



Processing Time Estimation in the Textile Warp Preparation Shop with Machine Learning Approaches

Yunus Demir^{1*} , Müzeyyen Göksel¹ 

¹ Bursa Technical University, Department of Industrial Engineering, 16310, Bursa, Türkiye

ARTICLE INFO

Received Date: 26/09/2025
Accepted Date: 4/11/2025

Cite this paper as:

Demir, Y., & Göksel, M. (2026). Processing Time Estimation in the Textile Warp Preparation Shop with Machine Learning Approaches. *Journal of Innovative Science and Engineering*. 10(1), 105-117.

*Corresponding author: Yunus Demir
E-mail:yunus.demir@btu.edu.tr

Keywords:

Processing time estimation
Production planning
Machine learning
Regression methods

© Copyright 2026 by
Bursa Technical University. Available
online at <http://jise.btu.edu.tr/>



The works published in Journal of Innovative Science and Engineering (JISE) are licensed under a Creative Commons Attribution-NonCommercial 4.0 International License.

ABSTRACT

Accurate determination of processing time has critical importance in the production planning process for several reasons, such as effective planning of production resources and ensuring customer satisfaction by meeting due dates. Traditionally, processing time is often determined through time studies or simple calculations and is usually assumed to be known prior to planning. However, in some special cases, processing time varies depending on many parameters. Such a situation occurs in the warp preparation process, which is one of the important steps of woven fabric production. In this study, supervised machine learning approaches were used to estimate the warp preparation process time based on data obtained from the ERP system of the enterprise where the application was implemented. Twelve different supervised machine learning algorithms were applied to both training and test datasets, and the results are presented comparatively. It was observed that boosting algorithms outperform others in terms of both training/tuning time and estimation accuracy.

1. Introduction

Production planning is a set of decision processes to ensure the correct and efficient flow of production processes according to specific objectives (Khaled et al., 2022). Production planning is a complex task involving many decision-making problems related to various production stages, such as aggregate planning, lot sizing and scheduling (Zarte et al.,

2021). Scheduling is more detailed than production planning, it maps each product/semi-product/raw material to productive resources, uses smaller units of time or even continuous time (Thomas & McClain, 1993).

Processing time has a significant impact on the key performance indicators of the scheduling process, such as total tardiness, total flow time, and makespan.

In scheduling activities, it is generally assumed that the processing time is known in advance and is often a fixed value. In some cases, the function for which processing time should be calculated is known in advance. Such situations often occur in chemical and metallurgical processes.

For example, in steel rolling mills, ingots are heated to the required temperature before rolling. The heating (processing) time depends on the current temperature of the ingot, that is, the time the ingot waits in the process. During the waiting period, the ingot cools; therefore, more heating time is required in the furnace (Gupta & Gupta, 1988). Examples of processing time varying according to a particular function can be found in various environments. In firefighting, a delay in the start of extinguishing activity can increase the time required to control the fire (Kunnathur & Gupta, 1990). In the dark or in deteriorating weather conditions, the processing time of the search/rescue operation may take longer (Mosheiov, 1996). In agriculture, as the crop dries up, the time required to harvest may shorten. Depending on the level of learning (experience), the processing time may be reduced. In these cases, the processing time may increase or decrease as time passes (Alidaee & Womer, 1999). In the literature, the processing time determined in this way is called the dependent processing time. Depending on the type of function, it can be linear, piecewise linear, or nonlinear (Gupta & Gupta, 1988).

In the actual production process, due to unexpected situations such as machine failure, energy shortages, urgent order additions, and tool loading or unloading, processing times cannot be recorded or collected precisely (Joo et al., 2018). It is observed that this situation, which is called uncertainty in processing times, is expressed in different ways in the literature.

Commonly, uncertain processing time is represented by fuzzy numbers based on probability theory (Joo et al., 2018). There are many examples of fuzzy processing time studies, especially in the field of scheduling (Gao et al., 2016). The most obvious difficulty in fuzzification of the processing time is to determine the membership function and an accurate probability distribution (Lei, 2011).

Another approach used to model uncertain processing time is interval numbers. The interval number, which gives the upper and lower bounds of a parameter, can be considered an extension of the real number concept or as a subset of the real line (Moore, 1979). This approach, which eliminates the disadvantages of the fuzzy number, is one of the simplest forms of

representing the uncertainty in the decision matrix and requires a minimum amount of information about the values of the features (Li et al., 2019).

Random (Nagasawa et al., 2015; Horng et al., 2012) or stochastic (He et al., 2021; Alzahrani, 2020) processing time is another way of expressing uncertainty in processing time. In this approach, the processing time of each job is a random variable with a known probability distribution.

As can be seen, the reason for the uncertainty of processing time in the literature is generally defined as incomplete/unreliable information or unavoidable stochastic variability (Jamrus et al., 2017). In this study, the uncertainty in processing time is not due to the probabilistic factors such as machine failure or power outages, but by a combination of factors with previously known fixed values. The problem addressed is inspired by the warp preparation operation, which is one of the important components of the woven fabric production process. Various machine learning algorithms have been used for processing time estimation. The structure of this work is organized as follows. In the second section, the literature on the subject was reviewed. In the third section, the production process in which the estimation study is carried out is explained. In the fourth section, the features used for estimation are introduced. Data preparation and preprocessing operations are explained in the fifth section. In the sixth section, the machine learning algorithms used for the estimation of the processing time in this study are briefly explained. In the seventh section, the model selection method and the performance metrics used to compare the models are presented. In the eighth section, the hyper-parameters and their levels tuned for the algorithms used in the study are given. In the ninth section, the processing time is estimated using 12 different machine algorithms and the numerical results obtained are presented in a comparative way. In the tenth section, a general evaluation of the study is made.

2. Literature Review

Statistical learning has been used as a tool for useful knowledge extraction since the early 90s (Cheng et al., 2017). Initially, it was heavily applied in fields such as medicine, biotechnology, finance, and marketing (Rainer, 2013). In recent years, there has been a significant increase in the number of articles discussing the use of different data analytics methods and techniques in production management. In the literature review conducted by Cheng et al. (2017), it was reported that such studies in production

management are mostly grouped under four headings: advanced planning and scheduling, quality improvement, fault diagnosis and defect analysis. In the same study, a new category was defined as flow time/cycle time estimation.

In the study published by Choudhary et al. in 2009, it was noted that there is a significant gap in the application of statistical learning techniques, especially in the field of production planning. In the literature review conducted by Cheng et al. (2017), it was revealed that few studies have been carried out on the related subject despite the intervening nine years. However, in recent years, there has been a substantial increase in the number of articles discussing the use of different data analytics methods and techniques in production management. In this section, studies focusing on statistical learning applications related to time in production management are examined. Studies are generally grouped under two groups as lead time or cycle/processing time.

The first of these is the lead time, which is defined as the time between the receipt of the customer's order and the delivery of the goods or services to the customer (Gunasekaran et al., 2001). Lead time includes all the time required to manufacture a product, including design, order preparation, purchasing for raw materials/finished parts, queuing, setup, processing, transportation, inspection, and delivery time (Business Dictionary, 2022; Schuh & Stich, 2012). In the literature, lead time can also be called product completion time, makespan, or flow time (Huang et al., 2020). Lead time may vary due to machine failures, missing materials, lack of personnel or insufficient employee qualification (Burggräf et al., 2017). This variability can lead to costs due to loss of customer trust and sanctions for late deliveries. However, these costs can be avoided if delivery dates are determined at an early stage and deviations from the schedule can be identified (Raaymakers & Weijters, 2003; Lödding, 2005). Therefore, the estimation of the lead time has attracted the attention of researchers especially in recent years.

Lingitz et al. (2018) used 11 different machine learning algorithms for lead time estimation in the semiconductor industry. Gyulai et al. (2018a) analyzed a flow shop production environment in the optical industry. In the related study, the authors estimated the uncertain lead time due to the variety of process parameters and especially the high impact of customer demands on the jobs in process. Gyulai et al. (2018b) applied machine learning techniques to proactively predict production lead times to make

decisions by applying a closed-loop production control. Huang et al. (2020) proposed a hybrid approach for lead time estimation by integrating a new mathematical model and deep learning technique for the multi-product mass production line. Extensive literature on lead time prediction can be found in Burggräf et al. (2017).

Another topic in which statistical learning applications related to time in production management concentrated is cycle time. The most important reason why researchers focus on this issue is that accurate determination of cycle time plays an important role in promising a good delivery time (Wang et al., 2018). In some studies (Chen, 2006, 2007), the cycle time, also called output time, is defined simply as the difference between a job's release time and its completion time. Cycle time, which is also defined as the time required for the machine to complete one cycle, includes periods such as waiting, control, rework and failure. Studies on the estimation of the cycle time are concentrated in the field of semiconductor manufacturing (Backus et al., 2006; Chen et al., 2001; Chen et al., 2009; Chien et al., 2005; Meidan et al., 2011; Tirkel, 2011).

In this section, brief explanations of relatively recent studies are shared. Wang et al. (2018a) focused on wafer fabrication, which is the most complex stage of semiconductor fabrication. In the related study, a three-part regression-based model was proposed for the selection of factors that may affect the cycle times of wafer lots. Next, a parallel computational model was applied to estimate the cycle times of wafer batches. Wang et al. (2018b) proposed a new data-intensive cycle time estimation system with parallel computation to quickly estimate the cycle time of wafer lots with large datasets in the semiconductor wafer fabrication system. In the study by Chen and Wang (2020), deep neural networks (DNNs) were applied to accurately predict the cycle times of jobs in the DRAM manufacturing factory. Sun et al. (2022) proposed a data-driven method (neural networks) to predict the feed rate and cycle time of jobs processed on NC machines. The authors demonstrated that the proposed method predicts cycle time with an accuracy greater than 90% compared to other analytical methods. To the best of our knowledge, this study is the first in the textile industry in terms of time estimation in general. In the study, the time required for the preparation of the warp beam to be sent to the weaving stage is estimated. Detailed information on the manufacturing process and the features used in the estimation is presented in the following section.

3. Material and Methods

3.1. Description of Manufacturing Process

Weaving, which is one of the fabric forming methods, is defined as the surface formed by the weft and warp yarns by changing their lower and upper positions in a certain order (Ebiç, 2012). In order for the yarns to become fabric producible in the weaving machine, they need to go through a number of pre-processes. Warp preparation is one of these pre-processes. In this process, the yarns unwound from the bobbins are transferred to beams with certain desired properties (Kumar, 2014). Bobbins are arranged on metal stands called creel in warping machines. The creel is the most important element that ensures that the yarns coming from the bobbins are transferred to the beam with equal tension, parallel to each other and smoothly (Tekstilsayfasi, 2021). The purpose of the warp preparation process is to combine the warp yarns in a beam so that they can be used in the weaving machine. In this way, the weaving machine

can be fed continuously. Another purpose is to maintain the elasticity of the warps while feeding the weaving machine.

For the warp preparation process, two different methods are used in the weaving industry as direct or beam warping and conical warping. In the beam warping machine, the warp yarns coming from the creel pass through the collecting reed and are wrapped directly on the beam with equal tension and parallel to each other. In conical warping machine, the yarns taken from the creel are wrapped in groups (warp section) by sliding them on a conical (sectional) drum in a certain order. The warp yarns wound on the drum are then transferred to the weaving beam. After reaching the planned number of warp section, the warps are transferred to the weaving beam and sent to the weaving process (Ebiç, 2012). The visuals related to the warp preparation process are given in Figure 1. In this study, the issue of estimating the required processing time for each beam prepared in conical warping machines is discussed.

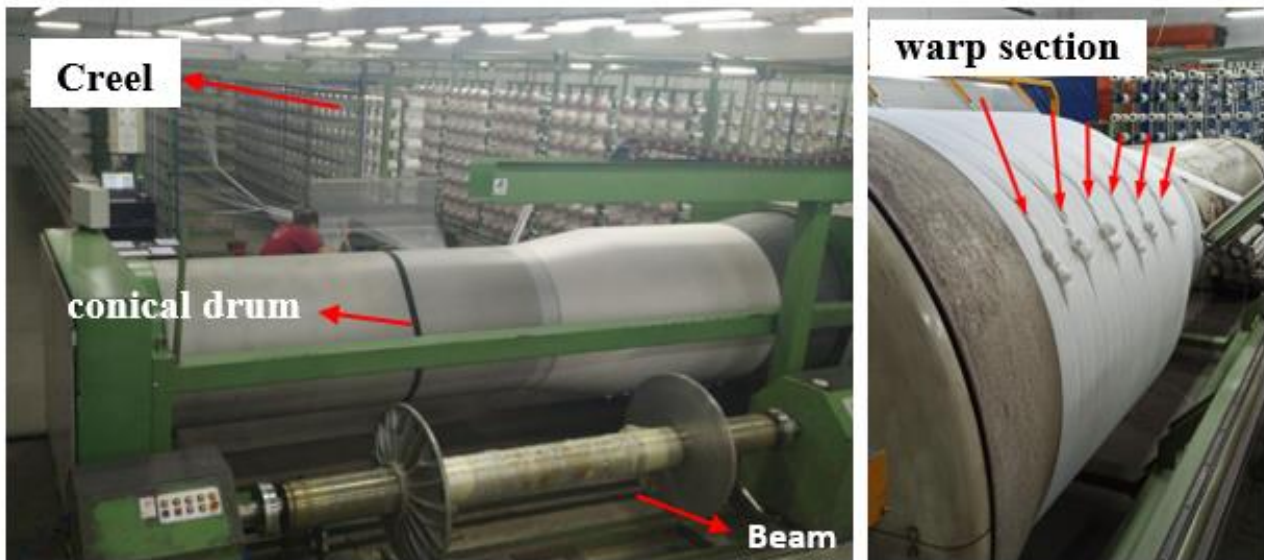


Figure 1: Conical warping machine and its components (left), warp section (right).

3.2. Feature Selection and Description

As the first step in the machine learning process, data collection is critical to building accurate models. When choosing input variables for machine learning methods, a balance must be made between completeness, data size and avoidance of noise. Incomplete data will cause a low rate of explanation of the dependent variable by the independent variables and therefore a high estimation error. In the opposite case, the use of excess data will provide more information, but will also contain a lot of information and noise that is not necessary for the

estimation. This will make the training process of the machine learning algorithm more difficult (Scher & Messori, 2018).

In this context, an evaluation meeting was held with a team from the company where the study was carried out, with the participation of all relevant experts. It has been determined that the processing time required to wind a beam in a conical warping machine may vary depending on the factors given in Table 1.

The winding time of the warp yarn specified in the work order in the conical warping machine starts with

the reading of the barcode number in the work order and ends with the printing of the label containing the information about the relevant beam after winding. Within this period, the creel loading and combing times (setup times), which are made once for each work order, are included. For example, if five beams are planned to be wound in a work order, the processing time of the first beam will be longer due to the specified setup times. In addition, the processing time of the beam winding process includes

the downtimes due to the yarn breakage, the stops during the installation of the new bobbin instead of the finished bobbin (bobbin change) and short-term machine breakdowns. The data set used for this study was taken from the production records of the enterprise between the years 2018-2022. The total number of historical records is 11.000 rows. The data was extracted from the company's ERP system and stored in the file in xlsx format.

Table 1: Estimation parameters.

Parameter	Explanation	Type	Range
Warp wire count	The total number of wire (yarn) in cm according to the width of the fabric	Int	(286, 44880)
Yarn type	The type of yarn used as warp yarn (20 types)	Categorical	
Yarn denier	Weight of 9000 meters long yarn (gr)	Int	(20, 2500)
Warp (reed) width	Fabric width (cm)	Int	(95, 360)
Beam winding speed	The transfer rate of the warp yarns wound on the drum to the weaving beam (m/min)	Int	(50, 250)
Unwinding speed	The winding speed of the yarns in the creel to the drum (m/min)	Int	(80, 700)
Length	The order quantity of the yarns wound on the weaving beam (m)	Int	(70, 19250)
Number of warp sections	It is the number of bands wound side by side to create the desired fabric width.	Int	(1, 40)
The creel loaded?	Indicates the situation where the bobbins are loaded to the creel (1) or not (0).	Int	(0, 1)
Combed?	If the warp width (warp section width) of the pre-job is incompatible with the post-job, reed replacement is performed before the post-job (1) Otherwise, it is not performed (0).	Int	(0, 1)
Machine-ID	The machine number of the warp preparation process. (5 machines)	Categorical	
Processing time	Dependent (output) variable (hour)		

3.3. Data Preprocessing

In data preprocessing, an important step towards creating more accurate estimators through machine learning, erroneous, incomplete or inconsistent data (outliers) are detected. The operations to be performed at this stage also vary depending on the preferred machine learning algorithm. For example, some algorithms (like the Random Forest family) do not support nulls in their inputs, while others can handle them. Or some machine learning algorithms are more robust than others in the presence of outliers (Chibani & Coudert, 2020). Processing times and data regarding related parameters are collected from the ERP system. Raw data sets for the years 2018-2022 require editing due to incorrect/incomplete data entry or unit conversion requirement. For example, denier was chosen as the yarn numbering system. However, data for English cotton yarn system (Ne) were determined in this area. These data in different units have been converted to denier. Inconsistencies caused by operators entering textual data in a numeric field

have been resolved. Records below 30 records for each yarn type were completely deleted.

The records of 2018, when the transition to the ERP system took place, were deleted because they contained missing or incorrect data for many factors.

3.4. Machine Learning Regression Algorithms

This study, which estimates the time required for the warp yarns to be wound on the beam in line with the characteristics specified in the work order in the conical warp preparation machine, can be evaluated within the scope of supervised learning, which is a sub-branch of machine learning. Machine learning algorithms can be divided into linear and nonlinear according to whether they need to make strong assumptions about the form of the mapping function. Linear methods assume that features affect the final result only through a linear model. In contrast, nonlinear methods are more general, and their performance is less dependent on the model

assumption (Ouyang et al., 2019). In this section, the machine learning regression algorithms (six linear and six nonlinear) used in the study are briefly explained.

3.4.1. Linear regression algorithms

Multiple Linear Regression (MLR): They are the most basic linear regression models that can be easily interpreted, assuming an approximately linear relationship between the variables (Lingitz et al., 2018).

Principal component regression (PCR): It is a very common dimension reduction technique that solves the multicollinearity problem and reduces the number of independent variables by creating linear combinations of independent variables. The main idea of PCR, which is a biased regression technique, is to create a regression model by reducing the dependent variable to the basic components of the independent variables, instead of regressing the dependent variable directly on the independent variables (Kurnaz, 2020).

Partial least squares regression (PLCR): It was developed by Wold (1966) to create effective predictive models with small datasets and missing values when the independent variables are large (even more than the number of observations) and highly multicollinearity (Phatak & Jong, 1997). The goal is to create components that capture most of the information in the independent variables that are useful for estimation while reducing the dimensionality of the regression problem (Garthwaite, 1994).

Ridge regression (RR): The ridge regression method is one of the most common methods for handling multicollinearity problems. It outperforms other approaches, especially when the sample size is small (El-Dereny & Rashwan, 2011; Fitrianto & Yik, 2014). The two previous techniques for solving the multicollinearity problem were based on the principle of constructing a predictive model with fewer sets of variables, which are a combination of the available independent variables. However, the linear methods that will be explained under this and the following headings are based on the penalization principle. These models are also called shrinkage models (Fidanoğlu & Akdeniz, 2010).

Lasso regression (LR): It is a suitable method for regression of high-dimensional data. RR includes all independent variables in the final model. In contrast, Lasso regression tends to completely remove the weights of the least important features (i.e., set them

to zero). In other words, Lasso regression also performs feature selection (Aytekin, 2021). Thus, Lasso produces interpretable models that help prevent overfitting (Tibshirani, 1996).

ElasticNet: It is a linear regression approach trained with the coefficients and the regularization norm of both algorithms, providing a bridge between RR and Lasso (Hans 2011; Scikit-Learn, 2020). It is so called because it acts like a net that does not miss the big “fish,” capturing the most important features (Zou & Hastie 2003).

3.4.2. Nonlinear regression algorithms

k nearest neighbors (KNN): It estimates the results using the closest k samples from the training set (Kuhn, 2013). The KNN algorithm is a non-parametric method for classification and regression (Altman, 1992). The only hyper parameter of the KNN algorithm is the k value. If this value is too low, overfitting occurs; if it is too high, underfitting occurs. The disadvantage of the KNN algorithm is that the processing load increases as the amount of data increases due to the distance calculation process (Seçkin et al., 2019).

Multilayer perceptron (MLP): MLPs are neural network models that work as universal predictors. That is, they can approximate any continuous function (Hornik, 1991). Multilayer perceptrons, which fall under the category of feedforward algorithms, have input and output layers, as well as one or more hidden layers where many neurons are stacked together (Bento, 2021).

Regression trees (RT): RT is an algorithm that works with simple if-then-else decision rules and is frequently used in supervised learning (Quinlan, 1986). RT creates classification or regression models in the form of a tree structure. The aim is to transform the complex structures in the data set into simple decision structures. Data sets are divided into homogeneous subgroups according to a determined target variable (Breiman et al., 2017). The result of the algorithm is a tree with decision nodes. RTs can handle both categorical and numerical data (Quinlan, 1987).

Random forest (RF): RF is a special case of a bagged regression tree, in which only randomly selected estimators are used in the tree generation process to reduce the correlation between estimators (Lingitz et al., 2018). It is designed as a forest consisting of many regression trees. RF, from the group of ensemble methods, is easy to interpret in most cases and has the ability to capture nonlinear correlations between

variables (James et al., 2013). However, it tends to overfit (Bishop, 2006).

LightGBM: It is a gradient boosting algorithm that uses tree-based learning algorithms (LightGBM documentation, 2022). This algorithm, developed by Microsoft, uses a histogram-based approach to reduce memory consumption and speed up the training process (Machado et al., 2019). It reduces the computational cost by converting the variables that have continuous values to discrete ones (Muratlar, 2020). Ke et al. (2017) concluded that it is 20 times faster than other models.

CatBoost: It is a new gradient boosting algorithm developed by Yandex engineers and researchers. This open-source algorithm can process categorical features with low information loss (Jabeur et al., 2021). CatBoost uses a more efficient strategy that allows the entire dataset to be used for training (Dorogush et al., 2017). In addition, CatBoost performs random permutations to estimate leaf values while selecting the tree structure to overcome the overfitting caused by traditional gradient boosting algorithms (Dorogush et al., 2018). Another advantage of CatBoost is that it provides great results with default parameters, reducing the time spent on parameter setting (CatBoost documentation, 2022).

3.5. Model Selection and Evaluation

Machine learning procedures learn from data to fit models and make predictions with the models created. However, there is no universal model suitable for every data, and different machine learning techniques can outperform each other according to the data. Therefore, a crucial step in a typical data analysis is to evaluate a set of candidate models and then select the most suitable one (Ding et al., 2018). In summary, this process, in which a final machine learning model is selected from a set of candidate models for a given training dataset, is called model selection (Brownlee, 2019).

In the last few decades, a wide variety of model selection methods have been proposed. Among these, percentage split and cross-validation are the most common. In the percentage split method, all data is manually divided into two sets: training and test. The training dataset is used for the learning process, and the test dataset is used for the performance evaluation. However, evaluation results may not be reliable due to differences in the distribution of training and test datasets, uneven distribution of outliers, etc. For this reason, the cross-validation method was developed. Being an iterative process, this method divides the data into k discrete clusters of approximately equal

size. In each iteration, nine discrete sets are used as training data and one set as test data. The cross-validation method gives more reliable results than the percentage split method. However, it may take a long time to result as learning and testing is done in each iteration (Seçkin et al., 2019).

In this study, 10-fold cross validation was used to determine the prediction accuracy of the models. The regression algorithms used in the study are compared in terms of five commonly used performance metrics (Table 2).

Table 2: Performance metrics.

Metric	Equation
Root mean squared error (RMSE)	$\sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2}$
Normalized root mean squared error (NRMSE)	$\frac{RMSE}{y_{max} - y_{min}} * 100$
Mean absolute error (MAE)	$\frac{1}{n} \sum_{i=1}^n y_i - \hat{y}_i $
Mean absolute percentage error (MAPE)	$\frac{1}{n} \sum_{i=1}^n \left \frac{y_i - \hat{y}_i}{y_i} \right * 100$
R^2	$1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2}$

Here y_i is the actual processing time, and \hat{y}_i are the estimated and average processing time, respectively. n is the number of observations considered in the test dataset. y_{max} and y_{min} are the largest and smallest values among the actual values, respectively

3.6. Hyper-Parameter Tuning

Hyper-parameters are parameters that are set before the learning process starts, whose value cannot be estimated from the data, and which define the model architecture. Hyper-parameter tuning/optimization is an important part of controlling the behavior of a machine learning model (Navas, 2022). The performance of machine learning algorithms is highly dependent on hyper-parameters. Therefore, tuning the hyper-parameters is a critical step in designing a model that can predict with greater accuracy. Also, tuning hyper-parameters prevents the model from suffering from overfitting, which is a serious problem in machine learning (Dash et al., 2021).

Grid search and random search methods were chosen to tune the hyper-parameters of 12 machine learning algorithms used in this study. Grid search evaluates every possible combination of hyper-parameters in

the search space (James et al., 2011). For each combination, a model is created and evaluated using these hyper-parameter values. The combination with the highest accuracy is reported (Villalobos-Arias et al., 2020). It can be computationally expensive from grid search, especially when too many hyper-parameters are again optimized for multiple levels. In this case, random search, another popular hyper-parameter tuning technique, is preferred. Random search follows an approach that samples a subset of the search space (James et al., 2012). Information

about the hyperparameters investigated for each technique used in this study is presented in Table 3.

Since the optimal limits of the parameter levels were not known in advance, tests were initially conducted over a rough range, and then the search area was scanned in a finer range. The parameter levels presented in Table 3 are levels defined within the fine range. Optimum values for each hyperparameter are also shared in the relevant table.

Table3: Hyper-parameters tuned for selected algorithms.

Approach	Hyperparameter	Possible Values	Optimal Values
PCR	num_components	(1 ,2 ,..., 39)	35
PLS	num_components	(1 ,2 ,..., 39)	36
RR	alpha	10^x , $x=(5, 4.9, 4.8, \dots, -0.8, -0.9, -1)$	0.005
Lasso	alpha	$10x$, $x=(5, 4.9, 4.8, \dots, -0.8, -0.9, -1)$	0.00004
KNN	k	(1 ,2 ,..., 30)	5
RT	max_depth	(8, 10, 20, 30, 40, None)	20
	max_features	auto, sqrt	auto
	max_leaf_nodes	(120, 140, 160, 190, 210)	210
	min_samples_leaf	(4, 6, 8, 10, 15)	4
	min_samples_split	(2, 5, 8, 10, 20)	10
MLP	learning_rate_init	(0.00001, 0.01, 0.1, 0.5)	0.001
	activation	relu, logistic, identity, tanh	relu
	hidden_layer_sizes	[(3), (6), (4, 4), (20,20), (40, 40), (50, 50), (60, 60)]	(50, 50)
	alpha	(0.1, 0.01, 0.001, 0.0001)	0.0001
Random Forest	max_depth	(2, 4, 6, 8, 10)	10
	max_features	(5, 10, 15, 20, 25, 30, 35)	35
	n_estimators	(200, 500, 1000, 2000)	2000
LightGBM	colsample_bytree	(0.2, 0.4, 0.6, 0.8, 1)	0.8
	learning_rate	(0.07, 0.09, 1.10, 1.30, 1.50)	0.09
	max_depth	(1, 3, 5, 7, 9)	3
	n_estimators	(1000, 1200, 1600, 1800, 2000)	1800
CatBoost	iterations	(200, 500, 1000, 2000)	2000
	learning_rate	(0.01, 0.03, 0.05, 0.1)	0.05
	depth	(3, 4, 5, 6, 7, 8)	5

4. Results and Discussion

In this section, 12 different machine learning algorithms are tested for the problem of estimating the warp preparation processing time. Comparative results according to five different metrics are presented in Table 4. The scikit-learn library and Google Colaboratory (Colab) notebook were used for estimation.

All algorithms used in the study were evaluated on both the training and test sets, and the results are presented comparatively in Table 4. In this way, an attempt was made to observe whether overfitting, one of the most basic problems of machine learning, occurs. An algorithm that performs well on the training set but underperforms on the test set is an indication of overfitting. This means that the

generalization ability of the created model is low and will not yield successful results for new data. The results presented in Table 4 show that all algorithms perform very close to each other for the training and test sets, except for the small difference in the results obtained with the KNN algorithm.

A MAPE value of less than 20% indicates that the model has good accuracy in fitting and in prediction. In this context, it seems that all the methods analyzed have the ability to give accurate results and each of them is applicable in real life conditions. There is no single model that is consistently better than any other in terms of all performance measures. However, in general, it has been observed that nonlinear machine learning algorithms yield better results across all metrics. Especially boosting machine algorithms produced quality results. The CatBoost and

LightGBM methods showed almost the same performance in terms of RMSE and NRMSE metrics. However, in the context of MAPE and MAE, LightGBM excelled. The KNN algorithm produced

the worst results across all metrics except NRMSE. The lowest NRMSE value was obtained with the PCR algorithm.

Table4: Performance comparison among different machine learning models.

Linear Machine Learning Methods												
	MLR		PCR		PLC		Ridge		Lasso		ElasticNet	
	Train	Test	Train	Test	Train	Test	Train	Test	Train	Test	Train	Test
RMSE	0,826	0,930	0,826	0,930	0,815	0,910	0,826	0,929	0,826	0,929	0,899	0,988
NRMSE	5,47%	4,55%	5,47%	7,16%	5,47%	4,55%	5,54%	4,65%	5,54%	4,65%	6,03%	4,95%
MAE	0,591	0,607	0,591	0,607	0,586	0,595	0,591	0,608	0,591	0,607	0,661	0,669
MAPE	12,70%	12,67%	12,69%	12,67%	12,60%	12,70%	12,69%	12,69%	12,69%	12,67%	14,54%	14,52%
R²	0,838	0,838	0,838	0,838	0,842	0,821	0,838	0,837	0,838	0,838	0,808	0,808
NonLinear Machine Learning Methods												
	KNN		RT		MLP		Random Forest		LightGBM		CatBoost	
	Train	Test	Train	Test	Train	Test	Train	Test	Train	Test	Train	Test
RMSE	0,999	1,235	0,702	1,099	0,491	0,945	0,536	0,836	0,427	0,732	0,416	0,730
NRMSE	6,70%	6,18%	3,70%	4,18%	3,29%	4,73%	3,59%	4,18%	2,86%	3,66%	2,79%	3,65%
MAE	0,663	0,802	0,510	0,680	0,366	0,529	0,376	0,504	0,238	0,345	0,310	0,395
MAPE	14,30%	17,25%	11,20%	14,22%	8,1%	10,75%	8,31%	10,7%	5,54%	7,58%	7,07%	8,53%
R²	0,762	0,762	0,882	0,882	0,969	0,987	0,932	0,932	0,957	0,957	0,959	0,959

5. Conclusion

In this study, the warp preparation process, which is one of the important components of woven fabric production stages, is the focus. The processing time required for the preparation of a beam in the relevant process, in which the warp yarns are wound on beams for use in weaving machines, varies depending on many parameters. First of all, interviews were held with the experts of the subject, and as a result, 11 features that could affect the processing time were determined. Then, a total of 12 different machine learning algorithms, six linear and six nonlinear, were tested to estimate the warp preparation processing time. The experimental study was carried out with the data obtained from the ERP system of the enterprise for the years 2018-2022. After data elimination during preprocessing, approximately 8000 rows of data were used. The results obtained as a result of the experimental study show that Boosting algorithms are clearly superior to the other algorithms.

Integration of the developed forecasting model into the company's ERP system is planned. With a more accurate estimation of the processing time, it is expected that the idleness of the weaving looms, caused by waiting for the warp beams required for weaving, will decrease. In addition, with more accurate planning, it is expected that the capacity

utilization rate of warping machines will increase, and as a result, contract manufacturing costs will decrease.

Article Information

Financial Disclosure: The author (s) has no received any financial support for the research, authorship or publication of this study.

Authors' Contribution: Concept: Demir, Göksel; Design: Demir, Göksel; Supervision: Demir, Göksel; Resources: Demir, Göksel; Data Collection: Demir, Göksel; Analysis: Demir, Göksel; Literature Search: Demir, Göksel; Writing Manuscript: Demir, Göksel; Critical Review: Demir, Göksel.

Conflict of Interest/Common Interest: The author declares that he has no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Ethics Committee Approval: This study does not require ethics committee permission or any special permission.

Declaration of the Author(s): The author(s) declare that there is no conflict of interest regarding the publishing of the paper by the Journal of Innovative

Science and Engineering, that the paper has been not published elsewhere, and not include any form of plagiarism. All the authors listed above have approved the manuscript and have agreed with the submission of the manuscript to the Journal of Innovative Science and Engineering.

Acknowledgements: I would like to thank all Küçükçalık Textile family in the presence of Planning Manager Zeynep Işıkdoğdu and department Chief İsmail Hakkı Demir for their valuable contribution to this study.

Artificial Intelligence Statement: The author bear full responsibility for the content and accuracy of their work, including any use of artificial intelligence (AI) technologies, and confirm that they have read the AI Policy, which is accessible on the journal's website.

References

- Alidaee, B., & Womer, N. K. (1999). Scheduling with time dependent processing times: Review and extensions. *Journal of the Operational Research Society*, 50(7), 711–720.
- Altman, N. S. (1992). An introduction to kernel and nearest-neighbor nonparametric regression. *The American Statistician*, 46(3), 175–185.
- Aytekin, H. T. (2021). Makine öğreniminin araştırmacıların veri analizi bağlamında potansiyel önemi. *Ufuk Üniversitesi Sosyal Bilimler Enstitüsü Dergisi*, 10(19), 85–106.
- Backus, P., Janakiram, M., Mowzoon, S., Runger, C., & Bhargava, A. (2006). Factory cycle-time prediction with a data-mining approach. *IEEE Transactions on Semiconductor Manufacturing*, 19(2), 252–258.
- Bento, C. (2021). *Multilayer perceptron explained with a real-life example and Python code: Sentiment analysis*. Towards Data Science. <https://towardsdatascience.com/multilayer-perceptron-explained-with-a-real-life-example-and-python-code-sentiment-analysis-cb408ee93141>
- Bishop, C. M. (2006). *Pattern recognition and machine learning* (Information Science and Statistics). Springer.
- Breiman, L., Friedman, J. H., Olshen, R. A., & Stone, C. J. (2017). *Classification and regression trees*. Routledge.
- Brownlee, J. (2019). *A gentle introduction to model selection for machine learning*. Machine Learning Mastery. <https://machinelearningmastery.com/a-gentle-introduction-to-model-selection-for-machine-learning/>
- Burggräf, P., Wagner, J., Lück, K., & Adlon, T. (2017). Cost-benefit analysis for disruption prevention in low-volume assembly. *Production Engineering*, 11(3), 331–342.
- Business Dictionary. (2022, August 15). <http://www.businessdictionary.com/>
- CatBoost Documentation. (2022). <https://catboost.ai/>
- Chen, C. Y., George, E. I., & Tardif, V. (2001). A Bayesian model of cycle time prediction. *IIE Transactions*, 33(10), 921–930.
- Chen, T. (2006). A hybrid SOM-BPN approach to lot output time prediction in a wafer fab. *Neural Processing Letters*, 24(3), 271–288.
- Chen, T. (2007). An intelligent hybrid system for wafer lot output time prediction. *Advanced Engineering Informatics*, 21(1), 55–65.
- Chen, T., Wang, Y. C., & Tsai, H. R. (2009). Lot cycle time prediction in a ramping-up semiconductor manufacturing factory with a SOM-FBPN-ensemble approach. *The International Journal of Advanced Manufacturing Technology*, 42(11), 1206–1216.
- Chen, T. C. T., & Wang, Y. C. (2020). *An explainable deep-learning approach for job cycle time prediction*. SSRN. <https://doi.org/10.2139/ssrn.4136028>
- Cheng, Y., Chen, K., Zhang, Y., & Tao, F. (2017). Data and knowledge mining with big data towards smart production. *Journal of Industrial Information Integration*. <https://doi.org/10.1016/j.jii.2017.08.001>
- Chibani, S., & Coudert, F. X. (2020). Machine learning approaches for the prediction of materials properties. *APL Materials*, 8(8), 080701.
- Chien, C. F., Hsiao, C. W., Meng, C., Hong, K. T., & Wang, S. T. (2005). Cycle time prediction and control based on production line status and manufacturing data mining. In *ISSM 2005: IEEE International Symposium on Semiconductor Manufacturing* (pp. 327–330). IEEE.
- Dash, R. K., Nguyen, T. N., Cengiz, K., & Sharma, A. (2021). Fine-tuned support vector regression

- model for stock predictions. *Neural Computing and Applications*, 1–15.
- Ding, J., Tarokh, V., & Yang, Y. (2018). Model selection techniques: An overview. *IEEE Signal Processing Magazine*, 35(6), 16–34.
- Dorogush, A. V., Gulin, A., Gusev, G., Ostroumova Prokhorenkova, L., & Vorobev, A. (2017). *CatBoost: Unbiased boosting with categorical features*. arXiv:1706.09516.
- Dorogush, A. V., Ershov, V., & Gulin, A. (2018). CatBoost: Gradient boosting with categorical features 1-7.
- Ebiç, O. (2012). *Tek bobinden çözgü hazırlama makinesinin tasarımı ve değerlendirilmesi* (Yüksek lisans tezi). Gazi Üniversitesi, Ankara.
- El-Dereny, M., & Rashwan, N. (2011). Solving multicollinearity problem using ridge regression models. *International Journal of Contemporary Mathematical Sciences*, 6, 585–600.
- Fidanoğlu, I., & Akdeniz, F. (t.y.). *İstatistiksel daraltıcı (shrinkage) model ve uygulamaları*.
- Fitrianto, A., & Yik, L. C. (2014). Performance of ridge regression estimator methods on small sample size. *Journal of Mathematics and Statistics*, 10, 25–29.
- Gao, K. Z., Suganthan, P. N., Pan, Q. K., Chua, T. J., Chong, C. S., & Cai, T. X. (2016). An improved artificial bee colony algorithm for flexible job-shop scheduling problem with fuzzy processing time. *Expert Systems with Applications*, 65, 52–67.
- Garthwaite, P. H. (1994). An interpretation of partial least squares. *Journal of the American Statistical Association*, 89(425), 122–137.
- Gunasekaran, A., Patel, C., & Tirtiroglu, E. (2001). Performance measures and metrics in a supply chain environment. *International Journal of Operations & Production Management*, 21(1/2), 71–87.
- Gupta, J. N., & Gupta, S. K. (1988). Single facility scheduling with nonlinear processing times. *Computers & Industrial Engineering*, 14(4), 387–393.
- Gyulai, D., Pfeiffer, A., Nick, G., Gallina, V., Sihn, W., & Monostori, L. (2018). Lead time prediction in a flow-shop environment. *IFAC-PapersOnLine*, 51(11), 1029–1034.
- Gyulai, D., Pfeiffer, A., Bergmann, J., & Gallina, V. (2018). Online lead time prediction supporting situation-aware production control. *Procedia CIRP*, 78, 190–195.
- Hans, C. (2011). Elastic net regression modeling with the orthant normal prior. *Journal of the American Statistical Association*, 106(496), 1383–1393.
- He, J., Chu, F., Zheng, F., & Liu, M. (2021). A green-oriented bi-objective disassembly line balancing problem. *Annals of Operations Research*, 296(1), 71–93.
- Horng, S. C., Lin, S. S., & Yang, F. Y. (2012). Evolutionary algorithm for stochastic job shop scheduling. *Expert Systems with Applications*, 39(3), 3603–3610.
- Hornik, K. (1991). Approximation capabilities of multilayer feedforward networks. *Neural Networks*, 4(2), 251–277.
- Huang, J., Chang, Q., & Arinez, J. (2020). Product completion time prediction. *Journal of Manufacturing Systems*, 57, 311–322.
- Jabeur, S. B., Gharib, C., Mefteh-Wali, S., & Arfi, W. B. (2021). CatBoost model. *Technological Forecasting and Social Change*, 166, 120658.
- Bergstra, J. S., Bardenet, R., Bengio, Y., & Kégl, B. (2011). Algorithms for hyper-parameter optimization. *Advances in Neural Information Processing Systems*, 2546–2554.
- James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). *An introduction to statistical learning*. Springer.
- Bergstra, J., & Bengio, Y. (2012). Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, 13, 281–305.
- Joo, B. J., Shim, S. O., Chua, T. J., & Cai, T. X. (2018). Multi-level job scheduling. *Computers & Industrial Engineering*, 120, 480–487.
- Ke, G., Meng, Q., Finley, T., et al. (2017). LightGBM. *Advances in Neural Information Processing Systems*, 30.
- Khaled, M. S., Shaban, I. A., Karam, A., Hussain, M., Zahran, I., & Hussein, M. (2022). Research trends in sustainability of production planning. *Energies*, 15(2), 483.

- Kuhn, M., & Johnson, K. (2013). *Applied predictive modeling*. Springer.
- Kumar, S. M. (2014). *Industrial practices in weaving preparatory*. Woodhead Publishing.
- Kunnathur, A. S., & Gupta, S. K. (1990). Minimizing the makespan with late start penalties added to processing times in a single facility scheduling problem. *European Journal of Operational Research*, 47(1), 56-64.
- Lei, D. (2011). Population-based neighborhood search. *Computers & Industrial Engineering*, 61(4), 1200-1208.
- Kurnaz, F. S. (2020). Temel bileşenlere dayalı bazı regresyon yöntemleri. *Gümüşhane Üniversitesi Fen Bilimleri Dergisi*, 10(3), 581-591.
- Lingitz, L., Gallina, V., Ansari, F., Gyulai, D., Pfeiffer, A., Sihn, W., & Monostori, L. (2018). Lead time prediction using machine learning algorithms: A case study by a semiconductor manufacturer. *Procedia CIRP*, 72, 1051-1056.
- Li, X., Gao, L., Wang, W., Wang, C., & Wen, L. (2019). Particle swarm optimization hybridized with genetic algorithm. *Computers & Industrial Engineering*, 135, 1036-1046.
- LightGBM Documentation. (2022). <https://lightgbm.readthedocs.io/en/v3.3.2/>
- Lödding, H. (2005). *Verfahren der fertigungssteuerung*. Springer.
- Machado, M. R., Karray, S., & de Sousa, I. T. (2019). LightGBM: Predicting customer loyalty. In *2019 International Conference on Computer Science & Education* (pp. 1111-1116). IEEE.
- Meidan, Y., Lerner, B., Rabinowitz, G., & Hassoun, M. (2011). Cycle-time key factor identification and prediction in semiconductor manufacturing using machine learning and data mining. *IEEE Transactions on Semiconductor Manufacturing*, 24(2), 237-248.
- Moore, R. E. (1979). *Methods and applications of interval analysis*. Society for Industrial and Applied Mathematics.
- Mosheiov, G. (1996). A-shaped policies for schedule deteriorating jobs. *Journal of the Operational Research Society*, 47, 1184-1191.
- Muratlar, E. R. (2020). *LightGBM*. <https://www.veribilimiokulu.com/lightgbm/>
- Nagasawa, K., Ikeda, Y., & Irohara, T. (2015). Robust flow shop scheduling with random processing times for reduction of peak power consumption. *Simulation Modelling Practice and Theory*, 59, 102-113.
- Navas, J. (2022). *What is hyperparameter tuning?* Anyscale. <https://www.anyscale.com/blog/what-is-hyperparameter-tuning>
- Ouyang, F. S., Guo, B. L., Ouyang, L. Z., Liu, Z. W., Lin, S. J., Meng, W., ... & Yang, S. M. (2019). Comparison between linear and nonlinear machine-learning algorithms for the classification of thyroid nodules. *European Journal of Radiology*, 113, 251-257.
- Quinlan, J. R. (1987). Simplifying decision trees. *International Journal of Man-Machine Studies*, 27, 221-234.
- Phatak, A., & De Jong, S. (1997). The geometry of partial least squares. *Journal of Chemometrics*, 11(4), 311-338.
- Rainer, C. (2013). Data mining as technique to generate planning rules for manufacturing control in a complex production system. In *Robust Manufacturing Control* (pp. 203-214). Springer.
- Raaymakers, W. H., & Weijters, A. J. M. M. (2003). Makespan estimation in batch process industries: A comparison between regression analysis and neural networks. *European Journal of Operational Research*, 145(1), 14-30.
- Seçkin, M., Seçkin, A. Ç., & Coşkun, A. (2019). Production fault simulation and forecasting from time series data with machine learning in glove textile industry. *Journal of Engineered Fibers and Fabrics*, 14, 1558925019883462.
- Scher, S., & Messori, G. (2018). Predicting weather forecast uncertainty with machine learning. *Quarterly Journal of the Royal Meteorological Society*, 144(717), 2830-2841.
- Scikit-learn. (n.d.). Elastic net. https://scikit-learn.org/stable/modules/linear_model.html#elastic-net
- Schuh, G., & Stich, V. (2012). Konzeptentwicklung in der Produktionsplanung. In *Produktionsplanung und-steuerung 2* (pp. 149-415). Springer.

Sun, C., Dominguez-Caballero, J., Ward, R., Ayvar-Soberanis, S., & Curtis, D. (2022). Machining cycle time prediction: Data-driven modelling of machine tool feedrate behavior with neural networks. *Robotics and Computer-Integrated Manufacturing*, 75, 102293.

Thomas, L. J., & McClain, J. O. (1993). An overview of production planning. In *Handbooks in Operations Research and Management Science* (Vol. 4, pp. 333–370).

Tibshirani, R. J. (1996). Regression shrinkage via the LASSO. *Journal of the Royal Statistical Society: Series B*, 58, 267–288.

Tirkel, I. (2011). Cycle time prediction in wafer fabrication line by applying data mining methods. In *IEEE/SEMI Advanced Semiconductor Manufacturing Conference* (pp. 1–5).

Villalobos-Arias, L., Quesada-López, C., Guevara-Coto, J., Martínez, A., & Jenkins, M. (2020). Evaluating hyper-parameter tuning using random search in support vector machines for software effort estimation. *Proceedings of the ACM Conference on Predictive Models and Data Analytics*.

Wang, J., Zhang, J., & Wang, X. (2018). A data driven cycle time prediction with feature selection in a semiconductor wafer fabrication system. *IEEE Transactions on Semiconductor Manufacturing*, 31(1), 173–182.

Wang, J., Yang, J., Zhang, J., Wang, X., & Zhang, W. (2018). Big data driven cycle time parallel prediction for production planning in wafer manufacturing. *Enterprise Information Systems*, 12(6), 714–732.

Wold, H. (1966). E). Estimation of principal components and related models by iterative least squares. In *Multivariate Analysis* (pp. 391–420). Academic Press.

Zarte, M., Pechmann, A., & Nunes, I. (2021). Fuzzy Inference Model for Decision Support in Sustainable Production Planning Processes: A Case Study. *Sustainability*, 13, 1355.

Zou, H., & Hastie, T. (2003). Regression shrinkage and selection via the elastic net, with applications to microarrays. *Journal of the Royal Statistical Society: Series B*, 67, 301–320.