fibers

MDPI

Article

A Controlled Study on Machine Learning Applications to Predict Dry Fabric Color from Wet Samples: Influences of Dye Concentration and Squeeze Pressure

Warren J. Jasper [1,*] and Samuel M. Jasper [2]

[1] Department of Textile Engineering, Chemistry & Science, North Carolina State University, Raleigh, NC 27695, USA
[2] Independent Researcher, Raleigh, NC 27609, USA
* Correspondence: wjasper@ncsu.edu

Highlights

What are the main findings?

- AI was used to predict the color of dry cotton fabric with measurements in the wet state knowing just L*a*b* and squeeze roll pressure.
- Deep neural nets and extreme gradient boosting significantly outperformed regression models in wet to dry color prediction.

What is the implication of the main finding?

- In continuous dyeing, it is now possible to measure the fabric in the wet state and implement real-time feedback control to achieve a desired color in the dry state.
- The ability to implement real-time feedback on color reduces waste and improves quality in continuous dyeing, which accounts for 60% of dyed fabrics.

Abstract: Most dyeing occurs when a fabric is in a wet state, while color matching is performed when the fabric is in a dry state. As water is a colorless liquid, it has been difficult to analytically map these two states using existing color theories. Machine learning models provide a heuristic approach to this class of problems. Linear regression, random forest, eXtreme Gradient Boosting (XGBoost), and multiple neural network models were constructed and compared to predict the color of dry cotton fabric from its wet state. Different models were developed based on squeeze pressure (water pickup), with inputs to the models consisting of the L*a*b* (L*: lightness; a*: red–green axis; b*: blue–yellow axis) coordinates in the wet state and the outputs of the models consisting of the predicted L*a*b* coordinates in the dry state. The neural network model performed the best by correctly predicting the final shade to under a 1.0 color difference unit using the International Commission on Illumination (CIE) 2000 color difference formula (CIEDE2000) color difference equation about 63.9% of the time. While slightly less accurate, XGBoost and other tree-based models could be trained in a fraction of the time.

Keywords: fabric; color difference and prediction; machine learning; deep learning; XGBoost; neural network

check for updates

Academic Editor: Martin J. D. Clift

Received: 28 February 2025
Revised: 31 March 2025
Accepted: 9 April 2025
Published: 15 April 2025

Citation: Jasper, W.J.; Jasper, S.M. A Controlled Study on Machine Learning Applications to Predict Dry Fabric Color from Wet Samples: Influences of Dye Concentration and Squeeze Pressure. *Fibers* **2025**, *13*, 47. https://doi.org/10.3390/fib13040047

Copyright: © 2025 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 (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Most commercial dyeing processes use water as a solvent to transfer dyes onto fabrics. One of the most common dyeing processes is continuous dyeing, where a fabric roll is fed

continuously into a dye range at speeds of between 50 and 250 m per minute. Continuous dyeing accounts for about 60 percent of the total yardage dyed in the industry [1]. To improve the dyeing process through feedback control, it is necessary to measure the color difference between the actual dyed fabric and the target shade in real-time. If the dyed fabric shade deviates significantly from the target shade, potentially hundreds of meters of fabric could be wasted. However, while dyeing occurs when a fabric is wet, the color of a fabric is generally measured when the fabric is dry. Although water itself is colorless, when a fabric is wet, its apparent color is different. In Figure 1, one drop of deionized (DI) water was deposited onto the center of three cotton fabrics to illustrate this effect. Because fabric samples are typically dried before colorists can compare them with a target shade, the ability to predict the dry color directly from a wet sample is an energy-efficient and convenient way to tell if two color samples match and can allow for real-time changes during the dyeing process.



Figure 1. Visual perception of color differences between wet and dry fabrics to the human eye with fabrics dyed with primary dyes: red, blue, and yellow. The center of each fabric was wetted with a drop of deionized water.

Although Kubelka–Munk theory and Goldfinger's model have been used to predict color-matching under wet and dry conditions [2–4], there has not been a reliable color prediction model or related theory to predict the final shade of a dry fabric from the wet state. Artificial Neural Network models have been applied to many textile applications in color prediction [5–8]. We previously published a detailed review of machine learning methods in textile dyeing [9]. Currently, colorists achieve a target color by formulating different dye recipes, which eventually converge to the desired shade. A neural network model can predict the target color through heuristic learning without knowing the fundamental physical theories [10,11]. In one of the first applications of neural networks to coloration, Jasper et al. used a neural network to predict the dye concentrations in multiple-dye mixtures from their spectrophotometric absorbances based on Beer's Law [12]. Jasper also used a neural network combined with near-infrared (NIR) spectrophotometry to identify different types of fibers by their infrared absorbance spectra [13].

Neural networks are used in textiles and in an increasing number of research projects, applying heuristic learning as a support tool to obtain better results. Neural network models have been applied to viscose, nylon, lycra, polyester, and various blends of pre-treated fabrics [14–16]. Recently, artificial neural networks (ANNs) have been combined with other regression or classification models with applications in inkjet printing and digital printing for better color precision [17,18]. Furferi et al. used a cascade neural network to predict the color and color solidity of a jigger-dyed cellulose-based fabric [19]. Jawahar et al. used statistical regression models to predict the color of leather from its wet state using a root-mean-squared error criterion [20]. However, more commonly accepted criteria for color differences are the International Commission on Illumination (CIE) 2000 color difference formula (CIEDE2000) or the color difference method of the Color Measurement Committee [21,22]. In 2014, Jawahar et al. compared a neural network model with a

Kubelka–Munk model to predict the final color of leather with three dye combinations. Using CMC as the color difference formula to evaluate the prediction accuracy, Jawahar et al. concluded that an ANN model can predict the final color of leather with a mean ΔE_{CMC} value of 0.78, compared to 2.65 for the Kubelka–Munk model [5].

The most common error function used in neural networks is the Euclidean norm, which corresponds to the color difference equation CIELAB. Over the years, different color difference equations were proposed to provide a more perceptually uniform metric. In 1984, the Color Measurement Committee of the Society of Dyers and Colorists proposed a color difference equation called CMC (*l*:*c*) [23]. Two additional parameters were added to the color difference formula based on the ratio of the lightness (*l*) and chroma (*c*). In the textile industry, it is common to set *l* equal to 2 and *c* equal to 1 because the eye is more sensitive to chroma than lightness. Later, the CIE94 formula was developed to address perceptual non-uniformities while retaining the CIELAB color space by introducing application-specific weights [24]; however, CIE94 still failed to adequately resolve the perceptual uniformity issue. The CIEDE2000 formula attempted to correct for errors in perceived color difference by adding five additional correction terms [25]. Thus, CIEDE2000 was optimized for pairwise comparison. This paper focuses on the use of CIEDE2000 as an error function to both train and evaluate the machine learning models. Although each person perceives color and color differences differently, there is a quantifiable theory of a *standard observer*, which represents the visual color perception of a standard observer with normal color vision. For example, Figure 2 shows two examples of two colors, which differ by 0.7 and 2.0 CIEDE2000 (ΔE_{2000}) color units, respectively. It is generally accepted that color differences below 1.0 are difficult to distinguish for a common observer. Therefore, a criterion for a successful color prediction model is one that can predict color from one state to another with an accuracy of better than a 1.0 color difference unit.



(**a**) (**b**)

Figure 2. A visual cue to the human eye for color differences of (**a**) $\Delta E_{2000} = 0.70$ and (**b**) $\Delta E_{2000} = 2.00$.

1.1. Color Difference Models

A Geometric Perception of Color Difference

Formally, given a differential manifold $\mathbf{R}^3{}_{wet}$, we seek to find a mapping f such that: f: $\mathbf{R}^3{}_{wet} \rightarrow \mathbf{R}^3{}_{dry}$ such that the norm in $\mathbf{R}^3{}_{dry}$ is CIEDE2000

This is a geometric view of a color space where colors are represented as points in a normed three-space, where the norm or metric is CIEDE2000. Note that the distance between two points (or colors) in the wet space is different from the distance between those same two mapped points when the fabric is dried. However, if the mapping function f is known, it is possible to predict their distances, which correspond to the color differences. The rest of the paper describes the process of finding and validating the mapping function f.

Over the years, different color difference equations have been proposed, and the four most popular ones are presented below. Note that the CIELAB color difference equation is just the Euclidean norm.

CIELAB [26] for $\left(L_1^*, a_1^*, b_1^*;\ L_2^*, a_2^*, b_2^*\right)$:

$$\Delta E_{ab}^* = \sqrt{\left(L_2^* - L_1^*\right)^2 + \left(a_2^* - a_1^*\right)^2 + \left(b_2^* - b_1^*\right)^2} \quad (1)$$

The following definitions are used:

- L_1^* and L_2^*: The lightness values of the two colors. Lightness represents how light or dark a color is on a scale from 0 (black) to 100 (white).
- a_1^* and a_2^*: The a* (green–red) values of the two colors. Positive values indicate more red and negative values indicate more green.
- b_1^* and b_2^*: The b* (blue–yellow) values of the two colors. Positive values indicate more yellow and negative values indicate more blue.

CMC (l:c) [24] for $\left(L_1^*, a_1^*, b_1^*;\ L_2^*, a_2^*, b_2^*\right)$:

$$\Delta E_{CMC}^* = \sqrt{\left(\frac{L_2^* - L_1^*}{lS_L}\right)^2 + \left(\frac{C_2^* - C_1^*}{cS_C}\right)^2 + \left(\frac{\Delta H_{ab}^*}{S_H}\right)^2} \tag{2}$$

The following definitions are used:

- ΔE_{CMC}^*: Color difference metric introduced by the CMC in 1984.
- $L_2^* - L_1^*$: Difference in lightness between the standard and measured colors.
- $C_2^* - C_1^*$: Difference in chroma (color intensity) between the standard and measured colors.
- ΔH_{ab}^*: Difference in hue between the standard and measured colors.
- l: Parametric factor for lightness to adjust the sensitivity due to lightness differences.
- c: Parametric factor for chroma to adjust the sensitivity due to chroma differences.
- S_L: Compensation for lightness to scale the lightness difference depending on the colors.
- S_C: Compensation for chroma, which scales the chroma difference.
- S_H: Compensation for hue, which scales the hue difference.

CIE94 [25] for $\left(L_1^*, a_1^*, b_1^*;\ L_2^*, a_2^*, b_2^*\right)$:

$$\Delta E_{94}^* = \sqrt{\left(\frac{\Delta L_{ab}^*}{k_L S_L}\right)^2 + \left(\frac{\Delta C_{ab}^*}{k_C S_C}\right)^2 + \left(\frac{\Delta H_{ab}^*}{k_H S_H}\right)^2} \tag{3}$$

The following definitions are used:

- ΔE_{94}^*: Color difference metric introduced by the CIE in 1994.
- ΔL_{ab}^*: Difference in lightness between the standard color and the measured color.
- ΔC_{ab}^*: Difference in chroma (color intensity) between the two colors.
- ΔH_{ab}^*: Difference in hue between the two colors.
- k_L, k_C, k_H: Weighting factors for lightness, chroma, and hue.
- S_L: Compensation for lightness to scale the lightness difference depending on the colors.
- S_C: Compensation for chroma, which scales the chroma difference.
- S_H: Compensation for hue, which scales the hue difference.

CIEDE2000 [26] for $\left(L_1^*, a_1^*, b_1^*;\ L_2^*, a_2^*, b_2^*\right)$:

$$\Delta E_{00}^* = \sqrt{\left(\frac{\Delta L'}{k_L S_L}\right)^2 + \left(\frac{\Delta C'}{k_C S_C}\right)^2 + \left(\frac{\Delta H'}{k_H S_H}\right)^2 + R_T \frac{\Delta C'}{k_C S_C} \frac{\Delta H'}{k_H S_H}} \tag{4}$$

The following definitions are used:

- ΔE_{00}^*: Color difference metric introduced by the CIEDE in 2000.
- $\Delta L'$: The difference in lightness between the two colors, modified to account for perceptual non-linearity.
- $\Delta C'$: The difference in chroma between the two colors, also modified.
- $\Delta H'$: The difference in hue between the two colors, corrected for perceptual factors.
- k_L, k_C, k_H: Weighting factors for lightness, chroma, and hue.

- S_L: Compensation for lightness to scale the lightness difference depending on the colors.
- S_C: Compensation for chroma, which scales the chroma difference.
- S_H: Compensation for hue, which scales the hue difference.
- R_T: A rotation term that accounts for the interaction between chroma and hue, especially in the blue region of the color space.

2. Materials and Methods

This section discusses the methodology used to generate color data from a dyed fabric and the development of machine learning models, including a neural network.

2.1. Experimental Design

To construct a model, the L*, a*, and b* color-coordinates of a cotton fabric in the wet state were used as input parameters, while the L*, a*, and b* coordinates of the cotton fabric in the dry state were used as output parameters. A flow chart of such an experimental setup is shown in Figure 3. Color difference equations may be used as the criterion to evaluate performance for various Machine Learning (ML) models and as the error function to train a neural network. However, note that different architectures and models affect convergence and performance. Typically, machine learning models use the Euclidean norm (Mean Square Error) as the error function for model training. However, since color space is not perceptually uniform (i.e., Euclidean}, the Euclidean norm is not the best metric to describe the color distance between points. This should be considered when training models and evaluating model performance.

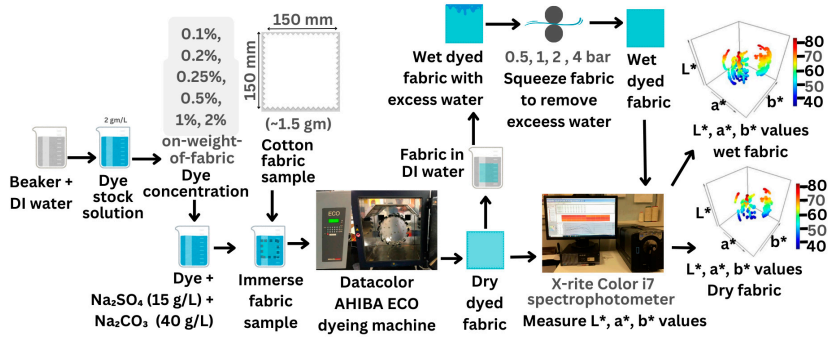


Figure 3. Fabric dyeing procedure.

2.2. Fabric Dyeing Procedure

Small woven cotton swatches were cut from an untreated cotton fabric roll (76 warp/54 weft; punch weight: 127.5 g/m^2). Three reactive dyes (red, blue, and yellow) from Huntsman Corporation (Charlotte, NC, USA) were selected: Novacron Red FN-3GL, Novacron Blue FN-G, and Novacron Yellow FN-2R. Sodium carbonate and sodium sulfate were obtained from Tronox Chemicals Company (Stamford, CT, USA) and BrenntagNorth America (Reading, PA, USA). The dyeing machine used was a Datacolor AHIBA ECO. The padding machine was from Werner Mathis USA Inc. (Concord, NC, USA). The spectrophotometer was an X-rite Color i7 benchtop spectrophotometer using Color iControl (v10.1) software.

A cotton fabric was cut into small pieces measuring 150 mm × 150 mm and weighing approximately 1.5 g each. The procedure outlined below follows the dye manufacturer's recommendation. Dyes were diluted with deionized water to make a 2 g/L stock solution. The concentrations of dyes were 0.1%, 0.2%, 0.25%, 0.5%, 1%, and 2% on-weight-of-fabric (owf) for each primary color. The concentrations of sodium carbonate (Na_2CO_3) and sodium sulfate (Na_2SO_4) were 15 g/L and 40 g/L, respectively. Cotton samples were weighed to three significant digits and the dyes and chemicals were calculated and measured following

the specified formula for each dye recipe. Measured DI water, dye stock solution, sodium sulfate, and sodium carbonate were slowly added into the machine beakers. The solution was mixed to ensure the chemicals dissolved evenly. The cotton samples were rinsed with DI water and then put through a padding machine to remove any excess water. The wet cotton samples were then placed into separate dyeing machine beakers with the lids closed to ensure that no leaking or evaporation occurred. The beakers were placed into the dyeing machine, and the temperature was then increased at a rate of 2 °C/min until reaching the target temperature of 60 °C. This temperature was held for 40 min before the bath was cooled down to 50 °C. The samples were then rinsed with warm water and dried in the oven at 50 °C.

2.3. Water Pickup

Water pickup is defined as the percentage of water remaining in a cotton fabric after squeezing out the water with two padding rollers and is measured by the pressure set at the padding machine. Padding machine pressures of 0.5 bar, 1 bar, 2 bar, and 4 bar corresponded to a water pickup of 110%, 89%, 72%, and 64%, respectively, with a coefficient of variation of 0.02. Wet pickup was measured gravimetrically by weighing wet and dry samples with ten repeats. The padding pressure was later treated as an input parameter along with the color coordinate values L*, a*, and b* in the wet state to train the machine learning (ML) models.

2.4. Color Measurement

The dyed samples were wetted with DI water and then passed through a padding machine at a speed of 2 m/min at four different squeeze pressures: 0.5, 1, 2, and 4 bar. After padding, the wet samples were immediately measured to obtain the color in CIE L*a*b* coordinates under a Daylight 65 (D65) light source and Large Area of View port (LAV 25 mm). The spectrophotometer used was an X-rite Color i7 benchtop spectrophotometer using Color iControl software with the performance specifications shown in Table 1. After completely drying the samples in the oven, the samples were cooled to room temperature. The dry color measurements were obtained under the same viewing conditions as the wet color measurements. Each set of dry and wet color measurements were recorded as one group for data analysis.

Table 1. Performance specifications of the X-rite Color i7 benchtop spectrophotometer [27].

Repeatability	0.01 RMS ΔE CIELAB White tile
Inter-Instrument Agreement	0.08 Avg. 13 BCRA Series II tiles SCI (LAV only)
Geometry	D/8 Tri-beam simultaneous SCE/SCI
Illumination	Pulsed Xenon, D65 Calibrated
Measurement time	2.7–4.0 s (flash and data acquisition)
Duty cycle	480 measurements per hour max
Spectral range	360 to 750 nm
Wavelength interval	10 nm
Photometric range	0.0% to 200%
Photometric resolution	0.001% reflectance

2.5. Color Data

A total of 762 dyed samples were obtained at each of the four pressures to train and test each model. Each dyeing took over 3.5 h to perform, resulting in 2667 h being required

to perform all 762 dyeings. About 37% of the samples were dyed with a single primary dye, 40% of the samples were dyed with two-dye combinations, and the remaining 23% of the samples were dyed with three-dye combinations. The total dyeing time for each sample was approximately 3.5 h, which included prep and post-scouring time. An 80/20 train/test split was used, resulting in a training dataset of 609 samples (under different pressures) and a testing dataset of 153 samples to evaluate the accuracy of the model. Separate models were developed using data points across all pressures (with pressure as an input parameter), which consisted of training and testing datasets that were roughly four times larger than those used for the individual pressure models. While stratification across pressures was not explicitly performed when creating the training dataset for the all-pressure models, it consisted of an even distribution of data points from all the individual pressures (i.e., the number of data points for an individual pressure was within 4% of any other pressure).

With these concentrations of dyes, the 762 colored samples covered the available color gamut, as shown in Figure 4. The black points represent each color sample in the dry state located in the xy surface of xyY color space.

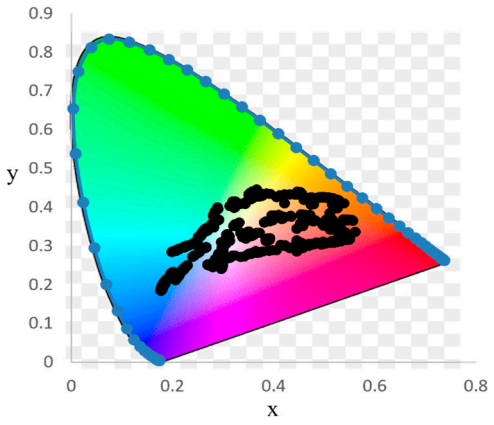


Figure 4. Color gamut for 762 samples in the CIE 1931 color space (CIE xyY) chromaticity plot. The *x*-axis represents the chromaticity coordinate x (ranging from 0 to 0.8), and the *y*-axis represents the chromaticity coordinate y (ranging from 0 to 0.9), showing the distribution of colors in the dataset.

Figure 5 shows 3D plots for the 762 color samples in the wet and dry states located in the L*a*b* color space. L* ranged from 0 to 100, while a* and b* ranged from −100 to 100. All the points were colored depending on their L*, a*, and b* values. While the shape of the 3D plot is not a completely perfect sphere due to the limitations of the three primary dyes, it covers most of the achievable color gamut. After data collection, all the color coordinates (L*, a*, and b*) were normalized in the range of 0 to 1 using Equation (5):

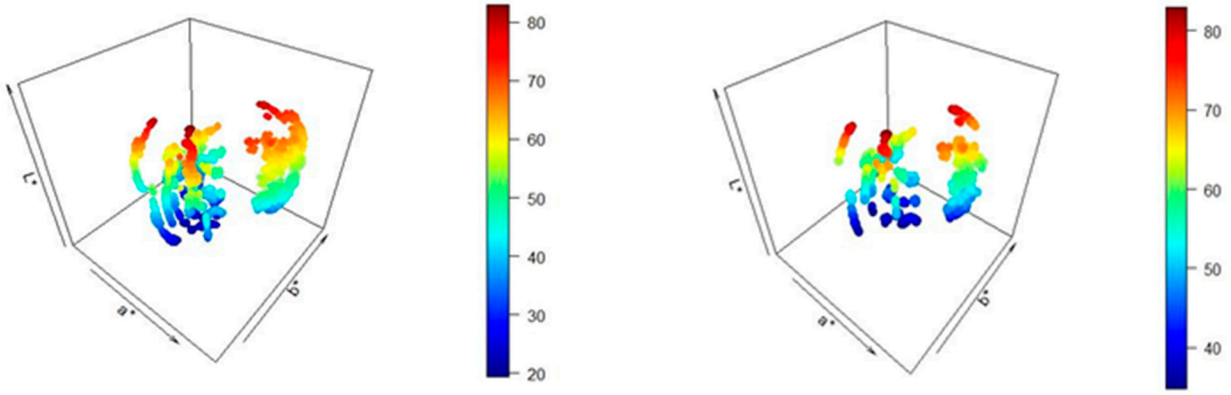
$$y_i^* = \frac{y_i - min(y)}{max(y) - min(y)},\ i = 1, 2, 3, \ldots, n \tag{5}$$

where the following definitions are used:

- y_i^* is the normalized value;
- y_i is the measured color (L*, a* or b*) in the wet state;
- n is the total number of samples.

Normalized values were only used for training neural network models, whereas unnormalized values were used for training all other models. The data were normalized to ensure that all parameters (L, a*, and b* for wet fabric) were scaled within the same range of 0 to 1. This prevents features with larger numerical ranges from dominating the training process, allowing each parameter to contribute equally. Additionally, normalization improved the model convergence speed and overall performance. Normalization was

specifically applied to neural networks, as these models are highly sensitive to feature scaling. In contrast, tree-based models (e.g., random forest, XGBoost) were trained on unnormalized data, as they rely on decision rules rather than distance-based calculations, making them inherently robust to differences in feature magnitudes.



(**a**) (**b**)

Figure 5. Three-dimensional plots of 762 color samples in the wet (**a**) and dry (**b**) states within the CIELAB (Lab) color space. The plots illustrate the shift in color coordinates due to the drying process, where L* represents lightness, while a* and b* indicate chromaticity along the red–green and blue–yellow axes, respectively.

2.6. Traditional Machine Learning Models

Five different model architectures were developed to predict the dry fabric color from the wet fabric color at each roller pressure: (1) linear regression, (2) random forest, (3) gradient boosted trees, (4) eXtreme Gradient Boosted trees (XGBoost), and (5) neural networks. A linear regression model was developed to evaluate the linear relationship between the wet and dry color states and provide baseline metrics for comparison with more advanced machine learning models. Tree-based models offer a robust and fast approach for predicting non-linear relationships, with the simplest ensemble tree-based model being a random forest. Random forests employ a sampling method known as bagging, where the training dataset is sampled at random with a replacement. Boosted trees employ a slightly different sampling method, known as boosting, where observations are not sampled randomly but are instead weighted by their difficulty to classify, which can result in ensembled trees that are more able to predict edge cases. Gradient boosting algorithms utilize boosted trees through an ensemble of simple decision trees called "weak learners", where each weak learner is trained to minimize the residuals of the previous learner. An XGBoost model further develops this concept by introducing a regularization term to prevent overfitting, which is more likely to occur while working with smaller datasets. Lastly, a neural network model architecture was selected due to its ability to model complex non-linear patterns and prioritize predictability over explainability, which is appropriate for this specific problem. A variety of models were chosen to compare not only performance, but also time to convergence.

Except for the neural network model architecture, three separate models were developed (one for predicting L*, one for predicting a*, and one for predicting b* in the dry state), which were then combined when calculating the ΔE_{2000} color difference between the actual and predicted dry L*a*b* values.

To maximize the performance of a machine learning model, proper hyperparameter tuning is needed [28]. A random grid search was used to test subsets of all possible combinations of hyperparameters by randomly selecting a value for each hyperparameter from the set list. Cross-validation was performed for each model to avoid overfitting

on the training dataset, and a seed was set prior to training to make the tuning reproducible. No hyperparameter tuning is needed for a linear regression model because a single mathematical solution exists to minimize the sum of the squared residuals.

Tables S4–S6 show the hyperparameters used in each tuning grid, along with their descriptions, a set of potential values, and the optimal values chosen by the best model(s) for each model type. For tree-based models, the *max_depth* and *n_estimator* parameters (the maximum depth of each tree and the number of ensembled trees per model, respectively) contributed the most to the training time as well as to overfitting.

2.7. Neural Networks

Neural networks from the deep learning subset of machine learning were also developed to predict the dry fabric color from the input color values of a wet fabric. The L*a*b* coordinates in the wet state under different pressures were used as inputs and the L*a*b* coordinates in the dry state were used as outputs. Figure 6 shows (as an illustrative example) a 3 × (50 × 10) × 3 model, which represents a neural network with 3 nodes in the input layer, 500 nodes in the hidden layer, and 3 nodes in the output layer.

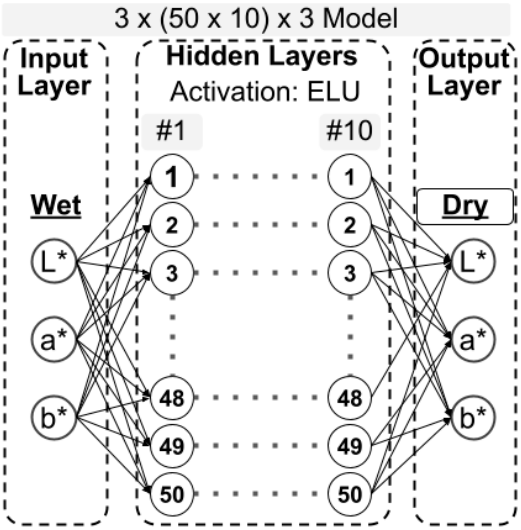


Figure 6. Architecture of the optimized neural network model with a structure of 3 × (50 × 10) × 3, where 3 represents the input neurons, 50 × 10 represents 50 neurons per hidden layer and 10 layers, and 3 represents the output neurons.

A total of 50 neural networks were studied by varying the number of hidden layers (1, 2, 3, 4, 5, 6, 7, 8, 9, 10), varying the number of neurons per hidden layer (5, 10, 20, 30, 40, 50), varying the activation functions (ELU: Exponential Linear Unit; SELU: Scaled Exponential Linear Unit), varying the learning rate (10^{-3}, 10^{-4}, 10^{-5}), and using two loss functions (MSE: Mean Squared Error; ΔE_{2000}: custom loss function). For comparison, 25 models were selected using mean squared error as the loss function, and 25 models were selected using custom ΔE_{2000} as the loss function. This custom ΔE_{2000} loss function was built in Tensorflow: 2.14.0 in order to integrate it with Keras 2.14.0, the Python (version 3.11.6) package chosen for building and training neural network models [29].

The number of hidden layers and the number of nodes in each hidden layer were modified and selected using a comparison of the final prediction results with the measured data. Although the specific problem and the data dictate the number of input and output nodes, the design of the optimal architecture is still an open research question. If too many hidden nodes and hidden layers are used, the model will overfit the data and fail to generalize. If too few hidden nodes/layers are used, the model will not converge or the model will not be robust. Hyperparameter tuning is the process by which the space of

possible neural network architectures and parameters (activation functions, convergence criteria, error function, etc.) is searched to arrive at an optimal design.

The optimal neural network architecture consisted of 3 neurons in the input layer, 50 neurons in each of the 10 hidden layers, and 3 neurons in the output layer (see Table S7). The total number of trainable parameters was calculated as follows:

1. From the input layer to the first hidden layer: (3 input neurons × 50 neurons in the first hidden layer) + 50 biases = 200 parameters.
2. Between each of the nine hidden layers: (50 neurons × 50 neurons) + 50 biases = 2550 parameters per connection, totaling 9 × 2550 = 22,950 parameters for all hidden layer connections.
3. From the last hidden layer to the output layer: (50 last hidden neurons × 3 output neurons) + 3 biases = 153 parameters.

In total, the neural network comprised 23,303 trainable parameters, calculated as follows: 200 (input to first hidden layer) + 22,950 (between hidden layers) + 153 (last hidden layer to output).

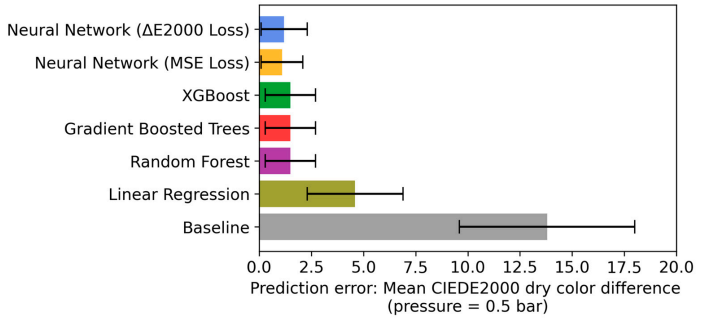
Therefore, a deep neural network with 15,000 epochs took 1 to 2 h to train. An epoch is made up of one or more batches, where part of the dataset is used to train the neural network. A batch size of 32 training observations was used, which is the default value for Keras' model training API. The best deep neural network model (previously determined via hyperparameter tuning) was then subjected to 250,000 additional epochs of training. The best all-pressure deep neural network model failed to converge during additional training, while the best 0.5 bar neural network model's performance degraded significantly (from 56.9% of predictions having a ΔE_{2000} color difference of less than or equal to 1 to 35.3% of predictions having a ΔE_{2000} color difference of less than or equal to 1). Similar model degradation was seen with the best 1-bar and 2-bar neural network models.

2.8. Computational Requirements

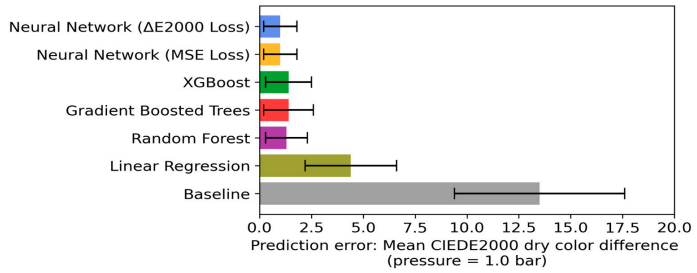
The model development and analysis was conducted on a desktop computer running Windows 10 (Version 22H2), equipped with an AMD Ryzen 7 3700X 8-Core Processor (3.59 GHz) and 32 GB of memory (2133 MHz). The data were stored on a Samsung SSD 860 EVO 1 TB. Note that all the model training was performed using CPUs; no GPUs were involved in the process. The programming environment was Python version 3.11.6. The scikit-learn library was used for training random forest and gradient boosting models, XGBoost was employed for training using the XGBoost algorithm, and Keras was utilized to train neural networks.

3. Results

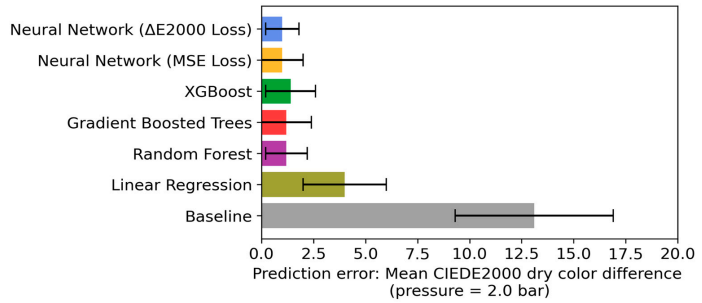
Figure 7 shows the performance of each model that was created using the dataset of the corresponding roller pressure. Each value represents the ΔE_{2000} color difference between the actual and predicted dry L*a*b* values using the test dataset. A separate set of models was developed using pressure as an input parameter, which is included in the "All" pressure rows. The Baseline naïve "model" is simply the ΔE_{2000} color difference between the wet and dry states for the test dataset, which provides a point of comparison for all the other models. The baseline is therefore the raw DE2000 reading between the wet sample and the dry sample and was used as a reference point to compare the performance of other models.



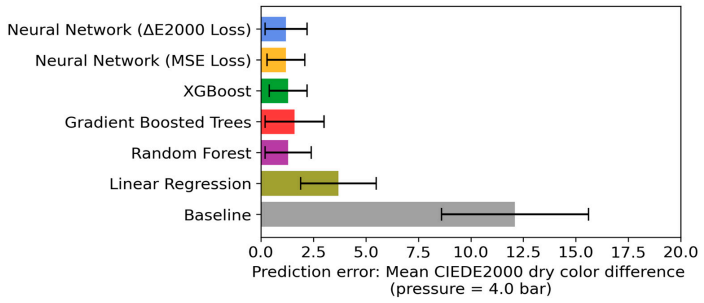
(**A**)



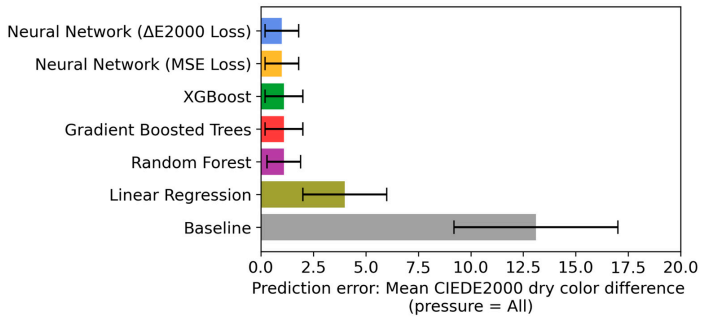
(**B**)



(**C**)



(**D**)



(**E**)

Figure 7. (**A**) Prediction error: mean CIEDE 2000 dry color difference (pressure = 0.5 bar). Note: for neural network, MSE (mean squared error) loss is also reported. The error bars indicate standard

deviation; (**B**) Prediction error: mean CIEDE 2000 dry color difference (pressure = 1.0 bar). Note: For neural network, MSE (mean squared error) loss is also reported. The error bars indicate standard deviation; (**C**) Prediction error: mean CIEDE 2000 dry color difference (pressure = 2.0 bar). Note: For neural network, MSE (mean squared error) loss is also reported. The error bars indicate standard deviation; (**D**) Prediction error: mean CIEDE 2000 dry color difference (pressure = 4.0 bar). Note: For neural network, MSE (mean squared error) loss is also reported. The error bars indicate standard deviation; (**E**) Prediction error: mean CIEDE 2000 dry color difference (pressure = all bar). Note: For neural network, MSE (mean squared error) loss is also reported. The error bars indicate standard deviation.

Figure 8 shows a histogram of the ΔE_{2000} color differences between the predicted and actual L*a*b* measurements for the best all-pressure neural network model. While the maximum ΔE_{2000} color difference for this model was 5.4, the median ΔE_{2000} color difference was only 0.7 and the standard deviation was only 0.8.

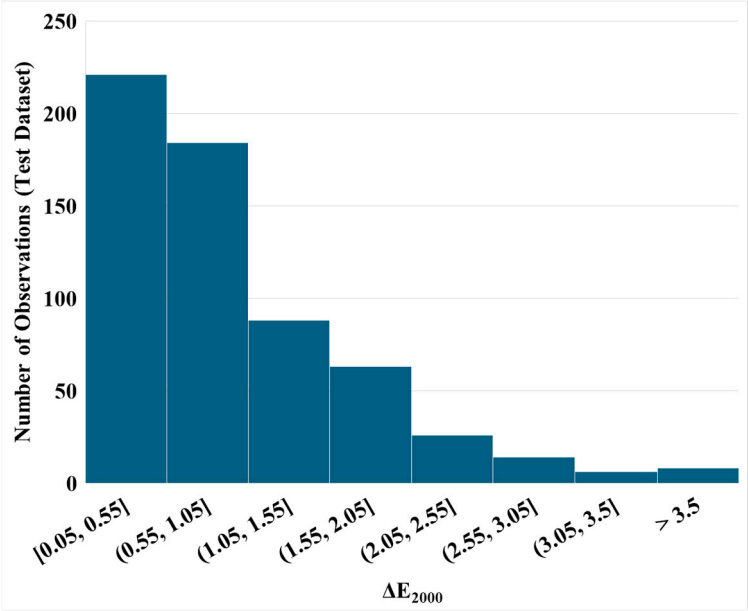
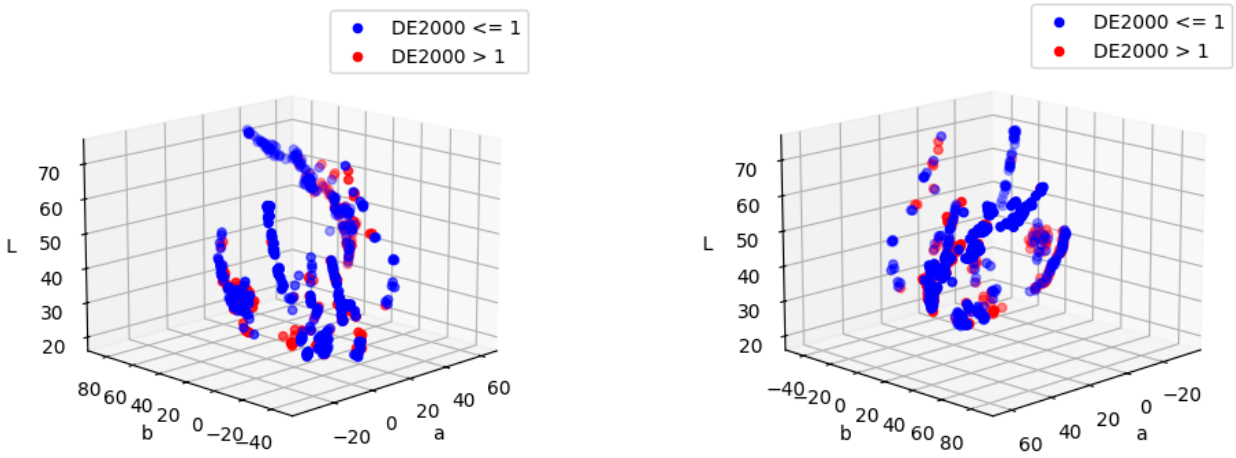


Figure 8. Histogram of ΔE_{2000} color differences between predicted and actual L*a*b* measurements for the best neural network (all pressures).

Figure 9 shows the L*a*b* values of each sample (in the wet state), with blue dots indicating that the dry state predictions had a ΔE_{2000} color difference of less than or equal to 1 and red dots indicating that the dry state predictions had a ΔE_{2000} color difference greater than 1 using the best deep neural network model (trained with data from all pressures). These 3D renderings were used to determine if the neural network model performed poorly in a particular region of the color space. Due to the even distribution of blue and red dots in this figure, the neural network model performed similarly across the entire color range. Figure 9a shows one orientation of the data, while Figure 9b shows the same data after rotating the figure 180°.

An important aspect of this study was the creation of a comprehensive wet and dry color dataset consisting of 762 samples, dyed using three primary colors: blue, red, and yellow. The resulting color range for the dry state of the dyed fabrics, shown in Figure 4, reveals that data points are densely clustered around achievable colors, while the white area is devoid of data points. This absence occurs because the dyes used in this study are not optical brighteners, meaning they cannot increase the perceived whiteness of the fabric. Dyes absorb light rather than emit it in a way that enhances brightness, making it impossible to achieve a 'whiter' fabric through dyeing. In the CIELAB color space, this limitation manifests as a void or a hole in the plot. Additionally, there is a practical limit

to how dark a fabric can be dyed, leading to a distribution of color data points that form a sphere-like shape with a central gap. This distribution reflects the impact of the dyeing process and dye combinations, which can be influenced by factors such as fluctuations in dye concentrations, subtle variations in squeeze roller pressure, and the inherent variability of the textile fabric.



(**a**) (**b**)

Figure 9. Rotated (**a**) left view, (**b**) right view of predicted wet L*, a*, and b* samples across all pressure levels. 'Blue' points represent samples with ΔE2000 (CIEDE2000 color difference) values ≤ 1, indicating high prediction accuracy. Red points represent samples with ΔE2000 values > 1, indicating greater color deviation. The L* axis represents lightness, while the a* and b* axes indicate chromaticity along the red–green and blue–yellow directions, respectively.

Despite these potential sources of variability, the dyeing process was tightly controlled, as demonstrated by the dense clustering of data points in the L*a*b* plot for both wet and dry states (Figure 5). Notably, the dry state data points are less dispersed, likely due to the absence of water, which minimizes light interaction effects during color measurements. Understanding these inherent limitations, even under controlled experimental conditions, is crucial for improving the predictive accuracy of models that rely on wet-state L*, a*, and b* values to estimate dry-state counterparts.

4. Discussion

In differential geometry, the metric, or the distance between points, determines the uniformity and hence the curvature of the space. Color spaces such as L*a*b* are non-uniform or non-Euclidean. For example, CIELAB (Equation (1)) is a Euclidean metric which assumes that a color space is uniform or Euclidean. In addition, because the mapping of points in the wet L*a*b* space to the dry L*a*b* space is non-linear, linear models struggle to capture the trends in the data and make accurate predictions. Consequently, as shown in Figure 7, the baseline and linear regression methods performed poorly, with average CIEDE2000 dry color difference errors ranging from 12.1 to 13.8 for the baseline and 3.7 to 4.6 for the linear regression across all pressure models. Typically, CIEDE2000 values exceeding 0.8 to 1.0 are generally outside of the accepted tolerances in the industry. The baseline consistently produced high CIEDE2000 error values, highlighting the inability of non-machine-learning approaches to capture the complex relationships between wet and dry color states. Linear regression showed an improvement over the baseline but still struggled to account for the non-linear dependencies in the data, leading to higher variability and fewer predictions achieving a ΔE_{2000} less than or equal to one.

To model the non-linear relationship between wet and dry color coordinates, it is essential to employ a method that is capable of capturing the complex regression surface that maps wet-state L*, a*, and b* values to their dry-state counterparts. In this study, traditional machine learning ensemble tree-based methods—such as Random Forest, Gradient Boosted Trees, and XGBoost—outperformed linear methods. The average CIEDE2000 dry color difference error range was 1.1 to 1.5 for Random Forest, 1.1 to 1.6 for Gradient Boosted Trees, and 1.1 to 1.5 for XGBoost across the models built at different squeeze roll pressures, as shown in Figure 7 and Table S8.

Over 100 models were tested for each of the tree-based methods with varying hyperparameters (see Tables S4–S6). The Random Forest method combines L*, a*, and b* predictions from multiple decision trees to enhance performance and reduce overfitting. Each tree is trained on a random subset of the color data to ensure diversity in the ensemble. Using the all-pressure model, Random Forest achieved mean ΔE_{2000} values of 1.1, with over 57% of predictions falling within $\Delta E_{2000} \leq 1$ and over 88% within $\Delta E_{2000} \leq 2$, as shown in Table 2. Its ensemble approach effectively captures non-linear interactions, offering a robust fit across varying pressure conditions, although the slight variability at higher pressures suggests limitations in its ability to resolve subtle nuances in the dataset.

Table 2. Accuracy of ML models in predicting CIEDE2000 dry color differences across varying pressures. The columns represent the percentage of rows in the test dataset with $\Delta E_{2000} \leq 1$ or ≤ 2.

Model	Pressure 0.5 Bar		Pressure 1.0 Bar		Pressure 2.0 Bar		Pressure 4.0 bar		All Pressures	
	≤1	≤2	≤1	≤2	≤1	≤2	≤1	≤2	≤1	≤2
Baseline	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Linear Regression	2.0	11.1	2.0	13.1	3.9	13.7	2.6	17.0	2.6	15.1
Random Forest	39.2	78.4	51.0	81.1	49.0	88.9	46.4	80.4	57.7	88.5
Gradient Boosted Trees	44.4	75.2	43.8	81.1	50.3	86.9	40.5	75.2	57.1	87.1
XGBoost	44.4	76.5	45.8	82.4	43.8	83.0	42.5	86.3	54.9	87.4
Neural Network (MSE Loss)	59.5	88.2	59.5	88.2	62.8	90.9	52.9	86.3	63.6	90.8
Neural Network (ΔE_{2000} Loss)	56.9	86.3	60.1	89.5	62.1	91.5	53.6	87.6	63.9	90.2

The Gradient Boosted Trees method builds an ensemble of decision trees sequentially, based on the CIEDE2000 dry color difference prediction error of the previous tree. Similarly, XGBoost employs regularization to minimize overfitting and is capable of modeling non-linear relationships. Both Gradient Boosted Trees and XGBoost achieved a comparable performance to Random Forest, with mean ΔE_{2000} values near 1.1.

The neural network models tested in this study were specifically designed to capture the non-linear relationships between wet and dry fabric color values, leveraging deep learning techniques to model the complex color transformations. A total of 50 neural network models were trained using the complete dataset, divided into two groups of 25 models based on the chosen loss function: the custom ΔE_{2000} loss function and the mean squared error (MSE) loss. Models were trained for up to 15,000 epochs, allowing for sufficient iterations to optimize the parameters and capture intricate patterns in the data.

The architecture of the neural networks was optimized through a random grid search hyperparameter tuning process, Table S7. Key parameters, such as the number of hidden layers, number of nodes per layer, and activation functions, were systematically tested. The tuning grid search space included values for the number of hidden layers ranging from

1 to 10, with the optimal value found to be 10 layers. Each hidden layer contained 50 nodes, which was determined to be the best configuration for capturing the data's complexity. For activation functions, the Exponential Linear Unit (ELU) was selected over the Scaled Exponential Linear Unit (SELU), as it demonstrated a better performance in handling the vanishing gradient problem typically encountered in deep networks. Additionally, the kernel initializer was optimized to use the LecunNormal initializer, which draws samples from a truncated normal distribution, ensuring that the model's weights were appropriately initialized for efficient learning. The learning rate, which controls the magnitude of weight updates during training, was set to 10^{-3} to obtain the best balance between training speed and model convergence.

The optimizer used in the final models was Nadam, a variant of the Adam optimizer, which incorporates Nesterov momentum for better convergence. The model's performance was evaluated using the "val_loss" objective, ensuring that the training process was guided by minimizing the validation loss. The custom ΔE_{2000} loss function, designed specifically for color difference prediction, was found to be the most effective for this task, as it is directly optimized for perceptual color differences. This tailored loss function was particularly valuable in ensuring that the neural network's predictions aligned well with human visual perception of color differences, a crucial aspect in textile color-matching.

The architecture of the optimized neural network model consisted of an input layer with three neurons, corresponding to the L*, a*, and b* values of the wet-state fabric. This was followed by 10 hidden layers, each containing 50 nodes, forming a dense network capable of capturing the complex interactions within the dataset. Finally, the output layer also contained three neurons, corresponding to the predicted L*, a*, and b* values for the dry state. This deep learning architecture significantly outperformed traditional models, including linear regression and tree-based ensemble methods, in predicting dry-state color values from wet-state measurements. The performance of a model is determined by how closely its predictions match the actual values. Neural networks, with their deep architectures and extensive trainable parameters (over 23,000 in this study), offer greater flexibility in capturing complex relationships between wet and dry color states. Unlike tree-based models, which rely on the discrete partitioning of data, neural networks can learn smooth, non-linear transformations, making them particularly effective for color prediction tasks. Additionally, the normalization of input data facilitated better convergence, and when CIEDE2000 was used as a loss function, it further aligned the training process with human perceptual differences, contributing to a superior performance. Thus, the network's ability to minimize ΔE_{2000} values demonstrated its capacity to learn and model the non-linear transformations between wet and dry fabric states. The all-pressure neural network model trained using a custom ΔE_{2000} loss function had a mean ΔE_{2000} value of 1.0 with over 63.9% of predictions falling within $\Delta E_{2000} \leq 1$. This improvement in performance highlights the neural network's ability to capture intricate, non-linear patterns and produce highly accurate, perceptually relevant predictions for textile color-matching.

The minimum MSE of 0.1 achieved by the neural networks in this study is comparable to a previous study that used ANN to predict CIE L*a*b* values of dry yarn from wet yarn, which reported an MAE of 0.5 [30]. Another study reported an average error of 1% using ANN to predict the CIE L*a*b* values of dyed fabrics after wet processing [31].

Lastly, it may be noted that the ΔE_{2000} color difference for most models was right-skewed, with a kurtosis of 3.8, meaning most of the predicted L*, a*, and b* measurements had a ΔE_{2000} color difference concentrated around lower values, as shown in Figure 8.

Overall, non-linear models, particularly neural networks, significantly outperform linear regression. These results highlight the importance of advanced machine learning

techniques in modeling the complexities of textile dyeing processes and predicting dry-state colors from wet-state measurements under varying pressure conditions.

5. Conclusions

The study successfully developed and evaluated multiple machine learning models to predict the dry color of fabric based on its wet state under varying roller pressures. A significant achievement was the creation of a comprehensive dyed fabric color dataset, which encompassed different dye combinations and pressures, facilitating the training and testing of several models. These included traditional machine learning approaches such as linear regression, random forests, and gradient-boosted trees, as well as deep learning neural networks.

Incorporating pressure as an input parameter significantly enhanced model performance, leading to lower color differences between the predicted and actual dry L*a*b* values. Among the models tested, neural networks exhibited superior predictive accuracy, particularly when employing the custom ΔE_{2000} loss function, achieving over 63.9% of predictions with a color difference of less than or equal to 1. The practical implications of this research extend beyond theoretical advancements, offering guidance for optimizing dyeing processes, improving color prediction accuracy, paving the way for real-time feedback control in continuous dyeing, and fostering sustainable practices in the textile industry.

Limitations and Future Directions

Water is a colorless liquid that can lead to significant and apparent color differences when applied to dry fabric. While a deep neural network can predict the relationship between colors in dry and wet states, the training process is highly dependent on the quality and quantity of the training data, the network topology, and the hyperparameter settings. Enhancing the model's performance would likely require a larger dataset of dyed fabrics. However, the process of obtaining each dyed fabric sample was time-intensive, with the total effort exceeding 2600 h.

Future research will focus on improving the model's generalizability. This will require systematic data collection encompassing various dyeing machines, process parameters, ambient conditions, material properties, and end-use requirements. Additionally, developing a modular neural network architecture—where specialized models handle specific tasks—may enhance both generalizability and real-time accuracy, enabling the model to operate effectively at production speed. Also, it might be worth looking at the effects of metamerism and different lighting conditions.

Supplementary Materials: The following supporting information can be downloaded at: https://www.mdpi.com/article/10.3390/fib13040047/s1, Figure S1: Correlation of dry vs. wet fabric weights; Table S1: Moisture regain in fabric at pressure of 1 bar; Table S2: Moisture regain in fabric at pressure of 2 bar; Table S3: Moisture regain in fabric at pressure of 4 bar; Table S4: Random Forest models. N = 100 models tested using all pressure dataset; Table S5: Gradient Boosted Tree models. N = 100 models tested using all pressure dataset; Table S6: XGBoost models. N = 200 models tested using all pressure dataset; Table S7: Neural network models. N = 50 models tested using all pressure dataset (25 models per loss function). Models were trained up to 15000 epochs; Table S8: CIEDE2000 dry color difference of ML model predictions across pressures.

Author Contributions: Conceptualization, W.J.J.; methodology, W.J.J.; software, S.M.J.; validation, W.J.J. and S.M.J.; formal analysis, S.M.J.; data curation, S.M.J.; writing—original draft preparation, W.J.J.; writing—review and editing, W.J.J. and S.M.J. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Datasets available on request from the authors.

Acknowledgments: The authors would like to thank Hanchi Zhu for the raw data used in this analysis and Jeffrey Krauss of the Wilson College of Textiles for the use of his equipment. We would also like to acknowledge the useful suggestions and comments given by Nilesh Ingle.

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

Abbreviations

The following abbreviations are used in this manuscript:

ANN	Artificial Neural Network
CIE	Commission Internationale de l'Éclairage
CIEDE	International Commission on Illumination 2000 Color-Difference Formula
CMC	Color Measurement Committee
DI	Deionized
NIR	Near-infrared

References

1. United States Environmental Protection Agency; Office of Compliance. *EPA Office of Compliance Sector Notebook Project. Profile of the Textile Industry*; Office of Compliance, Office of Enforcement and Compliance Assurance, U.S. Environmental Protection Agency: Washington, DC, USA, 1997; ISBN 0-16-049401-X.
2. Allen, E.H.; Goldfinger, G. The Color of Absorbing Scattering Substrates. I. The Color of Fabrics. *J. Appl. Polym. Sci.* **1972**, *16*, 2973–2982. [CrossRef]
3. Kubelka, P. New Contributions to the Optics of Intensely Light-Scattering Materials Part I. *J. Opt. Soc. Am.* **1948**, *38*, 448. [CrossRef] [PubMed]
4. Tsoutseos, A.A.; Nobbs, J.H. Alternative Approach to Color Appearance of Textile Materials with Application to the Wet/Dry Reflectance Prediction. *Text. Chem. Color. Am. Dyest. Rep.* **2000**, *32*, 38–43.
5. Jawahar, M.; Narasimhan Kannan, C.B.; Kondamudi Manobhai, M. Artificial Neural Networks for Colour Prediction in Leather Dyeing on the Basis of a Tristimulus System. *Color. Technol.* **2015**, *131*, 48–57. [CrossRef]
6. Shen, J.; Zhou, X.; Ma, H.; Chen, W. Spectrophotometric Prediction of Pre-Colored Fiber Blends with a Hybrid Model Based on Artificial Neural Network and Stearns–Noechel Model. *Text. Res. J.* **2017**, *87*, 296–304. [CrossRef]
7. Khataee, A.R.; Mirzajani, O. UV/Peroxydisulfate Oxidation of C. I. Basic Blue 3: Modeling of Key Factors by Artificial Neural Network. *Desalination* **2010**, *251*, 64–69. [CrossRef]
8. Furferi, R.; Governi, L.; Volpe, Y. Color Matching of Fabric Blends: Hybrid Kubelka-Munk + Artificial Neural Network Based Method. *J. Electron. Imaging* **2016**, *25*, 061402. [CrossRef]
9. Ingle, N.; Jasper, W.J. A Review of Deep Learning and Artificial Intelligence in Dyeing, Printing and Finishing. *Text. Res. J.* **2024**, *95*, 625–657. [CrossRef]
10. Shams-Nateri, A.; Amirshahi, S.H.; Latifi, M. Prediction of Yarn Cross-Sectional Color from Longitudinal Color by Neural Network. *Res. J. Text. Appar.* **2006**, *10*, 25–35. [CrossRef]
11. Saeed, U.; Alsadi, J.; Ahmad, S.; Rizvi, G.; Ross, D. Polymer Color Properties: Neural Network Modeling. *Adv. Polym. Technol.* **2014**, *33*, adv.21462. [CrossRef]
12. Jasper, W.J.; Kovacs, E.T.; Berkstresser, G.A. Using Neural Networks to Predict Dye Concentrations in Multiple-Dye Mixtures. *Text. Res. J.* **1993**, *63*, 545–551. [CrossRef]
13. Jasper, W.J.; Kovacs, E.T. Using Neural Networks and NIR Spectrophotometry to Identify Fibers. *Text. Res. J.* **1994**, *64*, 444–448. [CrossRef]
14. Hemingray, C.; Westland, S. A Novel Approach to Using Neural Networks to Predict the Colour of Fibre Blends. *Color. Technol.* **2016**, *132*, 297–303. [CrossRef]
15. Xiao, Q.; Wang, R.; Zhang, S.; Li, D.; Sun, H.; Wang, L. Prediction of Pilling of Polyester–Cotton Blended Woven Fabric Using Artificial Neural Network Models. *J. Eng. Fibers Fabr.* **2020**, *15*, 1558925019900152. [CrossRef]
16. Kuo, C.J.; Fang, C. Optimization of the Processing Conditions and Prediction of the Quality for Dyeing Nylon and Lycra Blended Fabrics. *Fibers Polym.* **2006**, *7*, 344–351. [CrossRef]
17. Hwang, J.P.; Kim, S.; Park, C.K. Development of a Color Matching Algorithm for Digital Transfer Textile Printing Using an Artificial Neural Network and Multiple Regression. *Text. Res. J.* **2015**, *85*, 1076–1082. [CrossRef]

18. Hajipour, A.; Shams-Nateri, A. Improve Neural Network-Based Color Matching of Inkjet Textile Printing by Classification with Competitive Neural Network. *Color Res. Appl.* **2019**, *44*, 65–72. [CrossRef]
19. Furferi, R.; Carfagni, M. Prediction of the Color and of the Color Solidity of a Jigger-Dyed Cellulose-Based Fabric: A Cascade Neural Network Approach. *Text. Res. J.* **2010**, *80*, 1682–1696. [CrossRef]
20. Jawahar, M.; Venba, R.; Jyothi, G.; Kanth, S.V.; Doss, M.J.; Chandra Babu, N.K. Dry Colour Prediction of Leather from Its Wet State. *Color. Technol.* **2013**, *129*, 252–258. [CrossRef]
21. Melgosa, M.; Cui, G.; Oleari, C.; Pardo, P.J.; Huang, M.; Li, C.; Luo, M.R. Revisiting the Weighting Function for Lightness in the CIEDE2000 Colour-Difference Formula. *Color. Technol.* **2017**, *133*, 273–282. [CrossRef]
22. Oulton, D.P.; Westland, S. Vector-Based Modelling of Colour Difference: A Pilot Study of the DE2000 Colour Difference Model. *Color. Technol.* **2017**, *133*, 15–25. [CrossRef]
23. Luo, M.R.; Rigg, B. Uniform Colour Space Based on the CMC(l:C) Colour-Difference Formula. *J. Soc. Dye. Colour.* **1986**, *102*, 164–171. [CrossRef]
24. CIE. *Industrial Colour-Difference Evaluation*; Publication No. 116; CIE Central Bureau: Vienna, Austria, 1995.
25. Luo, M.R.; Cui, G.; Rigg, B. The Development of the CIE 2000 Colour-Difference Formula: CIEDE2000. *Color Res. Appl.* **2001**, *26*, 340–350. [CrossRef]
26. Ohta, N. Correspondence Between CIELAB and CIELUV Color Differences. *Color Res. Appl.* **1977**, *2*, 178–182. [CrossRef]
27. Color I7 Benchtop Spectrophotometer Operation Manual 2013. Available online: https://www.xrite.com/-/media/xrite/files/manuals_and_userguides/c/o/color_i7_manual_en.pdf (accessed on 10 April 2025).
28. Arnold, C.; Biedebach, L.; Küpfer, A.; Neunhoeffer, M. The Role of Hyperparameters in Machine Learning Models and How to Tune Them. *Political Sci. Res. Methods* **2024**, *12*, 841–848. [CrossRef]
29. Dillon, J.V.; Langmore, I.; Tran, D.; Brevdo, E.; Vasudevan, S.; Moore, D.; Patton, B.; Alemi, A.; Hoffman, M.; Saurous, R.A. TensorFlow Distributions. *arXiv* **2017**, arXiv:1711.10604.
30. Şahin, C.; Balcı, O.; Işık, M.; Gökenç, İ. Artificial Neural Networks Approach for Prediction of CIELab Values for Yarn after Dyeing and Finishing Process. *J. Text. Inst.* **2023**, *114*, 1326–1335. [CrossRef]
31. Balci, O.; Noyan Oulata, S.; Sahin, C.; Turul Oulata, R. An Artificial Neural Network Approach to Prediction of the Colorimetric Values of the Stripped Cotton Fabrics. *Fibers Polym.* **2008**, *9*, 604–614. [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.