

Article

An Integrated Prediction and Optimization Model of a Thermal Energy Production System in a Factory Producing Furniture Components

Halil Akbaş ^{1,*}  and Gültekin Özdemir ²

¹ Department of Industrial Engineering, Graduate School of Natural and Applied Sciences, Süleyman Demirel University, 32260 Isparta, Turkey

² Department of Industrial Engineering, Faculty of Engineering, Süleyman Demirel University, 32260 Isparta, Turkey; gultekinozdemir@sdu.edu.tr

* Correspondence: akbas.halil@yahoo.com or d1540136507@stud.sdu.edu.tr; Tel.: +90-246-211-3132

Received: 12 September 2020; Accepted: 13 November 2020; Published: 17 November 2020



Abstract: Thermal energy is an important input of furniture components production. A thermal energy production system includes complex, non-linear, and changing combustion processes. The main focus of this article is the maximization of thermal energy production considering the inbuilt complexity of the thermal energy production system in a factory producing furniture components. To achieve this target, a data-driven prediction and optimization model to analyze and improve the performance of a thermal energy production system is implemented. The prediction models are constructed with daily data by using supervised machine learning algorithms. Importance analysis is also applied to select a subset of variables for the prediction models. The modeling accuracy of prediction algorithms is measured with statistical indicators. The most accurate prediction result was obtained using an artificial neural network model for thermal energy production. The integrated prediction and optimization model is designed with artificial neural network and particle swarm optimization models. Both controllable and uncontrollable variables were used as the inputs of the maximization model of thermal energy production. Thermal energy production is increased by 4.24% with respect to the optimal values of controllable variables determined by the integrated optimization model.

Keywords: machine learning; artificial neural network; particle swarm optimization; importance analysis; thermal energy; grate-fired boiler

1. Introduction

The world's fossil fuel reserves have gradually been depleted with the growth in industry, transportation, population, and urban areas. Accordingly, the environmental impacts of high fossil fuel consumption have also increased considerably. Biomass is a suitable renewable energy resource that can substitute for fossil fuels to restrain greenhouse gas emissions [1–3]. Biomass resources can be converted to a more useful energy form with thermal processes called combustion, gasification, and pyrolysis [4]. Combustion is the burning of biomass in the air to produce hot gases and ash. Then, stored chemical energy is converted into heat, which is transformed to kinetic energy through heating water to produce vapor used for gas engines [5]. Great diversity in biomass resources creates the need for adaptable combustion technology in order to achieve efficient and clean biomass combustion [6]. Grate-fired boilers are the leading technologies in biomass combustion for heat and power production. They can fire different types of fuels with variable moisture levels using efficient fuel preparation and less handling effort [1,7]. The stability of thermal energy production (TEP) can be maintained by closely monitoring and controlling the grate-firing process. Current control technologies of the grate-firing

process still need improvement to enhance system efficiency, emission control, and operational automation [8]. Mathematical and computational fluid dynamics (CFD) models provide useful tools for the simulation, optimization and control of grate-fired boilers [1,9–11]. However, the grate-firing process consists of complex operations with chemical and physical reactions. Developing analytical models of the grate-firing process is challenging because of its inherent complexity [12,13]. Data-driven methods offer an alternative way to tackle complex and non-linear problems.

In engineering studies, data are represented as attribute-value pairs with corresponding measurement units. When the experimental data of any industrial application are attained, hidden engineering knowledge in the dataset can be extracted with predictive models. Data-driven methods are promising approaches for modeling and optimization, and they have been applied in various industries such as production systems, healthcare, energy, pollution, emission control, and supply chains [14–20]. Kusiak and Smith (2007) [14] outlined the usefulness of data-mining algorithms for extracting knowledge from a large volume of data, leading to significant improvements in the next generation of products and services. Mishra et al. (2020) [15] presented an enhanced and adaptive genetic algorithm model used with a multilayer perceptron neural network (MLPNN) to detect the presence of diabetes in patients to assist medical experts. Sun et al. (2019) [16] used a backpropagation artificial neural network (ANN)-based extreme learning machine to estimate the remaining useful life of lithium-ion batteries. Rushdi et al. (2020) [17] employed machine learning regression methods to predict the power generated by airborne wind energy systems. Yoo et al. (2016) [18] proposed an integrated ordinal pairwise partitioning and decision tree approach to estimate patterns of groundwater pollution sensitivity with improved accuracy and consistency. Mihaita et al. (2019) [19] applied machine learning modeling using decision trees and ANNs and showed that humidity and noise were the most important factors influencing the prediction of nitrogen dioxide concentrations of mobile stations. Chen et al. (2019) [20] developed an inventory model considering the revenue of an online pharmacy and waste of resources caused by excessive inventory for online pharmacies green supply chain management. They used a genetic algorithm (GA)-based solution procedure to determine the optimal product image location, product price, and procurement quantity to maximize profit.

Modeling and simulation efforts in relation to biomass combustion and heat transfer in grate-fired boilers, oil heaters with convection and radiation systems, heat exchangers, etc. continue to be a research subject [1]. Predictive models play an important role in the management of renewable energy systems. The grate-firing process can be modeled by using data-driven methods to perform generalization at high speed with the availability of historical data in database-management systems. Data-mining algorithms are popular methods for this, and there are a couple of articles in the literature related to building data mining prediction models for industrial boilers using fossil and biomass fuels as energy resources [21–32]. A review of the application of artificial intelligence in modeling and prediction of the performance and control of the combustion process is given by Kalogirou (2003) [21]. Chu et al. (2003) [22] proposed a constrained optimization procedure using ANNs as the modeling tool for the emission control and thermal efficiency of combustion in a coal-fired boiler. Hao et al. (2003) [23] illustrated the ability of ANNs to model complex coal combustion performance in a tangentially fired boiler. They stated that ANNs may not have directly provided a better understanding of the complex combustion processes in boilers than the CFD method, but ANNs were alternative approaches for the engineers to evaluate the combustion performance of the boilers with easier application and shorter computation time than the CFD method. Rusinowski and Stanek (2007) [24] presented the back-propagation neural modeling of coal-fired steam boilers. Their model described the dependence of the operational parameters upon the flue gas losses and losses due to unburned combustibles. Chandok et al. (2008) [25] presented a radial basis function and back-propagation neural network approaches to estimate furnace exit gas temperature of pulverized coal-fired boiler for operator information. A least-squares support vector machine approach was proposed by Gu et al. (2011) [26] to track the time-varying characteristics of a boiler combustion system to improve the energy efficiency and reduce pollutants emissions of power units. Liukkonen et al. (2011) [27] provided a useful approach by combining self-organizing maps, k-means clustering, and ANNs for monitoring the combustion process to increase its efficiency and reduce harmful

emissions in a circulating fluidized bed boiler. Lv et al. (2013) [28] designed an ensemble method including a fuzzy c-means clustering algorithm, least-square support vector machine, and partial least square approaches to predict NO_x emission of coal-fired boiler. A black-box model for the prediction of fuel-ash-induced agglomeration of the bed materials in fluidized bed combustion was created through multivariate regression modeling by Gatternig and Karl (2015) [29]. Toth et al. (2017) [30] used image-based deep neural networks to predict the thermal output of a 3 MW step-grate biomass boiler. Böhler et al. (2019) [31] presented models for the prediction of emissions based on the measured flue gas oxygen concentration and combustion temperature in a small-scale biomass combustion furnace for wooden pellets. Fuzzy black-box models provided the most promising prediction results. Yang et al. (2020) [32] employed the least-squares support vector machine method as a real time dynamic prediction model of NO_x emission of a coal-fired boiler.

Optimization models of boiler systems are studied using evolutionary algorithms, swarm intelligence algorithms, etc. in the literature. Hao et al. (2001) [33] obtained optimal combustion parameters of pulverized coal-fired utility boiler by using combined ANN and GA. Hao et al. (2003) [23] suggested ANN and GAs as effective methods to model the carbon burnout behavior and to optimize the operational conditions in a tangentially fired utility boiler. Si et al. (2009) [34] proposed an integrated optimization approach based on a non-dominated sorting genetic algorithm-II for the optimal operation of a coal-fired boiler. Zhou et al. (2012) [35] modeled NO_x emissions from coal-fired utility boilers by using support vector regression (SVR) with an ant colony optimization algorithm. Wang et al. (2018) [36] applied a Gaussian process to model the relationship between NO_x emission and boiler operation by using GA for the optimization of the hyperparameters of the Gaussian process model. Shi et al. (2019) [37] developed an optimization method utilizing GA based on ANN in a coal-fired boiler to optimize the air distribution scheme for higher thermal efficiency and lower NO_x emission. In this article, supervised machine learning (SML) algorithms are used to predict TEP from operational data. The thermal energy production system (TEPS) is composed of a grate-fired boiler, an oil heater with convection and radiation zones, heat exchangers, and a steam generator [1,25] in a factory producing furniture components and includes complex processes. Regression tree (RT), random forest (RF), non-linear regression (NLR), SVR, and ANN are the SML methods used to create data-driven prediction models of TEP with rather low computational complexity and short computation times. Five different SML algorithms are theoretically explained with formulations. Afterwards, proposed SML methods are employed to build the prediction model of TEP. The convenient decision variables are chosen with tree-based algorithms to reduce the dimensionality of the inputs. Using the test data, the prediction accuracy of the models is evaluated with five different statistical indicators. Moreover, an integrated prediction and optimization model of TEPS is designed. An integrated model is used to maximize the objective function depending on controllable and uncontrollable decision variables. The optimal values of controllable variables for daily observations are researched with ANN simulation connected with particle swarm optimization (PSO) to maximize TEP for each day, while the values of uncontrollable variables are taken exactly as their observed values in the optimization model. To the best of the authors' knowledge, both SML algorithms for the TEP prediction and the integrated ANN-PSO model for the maximization of TEP in a factory producing furniture components are implemented and applied for the first time in the literature in this article.

The remainder of this article is organized as follows. In the next section, RT, RF, NLR, SVR, ANN, and PSO models and their application at a factory producing furniture components are explained. In Section 3, the obtained results are presented and discussed. Finally, Section 4 gives the conclusion and future research objectives.

2. Methodology

2.1. The Facility

In this article, the proposed data-driven approaches are applied to a private company serving the furniture components industry in Turkey and international markets. The company's product range includes wood-based composite panels, such as high-density fiberboard (HDF), medium-density fiberboard (MDF), and laminated MDF [38], and other wood products, such as wood and parquet

floorings, wooden profiles, wooden doors, etc., produced from wood-based composite panels. The wood-based panel production line in the factory involves debarkers, chippers, chips storage yard, screening machines, washer, synthetic adhesive production unit, refiner, dryer, cyclones, fiber sifter, mat former, precompressor, continuous press, saws, cooler, stacking station, sander and cut-to-size machines, and a supervisory control and data acquisition (SCADA) system. The SCADA system collects instantaneous signals from sensors, constitutes a database, and is used as a control interface. The fully automated production line for wood-based panels is connected with the TEPs. The flow chart of the fiberboard production line in the factory is presented in Figure 1.

The quality of wood-based panels mostly depends on the proper production of chips and fiber. The chips are attained from pine and beech wood using chipper machines. Chips are temporarily held in the chips storage yard prior to the screening process, which ensures the cleanliness and uniformity of wood chips. Then, chips are transferred into a washer to separate from foreign materials. The washing process helps to improve the performance of the refining system. Dewatered chips are moved to a pressurized refiner chamber for fiber production with a refining process. After resin, wax, and other additives are blended with the fibers inside the blow line, fibers are pneumatically conveyed through the dryer duct and dried to the required moisture content. Fiber cyclones separate the fibers from the flue gas stream. Fibers are discharged by airlocks and fed into the fiber sifter before the mat-forming process in the wood-based panel production line. The dried fibers are passed through a sifter to separate dense resin particles and defects from the fibers. Dried and resin-coated fibers are laid down on the mat former before they are sent to the precompressor and hot-press machine. The process of pressing is another crucial factor influencing the quality of the wood-based panels. The main influencers of the process of pressing are temperature and pressure. Oil, distributed from the oil collector and circulated in the system, is steered through the convection and the radiation zones of the oil heater to be heated with the flue gas. The oil heater operates as a single-pass system. The oil is directed into the inlet headers, and flows through the curved coils on the convection zone. Afterwards, it passes through the heat exchangers in the radiation zone. Then, heated oil is obtained to transfer heat to the hot press, melamine press, and impregnation machines at the same time. Melamine press and impregnation machines are located in a separate production line and are especially used for the production of laminated MDF products. The fiberboards are cut, sized, and sanded after the cooling process. The targeted thickness and surface smoothness of the wood-based panels are obtained with a sanding process.

The grate-fired boiler is one of the leading technologies used in biomass combustion, and released flue gas with the combustion of biomass is used for TEP. The design of the grate-fired boiler's air supply system has both primary and secondary air fans to improve the efficiency of the biomass combustion. The split ratio of primary air to secondary air in the grate-fired boiler is regulated at different rates for a stable TEP with better burnout and lower pollutant emissions. The primary air fan blows air to the combustion system for rapid combustion and to the grate for cooling. The secondary air fan supplies air to over-grate zones for better combustion performance and to the boiler drum for cooling [1,9]. A stationary sloping grate system is used in the grate-fired boiler [1]. The grate-fired boiler supplies flue gas to the convection and radiation zones of oil heater with the air suction of the thermal oil fan and to the dryer with the air suction of the main fan. The flue gas transferred from the grate-fired boiler and from the air suction line of the thermal oil fan is received in the mixing chamber. The ambient air fan also feeds the mixing chamber with atmospheric air to adjust the process temperature of the flue gas. The flue gas mixture in the dryer is generated by mixing flue gas received from the grate-fired boiler and atmospheric air fed by an air fan in the mixing chamber of the dryer. The flue gas mixture obtained is used to dry fibers by using the dryer and to transfer fibers to the wood-based-panel production line. Flue gas in the convection boiler and the radiation heat exchanger heats the oil circulating in the system. The heated oil is used to obtain steam with the heat exchanger system in the steam generator and is also sent to the continuous press machines to adjust the required pressing process temperature in the wood-based panel production system. Steam is used in the refiner for fiber production. The fuel of the grate-fired boiler is obtained from wood materials, including sawdust, sanding dust, wood chips,

wood logs, wood bark [1], and defective wood products in the factory. Temperature sensors are used for the measurement of temperature. Resistance temperature detectors measure the heated oil temperature, while thermocouple sensors measure the flue gas temperature. The pressure of the boiler and the dryer are measured with pressure transmitters. Automatic valves and dampers are used to control the flow of wood materials, flue gas, air, heated oil, steam, ash, and water in the TEPS. The process diagram of the TEPS in the factory producing furniture components is shown in Figure 2.

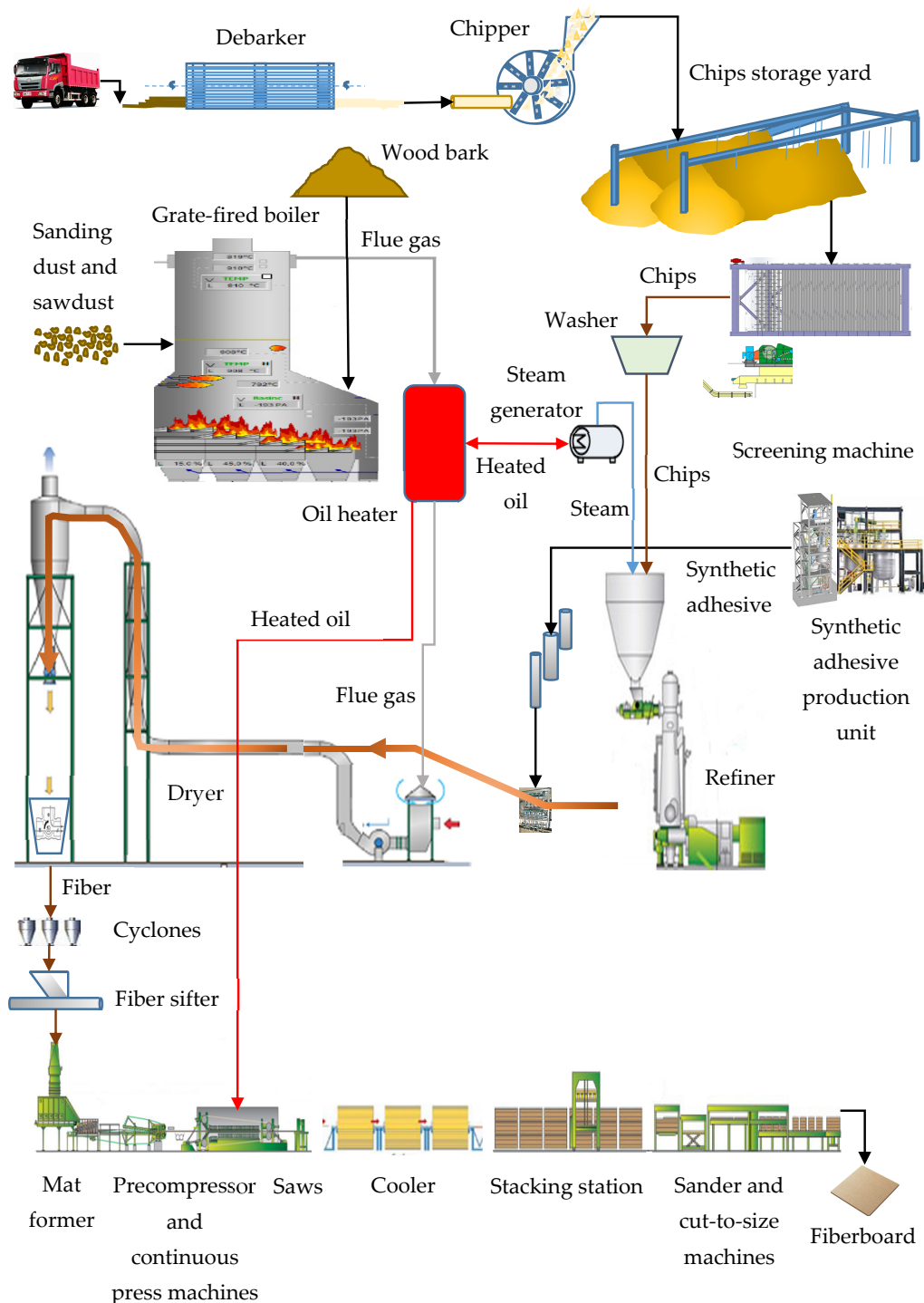


Figure 1. The flow chart of the fiberboard production line.

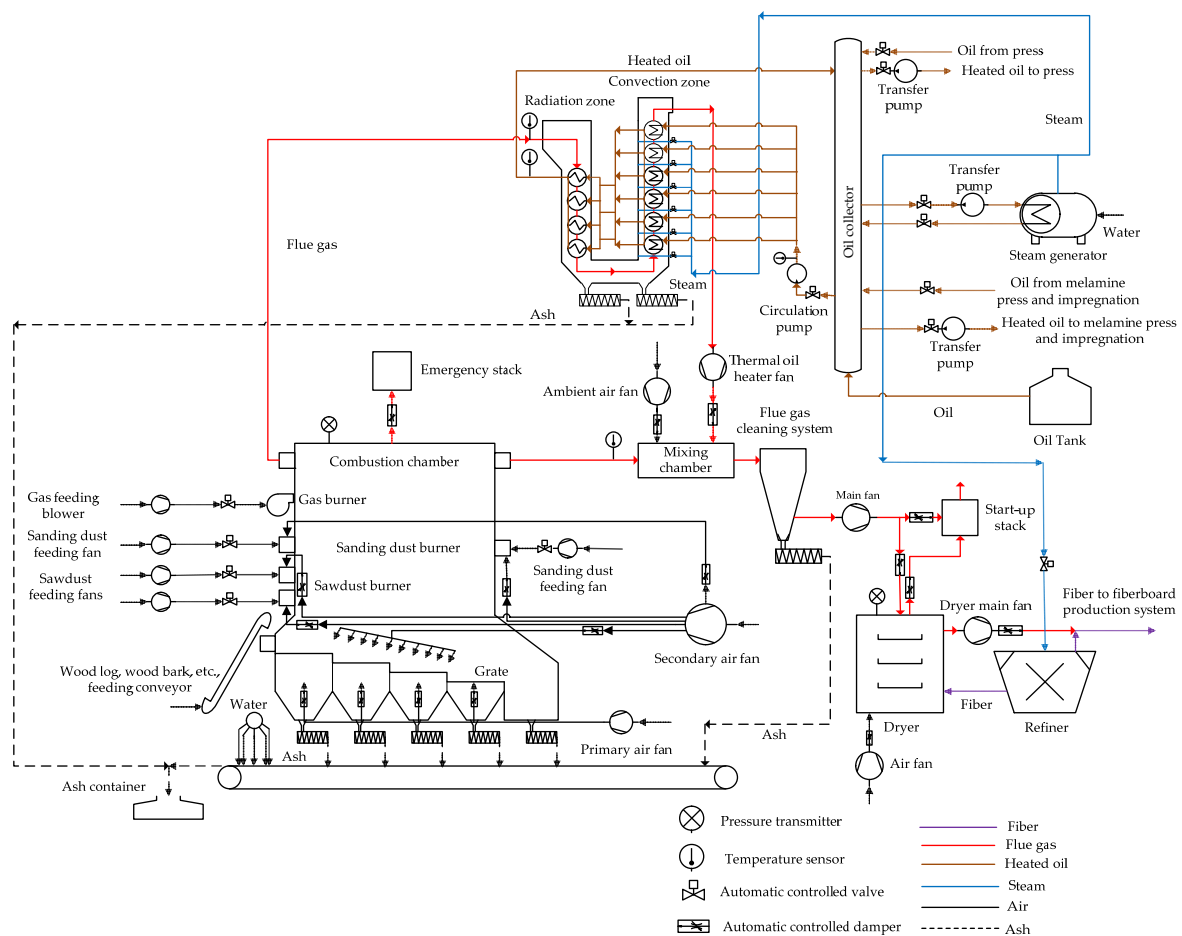


Figure 2. The process diagram of the thermal energy production system in the factory producing furniture components.

2.2. Data Description

The daily data of the TEPS are provided by the factory producing furniture components in Turkey. The data were collected for the period from 1 June 2018 to 31 December 2019. The averages of hourly observed values were used as the daily data. The dataset was preprocessed to remove errors and outliers. There were 482 convenient data points in the data set. The rates of training and test data were tried in both random and chronological order from 70% to 90% and 10% to 30% in turn until the highest accuracy of each predictive model was attained.

2.3. Variable Selection

The dataset includes 22 inputs: grate load (GL), primary air fan passage ratio (PFPR), sanding dust and sawdust load (SDSL), secondary air fan passage ratio (SFPR), flue gas inlet temperature to radiation (FGTR), flue gas inlet temperature to mixing chamber (FGTMC), ambient air fan passage ratio (AFPR), thermal fan passage ratio (TFPR), internal pressure of boiler (IPB), oil flow rate (OFR), oil inlet temperature to convection (OITC), oil exit temperature from radiation (OETR), boiler stack by-pass damper passage ratio (BSPR), total steam flow to refiner (TSFR), pressure of dryer (PD), dryer stack by-pass damper passage ratio (DSPR), main fan damper passage ratio (MFPR), steam control valve passage ratio (SVPR), heated oil flow rate to press (HOFP), heated oil flow rate to melamine press and impregnation (HOFMI), heated oil flow rate to steam generator (HOFSG), and fiber production rate (FPR). The output is called TEP. The list of the input and output variables and their minimum and maximum values are shown in Tables 1 and 2. The source of the values of the inputs and output is the SCADA system of TEPS. The SCADA system controls TEP by providing signals at

regular intervals. Basic information is collected with the use of sensors placed on the components of TEPS. Daily collected data are categorized for controllable and uncontrollable variables. Controllable variables are symbolized with x_i , while uncontrollable variables are represented with u_i . The values of uncontrollable variables are determined with respect to the demands of the production planning department. TEP is represented with y .

Table 1. List and notations of the inputs.

Input Variables (Unit) (x_i, u_i)	Minimum	Maximum	Input Variables (Unit) (x_i, u_i)	Minimum	Maximum
GL (kg/h) (u_1)	1835.00	23,062.22	OETR (°C) (x_{10})	236.00	303.26
PFPR (%) (x_1)	29.62	59.33	BSPR (%) (x_{11})	46.41	88.82
SDSL (kg/h) (u_2)	1275.80	7011.90	TSFR (Ton/h) (u_3)	5.33	30.46
SFPR (%) (x_2)	55.50	77.83	PD (Pa) (u_4)	201.39	435.05
FGTR (°C) (x_3)	755.40	969.00	DSPR (%) (x_{12})	0.20	63.64
FGTMC (°C) (x_4)	805.31	1028.80	MFPR (%) (x_{13})	13.37	88.95
AFPR (%) (x_5)	50.54	82.04	SVPR (%) (x_{14})	7.24	71.83
TFPR (%) (x_6)	25.79	85.16	HOFP (m ³ /h) (u_5)	463.41	869.04
IPB (Pa) (x_7)	162.45	207.83	HOFMI (m ³ /h) (u_6)	158.90	625.56
OFR (m ³ /h) (x_8)	189.00	1781.70	HOFSG (m ³ /h) (u_7)	125.22	1194.70
OITC (°C) (x_9)	211.41	352.15	FPR (Ton/h) (u_8)	2.62	61.50

Table 2. List and notation of the output.

Output (Unit) (y)	Minimum	Maximum
TEP (MW) (y)	14.61	72.50

An industrial grate-fired boiler with a capacity of 75 MW has been established in the factory for the production of thermal energy. Inputs are categorized as controllable variables x_i and uncontrollable variables u_i as presented in Table 1. Some of the system variables are important because they influence each other. For example, the airflow affects biomass combustion, which becomes rapid as the airflow rises. The decision variables PFPR and SFPR are used for the control of airflow to boost TEP. However, AFPR is adjusted with respect to the FGTMC in order to keep the flue gas at a suitable temperature for the fiber drying process. When the wood-based panel production increases, fiber production rises as well. FPR depends on TSFR in the refiner and MFPR of flue gas in the drier. TFPR and MFPR are the decision variables to control the flow of flue gas from convection zone to mixing chamber and from mixing chamber to drier. OITC and OETR are the decision variables to control the increase of oil temperature in TEPS. The quantities of GL and SDSL in the combustion chamber are decided with respect to the need of thermal energy for the production of wood-based panels in the factory. TEP is the output variable representing the sum of heat energy obtained from flue gas, heat energy obtained from heated oil, and steam energy production at the TEPS.

Prediction accuracy can be improved by reducing the number of predictors x_i . Therefore, the importance of predictors can be estimated with RTs by summing changes in the mean square error (MSE) due to splits on every predictor, and dividing the sum by the number of branch nodes. At each node, MSE is estimated as the risk for the node, which is defined as the node error weighted by the node probability. Predictor importance associated with a split is computed as the difference between the MSE for the parent node and the total MSE for the two child nodes. The training dataset is used to evaluate the importance of the predictors [39]. On the other hand, it is also possible to choose the most important variables with RFs. The main idea of predictor importance estimation with RFs is to permute out-of-bag (OOB) data to see how influential the predictor variables in the model are in predicting response [39–41].

2.4. Prediction Algorithms

SML algorithms, which include RT, RF, NLR, SVR, and ANN, are used to train and test the predictive models of TEPS. Regression models were chosen as the solution methodology because decision and response variables are continuous.

All of the SML techniques have advantages and disadvantages, which can be more or less important according to the data analyzed and thus have a relative relevance. RTs predict output variables by using the minimization of MSE as the split criterion. The best split is chosen among all variables at each node. RTs are easy to interpret, and their structure is transparent. However, overfitting is possible with deeper RTs. Minor changes in the training data have the potential to create large changes in the prediction results from RT modeling [39]. RF is the collection of independent RTs with a combined prediction result by averaging. RFs reduce the variance in the RTs by using different training samples. The best split is chosen among the subset of randomly selected predictors at each node in the RF algorithm. More RTs in the RF provide a more robust model and prevent overfitting. However, this might slow down the process [42]. NLR is a type of regression analysis in which the response variable is modeled with the successive approximations of non-linear fitting methods. The NLR method can speed up the prediction process, if there are non-linear and non-stationary relationships between the decision variables in the system [43]. SVR provides considerable advantages. The methodology of SVR is based on both mathematical modeling and statistical learning theory, so it can display a very good generalization performance. Its implementation requires relatively simple optimization techniques. However, large data sets require long training times in the SVR approach. It is necessary to choose the best kernel function and hyperparameters for the SVR model in order to improve its prediction accuracy [44]. ANNs offer a number of advantages. Using ANNs makes it possible to detect all of the interactions between predictors. They are capable of relating the inputs to the desired outputs through highly connected networks of neurons to model non-linear systems. They can also be trained with multiple training algorithms. However, the main disadvantages of ANNs are their black-box and experimental nature in modeling with greater computational burden and overfitting [45].

2.4.1. Regression Tree

The response values are predicted with RTs by following the decisions from the root node to leaf node as seen in Figure 3. The stump is the top node and includes all observations of the training data. A predicted response value \hat{y}_i where $i \in R_m$ is assigned to each terminal node, which is called a leaf.

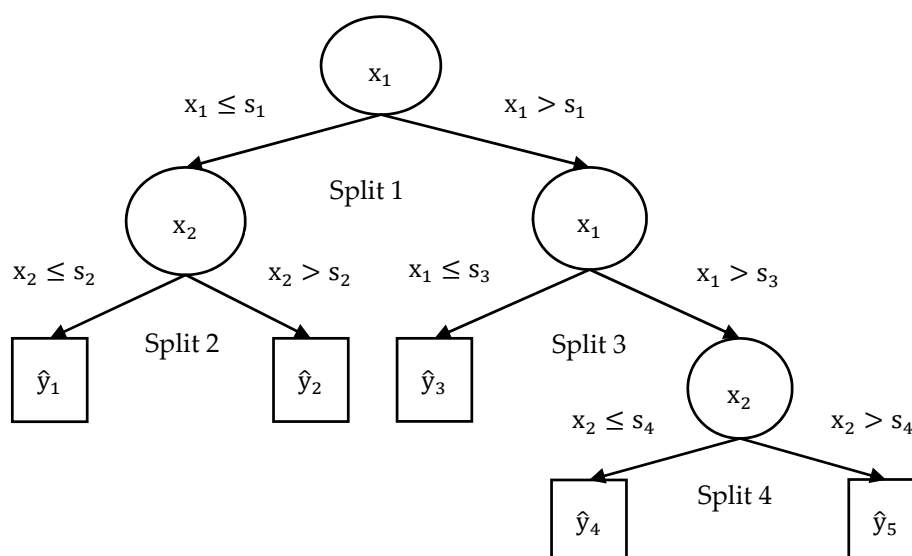


Figure 3. Geometrical representation of regression trees.

The basic idea of RTs is that predictor space is segmented into M regions R_1, R_2, \dots, R_m . The training data are used to calculate the predicted output \hat{y}_i as the mean of response variables y_i falling into each region. $T(x; \theta)$ is the predictor of the RT model given in Equation (1) [39]. θ characterizes parameters in terms of split variables, cut-off points at each node, and the terminal node values for RTs. γ_m is the mean of observed response values in the same region, as seen in Equation (2) [39].

$$T(x; \theta) = \hat{y}_i = \sum_{m=1}^M \gamma_m I(x \in R_m), \tag{1}$$

where $i = 1, 2, \dots, N$ and $m = 1, 2, \dots, M$.

$$\gamma_m = \bar{y}_m = \frac{1}{N_m} \sum_{i \in R_m} y_i, \tag{2}$$

At the tree-growing stage, split nodes are chosen by considering the highest reduction in the MSE between the response values for the training sample and their sample mean at each node. The problem of choosing regions $\{R_m\}_{m=1}^M$ is an NP-hard problem, and its solution is infeasible. For selecting regions in Figure 4, a training error estimate is calculated as the minimization of the MSE as given in Equation (3) [39]. N_m is the number of observations in R_m , while M is the number of regions.

$$\underbrace{\text{Min.}}_{\{R_m\}_{m=1}^M} \frac{1}{N_m} \sum_{i \in R_m} (y_i - \bar{y}_m)^2, \tag{3}$$

where $i = 1, 2, \dots, N$ and $m = 1, 2, \dots, M$.

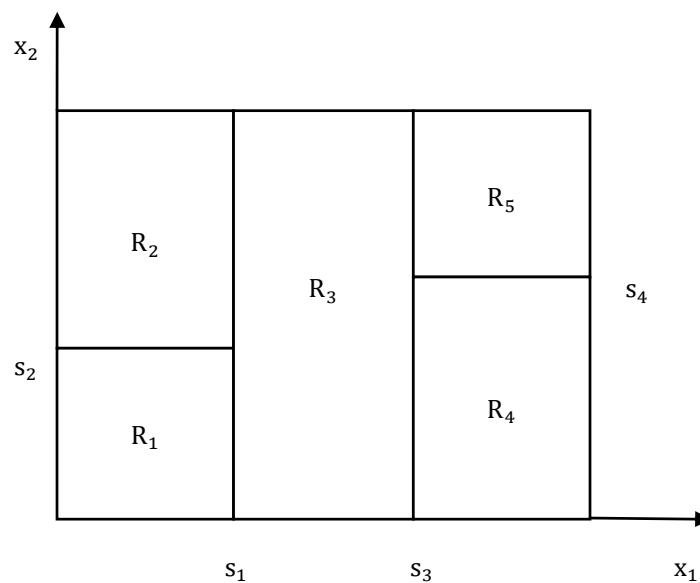


Figure 4. Graphical representation of recursive binary splitting.

Recursive binary splitting is a greedy algorithm to find the optimal way of partitioning the space of possible values of x_i for the split selection. At each iteration, the optimal cut-off point s for x_i is examined in order to choose the predictor that, yields the lowest MSE among p inputs. Iterations are terminated when the MSE gain becomes too small [39].

RTs can be pruned by considering the error-complexity measure as given in Equations (4) and (5) [39].

$$R_\alpha(T) = R(T) + \alpha|T|, \tag{4}$$

$$R(T) = \sum_{i \in R_m} (y_i - \bar{y}_m)^2, \tag{5}$$

$R_\alpha(T)$ is the error-complexity measure, $R(T)$ is the sum of square error of observed responses and predicted responses, $|T|$ is the number of terminal nodes, and α is the contribution to the error-complexity measure for each terminal node.

The $\alpha|T|$ term in Equation (4) penalizes the over-fitting of RTs. The overfitting emerges by partitioning x_i too finely to grow RTs. An independent test data or cross-validated data can be used to choose the right-size of RT. V -fold cross-validation can typically be with $V = 5, \dots, 10$ fold when applied. The proper value of α is examined to avoid overfitting. As long as more nodes are added to the RT, the training error becomes unrealistically low. Therefore, the RT with true complexity is the RT minimizing $R_\alpha(T)$ for the proper value of α [46]. The flow chart of RTs is presented in Figure 5.

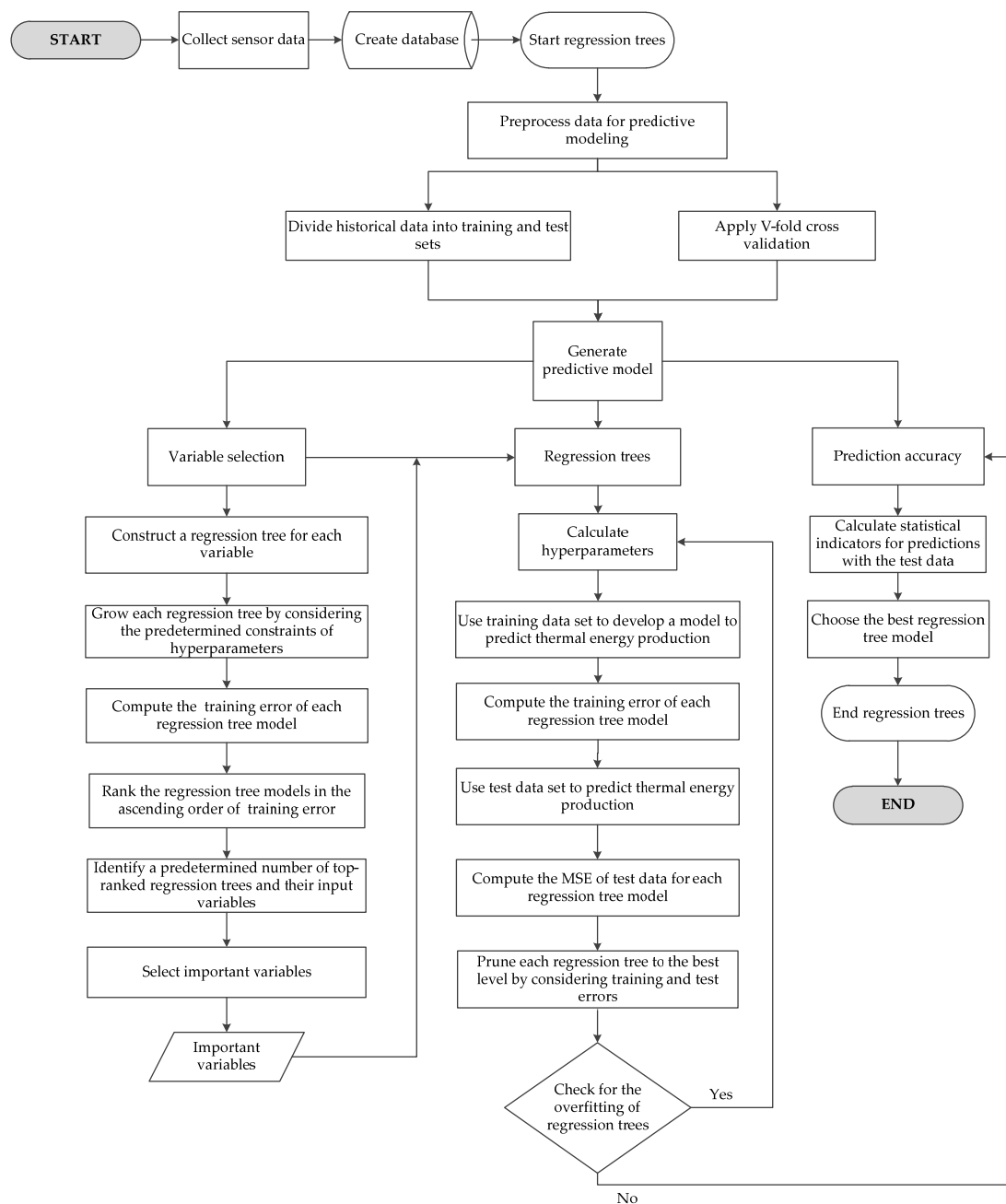


Figure 5. Flowchart of regression trees.

2.4.2. Random Forest

RF is composed of many RTs. It is a prediction technique derived with the modification of bagging. In bagging, a large collection of decorrelated RTs is built, bagged into one predictor, and averaged to obtain an overall prediction. The prediction $\hat{y}^{(b)}$ in Equation (6) [42,46] is obtained when the RT is applied with the bootstrap sample b drawn from the training sample with replacement.

$$\hat{y}^{(b)} = T(x; \mathfrak{D}_b), \quad (6)$$

where $b = 1, \dots, B$.

After growing B bagged trees $\{T(x; \mathfrak{D}_b)\}_1^B$, the RF predictor is formulated by averaging B predicted values as given in Equation (7) [42,46] to calculate the overall prediction. The variance of the predictor $\frac{1}{B} \sum_{b=1}^B \hat{y}^{(b)}$ can be calculated as presented in Equation (8) [42,46].

$$f_{RF}^B(x) = \frac{1}{B} \sum_{b=1}^B \hat{y}^{(b)}, \quad (7)$$

$$\text{Var}\left(\frac{1}{B} \sum_{b=1}^B \hat{y}^{(b)}\right) = \rho\sigma^2 + (1 - \rho)\frac{1}{B}\sigma^2, \quad (8)$$

In RFs, it is assumed that $\hat{y}^{(b)}$ are strongly correlated and identically distributed (ID) over bootstrap sample b . Each of the predictors has the same variance σ^2 . The correlation of different $\hat{y}^{(b)}$ predictors is equal to ρ . When the number of trees B of the bootstrap samples is increased, ρ becomes lower for different predictors, where $\rho > 0$. Then, a lower variance $\lim_{B \rightarrow \infty} \text{Var}\left(\frac{1}{B} \sum_{b=1}^B \hat{y}^{(b)}\right) = \sigma^2$ on the predictor exists [42,46].

OOB samples, which contain 37% of the whole dataset, are used as the test data of RFs. The error rate for observations that are left out of the bootstrap sample is called the OOB error rate, which is monitored for each RT grown on the bootstrap sample [42]. The flowchart of RFs is shown in Figure 6.

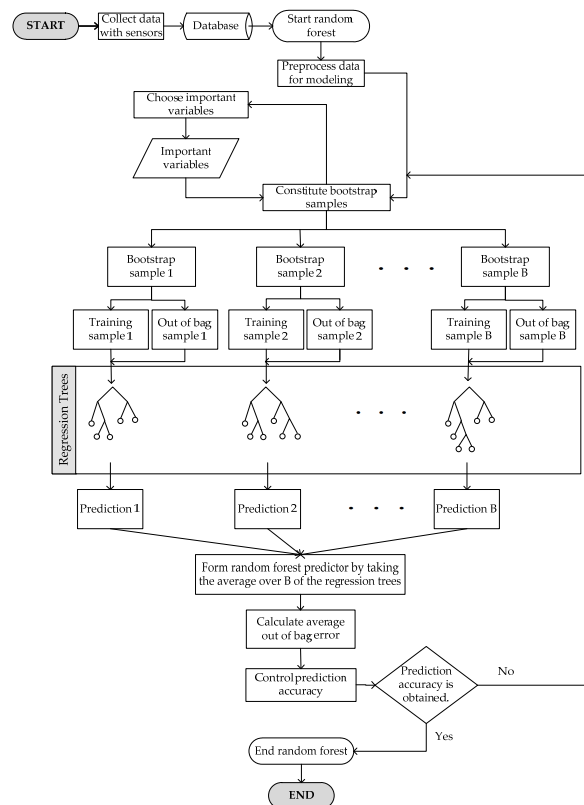


Figure 6. Flowchart of random forests.

2.4.3. Non-Linear Regression

NLR is a powerful modeling tool, which principally refers to the development of mathematical expressions to describe the behavior of random variables. The unknown parameters in a non-linear model are estimated by minimizing a statistical criterion between observed and predicted values. The relationships between y and x_i 's can be expressed with the function f , which can be written as in Equation (9) [43]. f includes the parameter vector $\theta = (\vartheta_1, \vartheta_2, \dots, \vartheta_k)$, which needs to be estimated.

$$y = f(x_1, x_2, \dots, x_n; \vartheta), \text{ where } i = 1, 2, \dots, n. \quad (9)$$

In much of statistical practice, there is a lack of information about the form of the relationship of variables, because the underlying processes are generally complex and not well understood. It is necessary to find some simple function f for which Equation (9) holds as closely as possible [43].

The general form of the NLR model is represented with Equations (10) and (11). $h(x, \vartheta)$ is the function of independent variables x and parameters ϑ . The error term ε is assumed to be an independently and normally distributed random variable. The non-linear functional form of the NLR model is used to create a prediction model with fewer parameters to obtain a better-characterized response. Equations (10) and (11) are iteratively fit by non-linear least-squares estimation [43]. The flowchart of NLR model is given in Figure 7.

$$y = h(x, \vartheta) + \varepsilon, \quad (10)$$

where $\varepsilon \sim N(0, \sigma^2)$.

$$h(x, \vartheta) = \vartheta_1 x_1^{\vartheta_2} + \vartheta_3 x_2^{\vartheta_4} + \dots + \vartheta_{k-1} x_n^{\vartheta_k}, \quad (11)$$

where $i = 1, 2, \dots, n$ and $j = 1, 2, \dots, k$.

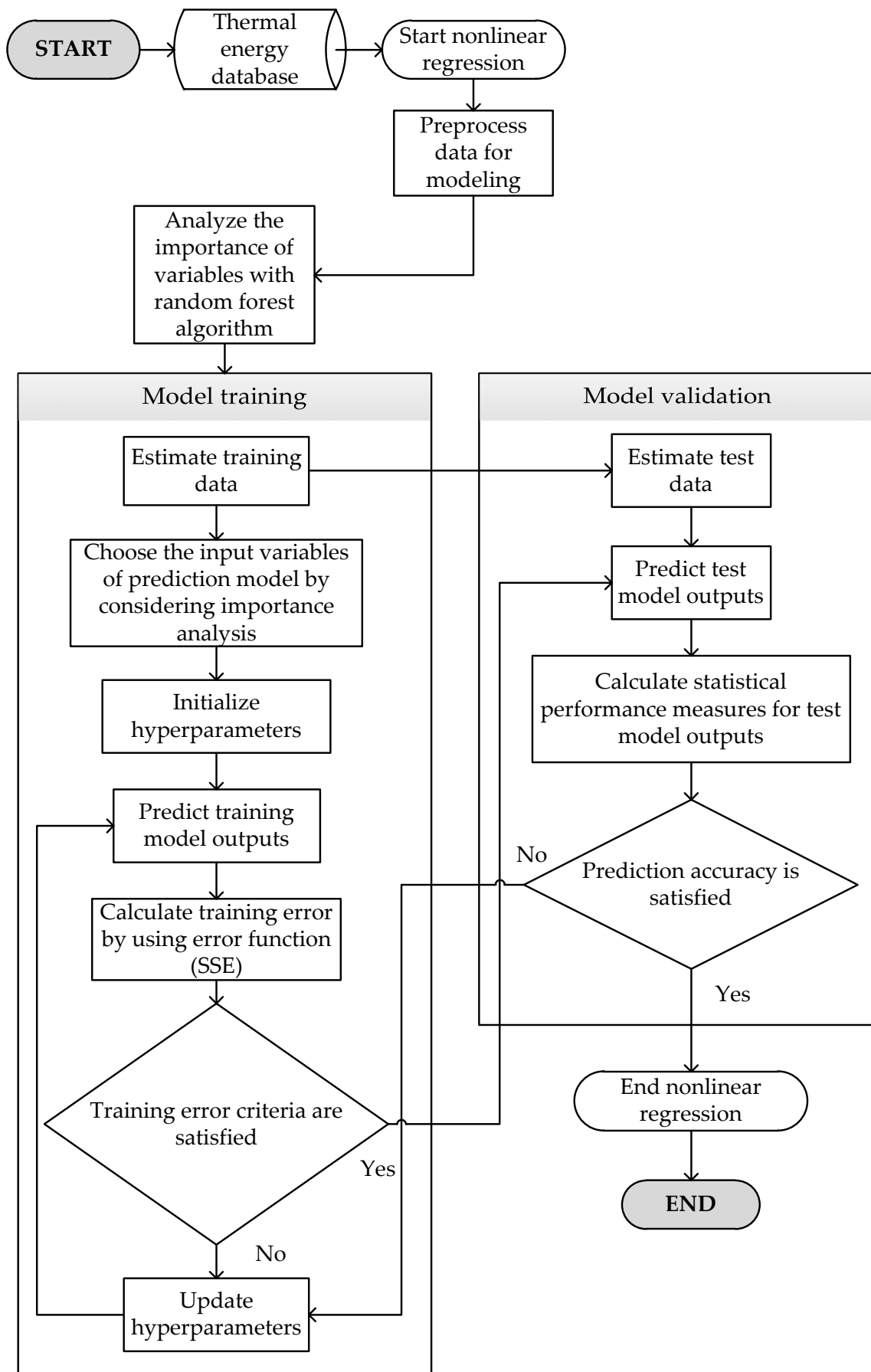


Figure 7. Flowchart of non-linear regression.

2.4.4. Support Vector Regression

SVR allows for designing prediction models with a large number of variables and small samples owing to its robust pattern. SVR can learn both simple and highly complex regression problems. Sophisticated mathematical principles can be employed with the SVR models to avoid overfitting [46]. It utilizes linear, polynomial, and Gaussian kernels to construct prediction models [44].

$D = \{(x_1, y_1), \dots, (x_i, y_i)\}$ denotes the training data where $x \in R^n$ and $y \in R$, assuming that $f(x)$ is a linear function as presented in Equation (12) [44]. Convex optimization problem (COP) with slack variables Ω_i and Ω_i^* searches for the possibility that $f(x)$ does not satisfy the constraints for all points as shown in Equation (13) [44].

$$f(x) = \langle w, x \rangle + b, \tag{12}$$

where $w \in X$ and $b \in R$.

$$\begin{aligned} \min \Phi(w, \Omega) &= \frac{1}{2} \|w\|^2 + \emptyset \sum_{i=1}^n (\Omega_i + \Omega_i^*), \\ \text{subject to } &y_i - \langle w, x_i \rangle - b \leq \varepsilon + \Omega_i : \forall i, \\ &\langle w, x_i \rangle + b - y_i \leq \varepsilon + \Omega_i^* : \forall i, \\ &\Omega_i, \Omega_i^* \geq 0 : \forall i. \end{aligned} \tag{13}$$

In Equation (13), the constant $\emptyset > 0$ is used to determine the trade-off between the flatness of $f(x)$ and the tolerable amount of deviations larger than ε . The COP with slack variables corresponds to the ε -insensitive loss function. Figure 8 illustrates the ε -insensitive loss function, which is formulated as given in Equation (14) [46]. $L_\varepsilon(y)$ is the ε -insensitive error measure. An $f(x)$ function, which deviates from y by a value less than ε for each training point x , is found to keep $L_\varepsilon(y)$ as flat as possible [46].

$$L_\varepsilon(y) = \begin{cases} 0 & \text{for } |f(x) - y| < \varepsilon \\ |f(x) - y| - \varepsilon & \text{otherwise} \end{cases}, \tag{14}$$

where $f(x) - y = \Omega$.

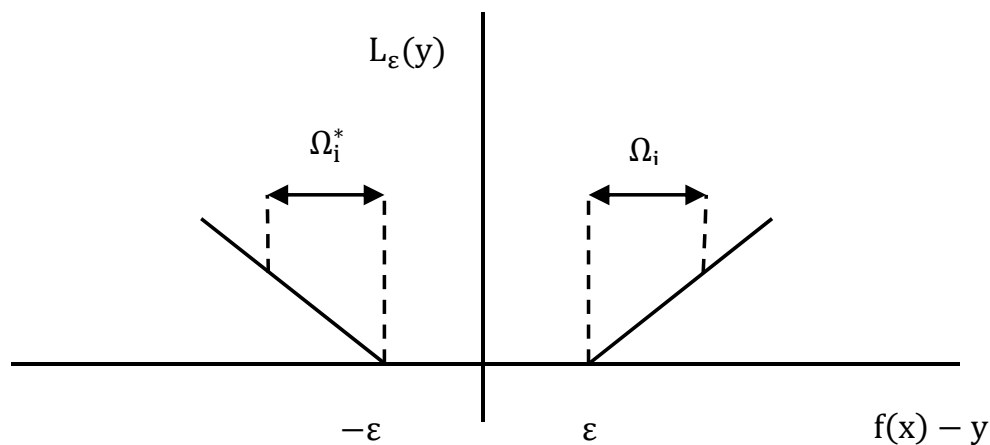


Figure 8. ε -insensitive loss function.

COP is computationally simpler to solve in its Lagrangian dual formulation (LDF). If COP satisfies Karush–Kuhn–Tucker (KKT) conditions [46], the solution of the dual problem gives the value of the

optimal solution to the primal problem. LDF is obtained from the primal function by introducing non-negative multipliers a_i and a_i^* for each observation x_i as presented in Equation (15) [44].

$$\begin{aligned}
 L(a) = \underset{a, a^*}{\operatorname{argmin}} & \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n (a_i - a_i^*)(a_j - a_j^*)k(x_i, x_j) \\
 & - \sum_{i=1}^n (a_i - a_i^*)y_i + \sum_{i=1}^n (a_i - a_i^*)\varepsilon, \\
 & \text{subject to } \sum_{i=1}^n (a_i - a_i^*) = 0, \\
 & 0 \leq a_i, a_i^* \leq \infty : \forall i
 \end{aligned}
 \tag{15}$$

Support vectors depending only on kernel functions $k(x_i, x_j)$ are given in Equation (16) [44]. They are used to predict response values in the SVR model. The flowchart of the SVR model is shown in Figure 9.

$$f(x) = \sum_{i=1}^n (a_i - a_i^*)k(x_i, x_j) + b,
 \tag{16}$$

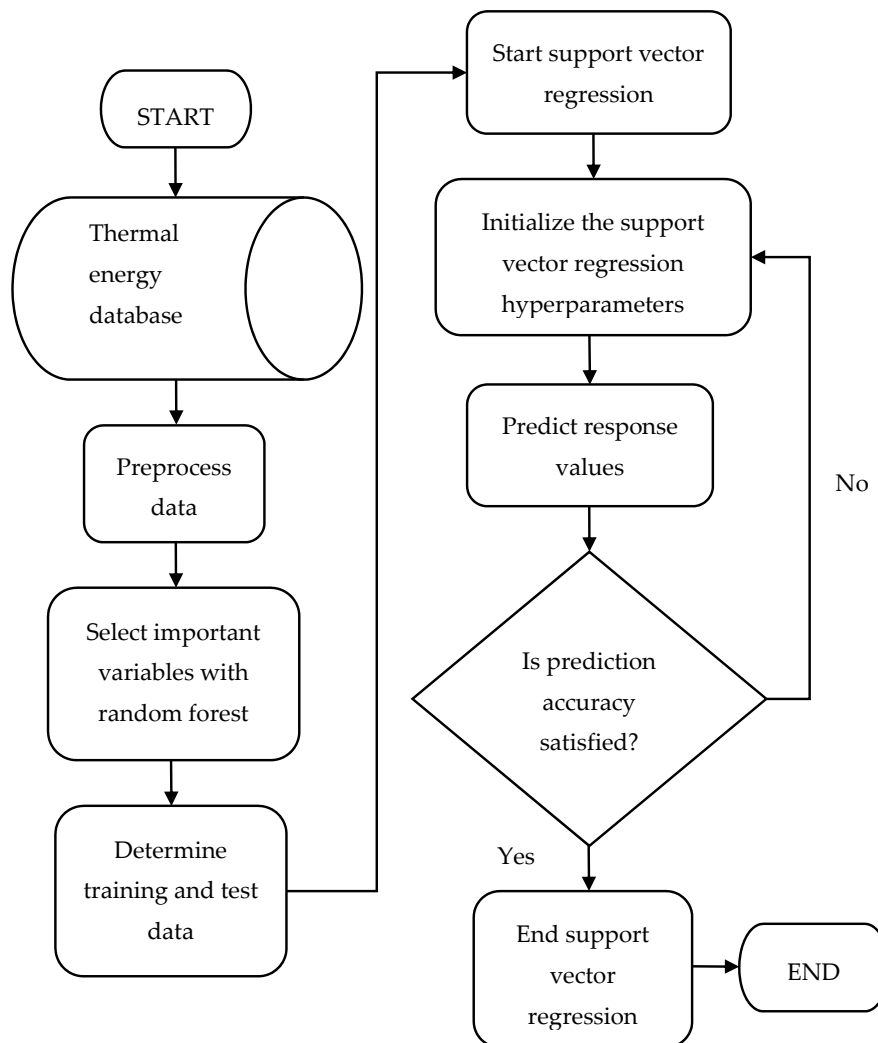


Figure 9. Flowchart of support vector regression model.

2.4.5. Artificial Neural Networks

ANNs are the mathematical representation of biological nervous systems and can execute complex computational tasks. This article uses MLPNNs, which involve multiple and fully connected input, hidden, and output layers as presented in Figure 10. Each node in the hidden and output layers represents a neuron with linear or sigmoid activation functions [44].

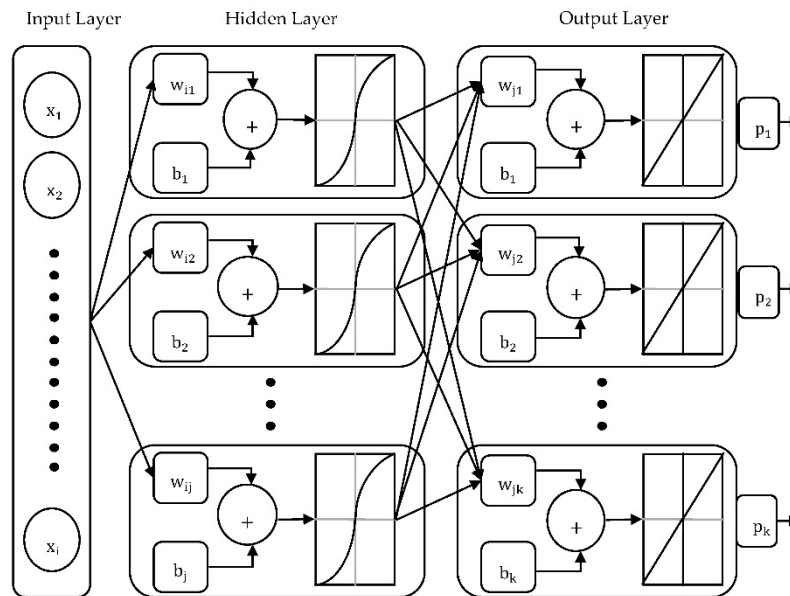


Figure 10. Schematic diagram of the multilayer perceptron neural network with one hidden layer.

The learning process of MLPNNs is influenced by setting the weight vector w on the basis of the comparison of the target and predicted outputs (t_i, p_i) . $T = \{(x_i, t_i) : i = 1, \dots, n\}$ denotes the training data with n argument value pairs of observed inputs x_i and target outputs t_i . The closeness of the target and predicted outputs can be measured with the non-linear error function (EF) as given in Equation (17), in which $p_i = f(x_i, w)$ are predicted outputs [45].

$$E(w) = \frac{1}{2} \sum_{i=1}^n [t_i - p_i]^2, \quad (17)$$

Firstly, all of the inputs x_i are multiplied by their corresponding weights w_{ij} to form preactivation functions y_j for each neuron. The preactivation values obtained from each neuron are input into the non-linear activation functions f_j . The outputs h_j of each neuron are multiplied by their corresponding weights w_{jk} . Then, weighted values are summed to form the preactivation function y_k of the output neuron. Finally, target output values p_k are computed using the preactivation values in the linear activation function f_k . b_j and b_k are the biases of each neuron in the hidden and output layers, respectively. Related formulations are shown with Equations (18)–(21) [45] with respect to Figure 10.

$$y_j = b_j + \sum_i x_i w_{ij}, \quad (18)$$

$$y_k = b_k + \sum_j f_j(b_j + \sum_i x_i w_{ij}) w_{jk}, \quad (19)$$

$$h_j = f_j(b_j + \sum_i x_i w_{ij}), \quad (20)$$

$$p_k = f_k(b_k + \sum_j f_j(b_j + \sum_i x_i w_{ij}) w_{jk}), \quad (21)$$

Back-Propagation Algorithm

The gradient descent algorithms (GDAs) can be used to compute the set of weights minimizing the EF. The EF's derivatives are taken iteratively with respect to each parameter in the GDAs until the best solution is obtained for error minimization. A back-propagation (BP) algorithm in MLPNNs can be used to determine the influence of weights on the prediction of outputs and to adjust the determined weights regarding the error minimization. The modification of the weights in the first and the second layers can be done with Equations (22) and (23), respectively, with the application of the BP algorithm [45]. σ represents the learning rate and t symbolizes the number of iterations in both equations.

$$w_{ij}(t) = w_{ij}(t-1) - \sigma \left(\frac{\partial E}{\partial w_{ij}} \right), \quad (22)$$

$$w_{jk}(t) = w_{jk}(t-1) - \sigma \left(\frac{\partial E}{\partial w_{jk}} \right), \quad (23)$$

The formulations of the amount of weight corrections in both layers using the gradient of the EF are given in Equations (24) and (25) as follows [45].

$$\frac{\partial E}{\partial w_{ij}} = x_i \nabla_j, \quad (24)$$

$$\frac{\partial E}{\partial w_{jk}} = h_j \nabla_k, \quad (25)$$

The biases of both layers are represented with Equations (26) and (27) as given below [45].

$$\frac{\partial E}{\partial b_j} = \nabla_j, \quad (26)$$

$$\frac{\partial E}{\partial b_k} = \nabla_k, \quad (27)$$

The error signals of both layers are formulated with Equations (28) and (29) [45].

$$\nabla_j = f'_j(y_j) \sum_k \nabla_k w_{jk}, \quad (28)$$

$$\nabla_k = f'_k(y_k) E'(p_k, t_k), \quad (29)$$

Levenberg–Marquardt Algorithm

The Levenberg–Marquardt (LM) algorithm is an approximation of Newton's method, which uses a Hessian matrix with the second partial derivatives. The Hessian matrix can be approximated with the formulation in Equation (30) [47].

$$H = J^T J + \mu I, \quad (30)$$

J is the Jacobian matrix that contains the first derivatives of the MLPNN errors with respect to the weights and biases. J can be computed through the BP algorithm. On the other hand, the gradient g can be computed with the formula in Equation (31), where e is the vector of the MLPNN errors [47].

$$g = J^T e, \quad (31)$$

The LM algorithm uses the approximation to the Hessian matrix to update the weights in the MLPNN with Equation (32), where μ is a constant value [47]. The flowchart of the MLPNN model is presented in Figure 11.

$$w_{k+1} = w_k - (J_k^T J_k + \mu I)^{-1} J_k e_k \tag{32}$$

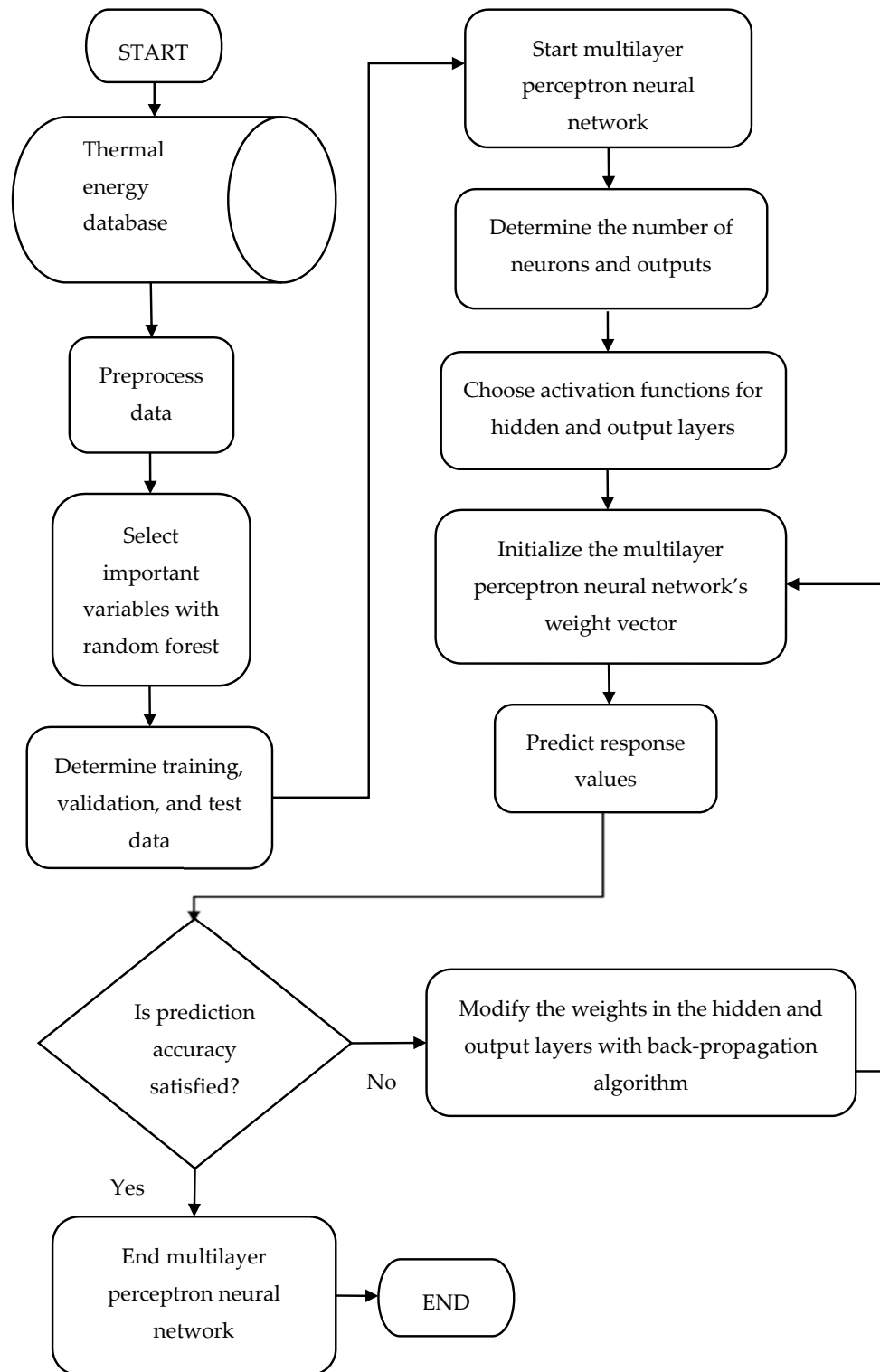


Figure 11. Flowchart of the multilayer perceptron neural network model.

2.5. Modeling Accuracy

Modeling accuracy of the data-driven models are evaluated with percentage error (PE), fractional bias (FB), root mean square error (RMSE), normalized mean square error (NMSE), and index of agreement (IA). Their formulations are given in Equations (33)–(37) [48]. \hat{y}_i represents predicted output values. y_i is the observed output value. \hat{y}_m symbolizes the mean of predicted output values, while y_m is the mean of observed output values. N is the number of observations in the test data. IA is an indicator that measures the correlation between the predicted and observed values. IA varies between 0 and 1. When IA takes the value of 1, it results in the best modeling accuracy under the condition of $FB = NMSE = 0$.

$$PE = \frac{1}{N} \sum_{i=1}^N \frac{|\hat{y}_i - y_i|}{y_i}, \quad (33)$$

$$FB = 2 \frac{\hat{y}_m - y_m}{(\hat{y}_m + y_m)}, \quad (34)$$

$$RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2}, \quad (35)$$

$$NMSE = \frac{\sum_{i=1}^N (y_i - \hat{y}_i)^2}{\sum_{i=1}^N (\hat{y}_i)^2}, \quad (36)$$

$$IA = 1 - \frac{\sum_{i=1}^N (y_i - \hat{y}_i)^2}{\sum_{i=1}^N (|y_i - y_m| + |\hat{y}_i - y_m|)^2}, \quad (37)$$

2.6. Optimization Model

This article aims to design an optimization model of TEPS in a factory producing furniture components to maximize TEP per unit of wood consumption. A PSO algorithm is proposed to calculate the optimal values of decision variables. The TEPS optimization model is designed as an integrated model of ANN and PSO.

PSO is a stochastic population-based metaheuristic inspired from swarm intelligence. Particles are simple and non-sophisticated agents cooperating by an indirect communication and moving in the decision space. PSO has many similarities with GAs, in which a population of random solutions is initialized to search for optimal solutions by updating generations. However, PSO does not use evolutionary operators such as crossover and mutation. The potential solutions are called particles in the PSO algorithm. A swarm of particles is arbitrarily arranged in the problem space by following the current optimal particles. The particles are progressively in search of different positions and another locally or globally best solution [49].

Shi and Eberhart (1998a, 1998b) [50,51] reported the formulas of the velocity and position of particles with the inclusion of inertia weight w in the PSO algorithm as presented in Equations (38) and (39). w is selected from the interval of [0.9,1.4]. $X_i = (x_{i1}, x_{i2}, \dots, x_{id})$ represents particles, which are potential solutions to problems in the D-dimensional search space. $V_i = (v_{i1}, v_{i2}, \dots, v_{id})$ is the velocity along each dimension. Each particle i is defined by its position vector x_{id} . The position of a particle's best solution is represented by p_{id} . p_{gd} is the position of the global best solution for all particles. c_1 and c_2 are cognitive and social parameters, which have positive values, while r_1 and r_2 denote random numbers uniformly distributed in the interval of [0, 1].

$$v_{id}^{new} = w * v_{id}^{old} + c_1 * r_1 * (p_{id} - x_{id}) + c_2 * r_2 * (p_{gd} - x_{id}), \quad (38)$$

$$x_{id}^{new} = x_{id}^{old} + v_{id}^{new},$$

where $i = 1, 2, \dots, N$. (39)

The proposed methodology is an integrated ANN–PSO model to maximize the objective function value. The flow chart of the integrated ANN–PSO model is presented in Figure 12. The PSO model begins by considering the controlling parameters and the initial positions and velocities of particles. After the initiation of the PSO model, an ANN simulation is run with the particle positions obtained by the PSO algorithm to calculate the objective function value for each iteration. Newer objective function values are compared with the existing values obtained in the previous iteration in order to capture the best objective function value. An overall loop is run until the termination criterion is met. The objective function value is identical to fitness value. Finally, global best fitness value is determined.

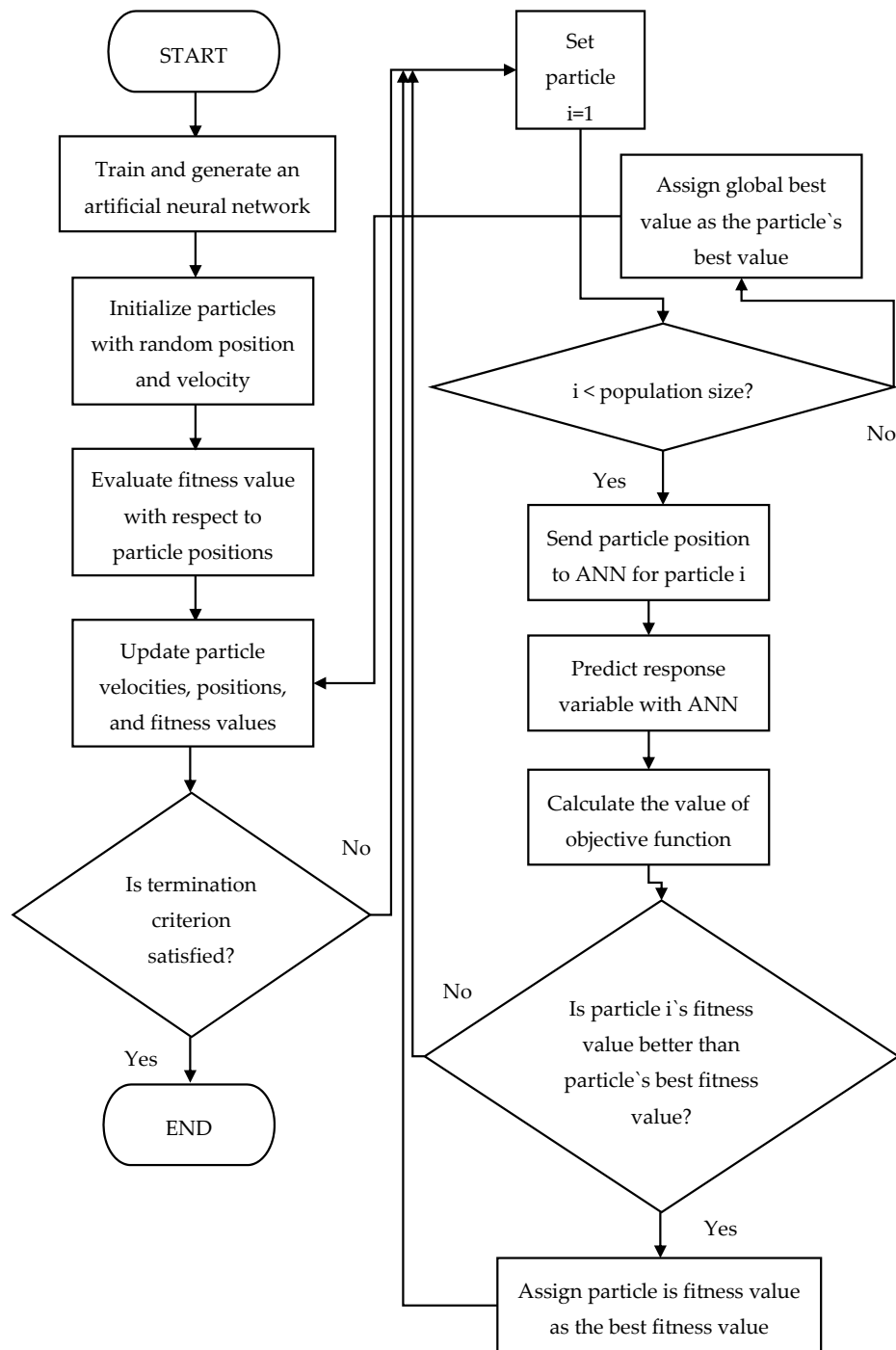


Figure 12. Flowchart of the integrated artificial neural network and particle swarm optimization algorithm.

3. Results and Discussion

Optimization problems encountered in a large spectrum of industrial applications have a complex pattern. Moreover, it is very difficult to solve them in an exact manner within a reasonable amount of computation time. Approximate algorithms are considered as the main alternative to solve the optimization problems of real-life. TEPS also has processes with non-linear relationships, and physical and chemical reactions. TEP processes must be carefully monitored and controlled to have a stable TEP.

The prediction and optimization of TEPS can be executed with three common approaches, namely analytical, hybrid, and data-driven methods. Analytical methods include mathematical models of processes proceeding in the machines and devices of TEPS. However, in many cases, these processes are complex, and the construction of analytical models of physical processes is not practical because of the expertise required, difficulties in making proper assumptions, very long computation time, and inability to adapt to environmental, economical, and social aspects, especially for the experiments of TEPS operators in the course of production. Hybrid methods of TEPS can be developed with the application of both analytical methods and artificial intelligence methods. Moreover, hybrid methods improve the prediction accuracy and reduce the computational complexity of analytical models. However, hybrid methods still contain the same limitations that usually exist in analytical methods. In contrast, data-driven methods of TEPS involve the ability to tackle the difficulties of analytical methods and hybrid methods. Data-driven methods can provide the possibility to explore statistical patterns even from noisy and incomplete data instead of onsite physical information. From this point of view, an integrated ANN-PSO model for the prediction and optimization of TEPS is designed in this article. After SML energy prediction models are applied for the TEP prediction, the SML model with the best prediction accuracy is determined as the ANN model. Then, daily optimal values of controllable variables and the maximized TEP values per day are calculated with the integrated ANN-PSO model. The ANN-PSO model is a data-driven approach particularly implemented for a real-life case of TEPS in a factory producing furniture components. It is possible to solve optimization problems of TEPS by modifying the implemented ANN-PSO model with respect to the characteristics of the problems to be handled in the design and optimization of TEPS.

Six different models are proposed to predict TEP in this article. The PSO algorithm is used as the single objective optimization model. Prediction models and the integrated ANN-PSO model are implemented in Matlab R2020b version. The experiments were conducted by using a notebook with 2.20 GHz Intel (R) Core (TM) i7-8750H processor and 16.00 GB RAM.

RTs, which grow deeply, are usually accurate for the training data. However, they might not have high accuracy on the test data. It is possible to change the depth of RTs by controlling hyperparameters, namely the maximum number of splits, minimum leaf size, and minimum parent size. For the RTs, minimum leaf size from 1 to 5 and minimum parent size from 5 to 50 are examined. If n is the training sample size, the maximum number of splits is limited to $n - 1$. Deep RTs can be pruned to the level with the minimum MSE for training and test data sets to grow shallower RTs. Importance analysis with the RT algorithm is a convenient option to reduce the high dimensionality of inputs in RTs. Training and test data sets of RTs can be chosen either in the chronological or random order. Alternatively, a cross-validation algorithm can be a useful technique to randomly divide the whole data set into V folds with approximate sizes to grow RTs. V -fold cross-validation grows V RTs, and predictions obtained from the cross-validated test data are aggregated with respect to the chronological order in the data set to match them with their observation values. Rates of training data are tried from 70% to 90%. The seed for the random number generator is chosen between 1 and 5.

Table 3 summarizes the RTs with the best prediction performance and their hyperparameters to predict TEP. A pruned regression tree model designed with important predictors (PRTDIP) is proposed to improve the modeling accuracy by using only important predictors GL, MFPR, BSPR, TSFR and HOFSG for the TEP response variable. The importance estimates of each predictor with RTs for TEP are given in Figure 13. The chosen threshold value is 0.5.

Table 3. Regression tree models to predict thermal energy production.

Model Name	Maximum Number of Splits	Minimum Leaf Size	Minimum Parent Size	Number of Inputs	Error Function	V-Fold Cross Validation
PRTDIP model	409	1	10	5	MSE	-
CVSRT model	79	1	5	22	MSE	10-fold

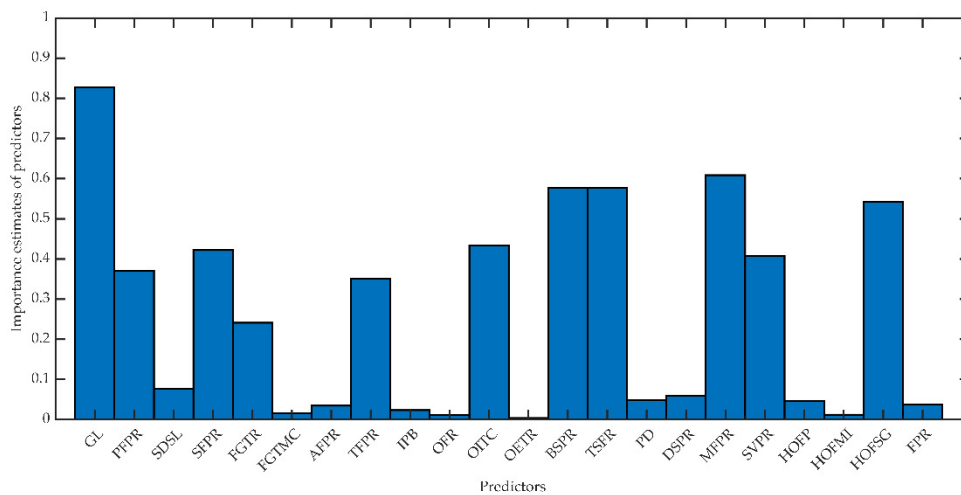


Figure 13. Importance estimates of predictors with regression trees for thermal energy production.

PRTDIP is grown by using 85% of the data points in the training data set. Training data are selected in random order. There are 77 pruning levels in PRTDIP. The minimum level of the training and test errors emerges at the pruning level of 59. The total computation time is 1.64 s.

A cross-validated shallow regression tree (CVSRT) is built with 10-fold cross-validation. After 10 RTs are grown, obtained predictions of TEP with the cross-validated test data are aggregated in chronological order. The histogram of the number of splits in the CVSRT shows that the number of imposed splits with the highest frequency is 160 among 10 CVSRT as given in Figure 14. Then, the maximum number of splits is reduced to 79 to search for the reduction of model complexity. The test error of CVSRT model is minimized with the hyperparameters in Table 3. The computation time of the CVSRT model is 1.32 s. The CVSRT model is much less complex and approximates the performance with the PRTDIP model.

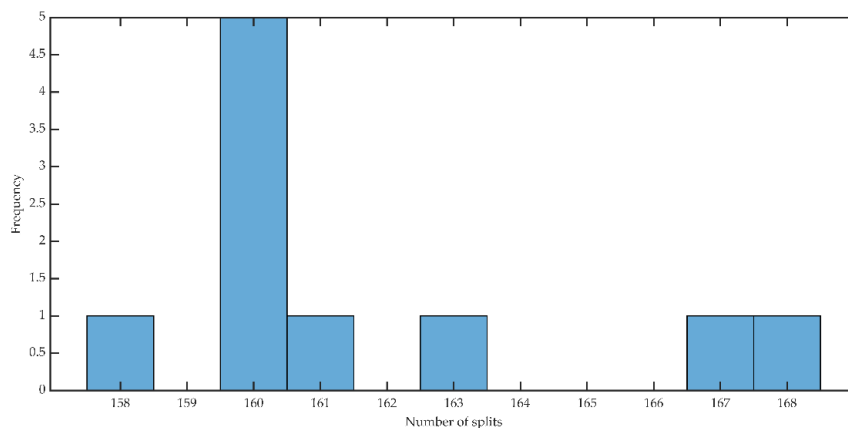


Figure 14. Histogram of the number of splits on the cross-validated shallow regression tree model for the prediction of thermal energy production.

RFs designed with important predictors (RFDIP) arise from RTs with quantities from 1 to 200 to predict TEP. For each RFDIP, OOB test error is calculated with respect to the minimum leaf size of 5, 10, 20, 50, and 100. Important predictors GL, MFPR, HOFSG, BSPR, OITC, SDSL, and TSFR are chosen with the RF algorithm for TEP. The importance estimates of each predictor are given in Figure 15. The threshold value is taken as 0.5 for importance analyses. The OOB test error is minimized when 200 RTs are grown with the minimum leaf size of 5 as presented in Figure 16. The computation time of the RFDIP model is 38.55 s. OOB predictions are estimated by averaging over predictions from all trees in the ensemble for which these observations are OOB. Then, the OOB predictions are brought together to obtain the holistic data set with 482 data points.

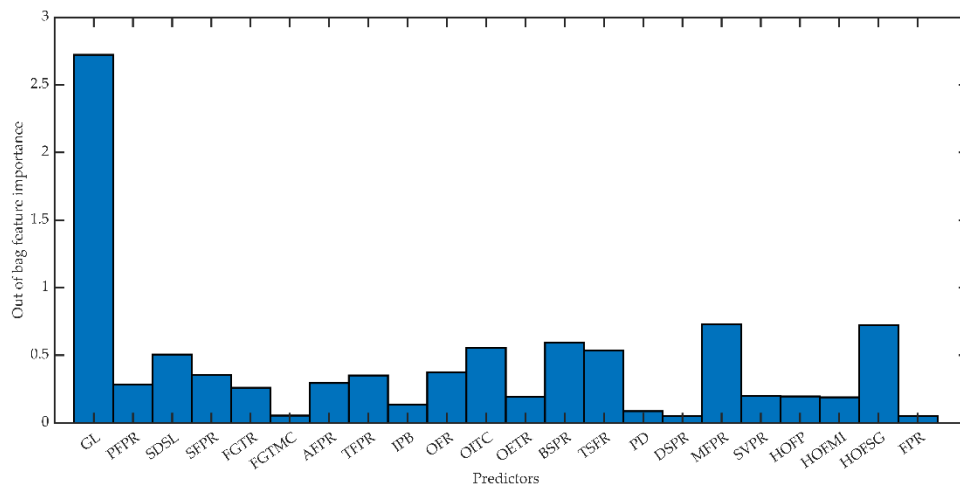


Figure 15. Importance estimates of predictors with random forests for thermal energy production.

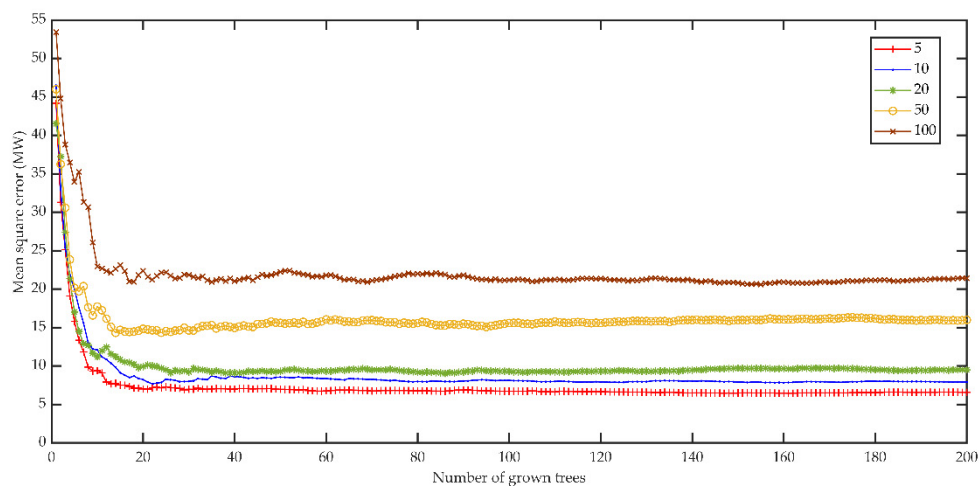


Figure 16. Choosing the best random forest designed with important predictors model by considering MSE with respect to the number of grown trees and leaf sizes.

The NLR model is designed with fewer parameters to obtain a better-characterized response without overfitting. The NLR models for the prediction of TEP are examined with respect to two predictors. They are randomly chosen among the most important predictors determined with the RF algorithm in Figure 15. The cross-validation approach is also tried with the whole data set. The most accurate NLR model to predict TEP is obtained as $y_1 \sim \vartheta_1 + \vartheta_2 x_1^{\vartheta_3} + \vartheta_4 x_1^{\vartheta_5}$, which is designed with the predictors GL and MFPR. The rates of training and test data are 80% and 20%, respectively in the chronological order for the best NLR model, for which parameters ϑ_1 , ϑ_2 , ϑ_3 , ϑ_4 and ϑ_5 are significantly estimated with p -values far less than 0.05. The total time of computation is 1.20 s.

The SVR models are designed with 22 predictors. Linear, polynomial, and Gaussian kernels are tried with the SVR models. Kernel functions are tested to specify the best parameter settings for the box constraint from 0.001 to 1000, kernel scale from 0.001 to 1000, epsilon from 0.0078 to 780, and polynomial order from 2 to 4. The SVR model designed with 22 predictors, a training data rate of 90%, and a test data rate of 10% provides the best prediction results for TEP. The best SVR model uses the polynomial kernel function with the parameters of polynomial order, box constraint, kernel scale, and epsilon as 3, 1, 1, and 0.7805, respectively. The total time of computation is 1.43 s.

The MLPNN with the BP algorithm is modeled with 22 predictors. Training, validation, and test data are constituted with both random and specified indices. The LM algorithm is used as the training algorithm. Batch training with all of the inputs is used to update weights and biases in the ANN model. MSE is the error function. Different numbers of hidden layers are tried from 1 to 15, and different numbers of neurons in hidden layers are tried from 5 to 50. Each neuron in hidden layers includes either hyperbolic tangent sigmoid or log-sigmoid transfer functions, while in output layers it uses a linear transfer function.

The MLPNN models were tried 50 times to find the ANN with the best prediction performance. The MLPNN model with the best prediction performance for TEP contains 22 inputs, 1 hidden layer, and 5 neurons in the hidden layer. Training, validation, and test datasets of the best MLPNN model are created with the rates of 70%, 10%, and 20%, respectively, with specified indices.

Table 4 summarizes the modeling performance of all prediction models tested for TEP. The MLPNN model, for which IA takes the value of 1.00 under the condition of $FB = NMSE = 0$, offers a better prediction accuracy than the other models. The MLPNN model is more prominent, with significantly lower RMSE and PE indicators than the values attributed to the other algorithms. The total computation time of the MLPNN model for TEP is 1.64 s. The ANN model providing the highest prediction accuracy without overfitting is saved to integrate with the PSO model.

Table 4. Modeling accuracy of the prediction models.

Model Name	PE	FB	RMSE	NMSE	IA
PRTDIP model	0.02	0.00	1.73	0.00	0.99
CVSRT model	0.01	0.00	2.41	0.00	0.99
RFDIP model	0.01	0.00	2.57	0.00	0.99
NLR model	0.03	0.00	2.70	0.00	0.99
SVR model	0.01	0.00	0.92	0.00	0.99
MLPNN model	0.01	0.00	0.96	0.00	1.00

Figure 17 shows the test observations against their predicted values by the MLPNN prediction model for TEP. The statistical indicators and MLPNN structure show that the MLPNN model perfectly predicts the TEP with time, as seen in Figure 17. All of the TEP patterns and peaks for 96 points in the test data are clearly recognized by the proposed prediction model.

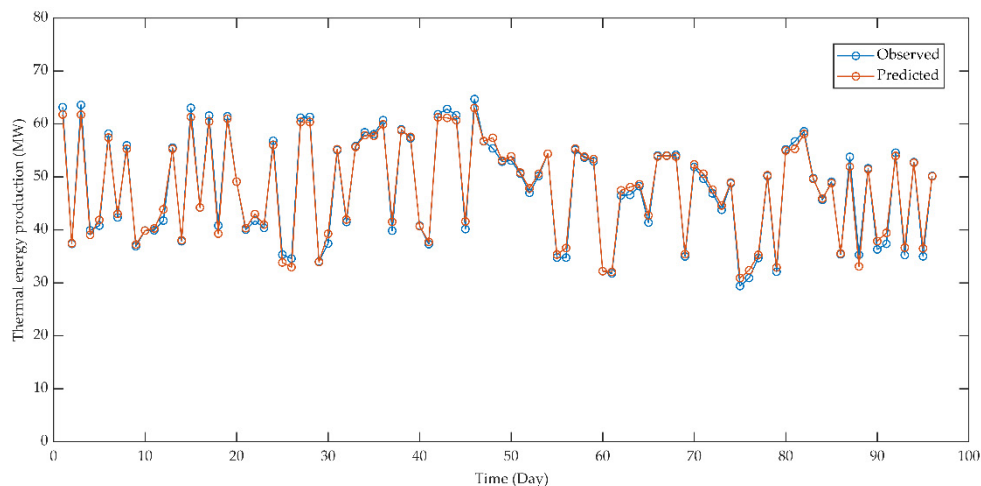


Figure 17. Observed and predicted values of thermal energy production for multilayer perceptron neural network model.

The control of a TEPS requires analytical models for heat transfer and combustion processes that initialize in grate-fired boilers. However, the construction of analytical models is difficult and time-consuming in the on-site experiments of operators because of the complexity of these processes. Thus, this article proposes an optimization model for TEPS using ANN and PSO to improve the capability of operators for system optimality. Regarding the good agreement between the observed and predicted values as presented in Figure 17, it is seen that the complexity of TEPS is clearly described with the MLPNN approach. The SCADA system and reliable sensors are extensively useful for monitoring a TEPS. The operators of a TEPS can control processes by adapting the system variables to any uncertainties of the characteristics of influent biomass, biomass feeding cycles, combustion stability, air disturbance, furnace wall conditions, etc. from the SCADA system in order to reach the maximum TEP with operator experience and process knowledge. However, an integrated ANN–PSO model is proposed to research the optimality conditions of TEPS for the maximization of TEP in this article. The proposed optimization model can be utilized to improve the control ability of operators for the maximization of TEP in practice.

The proposed optimization model aims to find the daily optimal values of controllable variables x_i to maximize TEP per hour for each day. The single objective function with constraints is formulated in Equation (40) as a maximization problem.

$$\begin{aligned} \max_{x_1, \dots, x_{14}} y = f(x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_{10}, x_{11}, x_{12}, x_{13}, x_{14}, u_1, u_2, u_3, u_4, u_5, u_6, u_7, u_8), \\ 29.62 \leq x_1 \leq 59.33; 55.50 \leq x_2 \leq 77.83; 755.40 \leq x_3 \leq 969.00; 805.31 \leq x_4 \leq 1028.80, \\ 50.54 \leq x_5 \leq 82.04; 25.79 \leq x_6 \leq 85.16; 162.45 \leq x_7 \leq 207.83; 189.00 \leq x_8 \leq 1781.70, \\ 211.41 \leq x_9 \leq 352.15; 236.00 \leq x_{10} \leq 303.26; 46.41 \leq x_{11} \leq 88.82; 0.20 \leq x_{12} \leq 63.64, \\ 13.37 \leq x_{13} \leq 88.95; 7.24 \leq x_{14} \leq 71.83. \end{aligned} \quad (40)$$

TEP is a function of both controllable and uncontrollable variables. The description of both types of variables is provided in Table 1. PFPR, SFPR, FGTR, FGTMC, AFPR, TFPR, IPB, OFR, OITC, OETR, BSPR, DSPR, MFPR, and SVPR are selected as controllable variables x_i for the model. The uncontrollable variables u_i are GL, SDSL, TSFR, PD, HOF, HOFMI, HOFSG, and FPR.

Each controllable variable is optimized subject to the constraint between its minimum and maximum observed values in the dataset. The minimum and maximum observed values of the inputs and the output of the TEPS are also presented in Table 1. Swarm size, maximum velocity, inertia weight, and maximum iteration number are sequentially set to 10, 0.09, 1.2, and 10, respectively, for the PSO algorithm. The steps shown in the flow chart of the integrated ANN–PSO model in Figure 12 are

repeated until the termination criterion is met. Operational conditions of TEPS in a factory producing furniture components remain unchanged as much as possible for a full season to obtain a stable TEP.

The daily optimization results of the TEP obtained with the proposed ANN–PSO model are presented in Figure 18. The test data including 96 observations with specified indices are used to solve the maximization problem represented with Equation (40) by using the integrated ANN–PSO model. With each iteration, the trained ANN is used to predict the TEP based on controllable and uncontrollable decision variables. Then, the PSO algorithm determines the best fitness value for the TEP by finding the optimal settings of controllable variables. Uncontrollable variables are also used in the ANN–PSO model with their observed values, but they are not optimized. The elapsed time to solve the TEP maximization problem with the integrated ANN–PSO model is 913 s.

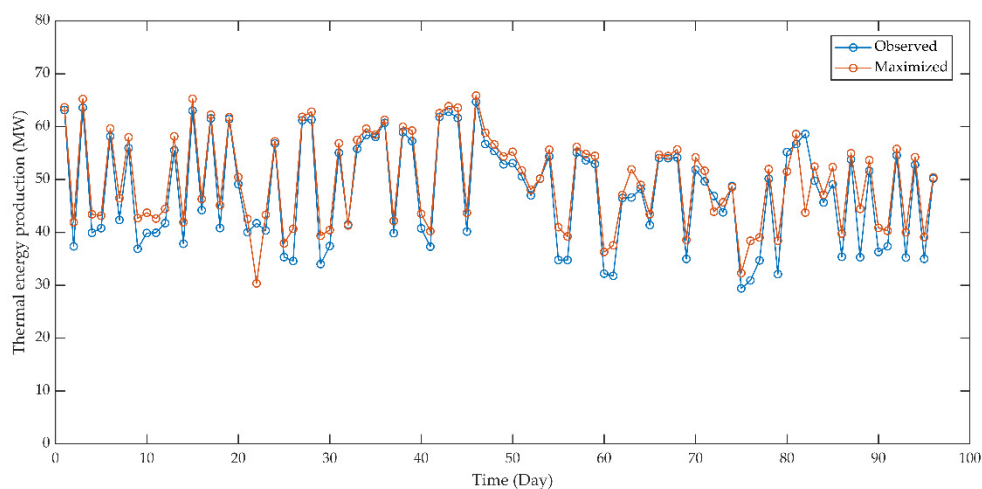


Figure 18. Observed and maximized values of thermal energy production.

Under the optimized operational conditions determined with the integrated ANN–PSO model, TEP can be improved by 4.24%. The increased TEP is owing to the optimization of controllable variables based on the maximization model in Equation (40). Figure 18 illustrates that most of the maximized TEP per hour for each day is larger than the daily observed values. Moreover, the production for the test period shows similar variability to the actual values on a daily basis. The stability of TEP is beneficial for the TEP process and factory operations. The integrated prediction and optimization model of TEPS in a factory producing furniture components is built and implemented by using ANN and PSO algorithms, and the maximized TEP is obtained in case of the optimization of all controllable variables at the same time for daily TEPS operations in this article.

4. Conclusions

The results obtained by this article prove that TEPS, which involves complex operations with chemical and physical reactions, can be modeled and optimized with very high prediction accuracy, low complexity, and short computation time by using the integrated ANN-PSO model. The solution of the single-objective optimization model has resulted in a 4.24% increase in the TEP under optimized operational conditions.

By putting the integrated ANN–PSO model into practice, it is expected that the TEPS will contribute to the cleaner production in the factory producing furniture components with maximized TEP, improved recycling of wood waste, and efficient usage of cleaner technologies at optimal input settings. It is recommended that education and orientation sessions about the application of the proposed optimization model are organized for the TEPS operators.

The integrated ANN-PSO model, which is implemented in this article, is intrinsic to the TEPS. Therefore, it is necessary to redesign the proposed optimization model with respect to the features of

another energy production system in the case of its application for optimization problems in other high-energy consumer industries.

The proposed single objective ANN–PSO model can be customized to design a multi-objective metaheuristic optimization model of TEPS as a future study.

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

Funding: This research was supported by the Research Fund of Süleyman Demirel University, project number FDK-2018-6845.

Acknowledgments: The authors would like to express appreciation to Süleyman Demirel University for their administrative, technical and equipment supports under project number FDK-2018-6845.

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

References

1. Yin, C.; Rosendahl, L.A.; Kaer, S.K. Grate-firing of biomass for heat and power production. *Prog. Energy Combust. Sci.* **2008**, *34*, 725–754. [[CrossRef](#)]
2. Kim, M.H.; Song, H.B. Analysis of the global warming potential for wood waste recycling system. *J. Clean. Prod.* **2014**, *69*, 199–207. [[CrossRef](#)]
3. Sulaiman, C.; Abdul-Rahim, A.S.; Ofozor, C.A. Does wood biomass energy use reduce CO₂ emissions in European Union member countries? Evidence from 27 members. *J. Clean. Prod.* **2020**, *253*, 119996. [[CrossRef](#)]
4. Bridgwater, A.V. Renewable fuels and chemicals by thermal processing of biomass. *Chem. Eng. J.* **2003**, *91*, 87–102. [[CrossRef](#)]
5. Osman, A.I.; Abdelkader, A.; Farrell, C.; Rooney, D.; Morgan, K. Reusing, recycling and up-cycling of biomass: A review of practical and kinetic modelling approaches. *Fuel Process. Technol.* **2019**, *192*, 179–202. [[CrossRef](#)]
6. Costa, M.; Massarotti, N.; Indrizzi, V.; Rajh, B.; Yin, C.; Samec, N. Engineering bed models for solid fuel conversion process in grate-fired boilers. *Energy* **2014**, *77*, 244–253. [[CrossRef](#)]
7. Cai, Y.; Tay, K.; Zheng, Z.; Yang, W.; Wang, H.; Zeng, G. Modeling of ash formation and deposition processes in coal and biomass fired boilers: A comprehensive review. *Appl. Energy* **2018**, *230*, 1447–1544. [[CrossRef](#)]
8. Rezeau, A.; Diez, L.I.; Royo, J.; Diaz-Ramirez, M. Efficient diagnosis of grate-fired biomass boilers by a simplified CFD-based approach. *Fuel Process. Technol.* **2018**, *171*, 318–329. [[CrossRef](#)]
9. Yin, C.; Rosendahl, L.; Kaer, S.K.; Clausen, S.; Hvid, S.L.; Hille, T. Mathematical modeling and experimental study of biomass combustion in a thermal 108 MW grate-fired boiler. *Energy Fuels* **2008**, *22*, 1380–1390. [[CrossRef](#)]
10. Haberle, I.; Skreiberg, O.; Lazar, J.; Haugen, N.E.L. Numerical models for thermochemical degradation of thermally thick woody biomass, and their application in domestic wood heating appliances and grate furnaces. *Prog. Energy Combust. Sci.* **2017**, *63*, 204–252. [[CrossRef](#)]
11. Bermudez, C.A.; Porteiro, J.; Varela, L.G.; Chapela, S.; Patino, D. Three-dimensional CFD simulation of a large-scale grate-fired biomass furnace. *Fuel Process. Technol.* **2020**, *198*, 106219. [[CrossRef](#)]
12. Yin, C.; Kaer, S.K.; Rosendahl, L.; Hvid, S.L. Co-firing straw with coal in a swirl-stabilized dual-feed burner: Modelling and experimental validation. *Bioresour. Technol.* **2010**, *101*, 4169–4178. [[CrossRef](#)] [[PubMed](#)]
13. Ranzi, E.; Pierucci, S.; Aliprandi, P.C.; Stringa, S. Comprehensive and detailed kinetic model of a traveling grate combustor of biomass. *Energy Fuels* **2011**, *25*, 4195–4205. [[CrossRef](#)]
14. Kusiak, A.; Smith, M. Data mining in design of products and production systems. *Annu. Rev. Control.* **2007**, *31*, 147–156. [[CrossRef](#)]
15. Mishra, S.; Tripathy, H.K.; Mallick, P.K.; Bhoi, A.K.; Barsocchi, P. EAGA-MLP- An enhanced and adaptive hybrid classification model for diabetes diagnosis. *Sensors* **2020**, *20*, 4036. [[CrossRef](#)] [[PubMed](#)]
16. Sun, T.; Xia, B.; Liu, Y.; Lai, Y.; Zheng, W.; Wang, H.; Wang, W.; Wang, M. A novel hybrid prognostic approach for remaining useful life estimation of lithium-ion batteries. *Energies* **2019**, *12*, 3678. [[CrossRef](#)]
17. Rushdi, M.A.; Rushdi, A.A.; Dief, T.N.; Halawa, A.M.; Yoshida, S.; Schmehl, R. Power prediction of airborne wind energy systems using multivariate machine learning. *Energies* **2020**, *13*, 2367. [[CrossRef](#)]

18. Yoo, K.; Shukla, S.K.; Ahn, J.J.; Oh, K.; Park, J. Decision tree-based data mining and rule induction for identifying hydrogeological parameters that influence groundwater pollution sensitivity. *J. Clean. Prod.* **2016**, *122*, 277–286. [[CrossRef](#)]
19. Mihaita, A.S.; Dupont, L.; Chery, O.; Camargo, M.; Cai, C. Evaluating air quality by combining stationary, smart mobile pollution monitoring and data-driven modelling. *J. Clean. Prod.* **2019**, *221*, 398–418. [[CrossRef](#)]
20. Chen, Y.K.; Chiu, F.R.; Chang, Y.C. Implementing green supply chain management for online pharmacies through a VADD inventory model. *Int. J. Environ. Res. Public Health* **2019**, *16*, 4454. [[CrossRef](#)]
21. Kalogirou, S.A. Artificial intelligence for the modeling and control of combustion processes: A review. *Prog. Energy Combust. Sci.* **2003**, *29*, 515–566. [[CrossRef](#)]
22. Chu, J.Z.; Shieh, S.S.; Jang, S.S.; Chien, C.I.; Wan, H.P.; Ko, H.H. Constrained optimization of combustion in a simulated coal-fired boiler using artificial neural network model and information analysis. *Fuel* **2003**, *82*, 693–703. [[CrossRef](#)]
23. Hao, Z.; Qian, X.; Cen, K.; Jianren, F. Optimizing pulverized coal combustion performance based on ANN and GA. *Fuel Process. Technol.* **2003**, *85*, 113–124. [[CrossRef](#)]
24. Rusinowski, H.; Stanek, W. Neural modelling of steam boilers. *Energy Convers. Manag.* **2007**, *48*, 2802–2809. [[CrossRef](#)]
25. Chandok, J.S.; Kar, I.N.; Tuli, S. Estimation of furnace exit gas temperature (FEGT) using optimized radial basis and back-propagation neural networks. *Energy Convers. Manag.* **2008**, *49*, 1989–1998. [[CrossRef](#)]
26. Gu, Y.; Zhao, W.; Wu, Z. Online adaptive least squares support vector machine and its application in utility boiler combustion optimization systems. *J. Process. Control.* **2011**, *21*, 1040–1048. [[CrossRef](#)]
27. Liukkonen, M.; Heikkinen, M.; Hiltunen, T.; Halikka, E.; Kuivalainen, R.; Hiltunen, Y. Artificial neural networks for analysis of process states in fluidized bed combustion. *Energy* **2011**, *36*, 339–347. [[CrossRef](#)]
28. Lv, Y.; Liu, J.; Yang, T.; Zeng, D. A novel least squares support vector machine ensemble model for NO_x emission prediction of a coal-fired boiler. *Energy* **2013**, *55*, 319–329. [[CrossRef](#)]
29. Gatternig, B.; Karl, J. Prediction of ash-induced agglomeration in biomass-fired fluidized beds by an advanced regression-based approach. *Fuel* **2015**, *161*, 157–167. [[CrossRef](#)]
30. Toth, P.; Garami, A.; Csordas, B. Image-based deep neural network prediction of the heat output of a step-grate biomass boiler. *Appl. Energy* **2017**, *200*, 155–169. [[CrossRef](#)]
31. Böhler, L.; Görtler, G.; Krail, J.; Kozek, M. Carbon monoxide emission models for small-scale biomass combustion of wooden pellets. *Appl. Energy* **2019**, *254*, 113668. [[CrossRef](#)]
32. Yang, T.; Ma, K.; Lv, Y.; Bai, Y. Real-time dynamic prediction model of NO_x emission of coal-fired boilers under variable load conditions. *Fuel* **2020**, *274*, 117811. [[CrossRef](#)]
33. Hao, Z.; Kefa, C.; Jianbo, M. Combining neural network and genetic algorithms to optimize low NO_x pulverized coal combustion. *Fuel* **2001**, *80*, 2143–2169. [[CrossRef](#)]
34. Si, F.; Romero, C.E.; Yao, Z.; Schuster, E.; Xu, Z.; Morey, R.L.; Liebowitz, B.N. Optimization of coal-fired boiler SCRs based on modified support vector machine models and genetic algorithms. *Fuel* **2009**, *88*, 806–816. [[CrossRef](#)]
35. Zhou, H.; Zhao, J.P.; Zheng, L.G.; Wang, C.L.; Cen, K.F. Modeling NO_x emissions from coal-fired utility boilers using support vector regression with ant colony optimization. *Eng. Appl. Artif. Intell.* **2012**, *25*, 147–158. [[CrossRef](#)]
36. Wang, C.; Liu, Y.; Zheng, S.; Jiang, A. Optimizing combustion of coal fired boilers for reducing NO_x emission using gaussian process. *Energy* **2018**, *153*, 149–158. [[CrossRef](#)]
37. Shi, Y.; Zhong, W.; Chen, X.; Yu, A.B.; Li, J. Combustion optimization of ultra supercritical boiler based on artificial intelligence. *Energy* **2019**, *170*, 804–817. [[CrossRef](#)]
38. Piekarski, C.M.; de Francisco, A.C.; da Luz, L.M.; Kovaleski, J.L.; Silva, D.A.L. Life cycle assessment of medium-density fiberboard (MDF) manufacturing process in Brazil. *Sci. Total Environ.* **2017**, *575*, 103–111. [[CrossRef](#)]
39. Breiman, L.; Friedman, J.H.; Olshen, R.A.; Stone, C.J. *Classification and Regression Trees*, 1st ed.; Pacific Grove: Wadsworth, OH, USA, 1984.
40. Loh, W.Y.; Shih, Y.S. Split selection methods for classification trees. *Stat. Sin.* **1997**, *7*, 815–840.
41. Loh, W.Y. Regression trees with unbiased variable selection and interaction detection. *Stat. Sin.* **2002**, *12*, 361–386.
42. Breiman, L. Random forests. *Mach. Learn.* **2001**, *45*, 5–32. [[CrossRef](#)]

43. Seber, G.A.F.; Wild, C.J. *Nonlinear Regression*, 1st ed.; John Wiley & Sons: Hoboken, NJ, USA, 2003.
44. Vapnik, V. *The Nature of Statistical Learning Theory*, 1st ed.; Springer: New York, NY, USA, 1995.
45. Haykin, S. *Neural Networks and Learning Machines*, 3rd ed.; Pearson Prentice Hall: Hoboken, NJ, USA, 2009.
46. Hastie, T.; Tibshirani, R.; Friedman, J. *The Elements of Statistical Learning: Data Mining, Inference and Prediction*, 1st ed.; Springer: New York, NY, USA, 2001.
47. Hagan, M.T.; Menhaj, M.B. Training feedforward networks with the marquardt algorithm. *IEEE Trans. Neural Netw.* **1994**, *5*, 989–993. [[CrossRef](#)] [[PubMed](#)]
48. Kusiak, A.; Wei, X. Prediction of methane production in wastewater treatment facility: A data-mining approach. *Ann. Oper. Res.* **2014**, *216*, 71–81. [[CrossRef](#)]
49. Kennedy, J.; Eberhart, R.C. Particle Swarm Optimization. In Proceedings of the IEEE International Conference on Neural Networks, Perth, Australia, 27 November–1 December 1995; pp. 1942–1948.
50. Shi, Y.; Eberhart, R. A Modified Particle Swarm Optimizer. In Proceedings of the IEEE International Conference on Evolutionary Computation, Anchorage, AK, USA, 4–9 May 1998; pp. 69–73.
51. Shi, Y.; Eberhart, R. Parameter Selection in Particle Swarm Optimization. In Proceedings of the 7th International Conference on Evolutionary Programming, San Diego, CA, USA, 25–27 March 1998; pp. 591–600.

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).