Paper technology

José L. Rodríguez-Álavrez*, Rogelio López-Herrera, Iván E. Villalón-Turrubiates, Jorge L. García-Alacaraz, José R. Díaz-Reza, Jesús L. Arce-Valdez, Osbaldo Aragón-Banderas and Arturo Soto-Cabral

Alternative method for determining basis weight in papermaking by using an interactive soft sensor based on an artificial neural network model

https://doi.org/10.1515/npprj-2022-0021
Received February 28, 2022; accepted June 24, 2022

Abstract: Currently, there are two procedures to determine the basis weight in papermaking processes: the measurements made by the quality control laboratory or the measurements made by the quality control system. This research presents an alternative to estimating basis weight-based artificial neural network (ANN) modeling. The NN architecture was constructed by trial and error, obtaining the best results using two hidden layers with 48 and 12 neurons, respectively, in addition to the input and output layers. Mean absolute error and mean absolute percentage error was used for the loss and metric functions, respectively. Python was used in the training, validation, and testing process. The results indicate that the model can reasonably determine the basis weight given the independent variables analyzed here. The R^2 reached by the model was 94 %, and *MAE* was 12.40 grams/m^2. Using the same dataset, the fine tree regression model showed an R^2 of 99 % and an *MAE* of 3.35 grams/m^2. Additionally, a dataset not included in the building process was used to validate the method's performance. The results showed that ANN-based modeling has a higher predictive capability than the regression tree model. Therefore, this model was embedded in a graphic user interface that was developed in Python.

Keywords: artificial neural network model; data science; graphic user interface; hyperparameters; papermaking.

***Corresponding author: José L. Rodríguez-Álavrez,** Department of Doctoral Program in Engineering Sciences, Instituto Tecnologico y de Estudios Superiores de Occidente, Tlaquepaque, JAL 45604, Mexico, e-mail: ng718130@iteso.mx
Rogelio López-Herrera, Iván E. Villalón-Turrubiates, Department of Doctoral Program in Engineering Sciences, Instituto Tecnologico y de Estudios Superiores de Occidente, Tlaquepaque, JAL 45604, Mexico, e-mails: rogelio@iteso.mx, villalon@iteso.mx
Jorge L. García-Alacaraz, José R. Díaz-Reza, Division of Research and Postgraduate Studies, TecNM/Ciudad Juárez, Ciudad Juárez, CHIH 32500, Mexico, e-mails: jorge.garcia@uacj.mx, al164440@alumnos.uacj.mx
Jesús L. Arce-Valdez, Osbaldo Aragón-Banderas, Mechatronic Engineering Department, TecNM/Región de los Llanos, Guadalupe Victoria, DGO 34700, Mexico, e-mails: jesus.av@regionllanos.tecnm.mx, osbaldo.ab@regionllanos.tecnm.mx
Arturo Soto-Cabral, Industrial Engineering Department, TecNM/Durango, Durango, DGO 34000, Mexico, e-mail: soto.cabral@itdurango.mx

Introduction

It is a standard practice to take measurements of product characteristics, either of a variable or attribute type, to study specific processes by changing variables suspected of contributing to the process variation. The resulting data are then analyzed in a certain way to determine if the changes occurred in these variables have had a significant effect either scientifically or economically (Kim et al. 2019). The papermaking processes are no exception since these processes have multiple variables that affect product quality, such as basis weight, caliper, moisture content, ash content, and fiber orientation (Merbold et al. 2016), to mention a few. In addition, whiteness is an essential property of coated paper (Tarasov et al. 2018). Therefore, it is vital to monitor these variables to identify the significant differences in reducing process or product variation and successfully improving product quality.

The traditional control chart technique has been widely used in different industries (Dudek-Burlikowska 2005, Camargo et al. 2010, Shamsuzzaman et al. 2015, Zhiyuan and Jinsheng 2015, Zaman et al. 2020, Rodríguez-Álvarez et al. 2021). Although, the fuzzy set theory proposed by (Zadeh 1965) has also been employed for at least three decades to develop control charts based on this theory (Chang and Aw 1996, Cheng 2005, Gülbay and Kahraman 2006, Gülbay and Kahraman 2007, Kaya and Kahraman 2011, Shu and Wu 2011). However, these approaches do not relate the operating conditions to the output variables (quality characteristics). This aspect is one of their main disadvantages. To overcome this problem, regression models are tools that relate input to output variables and have been widely used for predictive and process optimization purposes in the pulp and paper industry. For instance, (Adamopoulos et al. 2016) presented a predictive model for the mechanical properties of corrugated base papers from fiber and physical property data using multiple linear regression and artificial neural networks. (Kilulya et al. 2015) used a partial least squares (PLS) regression model to evaluate the effects and influence of the lipophilic extractive residues on quality parameters of dissolving pulp. Their findings indicate that sterols, fatty alcohol, and saturated and unsaturated fatty acids significantly influenced/affected viscosity, kappa number, and carbohydrates in the pulp. Meanwhile, (Marklund et al. 1998) modeled the influence of fiber properties on strength parameters for softwood kraft made from 20 different wood samples by using multivariate data analysis and the partial least squares method. Recently, (Rodriguez-Alvarez et al. 2021) used a regression tree model and experimental designs to find optimal operating conditions in the papermaking machine's complex rodsizer and spooner section.

On the other hand, neural network models have demonstrated almost the same or more predictive capability than traditional techniques (Costela and Castro-Torres 2020, Mahmoud Ali et al. 2021, Mohammadi et al. 2021, Saha et al. 2021, Shams et al. 2021). However, this approach has not been widely investigated to determine quality characteristics in papermaking processes.

ABB and Honeywell (companies dedicated to the sale of technology solutions for the pulp and paper industry) contribute to solutions based on data analysis oriented to the optimization and control of processes in various industries like the pulp and paper industry. For example, ABB is a leader in the papermaking market by offering innovative products for paper quality measurements, ranging from in-line scanning to test laboratories (ABB 2021). Among its robust solutions is a quality management system through its Quality Control System (QCS), while Honeywell offers a similar product with its Quality Control System 4.0 (HONEYWELL 2021).

The QCS is reliable for monitoring variables for quality assurance purposes. It keeps the output variables (base weight, caliper, and moisture content mainly) under control since the values of the input variables (process) are being controlled at all times. However, one of the significant disadvantages of these control systems is the self-calibration process since, at time intervals defined by the supplier or process characteristics. The system has to go offline to perform the self-calibration and maintenance process. During this time, the system is offline, and many linear meters (depending on the speed of the machine) may not be within specification. Thus, leave the control in manual mode and loss visibility of the behavior of the output variables.

Currently, ABB offers a solution for monitoring basis weight by virtually measuring this variable of the paper. If the QCS is offline, their approach helps operators keep the basis weight properties with the target by creating an initial static conditional weight model using historical data and thus establishing an initial expectation of the accuracy of the calculated basis weight. If the accuracy of the initial model is acceptable, then the computed basis weight measurement is implemented online through their platform called ABB AbilityTM. However, this solution is intended for output variable monitoring purposes while the QCS is offline and is not intended for experimental purposes.

Machine learning techniques are widely used in many industrial applications (Morala et al. 2021) including experimental designs (Mezgár et al. 1997, Wong et al. 2018, Heinisch et al. 2021, Moreira et al. 2021). In addition, several authors have worked with interactive user tools called soft sensors (Chang and Li 2021, Kamyar et al. 2021, Niño-Adan et al. 2021, Zeng and Ge 2021), which are employed as tools for visualization of output data from the developed models. However, there is no evidence of the development of such interactive tools to determine basis weight in papermaking processes.

Since there are only two approaches to determining basis weight in papermaking processes, this research presents a novelty model based on data science techniques to estimate basis weight as an alternative to the current methods. In addition, the model will be embedded in a user interface so that the user will interact by entering input variables values (process variables) to calculate with reasonable accuracy the basis weight value (output variable). Hence, the proposed method is an excellent alternative to the existing ones. Furthermore, the model could be used to perform experimental designs to find optimal operating conditions.

This paper is organized as follows: the next section presents a brief review of neural network architectures, including the common hyperparameters used to develop a model based on the neural networks approach. Following section presents the methodology, which corresponds to the phases of a typical data science project. Finally, the last two sections include the results and conclusions.

Neural networks architectures: A brief review

For a few decades, artificial neural networks have constituted one of the essential computational intelligence tools used in a wide variety of problems (Haykin 2004).

Two of the essential criteria in building up a neural system model are network architecture and parameter selection. The interconnections between the neurons in an artificial neural network define its architecture. The neural systems are intense nonlinear signal processors; however, the results are regularly a long way from satisfactory (Rosli et al. 2016). So, the number and type of neuron connectivity and the activation function are essential parameters, and their selection is a determinant of producing a good network for any case study (Karamichailidou et al. 2021).

Several neural systems architectures are available in the literature (Dayhoff 1990, Karayiannis and Venetsanopoulos 1992, Fausett 2006, Teuscher 2012). In the present work, we describe in detail the two most common architectures: the multilayer perceptron neural network (MLP-NN) and radial basis function neural network (RBF-NN), and, briefly, the generalized regression neural network (GR-NN) and Elman neural network (Elman-NN).

Multilayer perceptron neural network

One of the most well-known architecture researchers has used the multilayer perceptron (MLP). This architecture includes the input layer, hidden layer(s), and output layer. The architecture is shown in Figure 1.

This approach is essentially a combination of neurons, biases assigned to neurons, interconnections among them, and weights assigned to these interconnections. The learning process is performed according to input and target data sets and training algorithms (Hashemi Fath et al. 2020).

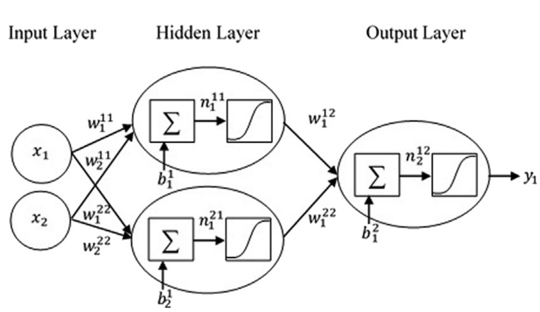

Figure 1: The architecture of MLP-NN (taken from Rosli et al. 2016).

Mathematically, a neuron K can be defined via the following equations:

$$y_k = f\,(\mu_k + b_k) \tag{1}$$

$$\mu_k = \sum_{i=1}^{N} w_{ki} x_i \tag{2}$$

where $x_1,\ x_2,\ x_3,\ \ldots,\ x_n$ denote the input signals, are the connection weights of the neuron, μ_k is the linear output of the linear combination among weighted inputs, b_k is the bias term, is the activation function, and y_k is the output signal of the neuron.

The multilayer perceptron is trained based on the backpropagation algorithm, which follows a learning procedure based on the error-correction rule. By comparing the target values and the network's output, the error value is calculated. Afterward, the weights and biases are adjusted to minimize the error, and the training process continues until the network reaches a predefined minimum allowable error. The error function typically used is the mean square error (*MSE*) (Hashemi Fath et al. 2020). Still, in the presence of many outliers in the training set, it is recommended to use the mean absolute error (*MAE*) (Géron 2019).

Radial basis function neural network

The radial basis function is widely used by many researchers in various sciences, mainly as function approximation and pattern classification (Moody and Darken 1989, Poechmuelloer et al. 1994, Nabney 1999, Fu and Wang 2003). This network is accessible to train, design, and robust tolerance to input noise (Hashemi Fath et al. 2020, Rosli et al. 2016).

This neural network architecture comprises an input layer, a hidden layer, and an output layer. The input layer serves only as an input distributor to the hidden layer. The hidden layer contains radial basis functions, and the output layer generates the network output by linearly combining the outcomes of the hidden neurons (Hashemi Fath et al. 2020). The architecture is shown in Figure 2.

The formulation of a radial basis function neural network is as follows:

$$y_i(x) = \sum_{i=1}^{k} w_{ij} \emptyset(\|x - C_j\|) \tag{3}$$

This parameter is a multidimensional radial basis function describing the difference between an input vector and a predefined center vector, where x is the input vector, y_i is the network's ith output, K is the number of neurons

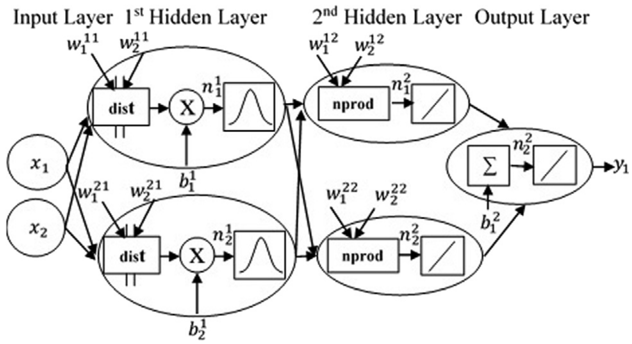

Figure 2: The architecture of RBF-NN (taken from Rosli et al. 2016).

in the hidden layer, C_j denotes the center of the jth hidden neuron, w_{ij} represents the weight of the link from the jth neuron in the hidden layer to the ith neuron in the output layer, and $\|.\|$ is the Euclidian norm. $\emptyset$ is the radial basis function used in the neurons of the hidden layer. The most common applications in the literature refer to the Gaussian Function, which is defined as follows:

$$\emptyset\left(\left\|x - C_j\right\|\right) = e^{\left(\frac{\left\|x - C_j\right\|^2}{2\sigma_j^2}\right)} \tag{4}$$

where σ_i is the width of the *jht* hidden neuron, finding the centers, widths, and the weights connecting hidden neurons to the output is the key to constructing and training the radial basis function neural network.

Generalized regression and Elman networks

Other types of neural network architectures are generalized regression and Elman neural networks. The generalized neural network has seldom been employed for addressing nonlinear process monitoring issues (Lan et al. 2020). However, due to its strong nonlinear mapping capability, simplicity of the structure, and high robustness, the generalized regression neural network has been demonstrated to be a powerful tool for nonlinear supervised learning (Baruník and Křehlík 2016). Furthermore, it can be trained to estimate the behavior of complex systems with a non-parametric technique (Antanasijević et al. 2015). This approach is treated as normalized (Konate et al. 2015) and belongs to radial basis function neural networks. Still, it can quickly perform fast learning and coverage to the optimal regression surface (Specht 1991), even when the number of training samples is limited (Amiri et al. 2010).

The generalized regression neural network architecture consists of four layers: the input, hidden, summation, and output layers, respectively. The process from the originating neurons is multiplied by their weights at each hidden neuron (Rosli et al. 2016). The weights are added with a bias to increment or decrement the input into the activation function defined (Rooki 2016).

Finally, the Elman neural network proposed by (Elman 1990) is a kind of feedforward neural network which is especially suitable for time series prediction. This approach better predicts performance because it has a load-bearing layer that other neural networks do not have. One of its main advantages is this approach can be regarded as a recurrent neural network with a local memory unit and local feedback connection (Zhao et al. 2020).

The Elman neural network architecture consists of two layers. A feedback connection is formed by feeding the output of the hidden layer back to the input layer, referred to as the context layer. A feedback loop with a single delay stores the information and retains the memory (Sundaram et al. 2016). Its structure is made simple, and the input parameters are minimized, thereby shortening the training time (Sun et al. 2015).

Neural networks hyperparameters

As it is well-known, multilayer perceptron can be used for regression tasks. If it is necessary to predict a single value, then need a single output neuron; thus, the output is the predicted value. When building an MLP for regression, it is unnecessary to use an activation function for the output neurons, so they are free to output any range of values. However, if it is required to guarantee that the output will always be positive, then the rectified linear unit activation function can be used (Géron 2019).

There are many hyperparameters when a neural network is designed. For instance, in a simple MLP, the number of layers, the number of neurons per layer, the type of activation function to use in each layer, the weight initialization logic, and much more can change. Hyperparameter tuning is still an active area of research. Recently, (Jaderberg et al. 2017) have worked on this topic.

For the number of hidden layers, a common practice is selecting a single hidden layer for many problems. It has been shown that an MLP with just one hidden layer can model even the most complex functions, provided it has enough neurons. These facts convinced researchers that there was no need to investigate deeper neural networks for a long time. However, deep networks have a much higher parameter efficiency. It is better to start with just one or two hidden layers for some problems, and it will work just fine (Géron 2019).

The input type determines the number of neurons per hidden layer and output required. A common practice is to form a pyramid, with fewer and fewer neurons at each layer, the rationale being that many low-level features can coalesce into far fewer high-level features. As for the number of layers, users can increase the number of neurons until the network starts overfitting. The user will get more bang for the buck by increasing the number of layers than the number of neurons per layer. Unfortunately, the perfect number of neurons is still somehow a dark art.

The learning rate is arguably one of the most critical hyperparameters. The optimal learning rate is about half of the maximum learning rate. So a simple approach for tuning the learning rate is to start with a large value that makes the training algorithm diverge, divide this value by three, try again, and repeat until the training algorithm stops diverging.

Training a vast deep neural network can be painfully slow. So, choosing a better optimizer is also vital to deal with this possible problem. Four ways to speed up training are common:

- Applying a good initialization strategy for the connection weights.
- Using a good activation function.
- Batch normalization.
- Reusing parts of a pre-trained network.

However, another considerable speed boost comes from using a faster optimizer. The most used in practice are momentum optimization, Nesterov accelerated gradient, AdaGrad, RMSProp, and Adam and Nadam optimization (Géron 2019).

Another hyperparameter is the batch size, which significantly affects the model's performance and training time. In general, the optimal batch size will be lower than 32. A small batch size ensures that each training iteration is very fast, and although a large batch size will give a more precise estimate of the gradients. A common practice is the use of a batch size greater than 10.

An issue related to training iterations does not need to be tweaked: use early stopping instead.

Finally, it has been well recognized that the type of activation functions plays a crucial role in the multi-stability analysis of the neural network. Different activation functions might lead to other equilibrium points and different dynamical behaviors of neural networks (Nie et al. 2019). The common activation functions available are relu (rectified linear unit), sigmoid, softmax, softplus, softsign, tanh (hyperbolic tangent), selu (scaled exponential linear unit), elu (exponential linear unit), and exponential.

Many practical recommendations for deep networks are presented by (Bengio 2012).

Methodology

For developing the interactive soft sensor, the present study will follow a data science project's typical process: framing the problem, collecting raw data, processing the data, exploring the data, performing in-depth analysis, and communicating the results.

Problem statement

Beyond the methods used by the quality control laboratory and the quality control system (QCS) measurements available on the market, there are no alternatives to determine the basis weight of the paper in the papermaking processes. So, in the present research, we intend to develop a novelty model based on data science techniques to determine the basis weight in papermaking processes. In addition, a graphical user interface will be designed so that the user can interact by entering values of the input variables (process variables) and calculate with acceptable precision the value of the basis weight (output variable).

Data collection

The most common output variables monitored in papermaking machines are the basis weight, caliper, and moisture content. These variables are automatically monitored through a scanner. Basis weight and moisture content are included in the database (Raunio and Ritala 2018).

The database was collected on a 6-meter wide papermaking machine which contains three forming tables (top, middle, and back). This papermaking machine works all year round, shutting down only when there is scheduled maintenance or a process problem. The dataset includes the independent variables monitored in the process, maintaining under control and specification the basis weight variable.

The complete dataset is a matrix of 342,501 rows by 25 columns. This dataset was extracted from the server that stores historical data covering a period from 01 January to 30 August, 2019. This period includes all paper grades (paper basis weight) that are manufactured in the process. The dataset was the readings taken by sensors at 1-minute time intervals in this period. The readings correspond to the main factors affecting the paper's basis weight (dependent variable, y): flow rates, consistencies, storage tank levels, output pressures, lip angle, and machine speed (dependent variables, x's).

Data processing

A first data cleaning operation was performed in Excel to eliminate all those data and variables that do not affect the basis weight of the paper. From the original dataset, a total of eighteen columns were removed.

As mentioned above, in this study, twenty-five were defined. One variable corresponds to the basis weight (dependent variable). The remaining twenty-four are distributed in six categories: flows, consistencies, pressures, levels, machine speeds, and lip position, corresponding to the independent variables. These categories are highlighted in bold for each independent variable in Table 1.

A second data cleaning operation was carried out with Excel to remove all data readings with no logic included. First, all negative and zeros values were removed because the process studied does not produce thin paper grades (basis weight). Only readings equal to or greater than a basis weight of 100 grams (dependent variable) are included. Finally, because basis weight is determined mainly by machine speed, only data readings obtained equal to or greater than 50 meters per minute were included.

Table 1: Parameters of each independent variable.

Variable Name	Mean	SD	Min	Max
Pulp *Flow* at Machine Top	2268.4	683.9	100.1	4012.3
Consistency at Machine Top	4.1	0.5	2.0	5.1
Consistency at Tank Top	4.0	0.7	2.3	5.1
Output *Pressure* at Machine Top	2.2	0.2	0.0	3.0
Level Top	97.0	2.5	24.6	101.3
Pulp *Flow* at Machine Middle	5281.0	783.0	143.7	7252.4
Consistency at Machine Middle	3.8	0.3	2.0	5.1
Consistency at Tank Middle	3.6	0.3	2.4	4.6
Output *Pressure* at Machine Middle	2.2	0.0	0.5	2.4
Level Middle	91.7	1.7	34.8	94.8
Pulp *Flow* at Machine Back	2611.1	314.0	105.3	3610.2
Consistency at Machine Back	2.9	0.6	1.8	4.6
Consistency at Tank Back	3.8	0.2	1.8	4.6
Output *Pressure* at Machine Back	1.6	0.3	0.1	3.1
Level Back	93.4	1.2	27.3	96.2
Machine *Speed* Top	406.9	101.2	135.2	590.5
Machine *Speed* Middle	407.0	101.2	59.5	590.9
Machine *Speed* Back	406.3	101.1	136.8	589.7
Horizontal *Lip Position* Top	27.7	4.1	9.6	36.2
Vertical *Lip Position* Top	78.4	2.8	72.2	83.0
Horizontal *Lip Position* Middle	28.2	3.5	18.7	35.2
Vertical *Lip Position* Middle	20.7	3.6	14.3	28.8
Horizontal *Lip Position* Back	28.7	3.3	17.0	36.8
Vertical *Lip Position* Back	57.9	3.3	50.9	63.2

Explore the data

In the presence of null data, these will be imputed with the average value of each column (variable) that has shown at least one null data (rows).

As an ideal tool for machine learning developments, the TensorFlow library (included in Python programming language) will be checked to identify a model capable of determining the basis weight of the paper with reasonable accuracy.

Perform in-depth analysis

In this research, the artificial neural network (ann) approach will be used to identify the model with good accuracy (at least an R^2 of 90 %). At a minimum, the ann network architecture comprises hidden layers, neurons, and activation functions (Rosli et al. 2016). The multilayer perceptron (MLP) system was a typical, well-known network architecture. MLP is one of most analysts' favored neural system topologies of most analysts (Sharma et al. 2015). These networks are essentially a combination of neurons, biases assigned to neurons, interconnections or links among them, and weights assigned to these interconnections. The learning process is performed according to input and target data sets and training algorithms (Hashemi Fath et al. 2020).

Since the multilayer perceptron (MLP) is simpler to design and faster to train, the knowledge is already spread well throughout different scientific communities (Canário et al. 2020). In addition, its straightforwardness and capability to predict precisely for a regression scenario; therefore, in the present research, the multilayer perceptron was selected as a baseline over other advanced strategies.

Firstly, the multilayer perceptron will be developed by following the sequential model. This model is one of the most straightforward neural network models composed of a single stack of layers connected sequentially (Géron 2019). Each layer will be of a Dense type; they will be fully connected layers. A typical architecture of a regression MLP to be used in this research is summarized in Table 2 (Géron 2019).

Therefore, in the present research, the trial and error strategy follows any path according to the hyperparame-

Table 2: Typical Regression MLP Architecture.

Hyperparameter	Typical Value
# input neurons	One per input feature.
# hidden layers	Depends on the problem. Typically 1 to 5.
# neurons per hidden layer	Depends on the problem. Typically 10 to 100.
# output neurons	1 per prediction dimension
Hidden activation	ReLU (o SeLU).
Output activation	None or ReLU/Softplus (if positive outputs) or Logistic/Tanh (if bounded outputs).
Loss function	MSE or MAE

ters and typical values summarized in Table 2. This process is applied until reach the desired results.

Gather the number of neurons per hidden layer; a common practice is to size them by forming a pyramid. However, a rhombus shape is proposed in the present work. Thus, the input data shape is a 24-dimensional vector with a 48-dimensional output vector. The next layer has a 12-dimensional output vector, and finally, the 1-dimensional output vector corresponds to the response value. The rectified linear unit activation function (relu) will be used for each layer to guarantee that the output will always be positive.

The RMSProp will be used as an optimizer learning rate. The main gist of this algorithm is to maintain a moving (discounted) average of the square of gradients and divide the gradient by the root of this average (Hinton et al. 2012). The mean absolute error (*MAE*) and mean absolute percentage error (*MAPE*) is considered the most proper for the loss and metric functions. In addition, the mean absolute error (*MAE*), the root mean square error (*RMSE*), and the explained variance score (R^2) would be calculated to evaluate performance in the building process of the model. A batch size of 32 will be used in the present work.

Related to the training iterations (epochs) and trying to find an inflection point, 1000 epochs will be used for the training process. Finally, to validate the method performance, a large dataset not included in the building process of the proposed model will be used.

The neural network model for determining the basis weight in papermaking processes and the performance evaluation of the model will be carried out in Python 3.8.5 programming language by using the TensorFlow 2.0 library.

The Matlab regression learner app will be used to explore alternative models for predicting basis weight with acceptable accuracy. The same dataset used in the building process of the artificial neural network model will be used in the building process of the alternative model. The used criteria to select the best model will be based on mean absolute error (*MAE*), mean square error (*MSE*), and the root mean square error (*RMSE*).

Communicate results

Since the neural network model developed is intended to be an alternative tool to determine the basis weight in papermaking processes. Therefore, a graphic user interface (GUI) was designed and developed in Python 3.8.5 using the streamlit app to enter the input variables data. Then, the user can know the basis weight for any paper grade with reasonable accuracy.

An overall scheme of the proposed methodology to develop an interactive soft sensor based on an artificial neural network model is shown in Figure 3.

Results

Using the collected data as described above, this section presents the model and graphic user interface development results. Until now, there is no evidence of models that can be used as alternatives to determine the basis weight of the paper in the papermaking processes.

Model development

The model development follows the typical data science process. The dataset collected included all paper grades, so the period was adequate to develop a robust model. After the cleaning process was applied to the dataset, the size of the resulting matrix was 215,103 rows by 25 columns. These twenty-five columns include the twenty-four independent variables corresponding to each physical input feature that must be monitored. In addition, one column corresponds to the physical output feature. In the actual papermaking process, all the physical input features presented here must be monitored by engineers to control the basis weight of the paper at the end of the process. Table 1 shows each independent variable's mean, standard deviation, minimum, and maximum values. However, the set point for each variable changes as a function of the paper grade to be manufactured. The typical paper grades manufactured go from 180 to 320 grams/m^2. In addition, other weighted paper grades as 440 grams/m^2 can be manufactured.

Fifteen percent of the dataset was extracted to perform different analyses with the proposed model, so these data

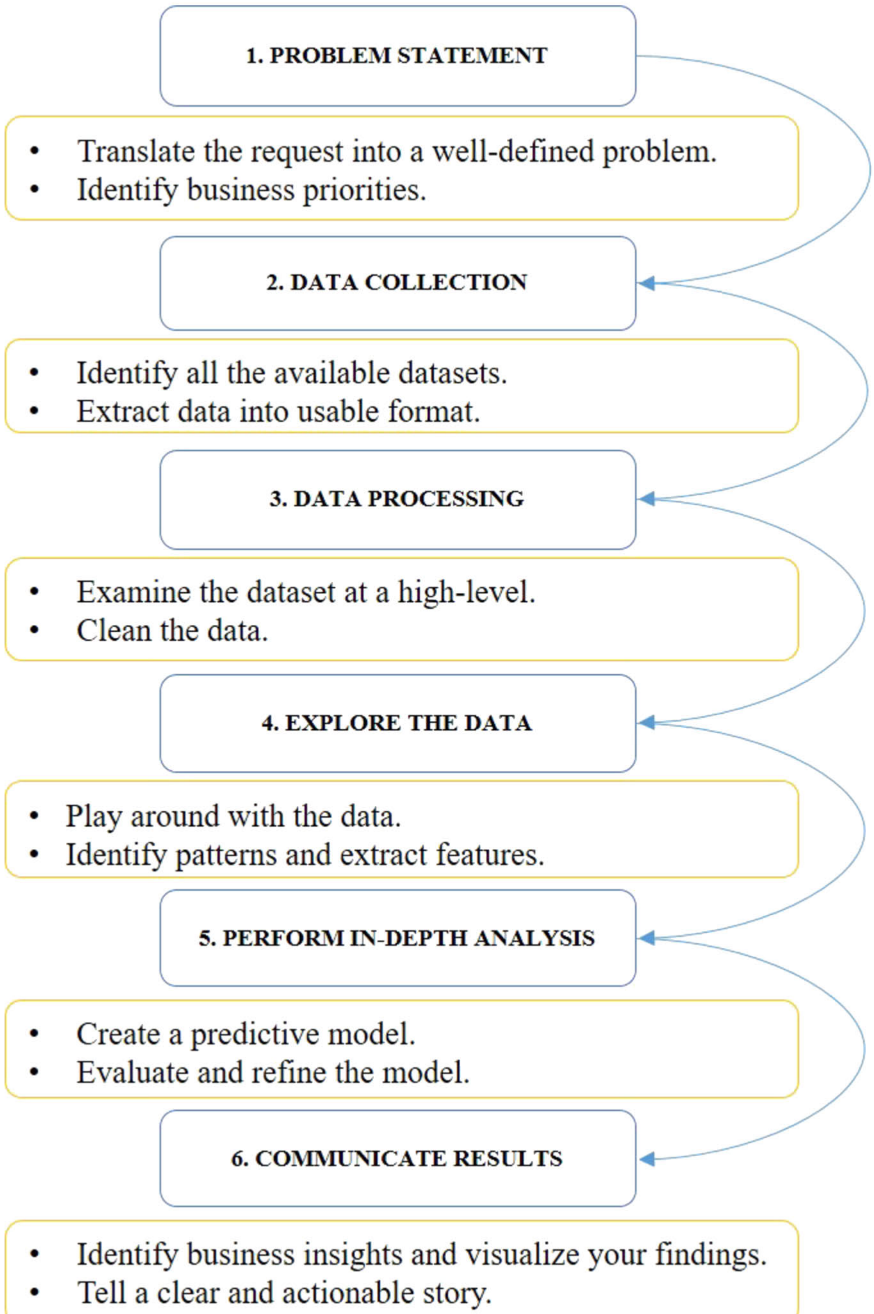

Figure 3: Proposed methodology to develop an interactive soft sensor based on an artificial neural network model.

were not included in the training, validation, and testing process. Thus, the size of the resulting array was 182,834 rows by 25 columns.

The independent variables that showed the presence of null data were: consistency of output at machine head top, consistency of output in mixing tank top, consistency of output at machine head middle, consistency of output in mixing tank middle, consistency of output at machine head back, and consistency of output in mixing tank back. These null data were imputed with the following average values: 4.12, 4.00, 3.84, 3.60, 2.92, and 3.80.

After the cleaning process, the whole dataset was segregated into two arrays. An input array size of 164,550 rows by 25 columns was used in the training and validation process, while an input array size of 18,284 rows by 25 columns was used in the test process.

Although all the parameters are defined in the training process addition, it is essential to pass a validation pro-

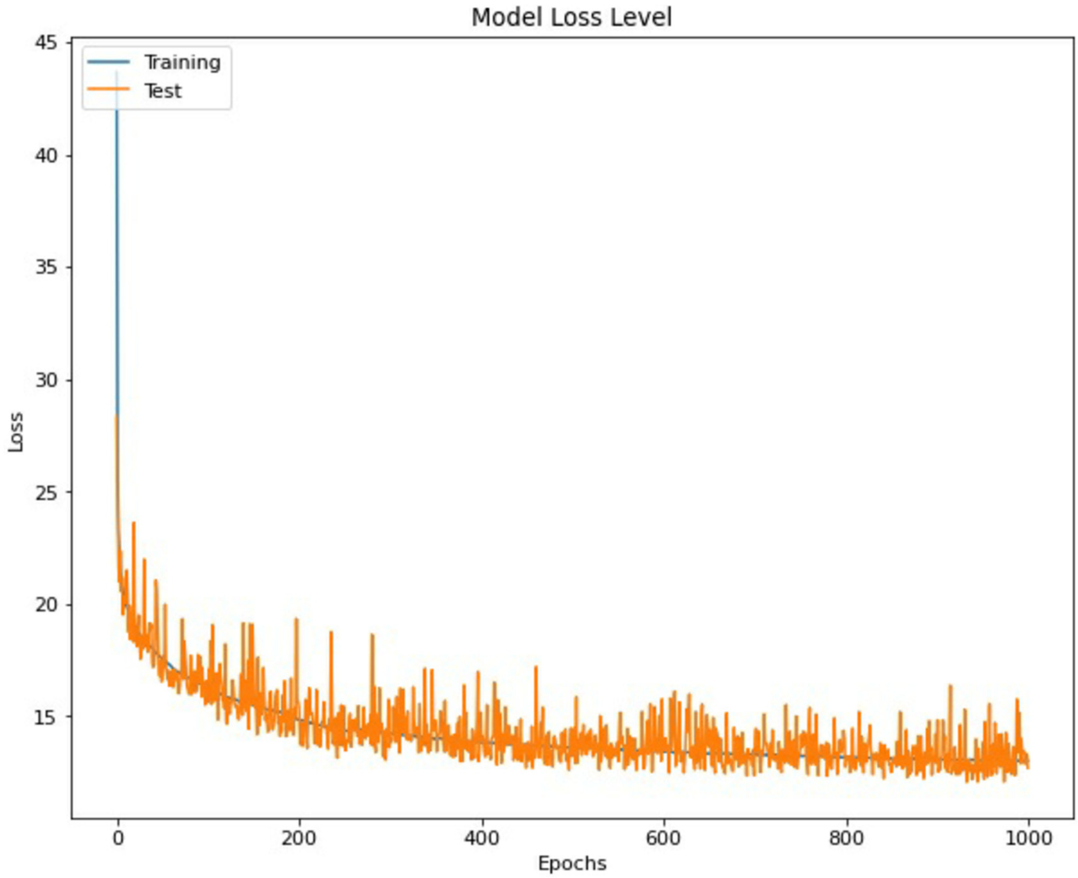

Figure 4: Training summary for the model loss level.

cess. For this purpose, fifteen percent was used from the input array size of 164,550. Measuring the loss function on this set at the end of each epoch helps see how well the model performs. When the performance on the training set is better than on the validation set, the model is probably overfitting the training set. If you can see that the loss function went down after 50 epochs (Géron 2019), this is a good sign. This research does not seem to be much overfitting, as shown in Figure 4.

However, suppose the results are not satisfied with the expected performance of the model. In that case, it is necessary to go back and tune the model's hyperparameters related to the number of layers, the number of neurons per layer, the types of used activation functions for each hidden layer, the number of training epochs, and the batch size by following alternative paths according to Table 2. Since the interest in this research is to find an accuracy model capable of explaining at least 90 % of the variability; therefore, this criterion was used to define the hyperparameters combinations in the training and validation process.

When the training and validation process has reached the desired accuracy, it is necessary to evaluate a test set to estimate the generalization error before any model deployment to production. Notice that it is common to get slightly lower performance on the test set than on the validation set because the hyperparameters are tuned on the training and validation set, not for the test set.

The best-found architecture and structure of the neural network are shown in Table 3. By following the above-described method, the activation functions and the number of neurons per layer were moved by trial and error for the neural network architecture design. Notice that this method could be tedious.

Using the loss and metric functions described above, a learning rate of 0.001 for the RMSprop optimizer, a batch size of 32, and 1,000 epochs, the resulting mean absolute error (*MAE*) were 12.40 grams/m^2. Meanwhile, the explained variance score (R^2) was 94 %. Figure 4 shows a training summary for the model loss level. Meanwhile, Figure 5 is shown the predicted vs. test data to evaluate the model performance in the building process of the model. Figure 6 shows in detail the predicted vs. test data for a ran-

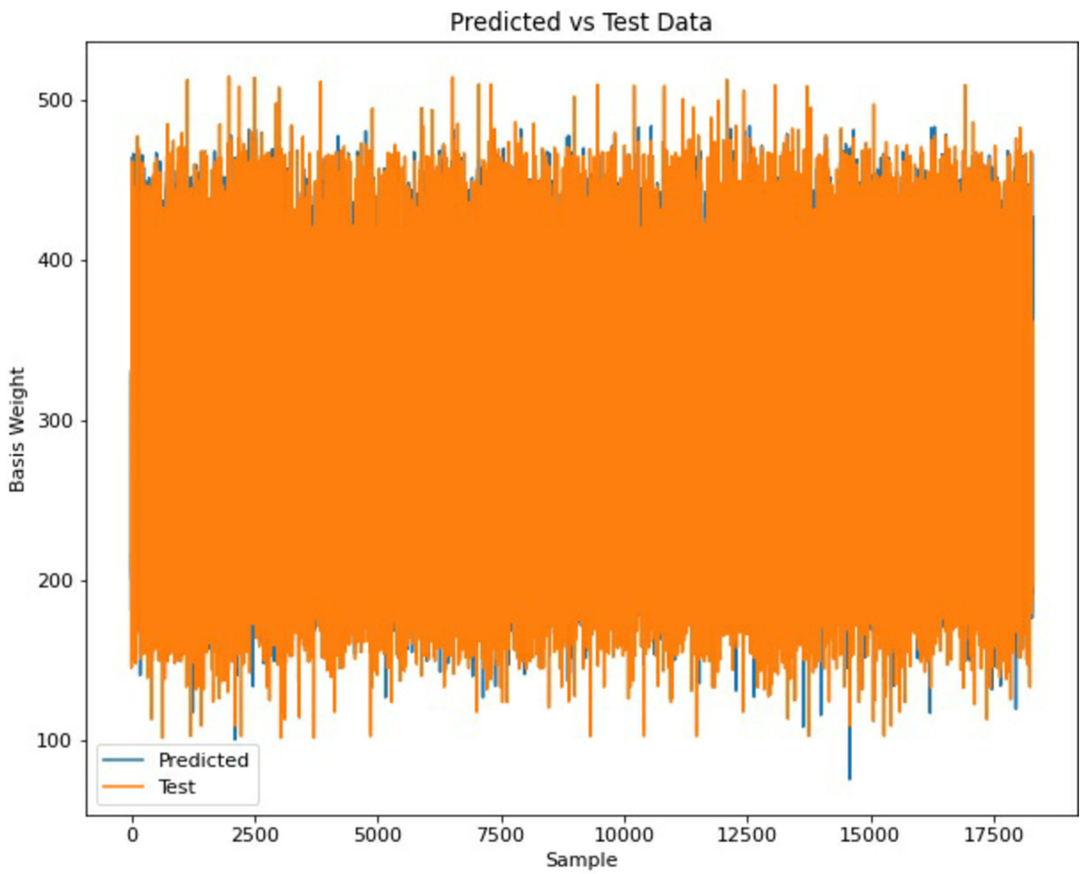

Figure 5: Model performance: predicted vs. test data.

Table 3: Neural Network Model Architecture.

Model: "sequential"		
Layer (type)	**Output Shape**	**Param #**
dense (Dense)	(None, 48)	1,200
dense_1 (Dense)	(None, 12)	588
dense_2 (Dense)	(None, 1)	13
Total params: 1,801		
Trainable params: 1,801		
Non-trainable params: 0		

dom range selected from 12000 to 12200. The results show that the model developed can determine the basis weight as an alternative to the quality control laboratory and the measurements made by the quality control system (QCS) available on the market.

Notice that the modeling process includes all independent variables monitored by the engineers in operation to control the dependent variable (basis weight). Because the primary purpose of this research is to offer an alternative method to estimate basis weight by using the soft sensor, any feature selection was not included.

The same input array size of 164,550 rows by 25 columns was used in the training and validation process to develop the best regression model, while an input array size of 18,284 rows by 25 columns was used in the test process. After the training and validation process, the best model was a fine tree. The resulting mean absolute error (*MAE*) was 3.35 grams/m^2. Meanwhile, the explained variance score (R^2) was 99 %. Figure 7 confirms that the fine tree model can predict the basis weight very well.

Finally, an external dataset not included in the building process of the model was used to validate the method's performance. Figure 8 shows the artificial neural networks model performance by comparing the predicted values against the real dataset selected for this purpose. The dataset used here corresponds to the paper grades from 180 to 250 grams/m^2. These are the most common manufactured paper grades. The input array size was 7,768 rows by 24 columns, while the output array size was 7,768 rows by 1 column. The resulting mean absolute error (*MAE*) was 12.10 grams/m^2, the mean square error (*MSE*) was 326.46, and the root mean square error (*RMSE*) was 18.07.

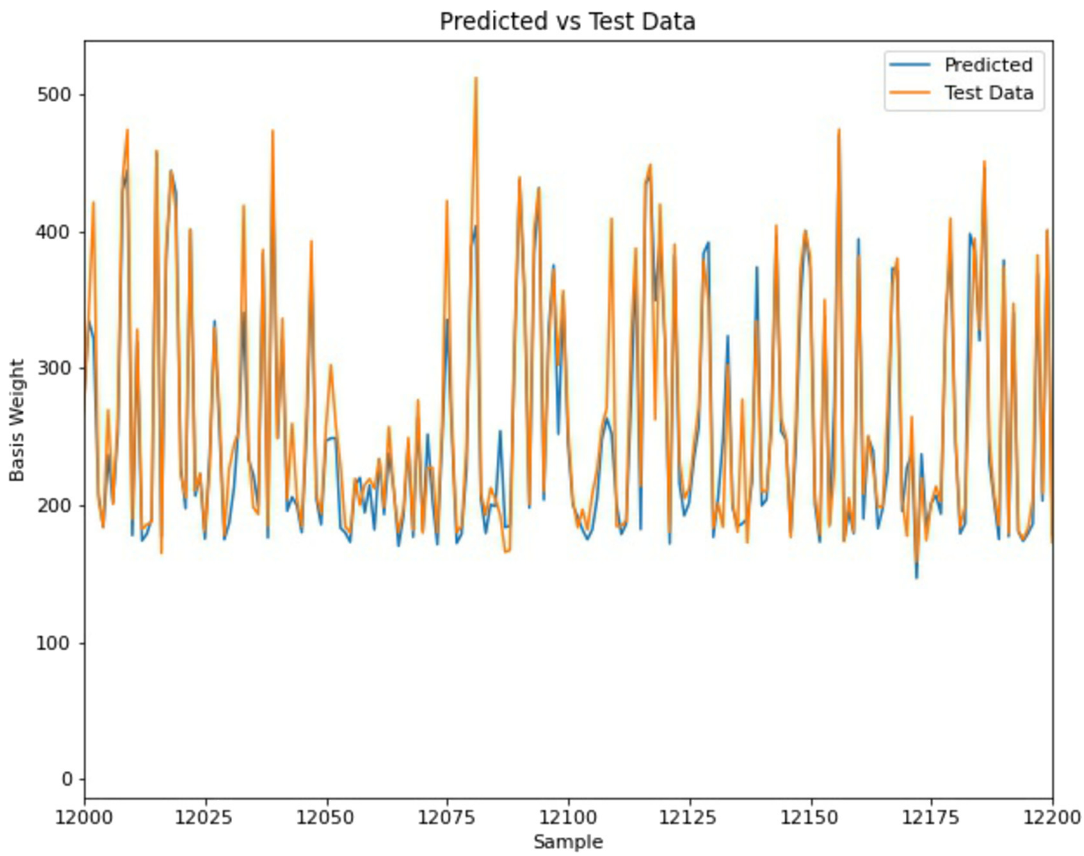

Figure 6: In detail the predicted vs. test data for a random range selected from 12000 to 12200.

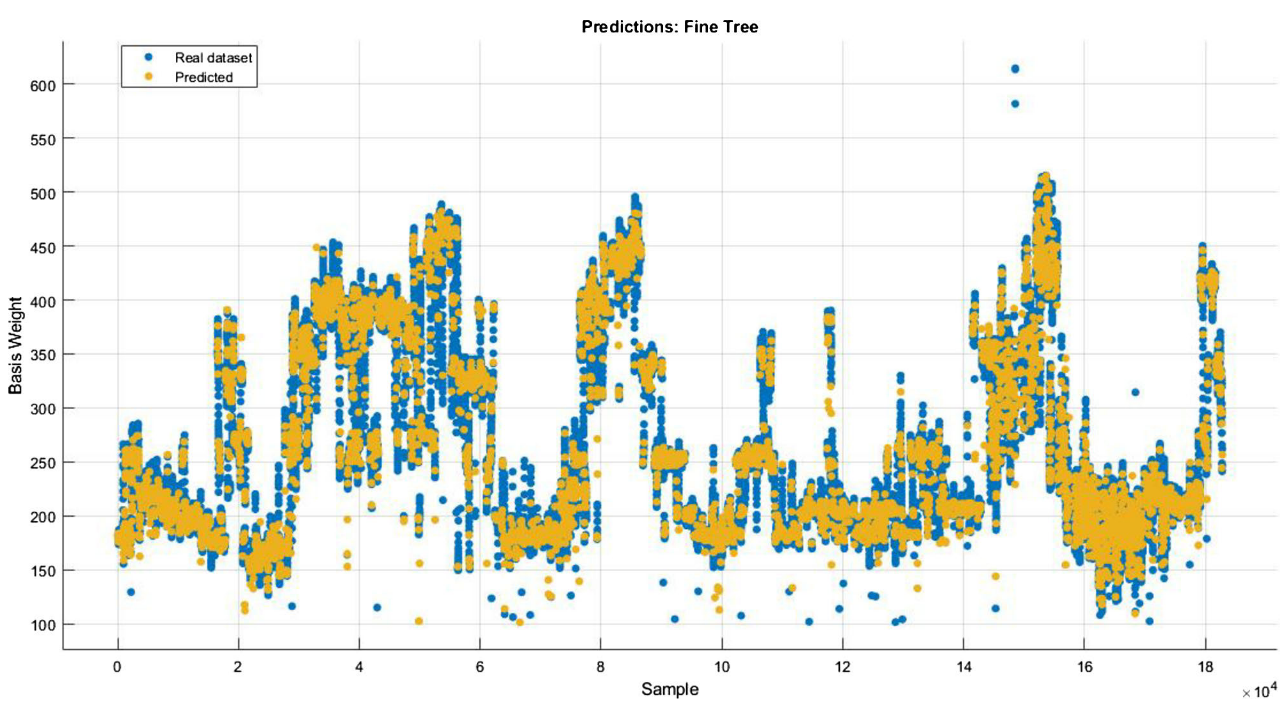

Figure 7: Fine Tree model performance: predicted vs. real dataset.

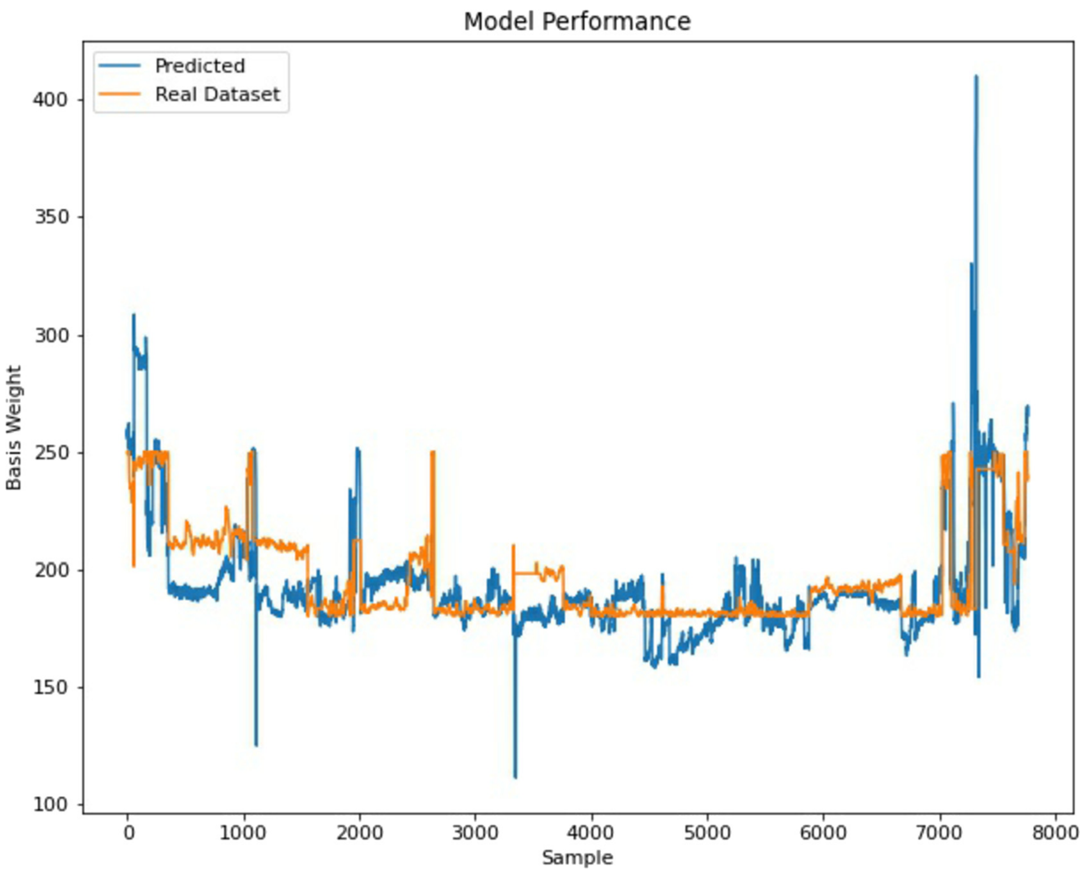

Figure 8: Artificial Neural Network model performance with an external dataset.

The same external dataset not included in the building process of the model was used to validate the fine tree performance. Figure 9 is shown the fine tree model performance by comparing the predicted values against the real dataset selected for this purpose. The resulting mean absolute error (*MAE*) was 36.47 grams/m^2, the mean square error (*MSE*) was 2309.8, and the root mean square error (*RMSE*) was 48.06.

Graphic user interface

A user interface was developed in Python programming language by using the Streamlit app. Firstly, the layout for input and output variables was designed, including the text labels. The user must specify the input values for each machine section and the paper grade and quality variable. The basis weight is displayed when these parameters are introduced, as shown in Figure 10.

Since the ANN-based model demonstrated an excellent predictive capability; therefore, the model was embedded in a GUI to be assessed in situ. For this purpose, the code of the GUI (name.py) and the file containing the trained model (name.hdf5) can be obtained upon request.

Note that the file model was developed in Python; therefore, this open-source software can be used to know the model code in detail and, if necessary, retrain the model with a new dataset or tune its hyperparameters. Additionally, feedback regarding the model performance may be provided upon request.

The generated package can be saved in any local host and then run to be used in-situ. In addition, the GUI can work in a web environment. Therefore, to run the GUI and see the app, as shown in Figure 10, follow these steps:
– Install Anaconda (free and open distribution).
– Open Powershell from Anaconda.
– Save the previously requested files on your computer.
– Copy the path of the file .py
– In Powershell, type the “cd” instruction before pasting the copied path.
– Run the app by typing the following instruction: “streamlit run name.py”

Finally, although this GUI presented as a soft sensor can estimate the basis weight reasonably, general assumptions

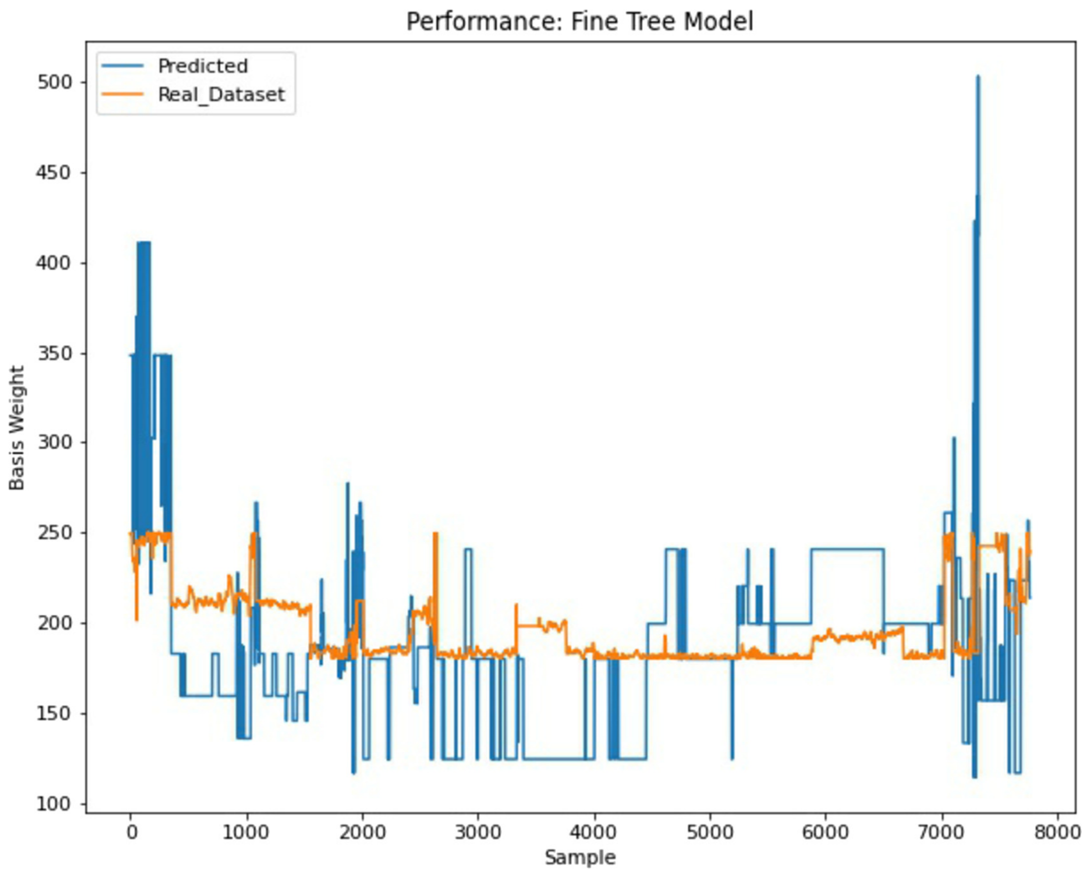

Figure 9: Fine Tree model performance with an external dataset.

must be verified. As described by (Fortuna et al. 2007) the soft sensors exploit the essential information behind the data to build models with excellent performance and robustness; however, this technology could be susceptible to measurement drift from long-term usage, challenging practical usability (Kim et al. 2020). Therefore, although the based-neural network model deployment presented here showed accuracy and reliability using data not included in the building process, the following considerations described in (Vinoth et al. 2022) will be required in advance.

- If drift is observed from a long-term usage, necessary compensation for the network o response variable must be done.
- The performance of the neural network model can be assessed by conducting periodical experimental trials with known data. Results must be compared with the earlier performance.
- The neural network model shall be trained with relevant additional process data, which can improve the generalization, robustness, precision, and accuracy of predictions.
- Since the neural network model performance relies heavily on the historian data for specific paper grades. If there is a significant change in the paper included during the design, rebuilding the neural network model will be recommended.

Discussion

A soft sensor is defined as the association of a hardware sensor enabling the online measurement of some process variables employing an algorithm to estimate unmeasured variables (De Assis and Maciel Filho 2000, Paggi et al. 2022). Likewise, (Napoli and Xibilia 2011) define soft sensors as systems composed of mathematical algorithms that produce reliable real-time estimates of unmeasured variables using correlation with available data. This definition describes how the soft sensor presented here was developed.

The main disadvantages of the soft sensor are the margin of error concerning either measurement for other sensors and the that the measure comes from predictions

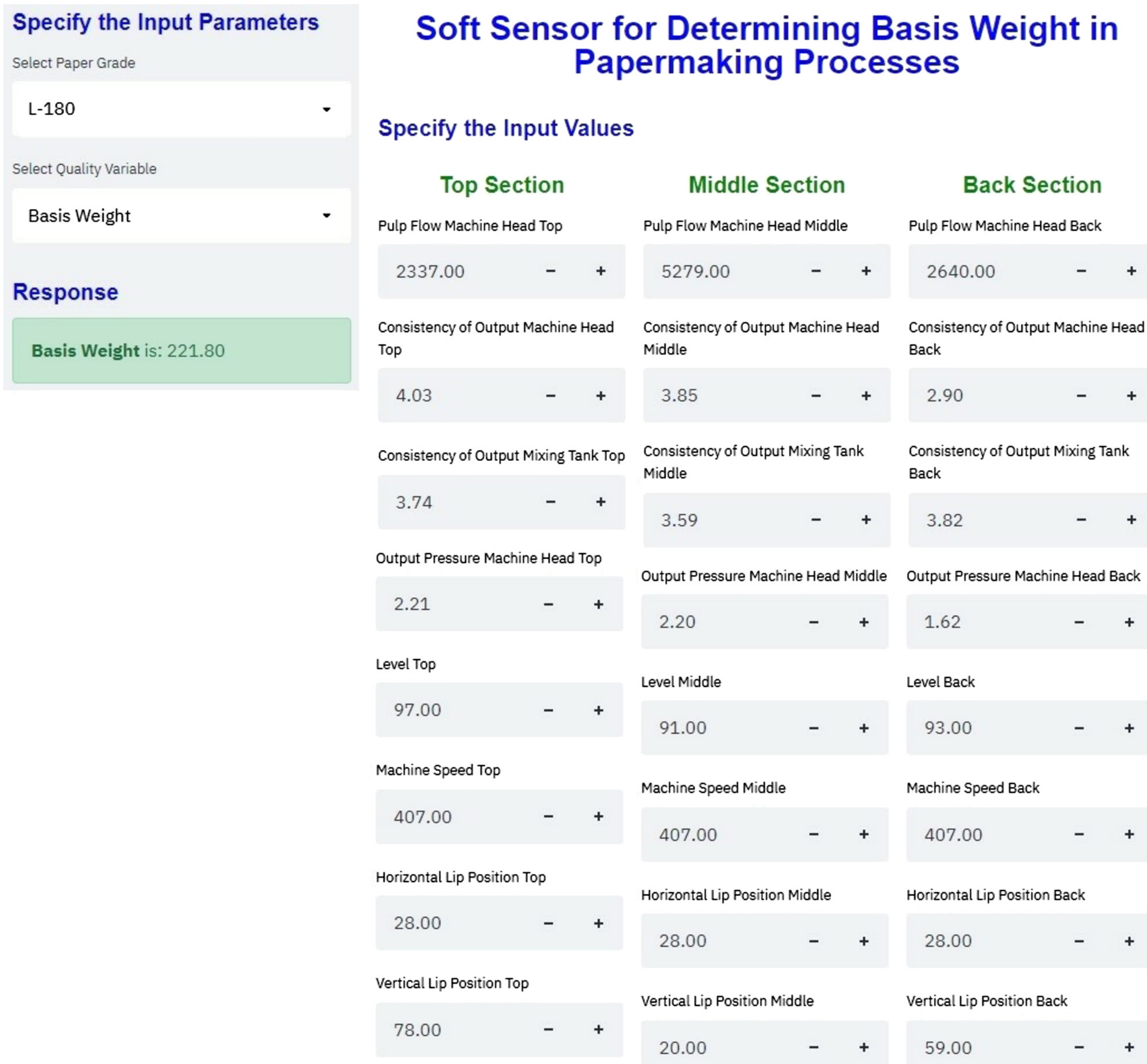

Figure 10: Graphic user interface.

(Gadeo-Martos et al. 2011). Although the commercial quality control system (QCS) could present a little margin of error, the results showed by the soft sensor presented here are significantly closer to QCS; therefore, the predicted measurements can be acceptable for almost basis weight.

Unlike the existing methods (instruments) to measure basis weight, this proposal does not use the final product or service measures. Instead, it uses a dataset coming directly from each independent variable; therefore, the predicted measurements made by a trained model with a reasonable precision embedded in a graphical user interface could represent an advantage. The reason is that the measures include the variability shown in each independent variable that affects the response. A great number of applied studies can be found in the literature, where the dataset in the analysis comes from measurements taken directly on the product or service.

Finally, the models based on neural networks have demonstrated better predictions than other regression techniques, mainly for data not included in the training and validation process. The soft sensor presented in this research had a lower *MAE* value than the regression tree model obtained. They demonstrated the potential of neural networks for prediction purposes.

Conclusions

The basis weight of paper is a critical quality characteristic that must be monitored and controlled in papermaking. Currently, in the papermaking process, the basis weight and other important quality characteristics are monitored using traditional control charts (Rodríguez-Álvarez et al. 2021) due to the methods used by the quality control laboratory. In addition, the commercial-quality control system (QCS) available on the market is the other alternative to monitor basis weight.

The present research proposed a new alternative to measuring basis weight for any papermaking process. The proposed model is advantageous since its development is based on an artificial neural network using the multilayer perceptron approach. Therefore, it can predict the basis weight with reasonable accuracy (greater than 90 %) for new operating conditions or data not included in the training and validation process. For this purpose, the performance of the proposed method was validated by using an external dataset not included in the building process of the model. The findings showed a mean absolute error of 12.10 grams/m^2. So, the model can predict the basis weight of paper, mainly for the grades from 180 to 250 grams/m^2, reaching an error from 4.8 to 6.7 %. On the other hand, although the fine tree model showed a mean absolute error of 3.35 grams/m^2 in the training and validation process when this model was validated using the external dataset, the results showed a mean absolute error of 36.47 grams/m^2. Therefore, it is demonstrated that artificial neural network-based modeling has a higher predictive capability than regression-based modeling for data not included in the training and validation process.

One of the most significant findings from this study is that model was embedded in a user interface. The generated package can be saved in any local host and then run to use the user interface. Moreover, the developed model can work in a web environment. Thus, the process engineers can calculate basis weight offline without waiting for the quality control laboratory results or the readings shown by the quality control systems (QCS). In addition, continuous improvement engineers can use the user interface mainly in the improvement phase for any six-sigma project. Moreover, the user interface can be used to perform non-invasive experimental designs (Rodriguez-Alvarez et al. 2021), which is one of the main advantages of avoiding the costs of any experimentation process.

Although the model predicts the basis weight with acceptable accuracy, determining optimal operating conditions becomes a complicated function due to the number of independent variables. So, in future work, the user interface will be used to conduct experimental designs to find optimal operating conditions, mainly in complex processes. Even we will try to link the output data provided by the model through the user interface directly to some specialized statistical software.

Since the explained variance score reached by the model was 94 %. Trying to improve this value, in future work, by using k-Means firstly to find the clusters (different types of paper grades), the radial basis function (RBF) approach will be used to train and validate a new model. In addition, a feature selection method based on experimental designs will be used to quantify the effect and direction of each independent variable.

Funding: This work was supported in part by the National Council of Science and Technology [Consejo Nacional de Ciencia y Tecnología (CONACYT)] under Grant 487109.

Conflict of interest: The authors declare no conflicts of interest.

References

ABB (2021) Weight Virtual Measurement for paper, packaging and tissue. Retrieved from https://new.abb.com/pulp-paper/abb-in-pulp-and-paper/papermaking.

Adamopoulos, S., Anthony, K., Rapti, E., Birbilis, D. (2016). Predicting the properties of corrugated base papers using multiple linear regression and artificial neural networks. Drewno 59:61–72. doi:10.12841/wood.1644-3985.144.13.

Amiri, M., Davande, H., Sadeghian, A., Chartier, S. (2010) Feedback associative memory based on a new hybrid model of generalized regression and self-feedback neural networks. Neural Netw. 23(7):892–904. doi:10.1016/j.neunet.2010.05.005.

Antanasijević, D., Pocajt, V., Ristić, M., Perić-Grujić, A. (2015) Modeling of energy consumption and related GHG (greenhouse gas) intensity and emissions in Europe using general regression neural networks. Energy 84:816–824. doi:10.1016/j.energy.2015.03.060.

Baruník, J., Křehlík, T. (2016) Combining high frequency data with non-linear models for forecasting energy market volatility. Expert Syst. Appl. 55:222–242.

Bengio, Y. (2012) Practical recommendations for gradient-based training of deep architectures. In: Neural networks: Tricks of the trade. Springer. pp. 437–478.

Camargo, M.E., Santos, G.M., Russo, S.L. (2010) Applied control charts for analysis of quality control. Paper presented at the 40th International Conference on Computers & Industrial Engineering.

Canário, J.P., Mello, R., Curilem, M., Huenupan, F., Rios, R. (2020) In-depth comparison of deep artificial neural network architectures on seismic events classification. J. Volcanol. Geotherm. Res. 401. doi:10.1016/j.jvolgeores.2020.106881.

Chang, P., Li, Z. (2021) Over-complete deep recurrent neural network based on wastewater treatment process soft sensor application. Appl. Soft Comput. 105. doi:10.1016/j.asoc.2021.107227.

Chang, S., Aw, C. (1996) A neural fuzzy control chart for detecting and classifying process mean shifts. Int. J. Prod. Res. 34(8):2265–2278.

Cheng, C.-B. (2005) Fuzzy process control: construction of control charts with fuzzy numbers. Fuzzy Sets Syst. 154(2):287–303.

Costela, F.M., Castro-Torres, J.J. (2020) Risk prediction model using eye movements during simulated driving with logistic regressions and neural networks. Transp. Res., Part F Traffic Psychol. Behav. 74:511–521. doi:10.1016/j.trf.2020.09.003.

Dayhoff, J.E. Neural network architectures: an introduction. Van Nostrand Reinhold Co, 1990.

De Assis, A.J., Maciel Filho, R. (2000) Soft sensors development for on-line bioreactor state estimation. Comput. Chem. Eng. 24(2-7):1099–1103.

Dudek-Burlikowska, M. (2005) Quality estimation of process with usage control charts type XR and quality capability of process Cp, Cpk. J. Mater. Process. Technol. 162:736–743.

Elman, J.L. (1990) Finding structure in time. Cogn. Sci. 14(2):179–211.

Fausett, L.V. Fundamentals of neural networks: architectures, algorithms, and applications. Pearson Education India, 2006.

Fortuna, L., Graziani, S., Rizzo, A., Xibilia, M.G. Soft sensors for monitoring and control of industrial processes. vol. 22, Springer, 2007.

Fu, X., Wang, L. (2003) Data dimensionality reduction with application to simplifying RBF network structure and improving classification performance. IEEE Trans. Syst. Man Cybern., Part B, Cybern. 33(3):399–409.

Gadeo-Martos, M.A., Fernandez-Prieto, J.A., Velasco, J.R. (2011) An architecture for performance optimization in a collaborative knowledge-based approach for wireless sensor networks. Sensors 11(10):9136–9159.

Géron, A. Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems. O'Reilly Media, 2019.

Gülbay, M., Kahraman, C. (2006) Development of fuzzy process control charts and fuzzy unnatural pattern analyses. Comput. Stat. Data Anal. 51(1):434–451.

Gülbay, M., Kahraman, C. (2007) An alternative approach to fuzzy control charts: Direct fuzzy approach. Inf. Sci. 177(6):1463–1480.

Hashemi Fath, A., Madanifar, F., Abbasi, M. (2020) Implementation of multilayer perceptron (MLP) and radial basis function (RBF) neural networks to predict solution gas-oil ratio of crude oil systems. Petroleum 6(1):80–91. doi:10.1016/j.petlm.2018.12.002.

Haykin, S. A comprehensive foundation. Neural Networks, vol. 2. p. 41, 2004.

Heinisch, J., Lockner, Y., Hopmann, C. (2021) Comparison of design of experiment methods for modeling injection molding experiments using artificial neural networks. J. Manuf. Process. 61:357–368. doi:10.1016/j.jmapro.2020.11.011.

Hinton, G., Srivastava, N., Swersky, K. (2012) Neural networks for machine learning lecture 6a overview of mini-batch gradient descent. Retrieved from https://www.cs.toronto.edu/~tijmen/csc321/slides/lecture_slides_lec6.pdf.

HONEYWELL (2021) Quality Control System 4.0. Retrieved from https://www.honeywellprocess.com/en-US/online_campaigns/QCS4.0/Pages/index.html.

Jaderberg, M., Dalibard, V., Osindero, S., Czarnecki, W.M., Donahue, J., Razavi, A., Vinyals, O., Green, T., Dunning, I., Simonyan, K. Fernando, C., Kavukcuoglu, K. (2017) Population based training of neural networks. arXiv preprint arXiv:1711.09846.

Kamyar, R., Lauri Pla, D., Husain, A., Cogoni, G., Wang, Z. (2021) Soft sensor for real-time estimation of tablet potency in continuous direct compression manufacturing operation. Int. J. Pharm. 602. doi:10.1016/j.ijpharm.2021.120624.

Karamichailidou, D., Kaloutsa, V., Alexandridis, A. (2021) Wind turbine power curve modeling using radial basis function neural networks and tabu search. Renew. Energy 163:2137–2152. doi:10.1016/j.renene.2020.10.020.

Karayiannis, N., Venetsanopoulos, A.N. Artificial neural networks: learning algorithms, performance evaluation, and applications. vol. 209, Springer Science & Business Media, 1992.

Kaya, İ., Kahraman, C. (2011) Process capability analyses based on fuzzy measurements and fuzzy control charts. Expert Syst. Appl. 38(4):3172–3184.

Kilulya, K.F., Mamba, B.B., Ngila, C., Bush, T., Msagati, T.A.M. (2015) Evaluation of the influence of lipophilic extractive residues on dissolving pulp quality parameters by partial least squares method of chemometrics. Nord. Pulp Pap. Res. J. 30(3):402–410. doi:10.3183/npprj-2015-30-03-p402-410.

Kim, D., Kwon, J., Jeon, B., Park, Y.-L. (2020) Adaptive calibration of soft sensors using optimal transportation transfer learning for mass production and long-term usage. Adv. Intell. Syst. 2(6).

Kim, J., Abdella, G.M., Kim, S., Al-Khalifa, K.N., Hamouda, A.M. (2019) Control charts for variability monitoring in high-dimensional processes. Comput. Ind. Eng. 130:309–316.

Konate, A.A., Pan, H., Khan, N., Yang, J.H. (2015) Generalized regression and feed-forward back propagation neural networks in modeling porosity from geophysical well logs. J. Pet. Explor. Prod. Technol. 5(2):157–166.

Lan, T., Tong, C., Yu, H., Shi, X., Luo, L. (2020) Nonlinear process monitoring based on decentralized generalized regression neural networks. Expert Syst. Appl. 150. doi:10.1016/j.eswa.2020.113273.

Mahmoud Ali, M., Omran, A.N.M., Abd-El-Hakeem Mohamed, M. (2021) Prediction the correlations between hardness and tensile properties of aluminum-silicon alloys produced by various modifiers and grain refineries using regression analysis and an artificial neural network model. Int. J. Eng. Sci. Technol. 24(1):105–111. doi:10.1016/j.jestch.2020.12.010.

Marklund, A., Hauksson, J.B., Edlund, U., Sjöström, M. (1998) Prediction of strength parameters for softwood kraft pulps: Multivariate data analysis based on physical and morphological parameters. Nord. Pulp Pap. Res. J. 13:211–219.

Merbold, H., Maas, D.J.H.C., v. Mechelen, J.L.M. (2016) Multiparameter sensing of paper sheets using terahertz time-domain spectroscopy: Caliper, fiber orientation, moisture, and the role of spatial inhomogeneity. Paper presented at the 2016 IEEE SENSORS (30 Oct.–3 Nov. 2016).

Mezgár, I., Egresits, C., Monostori, L. (1997) Design and real-time reconfiguration of robust manufacturing systems by using design of experiments and artificial neural networks. Comput. Ind. 33(1):61–70. doi:10.1016/S0166-3615(97)00011-0.

Mohammadi, F., Pourzamani, H., Karimi, H., Mohammadi, M.,

Mohammadi, M., Ardalan, N., Khoshravesh, R., Pooresmaeil, H., Shahabi, S., Sabahi, M., Sadat miryonesi, F., Najafi, M., Yavari, Z., Mohammadi, F., Teiri, H., Jannati, M. (2021) Artificial neural network and logistic regression modelling to characterize COVID-19 infected patients in local areas of Iran. Biomed. J. 44(3):304–316. doi:10.1016/j.bj.2021.02.006.
Moody, J., Darken, C. (1989) Fast learning in networks of locally-tuned processing units. Neural Comput. 1(2):281–294.
Morala, P., Cifuentes, J.A., Lillo, R.E., Ucar, I. (2021) Towards a mathematical framework to inform neural network modelling via polynomial regression. Neural Netw. 142:57–72. doi:10.1016/j.neunet.2021.04.036.
Moreira, M.O., Balestrassi, P.P., Paiva, A.P., Ribeiro, P.F., Bonatto, B.D. (2021) Design of experiments using artificial neural network ensemble for photovoltaic generation forecasting. Renew. Sustain. Energy Rev. 135. doi:10.1016/j.rser.2020.110450.
Nabney, I.T. (1999) Efficient training of RBF networks for classification. Int. J. Neural Syst. 210–215.
Napoli, G., Xibilia, M.G. (2011) Soft Sensor design for a Topping process in the case of small datasets. Comput. Chem. Eng. 35(11):2447–2456.
Nie, X., Liang, J., Cao, J. (2019) Multistability analysis of competitive neural networks with Gaussian-wavelet-type activation functions and unbounded time-varying delays. Appl. Math. Comput. 356:449–468. doi:10.1016/j.amc.2019.03.026.
Niño-Adan, I., Landa-Torres, I., Manjarres, D., Portillo, E. (2021) Soft-sensor design for vacuum distillation bottom product penetration classification. Appl. Soft Comput. 102. doi:10.1016/j.asoc.2020.107072.
Paggi, H., Soriano, J., Rampérez, V., Gutiérrez, R., Lara, J.A. (2022) A distributed soft sensors model for managing vague and uncertain multimedia communications using information fusion techniques. Alex. Eng. J. 61(7):5517–5528. doi:10.1016/j.aej.2021.10.060.
Poechmuelloer, W., Halgamuge, S., Glesner, M., Schweikert, P., Pfeffermann, A. (1994). RBF and CBF neural network learning procedures. Paper presented at the Proceedings of 1994 IEEE International Conference on Neural Networks (ICNN'94).
Raunio, J.-P., Ritala, R. (2018) Active scanner control on paper machines. J. Process Control 72:74–90.
Rodriguez-Alvarez, J., Lopez-Herrera, R., Villalon-Turrubiates, I., Grijalva-Avila, G., Garcia-Alcaraz, J. (2021) Modeling and parameter optimization of the papermaking processes by using regression tree model and full factorial design. Tappi J. 20(2):123–137. doi:10.32964/TJ20.2.123.
Rodríguez-Álvarez, J.L., López-Herrera, R., Villalon-Turrubiates, I.E., Molina-Arredondo, R.D., Alcaraz, J.L.G., Hernández-Olvera, Ó.D. (2021) Analysis and control of the paper moisture content variability by using fuzzy and traditional individual control charts. Chemom. Intell. Lab. Syst. 208:1–12.
Rooki, R. (2016) Application of general regression neural network (GRNN) for indirect measuring pressure loss of Herschel–Bulkley drilling fluids in oil drilling. Measurement 85:184–191. doi:10.1016/j.measurement.2016.02.037.
Rosli, N., Ibrahim, R., Ismail, I., Hassan, S.M., Chung, T.D. (2016) Neural network architecture selection for efficient prediction model of gas metering system. Paper presented at the 2016 2nd IEEE International Symposium on Robotics and Manufacturing Automation (ROMA) (25–27 Sept. 2016).
Saha, T.K., Pal, S., Sarkar, R. (2021) Prediction of wetland area and depth using linear regression model and artificial neural network-basedcellular automata. Ecol. Inform. 62. doi:10.1016/j.ecoinf.2021.101272.
Shams, S.R., Jahani, A., Kalantary, S., Moeinaddini, M., Khorasani, N. (2021) The evaluation on artificial neural networks (ANN) and multiple linear regressions (MLR) models for predicting SO2 concentration. Urban Clim. 37. doi:10.1016/j.uclim.2021.100837.
Shamsuzzaman, M., Alsyouf, I., Ali, A. (2015) Optimization design of $\bar{X}$ &EWMA control chart for minimizing mean number of defective units per out-of-control case. Paper presented at the 2015 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM).
Sharma, N., Zakaullah, M., Tiwari, H., Kumar, D. (2015) Runoff and sediment yield modeling using ANN and support vector machines: a case study from Nepal watershed. Model. Earth Syst. Environ. 1(3):1–8.
Shu, M.-H., Wu, H.-C. (2011) Fuzzy X and R control charts: Fuzzy dominance approach. Comput. Ind. Eng. 61(3):676–685.
Specht, D.F. (1991) A general regression neural network. IEEE Trans. Neural Netw. 2(6):568–576.
Sun, M., Ma, Z., Li, Y. (2015) Maneuvering target tracking using IMM Kalman filter aided by Elman neural network. Paper presented at the 2015 7th International Conference on Intelligent Human-Machine Systems and Cybernetics.
Sundaram, N.M., Sivanandam, S., Subha, R. (2016) Elman neural network mortality predictor for prediction of mortality due to pollution. Int. J. Appl. Eng. Res. 11(3):1835–1840.
Tarasov, D., Milder, O., Tyagunov, A. (2018) An Effect of the Paper Microelement Composition on Components of the Color Difference dE94 in Paper Whiteness Assesment. Paper presented at the 2018 2nd European Conference on Electrical Engineering and Computer Science (EECS).
Teuscher, C. Turing's connectionism: an investigation of neural network architectures. Springer Science & Business Media, 2012.
Vinoth, B., Uma, G., Umapathy, M. (2022) Recurrent Neural Network based Soft Sensor for flow estimation in Liquid Rocket Engine Injector calibration. Flow Meas. Instrum. 83. doi:10.1016/j.flowmeasinst.2021.102105.
Wong, P.K., Gao, X.H., Wong, K.I., Vong, C.M. (2018) Efficient point-by-point engine calibration using machine learning and sequential design of experiment strategies. J. Franklin Inst. 355(4):1517–1538. doi:10.1016/j.jfranklin.2017.02.006.
Zadeh, L.A. (1965) Fuzzy sets. Inf. Control 8(3):338–353.
Zaman, B., Lee, M.H., Riaz, M. (2020) An improved process monitoring by mixed multivariate memory control charts: An application in wind turbine field. Comput. Ind. Eng. 106343.
Zeng, L., Ge, Z. (2021) Bayesian network for dynamic variable structure learning and transfer modeling of probabilistic soft sensor. J. Process Control 100:20–29. doi:10.1016/j.jprocont.2021.02.004.
Zhao, X., Xuan, D., Zhao, K., Li, Z. (2020) Elman neural network using ant colony optimization algorithm for estimating of state of charge of lithium-ion battery. J. Energy Storage 32. doi:10.1016/j.est.2020.101789.
Zhiyuan, C., Jinsheng, S. (2015) Optimal design of AEWMA control chart with new sampling strategy. Paper presented at the 27th Chinese Control and Decision Conference (2015 CCDC).