuracy. Figure 7 shows that the points are distributed along the bisector line, indicating that the model accurately got the relationships between inter-stand tensions and final tube thickness. On the test set, the RMSE was equal to 3.98×10^{-4} mm, confirming the high performance of the proposed

network. Therefore, the MTNN was able to get the general thickness trend throughout the process, while also predicting small variations in the tube thickness values.

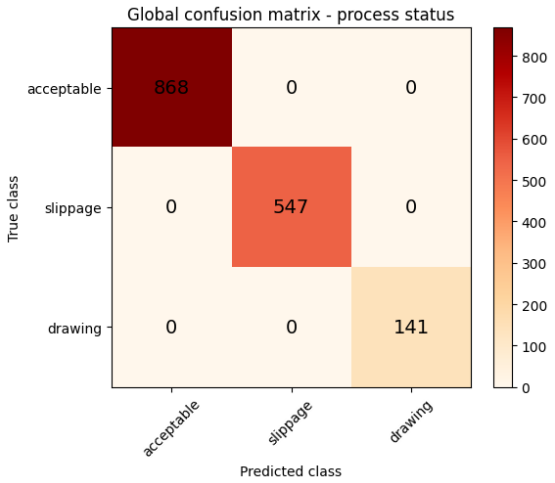

Figure 6. Global confusion matrix about process-status predictions.

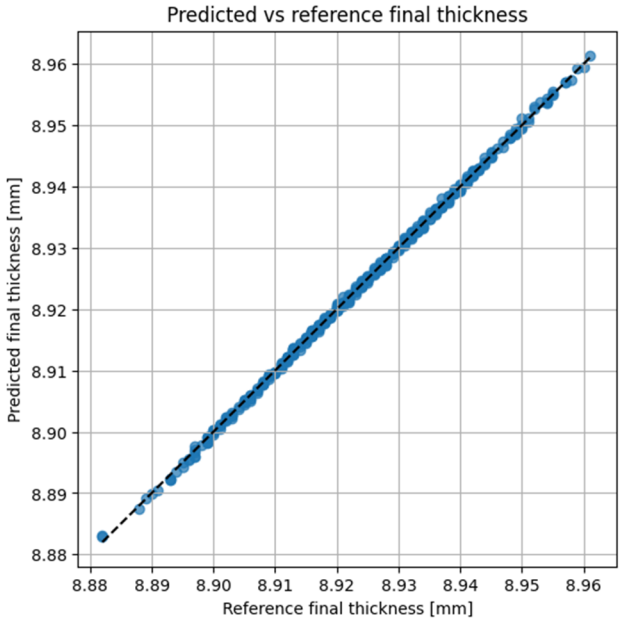

Figure 7. Comparison between the thickness values predicted by the MTNN and the reference final tube thickness values obtained from the simulation.

The global performance of the MTNN is summarized in Table 2. The results are reported separately for the training, validation, and test subsets using accuracy for process-status classification and RMSE and R^2 for the regression tasks. This table is used to assess the predictive accuracy and generalization of the network before applying it to the optimization procedure. The metrics show the model performance for both the classification of the status and the regression of thickness, angular speed, and neutral radius. The training, validation, and test values are similar, indicating that the model keeps a high predictive capability on unseen data, without evident overfitting. The status classification achieved a high accuracy on all the subsets, consistently with the global confusion matrix shown above. A future validation on experimental data or a more severe split would be useful to assess the classifier robustness. The regression tasks confirm the high level of performance of the proposed model. The thickness and the angular speed are correctly predicted throughout the entire

process. This is particularly relevant, as these variables constitute the most significant process parameters and are used for process control and monitoring purposes. The neutral radius has the highest RMSE, due to the physical scale of the variable: the absolute values of neutral radii are greater than the thickness or angular speed ones. Therefore, the RMSE value of 0.366198 is relatively small compared with the numerical scale of the neutral-radius variable. In addition, the R^2 values are close to one, indicating a high predictive capability for radius estimation. The global loss values further confirm the model stability. The practical interpretation of the reported errors must be considered in relation to the scale of each output variable and to the purpose of the model.

Table 2. Network performance across training, validation, and testing.

Task	Metric	Training	Validation	Test
Process status	Accuracy	1.0	1.0	1.0
Thickness	RMSE	0.000292	0.000312	0.000314
Thickness	R^2	0.999993	0.999993	0.999992
Angular speed	RMSE	0.002779	0.003052	0.003038
Angular speed	R^2	0.999829	0.999793	0.999790
Neutral radius	RMSE	0.328438	0.354048	0.366198
Neutral radius	R^2	0.999803	0.999758	0.999754
Overall model	Total loss	0.001552	0.001968	0.001959

A computational-cost analysis was performed for the reported optimization. Differential evolution required 41,006 objective-function evaluations before reaching the convergence criterion. The total execution time of the MTNN-based optimization was 54.2 s, corresponding to an average time of 1.3 ms per objective-function evaluation.

Compared with the separate-network approach previously used for the same SRM process [17], the proposed MTNN provides a unified model for the simultaneous prediction of process status, tube thickness, angular speed, and neutral radius. The shared representation preserves the relationships among the physically coupled outputs, while the task-specific heads allow each prediction task to be treated according to its own classification or regression nature.

Once the MTNN was tested, the differential evolution optimization algorithm was integrated to obtain the feasible inter-stand tension distribution to derive the optimal angular speeds to achieve the desired target thickness of 8.95 mm. The optimization results are reported in Tables 3–5: Table 3 reports the tension profile and the corresponding process-status predictions obtained for the selected feasible solution, while the subsequent tables provide the associated angular speeds, thickness evolution, and neutral-radius values. All the stands in the optimal solution have an acceptable process status, and the tensions between consecutive stands satisfy the imposed constraint of a maximum material stress $\Delta\sigma$ equal to 5. The tension between the central stands is higher than the others, and the tension distribution remains acceptable: the tension increases toward the central stands and then decreases toward the outlet stands, without strong variations between consecutive stands. This behavior is consistent with the imposed maximum difference between adjacent tensions and helps avoid sudden changes in the deformation conditions along the mill. Because the relationship between the inter-stand tensions and the final thickness is nonlinear and coupled across the stands, even relatively small changes in the tension profile may affect the predicted thickness evolution or the process status. It reaches the target final thickness while maintaining acceptable process states, bounded tension values, and admissible differences between consecutive tensions.

Table 3. Optimization results.

Stand	Status	Status Label	Tension Label	Tension to Next Stand	Angular Speed	Thickness	Neutral Radius
1	0.0	acceptable	$T_{1;2}$	2.85	1.000	8.60	123.30
2	0.0	acceptable	$T_{2;3}$	3.97	1.085	8.79	116.95
3	0.0	acceptable	$T_{3;4}$	8.95	0.936	8.87	137.72
4	0.0	acceptable	$T_{4;5}$	4.74	1.162	8.92	111.37
5	0.0	acceptable	$T_{5;6}$	0.90	1.193	8.94	108.33
6	0.0	acceptable	-	-	1.112	8.94	116.48

Table 4. Predicted versus target final tube thickness.

Target final thickness	8.95 mm
Predicted final thickness	8.94 mm
Absolute error	0.01
Percentage error	0.09%

Table 5. Inter-stand tension values in the optimal solution.

$\Delta\sigma$ between consecutive tensions	$T_{1;2}$, $T_{2;3}$	1.12
	$T_{2;3}$, $T_{3;4}$	4.98
	$T_{3;4}$, $T_{4;5}$	4.21
	$T_{4;5}$, $T_{5;6}$	3.84
Maximum $\Delta\sigma$	$T_{2;3}$, $T_{3;4}$	4.98

The corresponding angular speeds were obtained from the MTNN. The feasible optimized thickness shows an increase throughout the entire process, achieving the target. The error between the predicted and the target final thickness results as 0.01, corresponding to 0.09%. This value is small but still higher than the RMSE on the predictions during the testing phase because the optimization objective function also includes penalty terms related to the process feasibility, regularization, and maximum tensions between consecutive stands. The penalty terms guide the optimization toward a physically consistent solution among potentially multiple feasible profiles. This tension profile was selected because it reaches the target thickness while satisfying all feasibility requirements: all predicted process states are acceptable, all tension values remain within the selected range, and the maximum difference between consecutive tensions is lower than the threshold. The feasible inter-stand tensions identified were additionally tested in the structured pipeline used to generate the dataset to verify the consistency of the predicted results. The agreement between the MTNN predictions and the results of the original simulator verifies the surrogate consistency with the simulation model. This simulation confirmed the same values of angular speed, thickness and neutral radius obtained during the optimization procedure.

Overall, the optimization results confirmed the capability of the MTNN proposed to predict and to be used as a surrogate model supporting the surrogate-based process analysis and preliminary optimization and monitoring. The similar values of training, validation, and testing confirmed the generalization capability of the network on unseen data. Thickness errors demonstrated that the model captured the relationship between inter-stand tensions and the tube dimension. In addition, the MTNN integrated with the differential evolution algorithm showed that the model could be used as a surrogate tool for process control. The optimization algorithm allowed the identification of the feasible optimal inter-stand tension distribution to generate the associated angular speeds

associated with acceptable operating conditions. The objective function did not include only the thickness error, but also some penalty terms related to feasibility, regularization, and maximum tensions between consecutive stands. The predicted final thickness was close to the target value, showing that the proposed framework can provide feasible operating conditions while maintaining dimensional accuracy.

In conclusion, the results confirm the multi-task architecture as a valid alternative to separate neural networks. The shared network allowed us to capture the process interdependencies, while the task-specific heads make highly accurate predictions.

Since the MTNN was trained and tested using data generated by a simulation, its reported performance refers to the operating domain represented by the simulation model. A domain gap may exist between the simulated and real industrial environments. Once experimental data become available, some strategies could be used, such as transfer learning or fine-tuning, hybrid physics-informed models, uncertainty quantification, and out-of-distribution detection. Therefore, the present framework should be regarded as a simulation-based surrogate model and a preliminary decision-support tool.

4. Conclusions

In this study, a multi-task neural network approach was developed to model and optimize a multiple-stand stretch-reducing milling manufacturing process. The proposed architecture was designed to overcome the fragmentation of separate models, allowing us to better capture the complex and nonlinear relationships between process variables. Through a shared network with four task-specific heads, starting from the inter-stand tensions, the process status, tube thickness, angular speed, and neutral radius of each stand were predicted. The results demonstrated that the multi-task neural network provides high levels of accuracy across all the predicted parameters. In particular, the final tube thickness was estimated with a low error, confirming the ability of the model to describe the thickness evolution throughout the entire process. All the other metrics confirmed that the model was able to generalize on unseen data without evident overfitting within the simulated operating domain under the adopted split approach. The trained multi-task network was then employed as a surrogate model for the differential evolution optimization algorithm to identify the feasible inter-stand distribution for the target thickness. The optimization results showed that the proposed framework can provide suitable solutions, satisfying the physical constraint set, and maintaining a limited error on the desired thickness.

Overall, the study shows that a multi-task neural network represents a promising alternative to the multiple separate models for stretch-reducing mill monitoring and control. The results demonstrate the ability of the MTNN to reproduce the outputs of the first-principles simulator within the investigated operating domain. The integration with a constrained optimization algorithm further confirms its applicability to supporting the decision-making process, enhancing the identification of the feasible inter-stand tension distribution and corresponding angular speeds. The framework can support preliminary process analysis and optimization by reducing the need for repeated simulator evaluations.

Further developments may focus on another tuning of the objective function weights, introducing additional constrains and comparing different optimization and splitting techniques. Moreover, a validation of experimental data will be necessary to further assess the reliability and robustness of the proposed approach in real industrial environments.

Author Contributions: Conceptualization, L.A. and C.G.; methodology, L.A.; validation, C.G.; formal analysis, L.A.; investigation, C.G.; data curation, L.A. and C.G.; writing—original draft preparation, L.A.; writing—review and editing, C.G.; visualization, L.A.; supervision, C.G.; project administration, C.G. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding author.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

SRM	Stretch-reducing mill
MTL	Multi-task learning
MTNN	Multi-task neural network
RMSE	Root mean squared error

References

1. Yu, H.; Chen, Y.; Bai, X. Finite element analysis for wall thickness variation of seamless tube in stretch reducing rolling process. *Adv. Mater. Res.* **2011**, *291–294*, 532–536. [CrossRef]
2. Hu, J.; Yang, S.; Huang, Y.; Wang, X.; Chen, J. A New Correction Theory and Verification on the Reducing Rate Distribution for Seamless Tube Stretch-Reducing Process. *Crystals* **2022**, *12*, 1296. [CrossRef]
3. Li, X.; He, Y.; Ding, J.; Luan, F.; Zhang, D. Predicting hot-strip finish rolling thickness using stochastic configuration networks. *Inf. Sci.* **2022**, *611*, 677–689. [CrossRef]
4. Giardini, C.; Bugini, A.; Pellegrini, G. Creep Influence in a Stretch Reducing Mill Computer Simulation. In *Proceedings of the Twenty-Eighth International*; Macmillan Education: London, UK, 1990; pp. 455–460. [CrossRef]
5. Bayoumi, L.S. Analysis of flow and stresses in a tube stretch-reducing hot rolling schedule. *Int. J. Mech. Sci.* **2003**, *45*, 553–565. [CrossRef]
6. Giardini, C.; Maccarini, G.C.; Zavanella, L.; Bugini, A. A Stretch-Reducing Mill (SRM) Computer Simulation. In *Proceedings of the Twenty-Seventh International Matador Conference*; Macmillan Education: London, UK, 1988; pp. 333–339. [CrossRef]
7. Ojeda-López, A.; Botana-Galvín, M.; González-Rovira, L.; Botana, F.J. Numerical Simulation as a Tool for the Study, Development, and Optimization of Rolling Processes: A Review. *Metals* **2024**, *14*, 737. [CrossRef]
8. Udofia, E.; Messer, L.; Greivel, G.; Newman, A.; Thomas, B.G. A Statistical-Based Model of Roll Force During Commercial Hot Rolling of Steel. *Metals* **2025**, *15*, 1346. [CrossRef]
9. Gao, R.X.; Krüger, J.; Merklein, M.; Möhring, H.C.; Váncza, J. Artificial Intelligence in manufacturing: State of the art, perspectives, and future directions. *CIRP Ann.* **2024**, *73*, 723–749. [CrossRef]
10. Plathottam, S.J.; Rzonca, A.; Lakhnori, R.; Iloeje, C.O. *A Review of Artificial Intelligence Applications in Manufacturing Operations*; John Wiley and Sons Inc.: Hoboken, NJ, USA, 2023. [CrossRef]
11. Öznergiz, E.; Özsoy, C.; Delice, I.I.; Kural, A. Comparison of empirical and neural network hot-rolling process models. *Proc. Inst. Mech. Eng. B J. Eng. Manuf.* **2009**, *223*, 305–312. [CrossRef]
12. Martinetz, T.; Protzel, P.; Gramckow, O.; Sörgel, G. Neural Network Control for Steel Rolling Mills. In *Neural Networks: Artificial Intelligence and Industrial Applications*; Springer: London, UK, 1995; pp. 280–286. [CrossRef]
13. Park, C.Y.; Kim, J.W.; Kim, B.; Lee, J. Prediction for Manufacturing Factors in a Steel Plate Rolling Smart Factory Using Data Clustering-Based Machine Learning. *IEEE Access* **2020**, *8*, 60890–60905. [CrossRef]
14. Hu, Z.; Wang, S.; Li, Y.; Wei, L.; Sun, H. Optimisation of steel rolling schedule based on evolutionary multi-tasking transfer algorithm. *Comput. Oper. Res.* **2024**, *169*, 106743. [CrossRef]
15. Thakare, N.; Scharifi, E. Modelling of the cold rolling process for mild steel using physics-constrained neural network. *J. Intell. Manuf.* **2026**, *37*, 4363–4378. [CrossRef]
16. Hu, Z.; Wei, Z.; Ma, X.; Sun, H.; Yang, J. Multi-parameter deep-perception and many-objective autonomous-control of rolling schedule on high speed cold tandem mill. *ISA Trans.* **2020**, *102*, 193–207. [CrossRef]
17. Antonini, L.; Quarto, M.; Giardini, C.; Pellegrini, G.; Zambelli, M. Advanced Control of Stretch-Reducing Mills Using Artificial Intelligence. *Key Eng. Mater.* **2026**, *1049*, 1–11. [CrossRef]
18. Zhang, Y.; Yang, Q. An overview of multi-task learning. *Natl. Sci. Rev.* **2018**, *5*, 30–43. [CrossRef]
19. Crawshaw, M. Multi-Task Learning with Deep Neural Networks: A Survey. *arXiv* **2020**, arXiv:2009.09796.
20. Vandenhende, S.; Georgoulis, S.; Van Gansbeke, W.; Proesmans, M.; Dai, D.; Van Gool, L. Multi-Task Learning for Dense Prediction Tasks: A Survey. *IEEE Trans. Pattern Anal. Mach. Intell.* **2021**, *44*, 3614–3633. [CrossRef]
21. Yang, Y.; Hospedales, T. Deep Multi-task Representation Learning: A Tensor Factorisation Approach. *arXiv* **2017**, arXiv:1605.06391.

22. Abdelsamie, M.M.; Azab, S.S.; Hefny, H.A. Deep multi-task learning: A review of concepts, methods, and cross-domain applications. *Int. J. Data Sci. Anal.* **2026**, *21*, 77. [CrossRef]
23. Kendall, A.; Gal, Y.; Cipolla, R. Multi-Task Learning Using Uncertainty to Weigh Losses for Scene Geometry and Semantics. *arXiv* **2018**, arXiv:1705.07115. [CrossRef]
24. Vafaeikia, P.; Namdar, K.; Khalvati, F. A Brief Review of Deep Multi-task Learning and Auxiliary Task Learning. *arXiv* **2020**, arXiv:2007.01126.
25. Ruder, S. An Overview of Multi-Task Learning in Deep Neural Networks. *arXiv* **2017**, arXiv:1706.05098.
26. Vargas, J.M.; Pagliano, C.; Medina, F.; Muratori, F.; Paganin, F.; Paiuk, J. Process Optimization for the Manufacturing of Seamless Steel Tubes in a Continuous Mandrel Mill and a Stretch Reducing Mill. *IFAC Proc. Vol.* **1989**, *22*, 147–153. [CrossRef]
27. Storn, R.; Price, K. Differential Evolution-A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces. *J. Glob. Optim.* **1997**, *11*, 341–359. [CrossRef]
28. Zhang, K.; Cao, Z.; Wang, Y.; Rui, Y. A novel multi-task learning-based electrical energy consumption prediction method with application to a hot strip rolling mill process. *Electr. Power Syst. Res.* **2026**, *256*, 112878. [CrossRef]
29. Terven, J.; Cordova-Esparza, D.M.; Romero-González, J.A.; Ramírez-Pedraza, A.; Chávez-Urbiola, E.A. A comprehensive survey of loss functions and metrics in deep learning. *Artif. Intell. Rev.* **2025**, *58*, 195. [CrossRef]
30. de Amorim, L.B.V.; Cavalcanti, G.D.C.; Cruz, R.M.O. The choice of scaling technique matters for classification performance. *Appl. Soft Comput.* **2023**, *133*, 109924. [CrossRef]
31. Kingma, D.P.; Ba, J. Adam: A Method for Stochastic Optimization. *arXiv* **2014**, arXiv:1412.6980.
32. Smith, L.N. Cyclical Learning Rates for Training Neural Networks. In *IEEE Winter Conference on Applications of Computer Vision (WACV)*; IEEE: Santa Rosa, CA, USA, 2017; pp. 464–472.
33. SciPy API Optimization and Root Finding (Scipy.Optimize). Available online: https://docs.scipy.org/doc/scipy/reference/generated/scipy.optimize.differential_evolution.html (accessed on 26 May 2026).
34. Virtanen, P.; Gommers, R.; Oliphant, T.E.; Haberland, M.; Reddy, T.; Cournapeau, D.; Burovski, E.; Peterson, P.; Weckesser, W.; Bright, J.; et al. SciPy 1.0: Fundamental algorithms for scientific computing in Python. *Nat. Methods* **2020**, *17*, 261–272. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.