 doi:10.20944/preprints202210.0391.v1

Article

Application of Computational Intelligence Methods in Agricultural Soil-Machine Interaction : A Review

Chetan Badgujar [1,*], Dania Martinez Figueroa [2], Sanjoy Das [2], Daniel Flippo [1]

[1] Biological and Agricultural Engineering, Kansas State University, Manhattan, KS 66502, USA;
[2] Electrical & Computer Engineering, Kansas State University, Manhattan, KS 66502
* Correspondence: chetan19@ksu.edu

Abstract: Soil working tools, implements, and machines are inevitable in mechanized agriculture. The soil-tool/machine interaction is a multivariate, dynamic, and intricate process. The accurate interpretation, description, and modeling of a soil-machine interaction is key to providing a solution to sustainable crop production by reducing energy input, excessive soil pulverization, and compaction. The traditional method provides insight into soil-machine interaction but often provides inadequate solutions and lacks broad applicability. Computational intelligence (CI) is a comprehensive class of approaches that rely on approximate information to solve complex problems. The CI method has been extensively studied and applied in soil tillage and traction domain in recent decades. The study critically reviews the CI techniques implemented in soil-machine interactions, especially in the context of tillage, traction, and compaction. The traditional methods and their limitation are discussed. The fundamental of CI methods and a detailed overview of the most popular methods are provided. The study reviews and summarizes the 50 selected articles on soil-machine interaction studies where CI methods were employed. It discusses the strength and limitations of employed CI methods. It also suggests the emergent CI methods and future applications are discussed. The outlined study would serve as a concise reference and a quick and systematic way to understand the applicable CI methods that allow crucial farm management decision-making.

Keywords: Tillage; Traction; Compaction; Neural networks; Support vector regression

1. Introduction

Around 97% our food comes from the arable land and soil-engaging tools or machines are an indispensable part of mechanized agriculture. A soil-machine interaction deals with a behavior of tools or machines with a soil that results in either tillage, traction, or compaction. The soil-machine interaction is categorized into tillage, traction, and compaction [1]. In **traction**, a vehicle's powered traction element (wheel/track) operates on a deformable soil, causing soil shear to generates the traction [2]. The soil derived traction overcomes the vehicle's resisting forces, maintains its constant motion with its slip and terrain damage [3]. The slip is a principal form of vehicle power loss and one of the prime causes of the off-road vehicle's worst traction efficiency. Tractors are the prime movers in agriculture and mainly used for drawbar work. Hence, drawbar is the most used, but the least efficient, and approx. 20 to 55% of the tractor's available energy is wasted at the soil-tire interface, often causing a soil compaction and tire wear [4]. Therefore, a traction performance of the vehicle is quantified in-terms of traction, slip and power efficiencies on certain terrain. In off-road vehicle, it is critical to optimize or increase the working capacity, efficiency, and reducing slip and terrain damage. The vehicle traction performance is influenced by multiple variables such as traction element geometry, operating variables, and soil physical properties. Therefore, traction models are often proposed to optimize the off-road vehicle performance (e.g., drawbar, slip, traction).

Tillage alters the soil mechanically to create a favorable crop production conditions [2]. It employs either powered or unpowered mechanical devices (tool/implement) to apply

 doi:10.20944/preprints202210.0391.v1

forces to the soil, resulting in soil cutting, inversion, pulverization, displacement, mixing, or a combined action aiming to obtain desired conditions [5]. Most tillage devices are passive (unpowered), known as a conventional tillage, where drawbar is applied to the device and its movement through soil results in tillage. In contrast, active tillage, also known as a rotary tillage, employs a powered device to transmit power to the soil. The powered tool comparatively move greater soil volume than required, and energy cost increases with a working width and depth. Tillage is the most energy-intensive agricultural operation and accounts for nearly half of the total crop production energy [6].

Tillage energy (active or passive) is influenced by multiple factors, including soil conditions (initial condition, texture, bulk density, moisture content, crop residue cover), tool parameters (geometry, shape, size, cutting edge sharpness), and operating parameters (depth, speed) [5,6]. Therefore, an extensive literature is available which aimed to reduce tillage input energy by optimizing those factors. The research efforts were mainly revolved around the soil failure pattern, movement, and force or energy prediction models [5,6]. The information on tillage force or energy is critical to select tillage types, tools, control variables, energy management, optimization, and reducing excessive soil pulverization. For example, knowing the tool draft in specific soil helps select the tractor size with a matching implement, which reduces the operation costs and negative soil impacts. Therefore, tillage force or energy prediction models are necessary from energy optimization perspective.

Soil **compaction** is a leading factor in productive farmland degradation [7,8]. It has degraded an estimated 83 Mha of farmland [7,9] and affected around 45% of agrarian soil [10,11]. Natural and artificial activities are responsible for soil compaction. The artificial activities involved in crop production can severely affect a soil compaction. These activities include heavyweight machinery and its intense use, uncontrolled vehicle traffic, multiple passes, operating machines under unfavorable conditions (e.g. wet soil), repeated tillage operation, and bad crop rotation [7,12,13]. In addition to top soil compaction, a subsoil or plow pan are caused by a heavy vehicular movement, heavy plow weight, downward forces from a plow bottom/disk, and repeated tillage. Soil compaction resulting from soil-machine interaction influences the soil structure, its porosity, permeability, and density [7,14] which impacts the crop yield and may degrade the soil. The soil compaction evolved from soil-machine interaction is a complex process which involves a multiple interrelated factors. Hence, optimizing the vehicle parameters (e.g. tire type, orientation, inflation pressure, axle weight, traffic), tillage parameters (tool shape, weight, depth, speed, and tillage intensity), and assessing the initial field conditions (soil moisture) can minimize or eliminate the soil compaction.

The soil-machine interaction is a dynamic and intricate process that includes multivariate. But, understanding and accurately describing (models) the soil-machine interaction may provide a solution to sustainable agricultural production. For example, a slight improvement in the tillage tool design or practice could significantly reduce the input energy and avoid excessive soil pulverization or compaction. Similarly, improving the vehicle traction efficiency may increase the working capacity, save input energy, and avoid terrain damage or compaction.

In recent years, computational intelligence (CI) methods succeed in solving an intricate problems in agricultural food production system. The literature shows that a researchers, scholars, and engineers have implemented a cutting-edge CI methods, including neural networks, fuzzy logic, neuro-fuzzy systems, support vector machines, and genetic algorithms to solve a challenging problem in the soil tillage and traction domain. However, it lacks a comprehensive, curated source of reference, a detailed and well-organized discussion on application of CI methods on the soil-machine interaction. Therefore, this study aimed to survey, analyze the recent research efforts in the soil-machine interaction and critically review the existing methods with a detailed discussion. The article provides a brief information, progress, and future direction on implemented CI methods in soil tillage and traction domain. The proposed study would serve as a concise reference to the reader, engineers, researchers, and farm managers who are further interested in the soil-machine

 doi:10.20944/preprints202210.0391.v1

interaction. It is also a quick and systematic way to understand the applicable methods that further allows a crucial decision making on farm management. The review is organized into the following sections:

- Section 2: Discuss on a traditional modeling methods in soil-machine interaction domain.
- Section 3: Presents a brief overview on popular CI methods.
- Section 4: Provides a details discussion on the most popular CI methods.
- Section 5: Discuss the surveyed literature on soil-machine interaction where CI methods was employed.
- Section 6: Identifies the strengths and limitations of CI methods.
- Section 7: Discuss the emergent CI Models.
- Section 8: Discuss the future direction and scope.

2. Traditional Modeling Methods

In recent decades, a numerous methods were proposed to evaluate, analyze, model, and understand the soil-machine interaction, which aimed to optimize input energy, time, machine efficiency, and service life with reduced wear. The methods were explained as follows:

2.1. Analytical Method

Analytical method is based on physical principles of soil/terrain, machine parameters, and simple assumptions. The traction force is computed with a contact-surface interface geometry of soil-tyre and stress distribution (shear and normal) [5,15]. But, both soil and tyre deforms during the process, making it challenging to describe in mathematical form. Moreover, machine dynamics, varying soil, elastic-plastic nature of soil and inadequate information on boundary conditions makes the soil-machine interaction as a very complex problem to model accurately. These challenges raise a questions on its adaptability [3,5,15, 16].

Similarly in tillage, a soil resistance is computed with a logarithmic spiral method, and passive earth pressure theory [6]. These are assumption-based methods which do not include the actual soil failure patterns that vary with a tool parameters (shape, geometry, rake angle, speed) and soil parameters (moisture, density, structure) [17–20]. Also, the analytical methods are good for simple shape but difficult for the complex shape tool [5,6]. Thus, it exhibits a limited applicability for tool design and energy, or force prediction.

2.2. Empirical Method

Empirical method is derived from a large amount of experimental data, and the best-fit regression curve explains a relationship between the selected variables. Empirical equations are simple, consist of a few variables with a constants specific to the soil, track/wheel, tool type, machine configuration, and operating conditions. Thus, these equations can not be extrapolated to the other problems, that restricts its broad applicability [3,5,15]. Thus, precautions are necessary on a new tire, tillage tool, and test environment [3,5]. The method requires a large amount of experimental data, that is laborious and costly. Also, it is subjected to a multi-collinearity problem arising from not truly dependent factors [21].

2.3. Semi-Empirical Method

Semi-empirical method combines the experimental data, empirical formulations, and analytical method. In traction studies, the stress (normal and shear) and soil deformation is computed by assuming stress under a flat-plate, and bevameter is used [3,5,15]. The flat-plate is a non-flexible, but tire or track is flexible, working on a deformable soil. Thus, this method requires an improvement. Similarly, a passive earth pressure theory explains the soil-failure pattern for a simple shaped tillage tool [5,6,19]. But, adopting the earth pressure theory to other complex shaped tool is challenging [5,17,22,23]. Semi-empirical

 doi:10.20944/preprints202210.0391.v1

is a hybrid, reliable and the most common method, although, equation derived from an assumptions limit its accuracy in varying terrain.

2.4. Numerical Method

In recent years, a numerical method such as a Finite Element (FEM) and Discrete Element (DEM) were extensively studied in tillage and traction studies [3,5,15,19,24–28]. A detailed examples can be found here [25,26]. The FEM and DEM were successful in modelling the complex, dynamic, and non-linear soil-machine interaction problems with a greater accuracy and less assumptions [3,25,26]. It is highly computational method, consist of a virtual simulations with a commercial software installed on the high-speed computer. Therefore, it is time-consuming, requires special and costly resources. Also, the simulation setup needs an accurate description of a soil medium that varies on spatial-temporal basis, making it challenging.

In short, the traditional modeling methods has a few limitations and are very specific to machine or tool types and experimental conditions, which restricts its wide applicability.

3. Computational Intelligence: An Overview

Broadly speaking, the term "computational intelligence" refers to a wide class of approaches that rely on approximate information to solve complex problems [29–32]. There are a vast array of such problems (e.g. classification, regression, clustering, anomaly detection, function optimization), where CI models have been extensively used. In the available literature on soil-machine interactions, these models have been used for regression tasks. Accordingly, this article describes CI models from a regression standpoint. However, as some articles have used CI optimization approaches for training the models (i.e. model parameter optimization for best performance), CI based optimization algorithms are also addressed here, albeit in the context of training.

Many CI models apply metaheuristics that are derived from paradigms observed in the natural world. Artificial neural networks (NN), deep neural networks (DNN) and radial basis function networks (RBFN) are structures that loosely resemble the organization of neurons in higher animals. Fuzzy inference systems (FIS) performs computations in a manner analogous to verbal reasoning. Adaptive neuro-fuzzy systems (ANFIS) is designed to combine the attractive features of ANN and Fuzzy Logic (FL). These models are very well suited for regression tasks.

Other CI regression models, which are designed from purely mathematical considerations, do not have any natural counterparts. This class of machine learning methods include support vector regression (SVR) and Bayesian methods.

Natural phenomena also provide the backdrop of CI optimization methods. Genetic algorithms (GA) are modeled after Darwinian evolution, while particle swarm optimization (PSO) simulates the foraging strategy of a swarm of organisms. These methods have been routinely used to train other CI models [33–36].

CI models for regression are data-driven approaches. In soil-machine interaction studies, the data is typically collected from field experiments. Each sample in the data is a pair $(\mathbf{x}(n), \mathbf{t}(n))$, where n is a sample index. The quantity $\mathbf{x}(n) \in \mathbb{R}^M$ in a sample is an M-dimensional input, $\mathbf{t}(n) \in \mathbb{R}^N$ is its corresponding M-dimensional target (or desired output) vector. The symbol $\boldsymbol{\Theta}$ is used in this article to denote the set of all trainable parameters of any CI model. Wherever necessary, it may be treated as a vector. Note that throughout this article, italicized fonts are used to represent scalar quantities, and bold fonts for vectors (lowercase) and multi-dimensional arrays (uppercase).

3.1. Data Preprocessing

Preprocessing is often essential before using data to train a CI model .

(*i*) Data Normalization: This is the most rudimentary form of preprocessing. Each field of the data is normalized separately so that the entries lie in some desired range, usually $[0, 1]$ or $[-1, 1]$.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 26 October 2022 doi:10.20944/preprints202210.0391.v1

(*ii*) Data Cleaning: Experimental data may contain some missing entries. One option to deal with the issue is to remove every sample which contains a missing (scalar) field. This practice may be wasteful, particularly when the data is limited. If so, missing fields may be filled with means, medians, or interpolated values. Corrupted entries can also be treated in this manner. Noise reduction is another form of data cleaning. When the noise follows a non-skewed distribution around a zero mean, noise removal may not be necessary in regression tasks. Convolution with Gaussian or other filters is a common filtering tools for time series data [37].

(*iii*) Dimension Reduction: Dimension reduction is useful when the number of input dimensions, say M' is too high. Principal component analysis (PCA) is widely used for this purpose. More advanced techniques for dimension reduction include nonlinear PCA [38] and independent component analysis [39].

(*iv*) Spectral Transformation: This technique can be used with periodic data. The classical Fourier transform is regularly used to extract frequency components of such data; it does not preserve time information of the input. Wavelet transforms can be used when the data must incorporate frequency as well as temporal components.

The samples are randomly divided into three disjoint sets - the training set $\mathcal{S}_t$, the test set $\mathcal{S}_s$, and the validation set $\mathcal{S}_v$. Training samples are used directly to adjust the model parameters in small increments. Unlike them, samples in $\mathcal{S}_s$, are not used explicitly to compute parameter increments. Instead, testing samples are used intermittently during training to monitor progress. Validation samples in $\mathcal{S}_v$ are used as surrogates for the real world. The performance of the CI model is evaluated w.r.t. $\mathcal{S}_v$ only after training is completely accomplished. Approximately 60% - 80% of the samples are assigned to $\mathcal{S}_t$ and the remainder divided roughly equally between $\mathcal{S}_s$ and $\mathcal{S}_v$.

3.2. Loss Functions

The purpose of training any model is to minimize the differences between the targets and the true outputs, quantified in terms of its loss [40,41], which is the average of the penalties incurred by all samples. The symbol $\mathcal{L}$ is used to represent the loss. The model's loss w.r.t. samples in the dataset $\mathcal{S}$ is,

$$\mathcal{L}_{\mathbf{\Theta}}(\mathcal{S}) = \frac{1}{|\mathcal{S}|} \sum_{n \in \mathcal{S}} l(t(n) - y(n)) \tag{1}$$

The optional subscript $\mathbf{\Theta}$ in (1) above is used to highlight the loss's dependence on the model parameters. Each term $l(\cdot)$ is a sample penalty or error. Using this convention, $\mathcal{L}_{\mathbf{\Theta}}(\mathcal{S}_t)$, $\mathcal{L}_{\mathbf{\Theta}}(\mathcal{S}_s)$, and $\mathcal{L}_{\mathbf{\Theta}}(\mathcal{S}_v)$ are the training and validation losses.

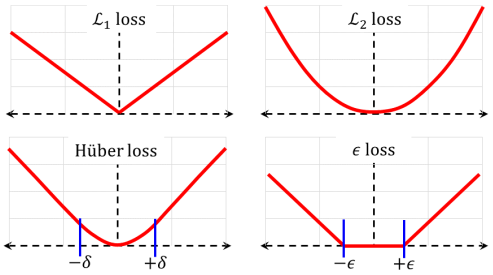


Figure 1. Loss Functions. Losses $\mathcal{L}$ as functions of the difference between the model output y, and the corresponding target (desired output) t.

Several loss functions have been proposed. The following are the most commonly used.

(*i*) Mean squared ($\mathcal{L}_2$) loss: For scalars, this loss is the average of the squared differences between the network's outputs $y(n)$, for inputs $\mathbf{x}(n)$ and the corresponding targets, so

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 26 October 2022 doi:10.20944/preprints202210.0391.v1

that, $\mathcal{L}_2 = |\mathcal{S}|^{-1} \sum_n [y(n) - t(n)]^2$. For vector outputs, the Euclidean norm $||\mathbf{y}(n) - \mathbf{t}(n)||$ is used, where $\mathbf{y}(n)$ is the model's vector output. The $\mathcal{L}_2$ loss is the most commonly used function. Using quadratic penalty terms makes the function quite sensitive to statistical outliers.

(*ii*) Averaged absolute ($\mathcal{L}_1$) loss: This is the average of the absolute difference, $\mathcal{L}_1 = |\mathcal{S}|^{-1} \sum_n |y(n) - t(n)|$. The $\mathcal{L}_1$ loss is used to avoid assigning excessive penalties to noisier samples. On the other hand, its effectiveness is compromised for data with copious noise.

(*iii*) Hüber loss: The Hüber loss represents a trade-off between the $\mathcal{L}_2$ and $\mathcal{L}_1$ losses [42]. Samples where the absolute difference is less than a threshold δ incur a quadratic penalty while the remaining ones, a linear penalty. It is obtained as the average $|\mathcal{S}|^{-1} \sum_n l_\delta(n)$, where $l_\delta(n)$ is the penalty of the n^{th} sample,

$$l_\delta(n) = \begin{cases} \frac{1}{2}[y(n) - t(n)]^2, & \text{if } |y(n) - t(n)| < \delta; \\ \delta |y(n) - t(n)| - \frac{1}{2}\delta^2, & \text{otherwise.} \end{cases} \tag{2}$$

As the Hüber loss function is not twice differentiable at $\pm\delta$, the similarly shaped log-cosh function below can be used in its place,

$$l_{lc}(n) = \log_e \frac{1}{2} \left(e^{y(n)-t(n)} + e^{t(n)-y(n)} \right). \tag{3}$$

(*iv*) ϵ-Loss: This loss does not apply a penalty when the difference $y(n) - t(n)$ lies within a tolerable range $[-\epsilon, +\epsilon]$, for some constant ϵ. A linear penalty is incurred whenever the numerical difference lies outside this range. In other words, $\mathcal{L}_\epsilon = |\mathcal{S}|^{-1} \sum_n l_\epsilon(n)$, where,

$$l_\epsilon(n) = \begin{cases} 0, & \text{if } |y(n) - t(n)| < \epsilon; \\ |y(n) - t(n)| - \epsilon, & \text{otherwise.} \end{cases} \tag{4}$$

Since the loss function is not differentiable at $\pm\epsilon$, if needed, a subgradient in $[0, 1]$ can be used as a substitute for its derivative at $\pm\epsilon$.

The shapes of the above loss functions are illustrated in Fig. 1. The log-cosh loss, which is similar to the Hüber loss, is not shown. There are several other loss functions, including those that are specific to the application, that have not been listed here.

3.3. Model Selection

Model complexity is a key concept in statistical learning theory, that is closely related to overfitting. The complexity of a model can be quantified as the number of independent scalar parameters used to compute its output, as well as their ranges. The V-C dimensionality of a model is one such measure of complexity [43] that has lead to the development of support vector machines.

When selecting a CI model, its complexity is a critical factor that should be considered. Low complexity models tend to exhibit a bias towards specific input-output maps. For instance a linear model, which is the regression model with lowest complexity, cannot be used to capture nonlinear input-output relationships. Conversely, as the CI model's complexity is increased, it is endowed with more degrees of freedom to fit the training data. Due to lower its lower bias, training the model yields significantly lower training error $\mathcal{L}_{\boldsymbol{\Theta}}(\mathcal{S}_t)$. Unfortunately, a model with too large a complexity, becomes too sensitive to extraneous artifacts present in its training dataset $\mathcal{S}_t$, such as random noise, sampling, or aliasing, which do not reflect any underlying input-output relationship. Stated in another manner, as the model's complexity increases, so does its variance. The model with higher variance performs poorly in the real world, with inputs outside $\mathcal{S}_t$. This is reflected in terms of its higher validation loss $\mathcal{L}_{\boldsymbol{\Theta}}(\mathcal{S}_v)$. In general, the model's effective loss can be decomposed into three components as,

$$\mathcal{L}_{\boldsymbol{\Theta}} = \text{bias}^2_{\boldsymbol{\Theta}} + \text{var}_{\boldsymbol{\Theta}} + \text{noise}. \tag{5}$$

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 26 October 2022 doi:10.20944/preprints202210.0391.v1

The square of the bias term is used in (5) as it can acquire positive and negative values. The noise component being an artifact introduced by the external environment is independent of the model $\boldsymbol{\Theta}$. Selecting a CI model with the optimal complexity is the well-known bias-variance dilemma in machine learning. This phenomenon is depicted in Fig. 2.

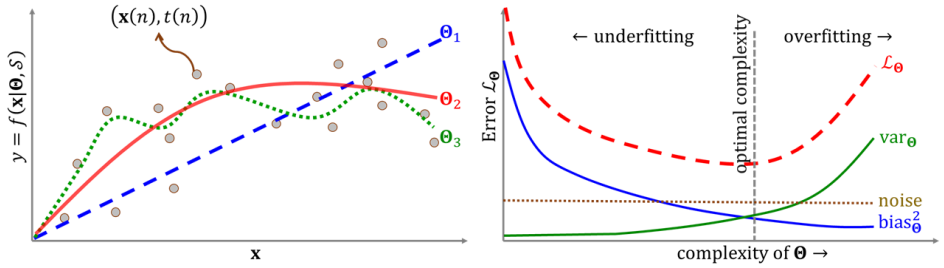


Figure 2. Bias, Variance and Model Complexity. (Left) Performance of three models, $\boldsymbol{\Theta}_1$ (dashed blue), $\boldsymbol{\Theta}_2$ (solid red), and $\boldsymbol{\Theta}_3$ (dotted green) with low, optimum, and high model complexities. Small grey circles are training samples $(\mathbf{x}, \mathbf{t}) \in \mathcal{S}$. (Right) Squared bias (solid blue), variance (solid green), noise (dotted brown) and loss (dashed red) as functions of model complexity.

A widely used approach to keep the model's complexity at lower levels is by adding a regularization term $\mathcal{R}(\cdot)$ to the loss function. Regularizers are routinely devised in terms of the model parameters in $\boldsymbol{\Theta}$. If $\boldsymbol{\Theta}$ is treated as a vector of parameters, $\mathcal{R}(\boldsymbol{\Theta}) = ||\boldsymbol{\Theta}||_1$ and $\mathcal{R}(\boldsymbol{\Theta}) = ||\boldsymbol{\Theta}||_2^2$ are used as LASSO and ridge regularizers. The elastic net function, which is the convex combination of the LASSO and ridge terms, so that $\mathcal{R}(\boldsymbol{\Theta}) = r||\boldsymbol{\Theta}||_2^2 + (1 - r)||\boldsymbol{\Theta}||_1$ (where $0 < r < 1$ is a constant), is another popular choice for regularization [44].

3.4. Training Algorithms

At present, almost all training algorithms rely on basic gradient descent. If $\mathcal{L}_{\boldsymbol{\Theta}}(\cdot)$ is the loss function (which may include a regularization term), the parameters of the model are incremented using the training samples in $\mathcal{S}_t$ as shown in the following expression,

$$\boldsymbol{\Theta} \leftarrow \boldsymbol{\Theta} + \eta \nabla_{\boldsymbol{\Theta}} \mathcal{L}_{\boldsymbol{\Theta}}(\mathcal{S}_t) \tag{6}$$

The quantity η in the above expression is the gradient descent step size, commonly referred to as the learning rate in CI terminology. The operator $\nabla_{\boldsymbol{\Theta}}$ is the gradient (vector derivative) w.r.t. $\boldsymbol{\Theta}$.

Since the loss $\mathcal{L}_{\boldsymbol{\Theta}}(\mathcal{S}_t)$ is the sum of all sample penalties $l(n)$ where $n \in \mathcal{S}_t$, a direct implementation of Eqn. (6) would require a pass over all samples in $\mathcal{S}_t$ before $\boldsymbol{\Theta}$ can be updated. As this is computationally burdensome (particularly for large datasets), training algorithms invariably use stochastic gradient descent (SGD). Before every training epoch of SGD, the samples in $\mathcal{S}_t$ are rearranged randomly. The vector parameter $\boldsymbol{\Theta}$ is incremented once for each sample n, using the gradient $\nabla_{\boldsymbol{\Theta}} l(n)$.

In theory, SGD can lead to a speed up the training algorithm by a factor $|\mathcal{S}_t|$. However, as the directions of the gradients $\nabla_{\boldsymbol{\Theta}} l(n)$ are not perfectly aligned with one another, the actual speed up is considerably less than $|\mathcal{S}_t|$. Adding a momentum term to the gradient step helps alleviate this situation. If in step $n - 1$ the parameter $\boldsymbol{\Theta}$ is incremented by an amount $\Delta\boldsymbol{\Theta}(n-1)$, in the next step n, the increment would be $\Delta\boldsymbol{\Theta}(n) = \eta \nabla_{\boldsymbol{\Theta}} l(n) + \mu \Delta\boldsymbol{\Theta}(n-1)$. The quantity μ $(0 \leq \mu < \eta)$ is the momentum rate.

The convergence rate of the training algorithm can be significantly improved by the Newton's algorithm, which requires the Hessian matrix $\nabla^2_{\boldsymbol{\Theta}}$. It can be readily established that the outer product $\nabla_{\boldsymbol{\Theta}} \nabla^{\mathrm{T}}_{\boldsymbol{\Theta}}$ is a close approximation of the Hessian. In the Levenberg-Marquardt algorithm [45], the diagonal elements of this matrix are incremented by an amount μ to improve the conditioning. Accordingly, incremental updates with the Levenberg-Marquardt algorithm are implemented as per the following rule,

$$\boldsymbol{\Theta} \leftarrow \boldsymbol{\Theta} + \left(\nabla_{\boldsymbol{\Theta}} \nabla^{\mathrm{T}}_{\boldsymbol{\Theta}} + \mu \mathbf{I}\right)^{-1} \nabla_{\boldsymbol{\Theta}} \mathcal{L}_{\Theta}(\mathcal{S}_t) \tag{7}$$

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 26 October 2022 doi:10.20944/preprints202210.0391.v1

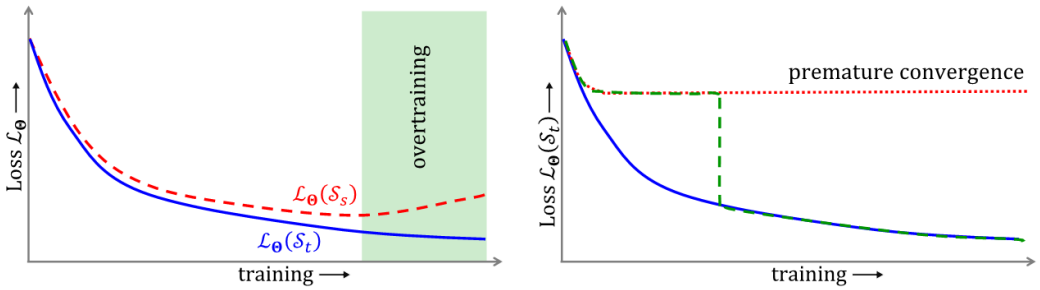


Figure 3. Overtraining and Premature Convergence. Overtraining is illustrated (left) showing how test loss (dashed red) begins to rise with overtraining (shaded green region) although training loss (solid blue) decreases. Premature convergence (right) of the training loss is shown (dotted red) in contrast to desired convergence (dashed green, solid blue). Due to "V" shaped narrow ridges in the loss function's landscape, there may no be any perceptible decrease in the loss for many training iterations (dashed green)

Overtraining - a problem that is closely related to overfitting, is frequently encountered during training. This is shown in Fig. 3. Since samples in $\mathcal{S}_t$ are used to compute the gradient, as long as η is small enough, the training loss decreases each time the parameters are incremented. Initially, the test loss $\mathcal{L}(\mathcal{S}_t)$ also drops with training. However, after the model has undergone a significant amount of training, $\mathcal{L}(\mathcal{S}_t)$ begins to rise. Applying (6) further will cause overtraining.

K-fold cross-validation [46] is an effective mean to prevent overtraining. The samples in $\mathcal{S}_t$ are randomly shuffled and split into K groups or folds of equal size. One of the folds is used as the test set $\mathcal{S}_s$ and the rest are used to increment $\mathbf{\Theta}$. This process is repeated K times, with each fold acting as the test set. The loss averaged over all K folds is a reliable estimate of its true (real-world) loss.

Premature convergence is another issue that is sometimes observed during training (see Fig. 3). This occurs if the training algorithm encounters a local minimum of the loss function's landscape, where the gradient $\nabla_{\mathbf{\Theta}}$ is very close to zero. Applying (6) or (7) would have little effect on the parameter $\mathbf{\Theta}$. A simple method to rectify the situation is to restart the training process from some other (randomly generated) initial point.

The presence of narrow ridges with "V"-shaped cross sections is another reason why the loss may remain unchanged (see Fig. 3), giving the appearance of a local minimum for several iterations. Although there is no perceptible drop in the loss, the amount of increment to $\mathbf{\Theta}$ is not negligible. Restarts are not necessary in such situations, for the algorithm eventually leaves such a ridge after multiple updates.

3.5. Optimization Metaheuristics

Existing training algorithms apply optimization metaheuristics, such as GA and PSO, to avoid getting trapped in local minima. These algorithms maintain a set of many candidate solutions, referred to as its population. In each step of the optimization algorithm, a new population is formed out of the existing one, using a variety of stochastic and heuristic search operators. Stochastic operators help the algorithm escape from local minima, while heuristics aid in its convergence towards the global maximum of the objective function.

GAs are useful in training CI regression models. Let the population of such a GA be the set $\{\mathbf{\Theta}^j | j = 1, 2, \cdots\}$, where each $\mathbf{\Theta}$ is a candidate model parameter. During each iteration, pairs of solutions are selected from the population in a random manner, but with better ones (in terms of the inverse loss function) being more likely to be picked. Using the crossover operator, a new pair of new solutions are generated from old ones. For example, in convex crossover, the existing pair $(\mathbf{\Theta}^i, \mathbf{\Theta}^j)$ can be used to generate a new pair, $(\mathbf{\Theta}^{i'}, \mathbf{\Theta}^{j'}) = \left(\mu\mathbf{\Theta}^i + (1-\mu)\mathbf{\Theta}^j, (1-\mu)\mathbf{\Theta}^i + \mu\mathbf{\Theta}^j\right)$.

In the mutation operator, a small amount of perturbation $\Delta\mathbf{\Theta}^i$ is added to each new candidate parameter, so that it becomes equal to $\mathbf{\Theta}^i + \Delta\mathbf{\Theta}^i$.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 26 October 2022 doi:10.20944/preprints202210.0391.v1

In Gaussian mutation, the perturbation $\Delta\Theta^i$ which follows a Gaussian distribution centered around the origin. This process is repeated many times until no further improvement can be found.

Although GA and PSO have found widespread use in many optimization applications, their use in machine-soil interaction studies is rather limited. GAs have been used during model training. In these cases, the GA is hybridized with (6), (7) or any other related method. Gradient descent steps can be incorporated in the GA in different ways. For instance, $\mathbf{\Theta}^i$ can be mutated into $\Theta^i + \Delta\Theta^i + \eta\nabla_{\mathbf{\Theta}_i}\mathcal{L}$, where $\Delta\Theta^i$ is the random perturbation and $\nabla_{\mathbf{\Theta}}\mathcal{L}$, the gradient of the training loss $\mathcal{L}$.

Similar hybrid techniques exist for PSO (cf. [47]). However, PSO has not been used in the existing literature on machine-soil interactions. A relatively unknown population based stochastic algorithm has been used in [48,49].

4. Current Computational Intelligence Models

4.1. Neural Networks

Neural network (NN) models have been routinely used in various regression applications since the mid eighties, wherever a significant amount of data is involved. Neural networks are layered structures consisting of an input layer, one or more intermediate layers, called hidden layers, and an output layer. Each layer comprises of elementary processing units or neurons. In a fashion resembling the mammalian cortex, the neurons in each hidden and output layer receive the outputs of those of the preceding layer as their inputs via weighted synaptic connections.

Fig. 4 shows the layout of a neural network with L layers. The indices of the input and output layers are 1 and L, where M and N are the number of neurons in the input and output layers. The vectors $\mathbf{x}$ ($\mathbf{x} \in \mathbb{R}^M$) and $\mathbf{y}$ ($\mathbf{y} \in \mathbb{R}^N$) denote the network's input and output.

The size of an NN can be written succinctly as $\prod_{l=1}^{L} N^{(l)}$ where $N^{(l)}$ is the number of neurons in layer l. For instance, a $3 \times 5 \times 6 \times 2$ NN has three input neurons, a hidden layer with five neurons, another hidden layer with six neurons, and two output neurons. Note that indices of layers (superscripts) are shown within parentheses so as not to confuse them with exponents.

Until recently, NNs were equipped with only one or two hidden layers (so that $L = 3$ or $L = 4$) - an approach used everywhere in the published literature on soil-machine interaction studies. In order to distinguish them from deep neural networks (DNNs) which have multiple hidden layers, models with only one hidden layer are referred to as shallow networks. However, for the purpose of this review, networks with two hidden layers is also included in this category. This section focuses on classical methods that are common to both shallow and deep networks. Advanced features that are relevant to DNNs are addressed separately in a subsequent section.

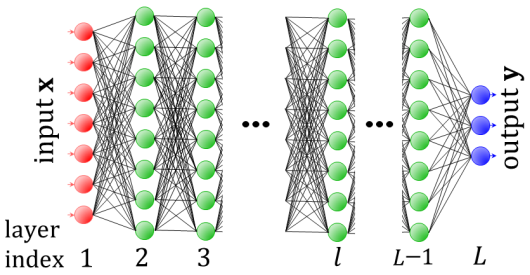


Figure 4. Neural Network. Neurons are depicted as small circles and synaptic connections as straight lines. The network has an input layer (red), hidden layers (green) and output layer (blue). Since the network shown has multiple hidden layers, it is a deep neural network.

The output of the k^{th} neuron in a layer indexed l ($l \in \{1 \cdots L\}$) is denoted as $y_j^{(l)}$. Thus the i^{th} element of $\mathbf{x}$ is $x_j = y_j^{(1)}$; similarly $y_j = y_j^{(L-1)}$. Fig. 4 shows all quantities associated

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 26 October 2022 doi:10.20944/preprints202210.0391.v1

with the k^{th} neuron in the layer l ($l > 1$). The neuron's input is the weighted sum of the outputs of all neurons in the preceding layer $(l-1)$ as shown in the following expression,

$$s_k^{(l)} = w_{k,0}^{(l)} + \sum_j w_{k,j}^{(l)} y_j^{(l-1)}. \tag{8}$$

The summation in (8) is carried out over the outputs $y_j^{(l-1)}$ of all neurons (indexed $j, j \geq 1$) of the previous layer, and the associated weight is $w_{k,j}^{(l)}$. The quantity $w_{k,0}^{(l)}$ is the neuron's bias. Fig. 5 shows a neuron in a hidden layer. The weights and biases are the trainable parameters of the neural network that are included in $\boldsymbol{\Theta}$.

Neurons in the input layer are linear elements; their role is merely to transmit the incoming vector to hidden neurons. However, those in the hidden layers, and optionally in the output layer as well, incorporate a monotonically increasing nonlinear function $f(\cdot)$, where either $f : \mathbb{R} \to (0,1)$ or $f : \mathbb{R} \to (-1,1)$, that is referred to as the activation function. The output of the neuron is,

$$y_k^{(l)} = f\left(s_k^{(l)}\right). \tag{9}$$

The logistic function $\sigma(s) = (1 + \exp(-s))^{-1}$ and the hyperbolic tangent function ($\tanh(\cdot)$) are the most commonly used activation functions used in shallow networks. Due to their characteristic 'S' shapes, such activation nonlinearities fall under the category of sigmoid functions.

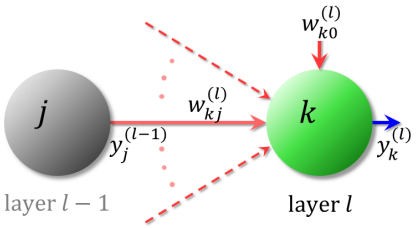


Figure 5. Neuron. Quantities associated with a neuron (green circle). Also shown is a neuron in the preceding layer (grey circle)

Historically, the popularity of NNs surged with the introduction of the back-propagation (BP) algorithm [50], which is a reformulation of SGD designed to train layered structures. The error $\delta_k^{(l)}$ of the k^{th} neuron in the l^{th} layer is defined as the derivative of the penalty term $l_{\boldsymbol{\Theta}}$ (in the loss $\mathcal{L}_{\boldsymbol{\Theta}}$) with respect to the neuron's input $s_k^{(l)}$ (see Eqn. (8)). Such penalties can be readily differentiated for neurons in the output layer ($l = L$). The back-propagation rule shows how $\delta_k^{(l)}$ can be computed for hidden neurons ($l < L$), using the errors of the next layer $l+1$. The schematic in Fig. 6 illustrates how errors back-propagate. The general expression to compute the errors is,

$$\delta_k^{(l)} = \begin{cases} \frac{\partial}{\partial s_k^{(L)}} l_{\boldsymbol{\Theta}}, & \text{if } l = L; \\ \sum_j w_{kj}^{(l+1)} \delta_j^{(l+1)}, & \text{otherwise.} \end{cases} \tag{10}$$

The weights in $\boldsymbol{\Theta}$ can be updated in the following manner,

$$w_{kj}^{(l)} = w_{kj}^{(l)} + \eta y_j^{(l-1)} \delta_k^{(l)}. \tag{11}$$

It is common practice to include a momentum term to BP. Additionally, BP can be extended to apply Levenberg-Marquardt updates. This is the Levenberg-Marquardt BP (LMBP) algorithm [51].

The VC-dimensionality of a neural network is typically specified in terms of the total number of weights and biases [52]. The number of training samples should be about ten

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 26 October 2022 doi:10.20944/preprints202210.0391.v1

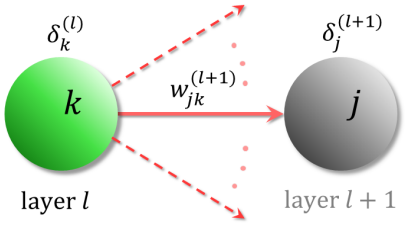


Figure 6. Back-Propagation. Back-propagation of errors. (solid circle).

times this quantity. The number of epochs to achieve training is independent of the data size.

The radial basis function network (RBFN) [53,54] is another popular computational intelligence regression model that is topologically identical to an $M \times K \times 1$ neural network. In other words, an RBFN has M input neurons, a single hidden layer of K neurons, and only one output neuron. The sole purpose of the input layer, which contains M linear neurons, is to pass on M dimensional inputs to the hidden layer. The K neurons in the hidden layer are incorporated with nonlinear activation functions. The output neuron computes the weighted summation of the outputs from the hidden layer. Due to their strong resemblance to shallow neural networks, RBFNs are sometimes classified as NNs. RBFs have been successfully used in agricultural applications [55–57].

Unlike in NNs, the hidden neurons of the RBFN are designed to produce localized responses. The activation function of any hidden neuron has an M dimensional parameter called its center. The closer an input is to its center, the higher is the neuron's output. In this manner the network's hidden neurons simulate sensory cells of the peripheral nervous system, which have localized receptive fields.

Suppose $\mathbf{x}$ ($\mathbf{x} \in \mathbb{R}^M$) is the network's input. Each hidden neuron k (where $k \in \{1 \cdots K\}$) receives $\mathbf{x}$ from the input layer, and produces an output $f(||\mathbf{x} - \mathbf{c}_k||)$, where $||\cdot||$ denotes a vector norm operator (e.g. length). Gaussian nonlinearities are the most widely used activation functions. In this case, the output of the k^{th} hidden neuron, denoted as ϕ_k, is obtained according to the following expression,

$$\phi_k = e^{\frac{1}{\sigma_k}||\mathbf{x}-\mathbf{c}_k||^2} \tag{12}$$

The quantity σ_k in (12) is an optional width parameter of the k^{th} hidden neuron. When dealing with training samples that are distributed evenly within the input space, all hidden neurons may be assigned the same widths σ.

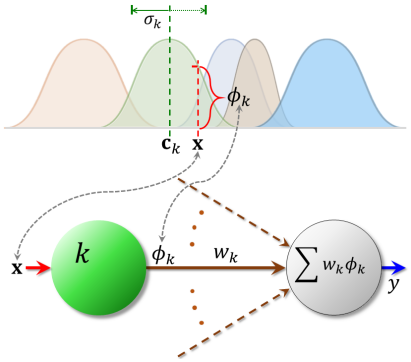


Figure 7. Radial Basis Function. The

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 26 October 2022 doi:10.20944/preprints202210.0391.v1

With w_k ($k \in 1 \cdots K$) being network weights, the RBFN's output y is the weighted sum $\sum_k w_k \phi_k$. Using (12), y can be expressed directly in terms of the input $\mathbf{x}$ as,

$$y = \sum_k w_k e^{\frac{1}{\sigma_k}||\mathbf{x}-\mathbf{c}_k||^2} \tag{13}$$

Fig. 7 depicts the main quantities of an RBFN.

The RBFN's parameters in $\mathbf{\Theta}$ are all its weights w_k and centers $\mathbf{c}_k$. If necessary to train the network's widths σ_k, they are also included in $\mathbf{\Theta}$. Due to the use of localized activation functions, the number of hidden neurons K required by the RBFN increases exponentially with the input dimensionality M. Hence the effectiveness of RBFNs is limited to tasks involving low dimensional data (up to $M = 6$ or 7). Even in such tasks, RBFNs require significantly more number of hidden neurons than NNs. As the trade-off for this limitation, RBFNs offer faster training, often by a few orders of magnitude. This speedup over Eqn. (6) is achieved when the centers, widths, and weights are trained separately [58].

A popular method to train the centers of the hidden neurons is by using K-means clustering [59]. For each hidden neuron k, a subset $\mathcal{N}_k$ of samples in the training set $\mathcal{S}_t$ is obtained. This subset consists of all samples in that are closer to the neuron's center $\mathbf{c}_k$ than to $\mathbf{c}_{k'}$ of any other neuron k', $k' \neq k$. The center of each hidden neuron is made equal to the average of all samples $\mathbf{x}(n)$ in $\mathcal{N}_k$. The two steps can be expressed as shown below,

$$\begin{aligned} &\mathbf{c}_k \leftarrow \frac{1}{|\mathcal{N}_k|} \sum_{n \in \mathcal{N}_k} \mathbf{x}(n), \quad \text{where,} \\ &\mathcal{N}_k = \{n \in \mathcal{S}_t | \, k' \neq k, ||\mathbf{x}(n) - \mathbf{c}_k|| < ||\mathbf{x}(n) - \mathbf{c}_{k'}||\} \end{aligned} \tag{14}$$

A relatively small number of iterations of (14) is enough to train the centers of all hidden neurons. Their widths can be fixed at some constant value such that $\sigma_k = \sigma$, $k \in \{1 \cdots K\}$. Alternately, a nearest neighbor heuristic can be applied to determine each σ_k separately, such as,

$$\sigma_k = c \underset{k' \neq k}{\text{argmin}} ||\mathbf{c}_k - \mathbf{c}_{k'}|| \tag{15}$$

The quantity c in (15) is an algorithmic constant.

For the $\mathcal{L}_1$ or the Hüber loss functions, the weight parameters $\mathbf{w}_k$ must be trained in an iterative manner using (6). When the $\mathcal{L}_2$ loss is used, the Moore-Penrose pseudoinverse formula provides a simpler method to obtain the weights. Let $\mathbf{w} \in \mathbb{R}^K$ be the vector of all weights. Similarly, let $\boldsymbol{\phi}(n) \in \mathbb{R}^K$ ($n \in \mathcal{S}_t$) be the vector of outputs of the hidden neurons, determined using (12)) with input $\mathbf{x}(n)$. It can be seen that the RBFN's output is $y(n) = \boldsymbol{\phi}^{\mathrm{T}}(n)\, \mathbf{w}$ (where $\cdot^{\mathrm{T}}$ is the transpose operator).

To see how the pseudoinverse formula works, let us construct an activation matrix $\mathbf{\Phi} \in \mathbb{R}_+^{|\mathcal{S}_t| \times K}$, whose n^{th} row is $\boldsymbol{\phi}^{\mathrm{T}}(n)$. Whence $\mathbf{y} = \mathbf{\Phi}\mathbf{w}$ is the $|\mathcal{S}_t| \times 1$ vector of all outputs of the RBFN. If $\mathbf{t} \in \mathbb{R}^{|\mathcal{S}_t|}$ is the corresponding vector of all target values, the mean squared $\mathcal{L}_2$ loss is the expression $\mathcal{L}_2 = ||\mathbf{y} - \mathbf{t}||^2$. The weight vector that minimizes this loss is $\underset{\mathbf{w}}{\text{argmin}} ||\mathbf{\Phi}\mathbf{w} - \mathbf{t}||^2$. If the number of training samples is more than the number of hidden neurons, (i.e. $|\mathcal{S}_t| > K$), which is always the case in a real application, the matrix $\mathbf{\Phi}^{\mathrm{T}}\mathbf{\Phi}$ is non-singular. In this case, the expression for the loss minimizing weight vector $\mathbf{w}$ is determined as,

$$\mathbf{w} = \left(\mathbf{\Phi}^{\mathrm{T}}\mathbf{\Phi}\right)^{-1} \mathbf{\Phi}^{\mathrm{T}}\mathbf{t} \tag{16}$$

The factor $\left(\mathbf{\Phi}^{\mathrm{T}}\mathbf{\Phi}\right)^{-1}\mathbf{\Phi}^{\mathrm{T}}$ in (16) is a matrix of size $K \times |\mathcal{S}_t|$. It is referred to as the pseudoinverse of $\mathbf{\Phi}$ and denoted as $\mathbf{\Phi}^{+}$.

In theory, all RBFN parameters can be trained iteratively using gradient descent. Although training the RBFN parameter vectors $[\mu_k]$ and $[\sigma_k]$ in this manner is fairly uncommon, gradient descent is often used to train the weight vector $\mathbf{w}$. This is carried out as in

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 26 October 2022 doi:10.20944/preprints202210.0391.v1

Eqn. (6), with $\boldsymbol{\Theta}$ replaced with $\mathbf{w}$. This method is applied to avoid numerical issues with matrix pseudoinversion, as well as wherever the training algorithm is not based on the mean squared loss function.

Recent RBFN models use multivariate Gaussians distributions, where Eqn. (12) is replaced with,

$$\phi_k = e^{\frac{1}{2}(\mathbf{x}-\mathbf{c}_k)^{\mathrm{T}}\boldsymbol{\Sigma}_k(\mathbf{x}-\mathbf{c}_k)} \tag{17}$$

In the above expression, the quantity $\boldsymbol{\Sigma}_k \in \mathbb{R}^{M\times M}$ is a covariance matrix.

4.2. Support Vector Regression

SVRs are another class of CI models [60,61] that are widely used in various engineering and other applications. [62]. SVRs have been used for regression applications in agriculture [63–66]. Unlike the other CI models discussed earlier, SVRs do not have any strong parallels in nature. Instead, they are specifically aimed at addressing the issue of model complexity, which is addressed below.

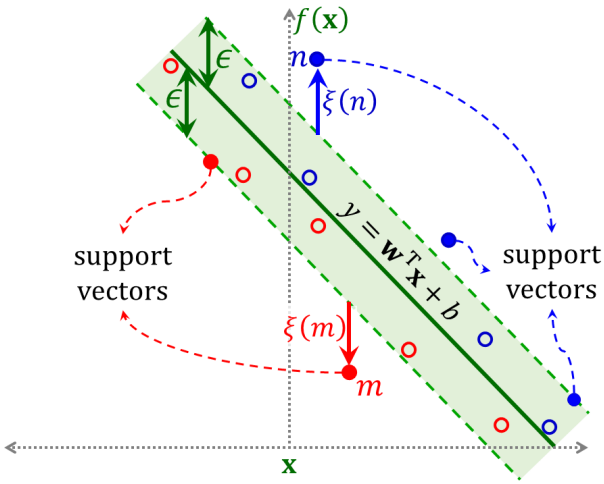


Figure 8. Linear Support Vector Regression. The regression line $y = \mathbf{w}^{\mathrm{T}}\mathbf{x} + b$ (solid green), and the ϵ region (shaded green) of zero penalty around it, are shown. Also shown are samples (small circles) including the support vectors (filled circles) indexed m and n.

The simplest formulation is the linear SVR with ϵ-loss, as shown in Fig. 8. Sample targets that lie within a margin of $\pm\epsilon$ from the regression line do not incur any penalty, while those outside the margin incur penalties. Hence the error arising from a sample pair $(\mathbf{x}(n), t(n))$ is obtained as shown in (4). Denoting this error as $\xi(n)$, it can be readily established that the following constraints are satisfied,

$$\begin{cases} \xi(n) \geq 0 \\ t(n) - \mathbf{w}^{\mathrm{T}}\mathbf{x}(n) - b \leq \epsilon + \xi(n) \\ \mathbf{w}^{\mathrm{T}}\mathbf{x}(n) + b - t(n) \leq \epsilon + \xi(n) \end{cases} \tag{18}$$

When the above conditions are satisfied, the loss is simply the sum of all errors, $\sum_n \xi(n)$.

It has been shown that the gap between the validation and training losses (i.e. $\mathcal{L}(\mathcal{S}_v) - \mathcal{L}(\mathcal{S}_t)$) can be lowered by increasing ϵ, or alternately by decreasing $||\mathbf{w}||_2$ while $\pm\epsilon$ is a constant [67]. This term can be recognized as the LASSO regularizer.

With C being an algorithmic constant, the optimal regression model can be obtained as the solution of the following constrained optimization problem,

$$\begin{cases} \min\limits_{\mathbf{w},b,\boldsymbol{\xi}} & \frac{1}{2}\mathbf{w}^{\mathrm{T}}\mathbf{w} + C\sum_{n\in\mathcal{S}_t}\xi(n) \\ \text{s.t.} & (18) \quad \text{is true} \end{cases} \tag{19}$$

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 26 October 2022 doi:10.20944/preprints202210.0391.v1

The above problem (19) to obtain of the SVR is in primal form. Classical optimization theory (cf. [68,69]) illustrates that for every primal problem, a dual problem can be constructed using the Lagrange multipliers of the primal constraints as its variables. Optimization theory establishes that under certain constraint qualification, the optima of the primal and dual problems coincide at a saddle point. The dual form of (19) can be derived readily [67]. Ignoring the constraints $\xi(n) \geq 0$ and using the symbols $\breve{}_+ \in \mathbb{R}_+^{|\mathcal{S}_t|}$ and $\breve{}_- \in \mathbb{R}_+^{|\mathcal{S}_t|}$ as the Lagrange multiplier vectors of the other constraints in (18). the dual problem can be formulated in the following manner,

$$\begin{cases} \min\limits_{\lambda_+,\lambda_-} & \frac{1}{2}(\breve{}_+ - \breve{}_-)^{\mathrm{T}}\mathbf{K}(\breve{}_+ - \breve{}_-) + \breve{}_+^{\mathrm{T}}(\epsilon\mathbf{1} - \mathbf{t}) + \breve{}_-^{\mathrm{T}}(\epsilon\mathbf{1} + \mathbf{t}) \\ \text{s.t.} & \mathbf{1}^{\mathbf{T}}(\breve{}_+ - \breve{}_-) = 0 \\ & \mathbf{0} \leq \breve{}_+, \breve{}_+ \leq C\mathbf{1} \end{cases} \tag{20}$$

The element in the m^{th} row and n^{th} column of the symmetric matrix $\mathbf{K} \in \mathbb{R}_+^{|\mathcal{S}_t|}$ in (20) is $\mathbf{x}(m)^{\mathrm{T}}\mathbf{x}(n)$. The bias b as well as the normal vector $\mathbf{w}$ can be obtained from the dual solution, although $\mathbf{w}$ is not required.

In more generalized settings, input samples can lie in any arbitrary Hilbert space. The inner product of the m^{th} and n^{th} samples is represented as $\langle \mathbf{x}(m), \mathbf{x}(n) \rangle$. The matrix $\mathbf{K}$ will contain pairwise inner products of such samples.

Nonlinear SVRs implicitly apply a transformation $\phi(\cdot)$ from the input space $\mathcal{S}$ to an unknown Hilbert space [60]. Under these circumstances, the $(m,n)^{\text{th}}$ element of $\mathbf{K}$, which we now denote as $K(\mathbf{x}(m), \mathbf{x}(n))$, is obtained as provided below,

$$K(\mathbf{x}(m), \mathbf{x}(n)) = \langle \phi(\mathbf{x}(m)), \phi(\mathbf{x}(n)) \rangle \tag{21}$$

The function $K : \mathcal{S} \times \mathcal{S} \to \mathbb{R}_+$ is referred to as the kernel function. Mercer's theorem states that as long as the kernel satisfies a few conditions, there must exist some transformation $\phi : \mathcal{S} \to \mathbb{H}$ satisfying (21). As long as these conditions are met, the matrix $\mathbf{K}$ obtained from every possible sample set, will be symmetric and positive definite. In other words, kernel functions can be devised without even considering the mapping $\phi(\cdot)$; this mapping along with its range in Hilbert space $\mathbb{H}$ can remain unknown. This is a remarkable feature of SVR models. In engineering applications, any symmetric, non-negative measure of similarity between pairs of samples can be adopted as the kernel function. For instance, Gaussian kernels $e^{\frac{1}{\sigma}||\mathbf{x}(m)-\mathbf{x}(n)||^2}$, or L_p-normed kernels $e^{\frac{1}{\sigma}||\mathbf{x}(m)-\mathbf{x}(n)||_p}$ can be adopted for inputs that lie in a Euclidean space $\mathbb{R}^M$. In bio-informatics, where samples may consist of DNA strands that are sequences of the letters C, T, G, and A, the kernel may vary negatively with the minimum edit distance between every pair of samples.

For each sample $\mathbf{x}(n)$ that is strictly within the $\pm\epsilon$ margin of the regression line (see Fig. 8), the corresponding dual variables $\lambda_\pm(n)$ obtained from (20) will be zeros. It is only when the sample lies either on the margin's boundaries or outside it, that yield either $\lambda_+(n) > 0$ or $\lambda_-(n) > 0$. These samples are the support vectors. The set of all support vectors is,

$$\mathcal{V} = \{n | \lambda_+(n) > 0 \text{ or } \lambda_-(n) > 0\}. \tag{22}$$

Given an unknown sample $\mathbf{x}$, the estimated output y can be obtained using the kernels of $\mathbf{x}$ and the support vectors in $\mathcal{V}$,

$$y = \sum_{n \in \mathcal{V}} (\lambda_+(n) - \lambda_-(n)) K(\mathbf{x}(n), \mathbf{x}) + b. \tag{4.14}$$

Although not provided in this article, the bias b can be obtained readily from the dual form in (20).

As long as the training set $\mathcal{S}_t$ is small enough so that it is computationally feasible to compute the matrix $\mathbf{K}$ and store in memory, quadratic programming can be applied directly to solve (20). Otherwise, there are a plethora of iterative training algorithms [70–72], that

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 26 October 2022 doi:10.20944/preprints202210.0391.v1

are well-equipped to train SVRs with larger data sets. SVRs can be formulated using other losses and regularizers as well.

4.3. Fuzzy Inference Systems

FIS are CI models that are inspired by decision making processes in humans. In order to capture the inherent vagueness in human verbal reasoning, the make use of fuzzy sets. Fuzzy set theory extends the classical concept of a set (called a 'crisp' set in fuzzy terminology) by incorporating such imprecision. The manner in which it does so is described next.

Any element x from the universe of discourse $\mathbb{U}$ can either be in a given crisp set A, where $A \subset \mathbb{U}$ (i.e. $x \in A$) or not in it (i.e. $x \notin A$). Accordingly, a binary membership function $\mu_A : \mathbb{U} \to \{0,1\}$ can be defined such that $\mu_A(x) = 1$ iff $x \notin A$, otherwise $\mu_A(x) = 0$ iff $x \notin A$. The membership function of a fuzzy set A is allowed to have any real value within the interval $[0,1]$, i.e. $\mu_A : \mathbb{U} \to [0,1]$. The numerical value of $\mu_A(x)$ indicates the degree to which x is included in A. For example, let T be the set of tall students in a class. If T is a crisp set, there must be a minimum cutoff for tallness. Let this cutoff is 5'10". Hence, Jack and Jill whose heights are 5'9" and 6'1" have memberships $\mu_T(\text{Jack}) = 0$, and $\mu_T(\text{Jill}) = 1$ in . On the other hand, if T is a fuzzy set, then memberships such as $\mu_T(\text{Jack}) = 0.7$, and $\mu_T(\text{Jill}) = 0.99$ are possible, indicating that Jack is very close to being tall, whereas Jill is definitely tall.

When the universe of discourse is a continuous variable, memberships can be defined in terms of functions of real arguments $\mu : \mathbb{R} \to [0,1]$. The Gaussian, trapezoidal, and triangular functions are commonly used for memberships. The Gaussian membership of a scalar input x to the fuzzy set $A \subset \mathbb{U}$ is $e^{-(x-\mu)/\sigma}$. The trapezoidal membership can be defined using four parameters, a, b, c, and d ($a \leq b < c \leq d$),

$$\mu_A(x) = \begin{cases} 0, & \text{if } x < a; \\ \frac{x-a}{b-a}, & \text{if } a \leq x < b; \\ 1, & \text{if } b \leq x < c; \\ \frac{d-x}{d-c}, & \text{if } c \leq x < d; \\ 0, & \text{if } d \leq x. \end{cases} \tag{23}$$

The triangle membership function requires only three parameters, a, b, and c ($a \leq b \leq c$),

$$\mu_A(x) = \begin{cases} 0, & \text{if } x < a; \\ \frac{x-a}{b-a}, & \text{if } a \leq x < b; \\ \frac{c-x}{c-b}, & \text{if } b \leq x < c; \\ 0, & \text{if } c \leq x. \end{cases} \tag{24}$$

Gaussian memberships as well as those in (23) and (24) have peak values of unity. Although this is common practice in real world applications, fuzzy sets can also admit any other membership function as long as its maximum lies anywhere in $(0,1]$. The complement $\bar{A}$ of the fuzzy set A can be readily defined in terms of the membership function as, $\mu_{\bar{A}} = 1 - \mu_A$. A fuzzy singleton - say B, is a fuzzy set that is fully parametrized by a constant v_B, where $v_B \in \mathbb{R}$ such that for the input $y \in \mathbb{R}$,

$$\mu_B(y) = \begin{cases} 1, & \text{if } y = v_B; \\ 0, & \text{otherwise.} \end{cases} \tag{25}$$

The operations of union ($\cup$) and intersection ($\cap$) in crisp sets correspond to conjunction (AND) and disjunction (OR) in Boolean algebra. In terms of membership functions, the union $A \cup B$ and intersection $A \cap B$ of the sets A and B are $\mu_{A \cup B} = \mu_A \,\text{OR}\, \mu_A$, and $\mu_{A \cap B} = \mu_A \,\text{AND}\, \mu_A$. Union and intersection of fuzzy sets can be realized in various ways [73], using t-conorms and t-norms. A popular choice is to use $\max(\cdots)$ as the t-conorm operator and $\min(\cdots)$ as the t-norm. In this case, $\mu_{A \cup B} = \max\{\mu_A, \mu_B\}$ and $\mu_{A \cap B} = \min\{\mu_A, \mu_B\}$.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 26 October 2022 doi:10.20944/preprints202210.0391.v1

In our previous example, suppose S is the fuzzy set of smart students and $\mu_S(\text{Jill}) = 0.75$, then $\mu_{S\cup T}(\text{Jill}) = \max\{0.75, 0.99\} = 0.99$ and $\mu_{S\cap T}(\text{Jill}) = \min\{0.75, 0.99\} = 0.75$.

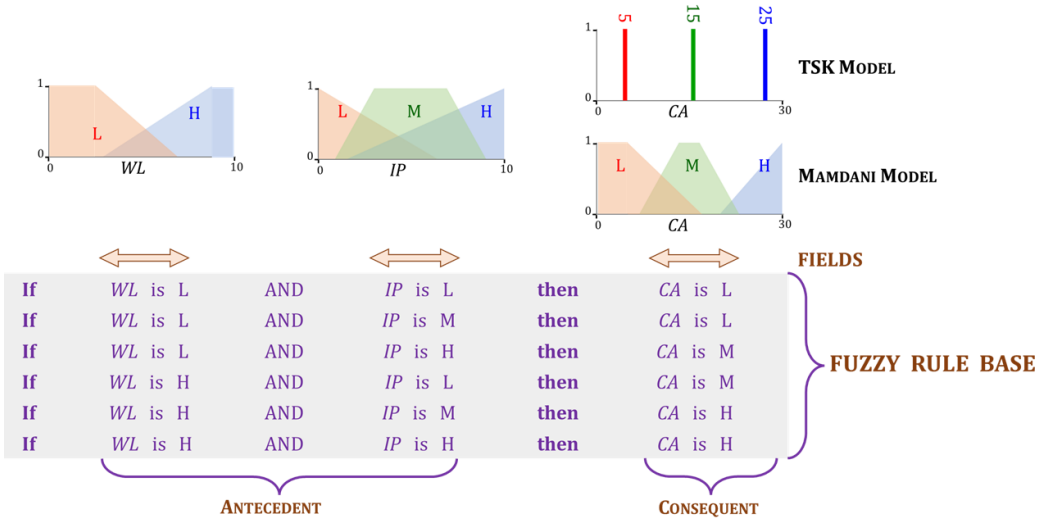


Figure 9. Fuzzy Inference System. The figure shows membership functions (top), and fuzzy rule base (bottom). The input **x** has two elements, $WL \in [0, 10]$ (wheel load) that can be L (Low), M (Medium) or H (High), and $IP \in [0, 10]$ (inflation pressure) that can be L (Low) or H (High). The output $y \in [0, 30]$ is a scalar ($N = 1$). This is the quantity CA (contact area), which can be L (Low), M (Medium), or H (High). The membership functions, $\mu_{\text{L}}(CA)$, $\mu_{\text{M}}(CA)$, and $\mu_{\text{H}}(CA)$ are trapezoids/triangle (Mamdani) or singletons (TSK).

A FIS encapsulates human knowledge through a fuzzy rule base. Each rule in the base consists of two parts, an antecedent and a consequent; and is written in the format, "**If** ANTECEDENT **then** CONSEQUENT". If the input to the model is an M-dimensional vector **x** and its output is an M-dimensional vector **y**, the antecedents and consequents are made up of M and N fields. The generic format of a rule with index $k \in 1, 2, \cdots, K$ is as shown below,

$$\textbf{If} \underbrace{x_1 \text{ is } A_1^k \diamond x_2 \text{ is } A_2^k \cdots \diamond x_M \text{ is } A_M^k}_{\text{ANTECEDENT}} \textbf{ then } \underbrace{y_1 \text{ is } B_1^k \cdots \diamond y_N \text{ is } B_N^k}_{\text{CONSEQUENT}}. \tag{26}$$

Each diamond symbol ($\diamond$) in (26) represents an AND or an OR operator.

The order in which these operations are applied may either be in accordance with an established convention, or alternately, specified explicitly by inserting brackets at appropriate places. Mathematically speaking, the j^{th} field in the antecedent of the fuzzy rule in Eqn. (26), "x_j is A_j" is the membership, $\mu_{A_j}(x_j)$. In a similar fashion, the i^{th} field in the consequent is $\mu_{B_i}(y_i)$. Fig. 9 shows a simple rule base with $K = 6$ rules.

There are two kinds of FIS, differing only in the way the sets B_i in the consequent's i^{th} field ($i \in \{1, 2, \cdots, N\}$) are defined. In the Mamdani model [74], they are allowed to be fuzzy sets. As a result of this flexibility, a Mamdani FIS can easily apply verbal descriptions in the consequents. On the other hand, in the Takagi-Sugeno-Kang (TSK) model [75,76], each B_i must be a singleton as in Eqn. (25). A TSK model is used to render the FIS more amenable to mathematical treatment. Fig. 9 shows examples of the Mamdani and TSK models.

The various steps involved in mapping an input to its output, will be illustrated using the examples shown in Fig. 10 (Mamdani model) and Fig. 11 (TSK model). The steps are briefly described below.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 26 October 2022 doi:10.20944/preprints202210.0391.v1

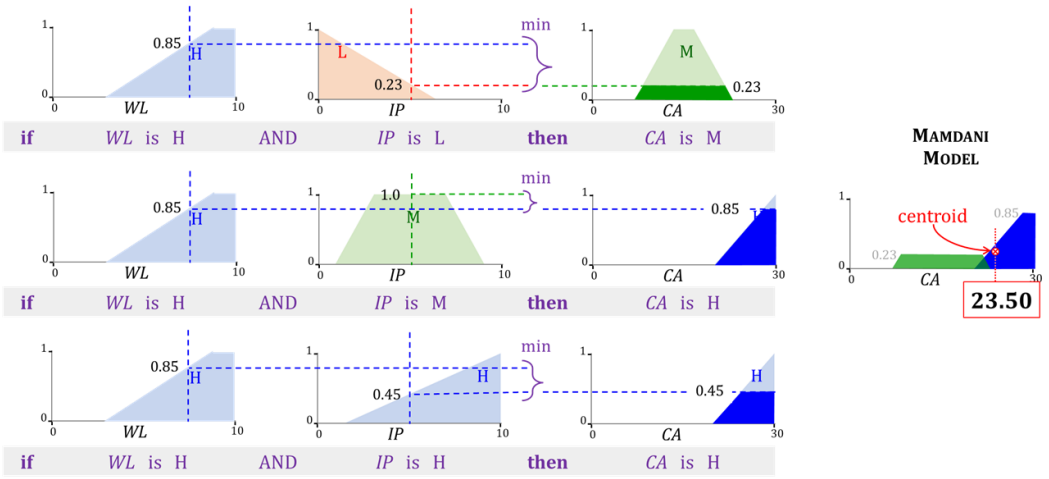


Figure 10. Mamdani FIS. The inputs to the FIS in 9 are $WL = 7.5$ and $WL = 5.0$ (dotted vertical lines), and the output is $y = 23.50$). The first three rules in 9 with $WL = \text{H}$ are ignored since $\mu_{\text{L}(7.5)=0)}$. The dark shaded regions are $\mathcal{R}^k$ of the relevant rules.

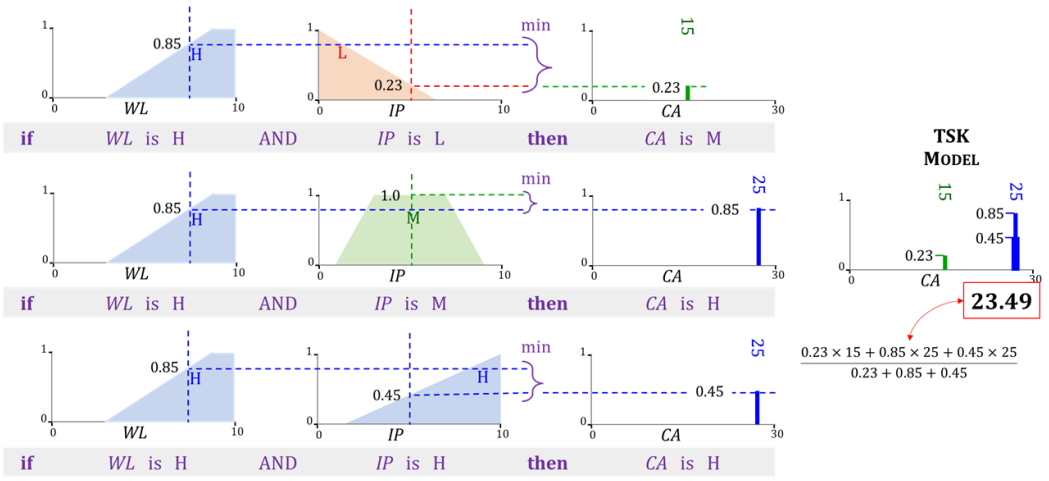


Figure 11. TSK FIS. The inputs to the FIS in 9 are $WL = 7.5$ and $WL = 5.0$ (dotted vertical lines), and the output is $y = 23.49$. The first three rules in 9 with $WL = \text{H}$ are ignored since $\mu_{\text{L}}(7.5) = 0$). In the other rules, the values of v_{B^k} are 15, 25, and 25.

(i) Fuzzification : This step is carried out separately in each antecedent field "x_j is A_j^k" and for each rule k. It involves computing the values of the memberships $\mu_{A_j^k}(x_j)$ using the numerical values of the input element x_j.

(ii) Aggregation : In this step, AND and OR operations are applied as appropriate, to each rule in the FIS. The rules in the FIS shown in Fig. 10 and Fig. 11 only involve conjunctions (AND) that are implemented through the min$(\cdot)$ t-norm. The aggregated membership is referred to as its rule strength. The strength of rule k is,

$$\mu_A^k = \bigcup_j \mu_{A_j^k}(x_j). \tag{27}$$

(iii) Inference : The strength of each rule is applied to its consequent. Each rule k in our example contains only one consequent field. Its membership function μ_{B^k} is limited to a maximum of μ_{A^k}. For every rule k in k, a two dimensional region $\mathcal{R}^k$ is identified in

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 26 October 2022 doi:10.20944/preprints202210.0391.v1

the Mamdani model. Since the TSK model involves only singletons at this step, only a two-dimensional point $\mathcal{R}^k$ is necessary. Accordingly,

$$\mathcal{R}^k = \begin{cases} \{(y^k, z^k) | y^k \in [0, y_{max}],\ z^k \in [0, \max(\mu_{B^k}(y), \mu_{A^k})]\}, & \text{Mamdani;} \\ (v_{B^k}, \mu_{A^k}), & \text{TSK.} \end{cases} \tag{28}$$

In the example shown in Fig. 10, the upper limit $y_{max} = 30$.

(iv) Defuzzification : The value of the FIS's output is determined in the last step. The Mamdani FIS in Fig. 10 uses the centroid method of defuzzification. The regions $\mathcal{R}^k$ are unified into a single region $\mathcal{R}$. The x-coordinate of the centroid of $\mathcal{R}$ is the final output. The TSK model in Fig. 11 uses a weighted sum to obtain the output y of the FIS. Mathematically,

$$y = \begin{cases} [\int_{\mathcal{R}} d\mathcal{R}]^{-1} \int_{\mathcal{R}} z^k \, d\mathcal{R}, & \text{Mamdani;} \\ [\sum_k \mu_{A^k}]^{-1} \sum_k v_{B^k} \mu_{A^k}, & \text{TSK.} \end{cases} \tag{29}$$

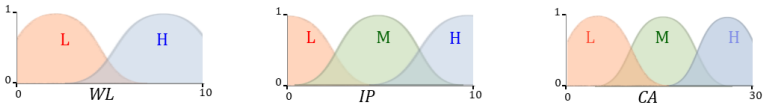
In the above expression, $\mathcal{R} = \bigcup_k \mathcal{R}$. It is evident from the above description, that the inference and defuzzification step in a Mamdani FIS is more computationally intensive in comparison to that in the TSK model. There are several other methods to obtain the output of a FIS. For details, the interested reader is referred to [77,78]. The Mamdani model [79–81] as well as the TSK model [82–86] have been used frequently in agricultural research.

4.4. Adaptive Neuro-Fuzzy Inference Systems

A TSK model with fuzzy rules as in (26) is often referred to as a zero order FIS. An ANFIS is a based on a zero or higher order TSK model, that is arranged in a manner resembling an NN [87–89]. ANFIS models frequently use first order TSK rule bases. Assuming a scalar output y, the format of the k^{th} rule in such a first order TSK model is,

$$\textbf{If } x_1 \text{ is } A_1^k \diamond \cdots \diamond x_M \text{ is } A_M^k \textbf{ then } y = b_0^k + b_1^k x_1 + \cdots + b_M^k x_M. \tag{30}$$

The consequent in Eqn. (30) is a linear expression for y in terms of $\mathbf{x}$, with $M \times K$ coefficients, b_j^k (where $j \in 1, \cdots, M$ and $k \in 1, \cdots, K$). To simplify its training, the membership functions in the ANFIS rules' antecedents are usually restricted to Gaussians [89]. Fig. 12 shows an example of a first order TSK model.



If	*WL*	is L	AND	*IP*	is L	**then**	$CA = 5WL + 5IP$
If	*WL*	is L	AND	*IP*	is M	**then**	$CA = 8WL + 7IP$
If	*WL*	is L	AND	*IP*	is H	**then**	$CA = 10WL + 10IP$
If	*WL*	is H	AND	*IP*	is L	**then**	$CA = 5WL + 5IP + 10$
If	*WL*	is H	AND	*IP*	is M	**then**	$CA = 8WL + 7IP + 10$
If	*WL*	is H	AND	*IP*	is H	**then**	$CA = 10WL + 10IP + 10$

Figure 12. Type 1 TSK FIS. Shown are the membership functions of the fields (top) and the fuzzy rule base (bottom). This antecedents of the rules are the same as those in Fig. 9.

Fig. 13 illustrates the ANFIS corresponding to the first order TSK rule set shown in Fig. 12. The parameters of the membership functions of each input variable are trainable quantities. For Gaussian memberships, they are σ_j^k, μ_j^k, where $j \in \{1, \cdots, M\}$, and k is the index of a rule). The coefficients in the consequent side of each such rule, which are b_j^k, $j \in \{0, 1, \cdots, M\}$ are also trainable. All trainable quantities constitute the parameter vector $\boldsymbol{\Theta}$ of the ANFIS.

There are five layers in the ANFIS model, which are as follows.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 26 October 2022 doi:10.20944/preprints202210.0391.v1

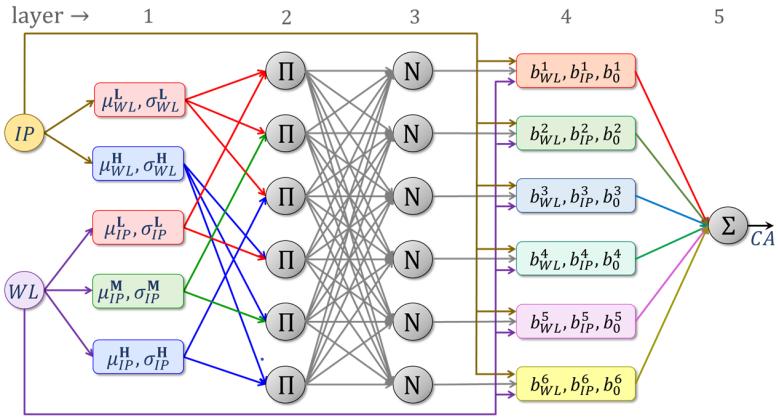


Figure 13. Adaptive Neuro-Fuzzy Inference System.

(i) Fuzzifying layer: The role of the first layer is to fuzzify scalar elements $x_j, j \in \{1 \cdots M\}$ of the input $\mathbf{x}$. It involves computing the memberships $\mu_{A_j^k}(x_j)$ in (30).

(ii) Aggregating layer: This layer performs aggregation. When all $\diamond$ operators in (30) are conjunctions, the output of the k^{th} unit in the second layer is obtained using the expression,

$$\mu_A^k = \prod_j \mu_{A_j^k}(x_j). \tag{31}$$

(iii) Normalizing layer: This is the third layer of the ANFIS, whose role is to normalize the incoming aggregated memberships, μ_A^k from the previous layer. The output of its k^{th} unit is,

$$\hat{\mu}_A^k = \frac{\mu_A^k}{\sum_{k'} \mu_A^{k'}}. \tag{32}$$

(iv) Consequent layer: The output of the k^{th} unit of the fourth layer is,

$$y^k = \hat{\mu}_A^k \left(b_0^k + \sum_j b_j^k x_j \right). \tag{33}$$

(v) Output layer: The final layer of the ANFIS performs summation of the consequent outputs y^k,

$$y = \sum_k y^k. \tag{34}$$

The quantity y is the output of the ANFIS.

Several methods have been proposed to train the parameters of an ANFIS model [90]. Much research has been directed towards gradient descent approaches ((6)) resembling BP [88,91]. Such approaches ave been used in agriculture [92–94]. A Levenberg-Marquardt approach has been suggested recently [95]. Stochastic metaheuristics such as GAs [96] and PSO [97] have also been investigated. Hybrid approaches combining them are a widely used to train ANFISs [98]. A comparison of three metaheuristics has been reported in [99] for an agriculture related application.

5. Soil-Machine Interaction studies: A brief survey

In recent decades, CI methods have been extensively studied in agriculture and a numerous applications can be found: crop management, insect-pest management, irrigation scheduling, precision agriculture, input application optimization, yield prediction, and so on [100]. However, this study is only focused on the soil-machine interaction centered around traction, tillage, and soil compaction. Initially, the research articles were collected from the multiple online database: Web of Science, Scopus, Science Direct, Google Scholar,

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 26 October 2022 doi:10.20944/preprints202210.0391.v1

Wiley, and Springer Link. A more than 150 research articles were collected in preliminary screening stage. Out of the 150 articles, only 50 articles directly related to the CI application on traction, tillage, and compaction were selected. Figure 14 showed the year-wise and categorical distribution of the selected articles.

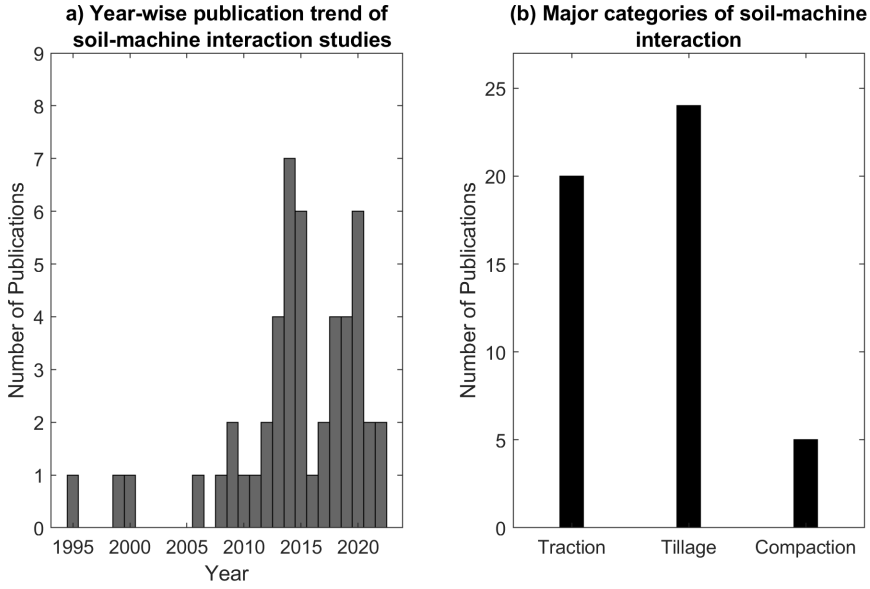


Figure 14. Soil-machine interaction studies: a) Year-wise publication trend, b) Major categories.

5.1. Traction

In traction studies, either an individual traction element (wheel, track, tire) or an entire off-road vehicle is tested in controlled laboratory setup or in prepared or un-prepared fields for its performance optimization. The performance parameters includes drawbar pull, traction efficiency, slip, and fuel consumption which are optimized as a function of numerous variables pertaining to machine, operational setup or parameters and soil properties. A brief summary of CI models developed in selected traction studies are shown in Table 1.

An off-road vehicle (Tractor or Skidder) used in agriculture and forestry are specially designed for drawbar work, i.e., pulling or pushing the implements. Drawbar power is a product of drawbar pull and vehicle velocity in travel direction. The vehicle tire size and its inflation pressure increase the soil contact area, that improves the drawbar performance. Hassan and Tohmaz (1995) [101] studied the drawbar performance of forestry skidders in soft soil to develop a MLR and fully connected NN models to predict the drawbar pull. For tractor energy management and optimization, Taghavifar et al. (2015) implemented an hybridized NN-GA and ANFIS to predict the drawbar pull energy [49], and net traction power [102]. The tractor's drawbar pull varies with a vehicle configuration, weight, and operating mode (2WD and 4WD). Thus, a FIS was proposed to estimate a drawbar pull [103]. In addition to tire size, the drawbar pull is also influenced by the tire geometrical parameters which can be defined with a 3D footprints. Thus, NN was implemented to understand the complex relationship between 3D tire footprints and generated drawbar pull [104].

Traction device develops a force parallel to travel direction and transfers to the vehicle. A traction efficiency is a ratio of output to input power to the device [105]. It is one of the most critical factors in traction studies and relates to energy saving. A several studies were conducted in a laboratory setup (single wheel tester) to study the influence of traction device's operational parameters and soil conditions on traction efficiency. Table 1 listed a various CI methods proposed to model the traction efficiency [106–112].

 doi:10.20944/preprints202210.0391.v1

Motion resistance, is an opposing force, works against the forward motion of the traction device and accounts for all energy loss unrelated to slip [105]. Motion resistance is the difference between gross traction and net traction. Taghavifar et. al (2013) conducted a series of experiments on a driven wheel in a soil bin (clay loam) setup. The experiments were aimed to study the motion resistance influenced by various operating parameters and CI methods (NN & FL) were also proposed to predict motion resistance [113–115].

Tractors are the major power source in a field agriculture. Therefore, for the efficient tractor operation, it is essential to understand how tractor power can be best utilized and optimized under different field conditions. The tractor loses most power at soil-machine interface and its performance is influenced by both operational and soil/terrain parameters. Therefore, Almaliki et. al. (2016) [116] evaluated the 75 HP tractor performance and developed a NN to model its performance as a function of soil and tractor-implement variables. Further, Shafaei et al. (2018) implemented the NN and ANFIS-based models to study the performance of tractor-implement operational parameters on tractive efficiency [117] and wheel slip [118]. The specific fuel consumption is one of the tractor performance indicators, thus, Hanifi et. al., (2021) [119] proposed NN to predict fuel consumption of 60 HP 2WD tractor.

In recent years, mobile robots and autonomous ground vehicles (AGV) are becoming popular on smart farms. Thus, traction behavior of the ground vehicle was studied on a sloped soil bin and NN models was developed to predict traction, mobility, and energy requirement of AGV [120].

Table 1. Traction studies

Author & Year	Traction device	Method	Input	Output
Hassan & Tohmaz (1995) [101]	Rubeer-tire skidder	NN	Tire size, tire pressure, normal load, line of pull angle	Drawbar pull
Carman & Taner (2012) [106]	Driven wheel	NN	Travel reduction	Tractive efficiency
Taghavifar et. al., (2013) [113]	Driven wheel	NN	Velocity, tire pressure, normal load	Rolling Resistance
Taghavifar & Mardani (2013) [114]	Driven wheel	FL	Velocity, tire pressure, normal load	Motion resistance coeff.
Taghavifar & Mardani (2014) [107]	Driven wheel	ANFIS	Velocity, wheel load, slip	Energy efficiency indices (Traction coeff. & traction efficiency)
Taghavifar & Mardani (2014) [108]	Driven wheel	NN	Velocity, wheel load, slip	Energy efficiency indices (Traction coeff. & traction efficiency)
Taghavifar & Mardani (2014) [109]	Driven wheel	NN	Soil texture, tire type, wheel load, speed, slip, inflation pressure	Traction force
Taghavifar & Mardani (2014) [115]	Driven wheel	NN & SVR	Wheel load, inflation pressure, velocity	Energy wasted
Taghavifar & Mardani (2015) [49]	Driven wheel	ANFIS	Wheel load, inflation pressure, velocity	Drawbar pull energy
Taghavifar et. al., (2015) [102]	Driven wheel	NN-GA	Wheel load, inflation pressure, velocity	Available power
Ekinci et. al., (2015) [110]	Single wheel tester	NN & SVR	Lug height, axle load, inflation pressure, drawbar pull	Tractive efficiency
Almaliki et. al., (2016) [116]	Tractor	NN	Moisture content, cone index, tillage depth, inflation pressure, engine speed, forward speed	Tractive efficiency, drawbar pull, rolling resistance, fuel consumption
Pentos et.al., (2017) [111]	Micro tractor	NN	Vertical load, horizontal deformation, soil coeff., compaction, moisture content	Traction force & traction efficiency
Shafaei et.al., (2018) [117]	Tractor	ANFIS, NN	Forward speed, plowing depth, tractor mode	Tractive efficiency
Shafaei et.al., (2019) [118]	Tractor	ANFIS, NN	Forward speed, plowing depth, tractor mode	Wheel slip
Shafaei et.al., (2020) [103]	Tractor	FL	Tractor weight, wheel slip , tractor driving mode	Drawbar pull
Pentos et.al., (2020) [112]	Micro tractor	NN, ANFIS	Vertical load, horizontal deformation, soil coeff., compaction, moisture content	Traction force & traction efficiency
Hanifi et. al., (2021) [119]	Tractor (60 HP)	NN	Inflation pressure, axle load, drawbar force	Specific fuel consumption
Badgujar et. al., (2022) [120]	AGV	NN	Slope, speed, drawbar	Traction efficiency, slip & power number
Cutini et. al., (2022) [104]	Tractor	NN	Tire geometric parameters (area, length, width, depth), slip	Drawbar pull

doi:10.20944/preprints202210.0391.v1

 doi:10.20944/preprints202210.0391.v1

5.2. Tillage

Tillage is classified into two major categories: (1) primary, and (2) secondary tillage, based on purpose, tillage depth, and energy requirement. Primary tillage is an initial major soil working operation, aiming to open up any cultivable land, reduce soil strength, cover plants/residues, and rearrange soil aggregates [2]. It manipulates soil at a greater depth (15 to 30 cm) and moldboard, disk, chisel plow, and subsoiler are commonly used primary tillage tools.

The moldboard (MB) plow shatters the soil, inverts the furrow slice, and covers the crop residues or grasses. As the plow bottom advances, it cuts and fails the ground in the form of furrow slices which shows a periodic variation in draft force. Therefore, a time-lagged recurrent neural network (RNN) was developed to predict the dynamic draft as a function of one step ahead prediction [121] for various shaped tillage tools (MB, Korean, model plow). The MB plow operation consumes the highest energy compared to other tillage tools for a given depth [122,123]. Therefore, several other researchers studied the various types of MB plow performance in varying soil conditions to optimize the energy requirement. The developed a CI-based model were listed in Table 2 and explained in brief as follows: The ANFIS based models were proposed for three-bottom MB plow draft parameters (draft and specific draft) prediction [124]. Carman et al. (2019) [125] developed NN to predict a specific draft and fuel consumption of tractor-mounted MB plow under varying operating conditions. Similarly, a NN was proposed for general-purpose MB plow draft and energy requirement [123].

The MB plow consist of a sliding plow bottom, that slides through the soil. The sliding friction is one of the primary reasons for the MB plow's higher draft and energy requirement. On the contrary, a disk plow is equipped with a concave rolling disks, i.e., a rolling plow bottom designed to reduce friction through rolling action. The energy requirement of disk plow is significantly lower than MB plow. The NN were proposed to predict the disk plow draft and energy requirement [126,127].

Deep tillage (depth <30 cm), is designed to shatter soil, breaking up hardpans and compacted soil layers to ease water and plant root movement. A chisel plow and subsoiler are mainly used for deep tillage. The chisel plow has a series of shovels or teeth spaced on the frame. Its draft requirement is comparatively low and varies with a soil type and depth of operation. Hence, Aboukarima & saad (2006) [128] proposed a various soil texture indices for predicting the chisel plow draft using NN. Shefaei et al. (2017) [129] proposed TSK type ANFIS for chisel plow draft prediction. Similarly, Marey et al. (2020) [130] presented NN for modeling the chisel plow performance parameters. The more details on model inputs and output can be found in Table 2.

A subsoiler has a narrow straight shank to break and fracture the deep compacted soil zone at a greater depth (60-90 cm). The subsoiling demands the high horsepower, often ranging from 30 to 50 hp per shank [131]. The NN was proposed to predict the draft and energy requirement of the subsoiler as a function of soil parameters and operational variables [132]. The subsoiler is a non-inversion tillage tool, available in various shaped shank, and selecting the right shank could reduce the draft [133]. The conventional straight shank type subsoiler requires a significantly higher draft and is often replaced with a parabolic, bent leg, or paraplow [134]. Therefore, the CI based models (ANFIS, MLR, RSM) were developed for predicting the draft of three types of subsoiler shank (subsoiler, paraplow, and bentleg) [132]. Similarly, the ANFIS based model were proposed to predict the forces acting on three design configurations of paraplow (with forward, backward and without bent wing) [135].

The secondary tillage is performed for seedbed preparation, crop production practices, and moisture conservation. The examples of a secondary tillage tools includes, harrow (disk, spring or spike tooth, chisel), cultivator, and clod crushing roller. The energy requirement in secondary tillage tools is comparatively less than the primary. The cultivator and harrow are often operated at a higher ground speed to produce the finer tilth, soil pulverization and weed control. Thus, its operational parameters (tool type, speed, depth) are often

 doi:10.20944/preprints202210.0391.v1

investigated to achieve finer tilth, prevent soil degradation and optimize the tillage energy. The NN was developed to predict draft of cultivator, disk harrow and MB plow in a soil bin setup [21]. The FIS based model was proposed to predict the soil fragmentation resulted form a combination of primary and secondary tillage implements during the seedbed preparation [136]. The draft efficiency and soil loosening of duckfoot cultivator was predicted with the FIS in soil bin [137]. Similarly, Gilandeh et al. (2020) [138] proposed the NN to predict the draft force of a chisel cultivator. A RBF neural network was presented to simulate the soil-machine interaction of five narrow blades in field conditions [139].

Reduced tillage offers several benefits, such as reduced energy and soil disturbance over the traditional tillage. Winged share is a reduced tillage tool and the CI models (NN and FIS) was proposed for predicting the draft force of two diffrent type of winged share tillage tools in a soil bin (loam soil) [140,141]. Likewise, a combined tillage implement is equipped with multiple tillage tools on a single frame to reduce the tractor passes compared to conventional method. The combined tillage saves time, fuel, and energy to obtain the desired soil conditions [142,143]. Therefore, Shafaei et al. (2019) [144] proposed the CI models (NN and ANFIS) to predict the energy indices of the tractor-implement system during a combined tillage operation.

A model tool is a miniature scale replica of actual tool and is often studied in a laboratory environment. The NN models were developed for predicting the energy requirement of rectangular cross-sectional model tool in a soil bin [145]. Similarly, NN was proposed to understand the design and technical insight of the plowing process of a multi-flat plate (model tool) and resulting soil fineness [146].

Table 2. Tillage studies

Author & Year	Tillage tool	CI method	Input	Output
Zhang & Kushwaha (1999) [139]	Narrow blades (Five)	RBF neural network	Forward speed, tool types, soil type	Draft
Choi et.al., (2000) [121]	MB plow, Janggi plow, model tool	Time lagged RNN	One step ahead prediction	Dynamic draft
Aboukarima (2006) [128]	Chisel plow	NN	Soil parameters (textural index, moisture, bulk density), tractor power, plow parameters (depth, width, speed)	Draft
Alimardani et.al., (2009) [132]	Subsoiler	NN	Travel speed, tillage depth, soil parameters (physical)	Draft & tillage energy
Roul et.al., (2009) [21]	MB plow, cultivator, disk harrow	NN	Plow parameters (depth, width, speed), bulk density, moisture	Draft
Marakoglu & Carman (2010) [137]	Duckfoot cultivator share	FL	Travel speed, working depth	Draft efficiency & soil loosening
Rahman et.al., (2011) [145]	Rectangular tillage tool	NN	Plow depth, travel speed, moisture	Energy requirement
Mohammadi et.al., (2012) [140]	Winged share tool	FL	Share depth, width, speed	Draft requirement
Al-Hamed et.al., (2013) [126]	Disk plow	NN	Soil parameters (texture, moisture, soil density), tool parameters (disk dia., tilt & disk angle), plow depth, plow speed	Draft, Unit draft & energy requirement
Saleh & Aly (2013) [146]	Multi-flat plowing tines	NN	Plow parameters(geometry, speed, lift angle, orientation, depth), soil conditions (moisture, density, strength)	Draft force, vertical force, side force, soil finess
Akbarnia et.al (2014) [141]	Winged share tool	NN	Working depth, speed, share width	Draft force
Gilandeh & Sedghi (2015) [136]	Combine tillage	FL	Moisture, speed, soil sampling depth	Median weight diameter
Shafaei et.al (2017) [129]	Chisel plow	ANFIS	Plowing depth, speed	Draft force
Shafaei et.al (2018) [147]	MB plow	ANFIS	Plowing depth, speed	Draft (specific force & draft force)
Shafaei et.al (2018) [124]	Disk plow	NN, MLR	Plowing depth, speed	Draft
Shafaei et.al (2018) [127]	Disk plow	ANFIS, NN	Plowing depth, speed	Fuel efficiency
Shafaei et.al (2019) [118]	Conservation tillage	NN, ANFIS	Plowing depth, speed, tractor mode	Energy indices
Askari & Gilandeh (2019) [134]	Subsoiler tines	MLR, ANFIS, RSM	Tine type, speed, working depth, width	Draft
Carman et.al (2019) [125]	MB plow	NN	Tillage depth, speed	Draft, fuel consumption
Marey et.al, (2020) [130]	Chisel plow	NN	Tractor power, soil texture, density, moisture, plow speed, depth	Draft, rate of soil volume plowed, fuel consumption
Al-Janobi et.al., (2020) [123]	MB plow	NN	Soil texture, field working index	Draft, energy
Gilandeh et.al.,(2020) [138]	MB plow, para-plow	ANFIS	Velocity, depth, type of implement	Draft, vertical & lateral force
Gilandeh et.al.,(2020) [135]	Chisel cultivator	NN, Reg	Depth, moisture, cone index, speed	Draft
Shafaei et.al (2021) [148]	MB plow	FL	Tillage depth, speed, tractor mode	Power consumption efficiency

doi:10.20944/preprints202210.0391.v1

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 26 October 2022 doi:10.20944/preprints202210.0391.v1

5.3. Compaction

Vehicular traffic is common during the field operation and the wheels often traffic the soil more than five times a year. The vehicular traffic affects the soil structure, void ratio, and bulk density which further influence the crop yield. Therefore, the soil compaction resulting from vehicular traffic needs to be optimized. Carman (2008) [149] studied the two different agricultural tires in a soil bin and developed the FIS to predict bulk density, penetration resistance, and soil pressure at 20 cm depth.

Wheel contact area with a soil varies with a wheel parameters (vertical load, inflation pressure, thread etc.,) that determine the forces acting on soil and resulting stress-strain. Therefore, Taghavifar et. al., (2013) conducted a series of experiments in soil bin and developed the CI models (NN, FES and Wavelet NN) to predict the wheel contact area, contact pressure, soil strength and soil density based on tire parameter [150–152]. A multiple wheel passes cumulatively compacts the soil. Hence, the NN was proposed for predicting the penetration resistance and soil sinkage as a function of wheel pass and other wheel operating parameter [153] mentioned in the Table 3.

Table 3. Soil compaction studies

Author & Year	Traction device	CI Method	Input	Output
Carman (2008) [149]	Radial tire (2)	FL	Tire contact pressure, velocity	Bulk density, penetration resistance, soil pressure at 20 cm depth
Taghavifar et al., (2013) [150]	Tire	NN	Wheel load, inflation pressure, wheel pass, velocity, slip	Penetration resistance, soil sinkage
Taghavifar & Mardani (2014) [151]	Tire	FL	Wheel load, inflation pressure	Contact area, contact pressure
Taghavifar & Mardani (2014) [152]	Tire (size 220/65R21)	WNN, NN	Wheel load, velocity, slip	Contact pressure
Taghavifar (2015) [153]	Tire (size 220/65R21 & 9.5L-14)	NN	Soil texture, tire type, slip, wheel pass, load, velocity	Contact pressure, bulk density

5.4. Implemented CI methods

A summary of CI methods proposed in a selected articles (50) is presented in Figure 15. The NNs were the most frequently employed, followed by regression, ANFIS, and FL. The NN-based models were proposed in 36 studies (50.7 %), out of which 34 studies employed, a fully connected feedforward (FF) NN type. The other types of NN, such as RNN, Wavelet NN, and RBF-NN, were reported once (Figure 16a). The most studies reported that shallow NN (single or double-layer feedforward NN) was sufficient to model the complex soil-machine interaction (Figure 16a). In most of the studies, NNs were trained with the backpropagation method such as Levenberg-Marquardt and Gradient Descent algorithms (Figure 16b). The metaheuristics methods such as GA and ICA were reported once.

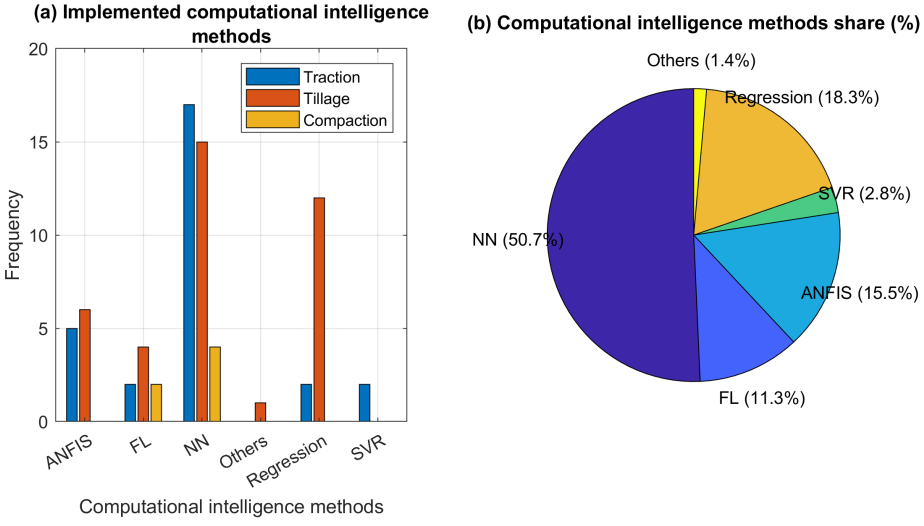


Figure 15. CI methods used in soil machine interaction studies a) soft computing methods and its frequency, b) percentage share of each method.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 26 October 2022 doi:10.20944/preprints202210.0391.v1

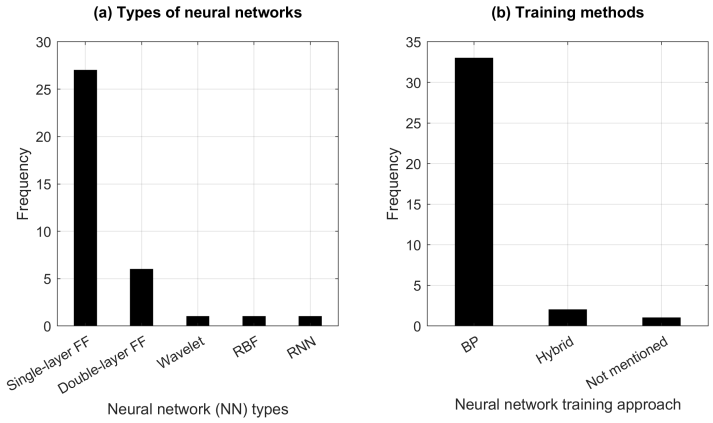


Figure 16. Neural network: (a) types of neural network, (b) training methods.

Subsequently, the FL was implemented in total of eight studies (11.3%), where six studies followed the knowledge-based and remaining two followed the intelligent fuzzy system. The triangular, gaussian, and linear are the popular membership functions. The ANFIS was proposed in eleven studies (15.5%) and the first-order Takagi-Sugeno-Kang fuzzy inference system was the most popular. The ANFIS models were often trained with a hybrid method that combined the least-squares and back propagation.

Additionally, the traditional regression methods were implemented in thirteen (11.3%) studies and are often compared with the other CI methods for prediction accuracy. The regression methods included the MLR and the standard ASABE equations (tool draft eqn.). Besides this, the SVR with different kernel function was reported in two studies.

The CI models performance is evaluated on a predefined matrices and the Figure 17(a) showed the most commonly cited matrices. The CI-based models are able to capture and simulate the complex and non-linear soil machine interaction, with a better accuracy ($\geq$ 90%) as shown in Figure 17(b). Although, the performance of traditional regression method was comparatively lower. MATLAB has emerged as the best platform for the CI based model development (Figure 17(c)).

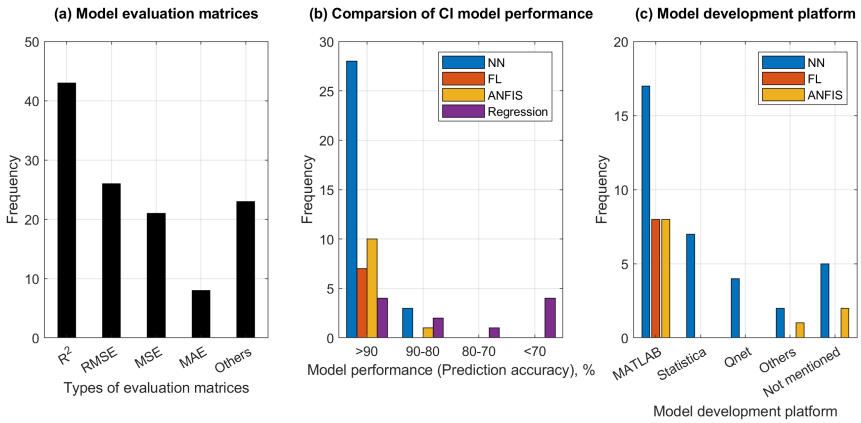


Figure 17. CI models: (a) evaluation matrices, (b) performance comparison, (c) development platform.

6. Strengths and limitations of CI methods

CI models offer manifold advantages over traditional methods described earlier. The features that make these models so attractive are enumerated below.

(*i*) Data driven models can handle copious amounts of data with relative ease [154]. With increasing data size, the corresponding growth in computational overheads is generally between linear and quadratic order of magnitude. For instance, the number of iterations (called epochs) needed to train a neural network is fixed regardless of

 doi:10.20944/preprints202210.0391.v1

data size [52]. On the other hand, traditional methods regularly witness quadratic or higher growths.

(*ii*) In order to further enhance their performances after initial offline training, data driven CI models (e.g. NNs and DNNs) are able to learn online during actual deployment [155]. In other words they are capable of learning from experience.

(*iii*) FIS models can directly benefit from human domain experts; their expert knowledge can be incorporated into the model [156].

(*iv*) Conversely, FIS model outputs are amenable to direct human interpretation. NNs endowed with such capability have been proposed recently [157].

(*v*) CI based algorithms can easily be hybridized with traditional algorithms as well as with one another, thereby offering the benefits of both (c.f. [47,158]). For instance, ANFIS is a combination of FIS and NN approaches.

(*vi*) These models offer the advantage of flexibility. A model that is developed for a specific task, can be adapted to handle another similar task [159,160].

(*vii*) CI models are robust to various forms of imprecision, such as incomplete information, noise, and statistical outliers [154,161–163]. Some such methods that are inherently stochastic, may even benefit from the presence of noise.

(*viii*) CI models (NN, FIS & ANFIS) are self-learned (automatically capture the input-output relationship from the data) and prior assumptions on data distribution are not necessary.

(*ix*) A nominal data can be used as CI model input and single model particularly, the NN can make predictions on multiple outputs, hence referred as a universal function approximator.

It is of little surprise that CI approaches have become very popular in agricultural soil-tillage and traction domain, and many other applications. However, there are a few limitations of CI methods which are discussed as follows.

(*i*) Black box: CI models (NN, ANFIS, FIS) are often considered as a black box. Unlike statistical and mathematical models, the input-output relationship expressed by these models are not self-explanatory. However, in recent years, the researcher has proposed various methods to quantify the variable contribution particularly in NN [164,165].

(*ii*) Computational Resources: CI model development requires a specialized commercial platform (e.g., MATLAB). Also, the model training is highly computational operation and requires a powerful computing resources, often GPUs, to work in parallel. The training time significantly decreases with an increase in computational resources. But, the computational resources are costly and one of the major hurdles in the adoption of CI methods.

(*iii*) Data sources: CI models require a very large, diverse and quality database for model building and training purposes. But, generating such dataset is expensive and time-consuming.

(*iv*) Model overfitting and underfitting: The model overfitting and underfitting results in poor generalization on a new dataset. The training runs (over-training or under-training), size of hidden layers, and model structure are the identified problems that result in model overfitting or underfitting. However, this problem could be potentially addressed with early stopping criteria, regularization methods, dropout method, and observing the model training for given hidden neurons.

(*v*) Local minima: The CI model (NN) often gets trapped at the local minima of error function during the training with the traditional BP algorithm. However, employing a hybrid approach with global search or optimization (GA) has helped solve this problem.

(*vi*) The fuzzy models are not dynamic and flexible and are often restricted to a specific situation for which it is designed. These models are not scalable, i.e., increasing the input variables or input range exponentially increases the number of rules which becomes tedious for human experts. They do not generalize, and updating knowledge

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 26 October 2022 doi:10.20944/preprints202210.0391.v1

(adding new variables or levels) requires changing the model rules. It also needs an expert's knowledge to decide the rules.

(*vii*) CI models such as ANFIS , FIS and SVR are time consuming since each output variable requires an independent model building.

7. Emergent Computational Intelligence Models

This review study critically analysed the most popular CI methods found in literature particularly in soil-machine interaction domain. Further, we also suggest the emergent CI methods that may provide a better results and can be considered as a alternative to CI methods. Those methods are described in brief here.

7.1. Deep Neural Networks

DNNs are NN models with multiple hidden layers [166–168]. In the past few years, this class of CI models has witnessed explosive growth in popularity. DDNs have emerged as a popular tool in a wide range of applications in agriculture [169–174], where they have been used for various image recognition tasks, Unfortunately, DNNs have yet to be explored in any soil-machine interaction application.

Fig. 4 illustrates the layout of such a DNN with fully connected layers. State-of-the-art DNNs incorporate various other types of layers, including RBFN [175], SVR [176], and TSK fuzzy [177,178] layers. DNNs can be endowed with the ability to handle time series data by incorporating of long short-term memory (LSTM) or gated recurrent unit (GRU) layers [173,179]. At each time step t, these layers can hold in memory, essential features from earlier time steps (i.e. $t-1$, $t-2$, etc.), by means of time-delayed feedback. Such DNNs are called recurrent neural networks. An alternate to LSTM and GRU in DNNs is the attention mechanism [180], which has been applied in agriculture [181]

7.2. Regression Trees and Random Forests

Decision trees are CI methods that use graphical tree based representations [182,183], with binary trees [184] being most frequently used. During training, each node in a binary tree is used to split sample pairs $(\mathbf{x}(n), t(n))$ $(n \in \mathcal{S}_t)$ into two subsets $\mathcal{S}_t^{\mathrm{L}}$ and $\mathcal{S}_t^{\mathrm{LR}}$. A threshold θ_j is applied to an element x_j. Hence,

$$y = \begin{cases} \mathcal{S}_t^{\mathrm{L}} = \{n \in \mathcal{S}_t | x_j(n) \leq \theta_j\}; \\ \mathcal{S}_t^{\mathrm{R}} = \{n \in \mathcal{S}_t | x_j(n) > \theta_j\}. \end{cases} \tag{35}$$

The threshold is computed so that at each node, the split is as evenly balanced as possible. Information theoretic and heuristic methods using values of the targets $t(n)$ in the training dataset. Regression trees have found agricultural applications in the past few years [185, 186].

Random forests are CI methods that use multiple trees to obtain outputs [187,188]. There has been a steep rise in the use of this approach for various applications in agriculture [189–199]. An excellent survey of decision trees, random forests, and other CI models has been published in [200].

7.3. Extreme Learning Machines

Extreme learning machines (ELM) are CI models that are useful in regression problems [201–203]. Although in comparison to some other CI models (NNs, RBFNs, and SVRs) ELMs have not been as widely used in other engineering domains, surprisingly they are very popular in various agricultural applications [204–211].

An ELM is structurally equivalent to an $M \times K \times N$ NN. The neurons in the hidden layer incorporate nonlinear activation functions in the same manner as in Eqn. (9). However, unlike in NNs, the hidden layer in an ELM is not fully connected to the input. The hidden weights of an ELM can be arranged as a $K \times M$ sparse matrix. These weights are assigned randomly and do not undergo any training. Only the output weights are trained using

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 26 October 2022 doi:10.20944/preprints202210.0391.v1

a matrix form of the pseudoinverse rule in Eqn. (16). This allows ELMs to be trained significantly faster than equivalent NNs. Hybrid training algorithms for ELMs have also been proposed in [212–214] for agricultural applications. DNN architectures that contain ELM layers are being investigated (cf. [215,216]).

7.4. Bayesian Methods

Bayesian methods are CI paradigms where the outcome renders itself to a probabilistic interpretation. Central to these methods is Bayes rule. The rule can be applied to a parametric Bayesian model in the following manner,

$$\mathrm{p}(\mathbf{\Theta}|\mathcal{S}_t) = \frac{\mathrm{p}(\mathcal{S}_t|\mathbf{\Theta})\mathrm{p}(\mathbf{\Theta})}{\mathrm{p}(\mathcal{S}_t)}. \tag{36}$$

In this expression, $\mathrm{p}(\cdot)$ is the probability of the argument. The left hand side of Eqn. (36) is the posterior probability. The factors $\mathrm{p}(\mathcal{S}_t|\mathbf{\Theta})$, and $\mathrm{p}(\mathbf{\Theta}_t)$ in the right hand side's numerator are the likelihood and the prior probability. In can be shown that LASSO and ridge regularization discussed earlier in Section 3.3 are instances of Bayesian methods where the prior probabilities follow Laplacian and Gaussian distributions.

Since the training data $\mathcal{S}_t$ is independent of the model, it can be dropped from Bayes rule. The model parameter is obtained as the one that has the highest probability, $\mathrm{argmax}_{\mathbf{\Theta}}\mathrm{p}(\mathbf{\Theta}|\mathcal{S}_t)$. Given any unknown input $\mathbf{x}$, the output probability $\mathrm{p}(y|\mathbf{x}\mathbf{\Theta})$ can be obtained from $\mathbf{\Theta}$. Bayesian approaches have been used in several areas of agriculture [217–220].

A Bayesian network is specific Bayesian modeling approach that uses a graphical structure that resembles an NN [221]. Inferring the output in this model relies heavily of statistical sampling techniques [222]. Bayesian networks have been used in [181,223,224].

A mixture of Gaussians[225,226] is a Bayesian model that makes use of hidden variables $z_i, i = 1, 2, \cdots$, which play an intermediate role between the inputs and outputs. Given any input $\mathbf{x}$, the output probability $\mathrm{p}(y)$ is determined as the summation $\sum_i(\mathrm{p}(y|z_i)\mathrm{p}(z_i|\mathbf{x}))$. The use of such methods have begun to be explored in agriculture [37,227,228].

Gaussian process regression [229–231] is a Bayesian approach which assumes the presence of Gaussian noise. As in SVR, kernel matrices are applied in this method. Gaussian process regression has been extensively used in various applications related to agriculture [232–236].

7.5. Ensemble Models

Ensemble models are approaches that combine multiple CI models for decision making [237–239]. Bagging and boosting are two commonly used ensemble approaches. Random forests as well as Gaussian mixtures discussed earlier in this section are also ensemble models.

There has been a surge in the use of these methods in the domain of agriculture [240–247]. Recent research has been directed towards using bagging [248–251] and boosting [186]. Ensemble of NNs have been investigated in [252–258]. GA and PSO have also been studied in this context [243,252].

8. Future direction and scope

8.1. Online Traction control

A sensing technology has reached to its maturity, and ample research material is available, where numerous sensors were employed to sense, measure, and provide real-time information on the biological material (e.g., plant, soil, and field conditions). This review article taught us that CI methods can accurately and precisely model or predict the complex soil-machine interaction. Therefore, the future research efforts should target automatic and real-time traction-tillage control with the help of a sensing and prediction model. The online traction control system would optimize the machine parameters in

real-time to increase the traction efficiencies and reduce soil compaction. For example, traction control is a standard safety feature in today's automotive vehicles. The wheel sensor senses the road conditions (icy or slippery), and the control algorithm enables the traction control to adapt to road conditions in real-time. Moreover, the planetary rover developed by NASA is also equipped with a traction control algorithm that senses the terrain driving condition and predicts the chance of getting trapped in soil (immobility condition) [259,260].

8.2. Online Tillage control

The agricultural soil characteristics and field conditions are very dynamic and vary on a spatial and temporal scale. Hence, a single tillage tool or management system operating uniformly throughout the field would not be sufficing. Multiple factors, including soil type, texture, structure, moisture, field topography, slope, and crop rotation, play a vital role when deciding which implement is best for the field. The current tillage management approach involves employing a single tillage tool for the entire area. The soil moisture is the only parameter, checked before performing the tillage operation. Therefore, future research should develop variable-depth, variable-intensity, and adaptive tillage implements that can be controlled in real-time. This site-specific tillage management would collect real-time information on soil and operating terrain, and CI models would serve as decision-support tools, creating a fully automated tillage management system. Site-specific tillage has excellent potential since the intensity of the operation is adapted to the local needs, which can dramatically improve tillage. Recently, adaptive tillage has become a significant research focus where the tillage tool adapts or changes its shape in real-time.

Author Contributions: Conceptualization, C.B., S.D., and D.F.; Methodology, C.B., D.M.F., S.D., and D.F.; Formal analysis, C.B., D.F., S.D; Resources, C.B., D.M.F., S.D., and D.F.; Data curation, C.B.., D.M.F., S.D; Writing—original draft preparation,C.B., S.D.; Writing—review and editing, C.B., D.M.F., S.D., and D.F.; Supervision, S.D., and D.F.; project administration, S.D., and D.F.; funding acquisition, S.D., and D.F.; All authors have read and agreed to the published version of the manuscript.

Funding: This study was financially supported by a National Institute of Food and Agriculture (NIFA-USDA). The project is titled "National Robotics Initiative (NRI): Multi-Robot Farming on Marginal, Highly Sloped Lands" Project number- KS4513081

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

References

1. Ani, O.A.; Uzoejinwa, B.; Ezeama, A.; Onwualu, A.; Ugwu, S.; Ohagwu, C. Overview of soil-machine interaction studies in soil bins. *Soil and Tillage Research* **2018**, *175*, 13–27. https://doi.org/10.1016/j.still.2017.08.002.
2. ASABE. Terminology and Definitions for Soil Tillage and Soil-Tool Relationships. Technical Report ASAE EP291.3 FEB2005 (R2018), American Society of Agricultural and Biological Engineers, 2018.
3. Sunusi, I.I.; Zhou, J.; Zhen Wang, Z.; Sun, C.; Eltayeb Ibrahim, I.; Opiyo, S.; korohou, T.; Ahmed Soomro, S.; Alhaji Sale, N.; T.O., O. Intelligent tractors: Review of online traction control process. *Computers and Electronics in Agriculture* **2020**, *170*, 105176. https://doi.org/10.1016/j.compag.2019.105176.
4. Zoz, F.; Grisso, R. Traction and Tractor Performance. *ASAE Distinguish. Ser.* **2012**, *27*.
5. Shrini K. Upadhyaya.; Thomas R. Way.; Shrini K. Upadhyaya.; William J. Chancellor. Chapter 2. Traction Mechanics. Part V. Traction Prediction Equations. In *Advances in Soil Dynamics Volume 3*, 1 ed.; American Society of Agricultural and Biological Engineers: St. Joseph, MI, 2009; pp. 161–186. https://doi.org/10.13031/2013.26875.
6. Karmakar, S.; Kushwaha, R.L. Dynamic modeling of soil–tool interaction: An overview from a fluid flow perspective. *Journal of Terramechanics* **2006**, *43*, 411–425. https://doi.org/10.1016/j.jterra.2005.05.001.
7. Johnson.; C. E..; and A. C. Bailey. Soil Compaction. In *Advances in Soil Dynamics Volume 2*, 1 ed.; American Society of Agricultural and Biological Engineers: St. Joseph, MI, 2002; pp. 155–178. https://doi.org/10.13031/2013.9452.
8. Acquah, K.; Chen, Y. Soil Compaction from Wheel Traffic under Three Tillage Systems. *Agriculture* **2022**, *12*, 219. https://doi.org/10.3390/agriculture12020219.
9. Soane, B.; van OUWERKERK, C. Soil Compaction Problems in World Agriculture. In *Developments in Agricultural Engineering*; Elsevier, 1994; Vol. 11, pp. 1–21. https://doi.org/10.1016/B978-0-444-88286-8.50009-X.
10. Brus, D.J.; van den Akker, J.J.H. How serious a problem is subsoil compaction in the Netherlands? A survey based on probability sampling. *SOIL* **2018**, *4*, 37–45. https://doi.org/10.5194/soil-4-37-2018.

11. Zabrodskyi, A.; Šarauskis, E.; Kukharets, S.; Juostas, A.; Vasiliauskas, G.; Andriušis, A. Analysis of the Impact of Soil Compaction on the Environment and Agricultural Economic Losses in Lithuania and Ukraine. *Sustainability* **2021**, *13*, 7762. https://doi.org/10.3390/su13147762.
12. Keller, T. Soil Compaction and Soil Tillage – Studies in Agricultural Soil Mechanics. Doctoral dissertation, Swedish University of Agricultural Sciences, Uppsala, 2004.
13. DeJong-Hughes, J. Soil compaction, 2018.
14. Badalíková, B. Influence of Soil Tillage on Soil Compaction. In *Soil Engineering*; Dedousis, A.P.; Bartzanas, T., Eds.; Springer Berlin Heidelberg: Berlin, Heidelberg, 2010; Vol. 20, pp. 19–30. https://doi.org/10.1007/978-3-642-03681-1_2.
15. Tiwari, V.; Pandey, K.; Pranav, P. A review on traction prediction equations. *Journal of Terramechanics* **2010**, *47*, 191–199. https://doi.org/10.1016/j.jterra.2009.10.002.
16. Wong, J.Y. *Theory of ground vehicles*, 3rd ed ed.; John Wiley: New York, 2001.
17. Godwin, R.; Spoor, G. Soil failure with narrow tines. *Journal of Agricultural Engineering Research* **1977**, *22*, 213–228. https://doi.org/10.1016/0021-8634(77)90044-0.
18. Makanga, J.; Salokhe, V.; Gee-Clough, D. Effect of tine rake angle and aspect ratio on soil failure patterns in dry loam soil. *Journal of Terramechanics* **1996**, *33*, 233–252. https://doi.org/10.1016/S0022-4898(97)00007-4.
19. Karmakar, S. Numerical modeling of soil flow and pressure distribution on a simple tillage tool using computational fluid dynamics. Dissertation, University of Saskatchewan, Saskatoon, 2005.
20. Tagar, A.; Ji, C.; Ding, Q.; Adamowski, J.; Chandio, F.; Mari, I. Soil failure patterns and draft as influenced by consistency limits: An evaluation of the remolded soil cutting test. *Soil and Tillage Research* **2014**, *137*, 58–66. https://doi.org/10.1016/j.still.2013.12.001.
21. Roul, A.; Raheman, H.; Pansare, M.; Machavaram, R. Predicting the draught requirement of tillage implements in sandy clay loam soil using an artificial neural network. *Biosystems Engineering* **2009**, *104*, 476–485. https://doi.org/10.1016/j.biosystemseng.2009.09.004.
22. Fielke, J.; Riley, T. The universal earthmoving equation applied to chisel plough wings. *Journal of Terramechanics* **1991**, *28*, 11–19. https://doi.org/10.1016/0022-4898(91)90003-O.
23. Godwin, R.; Seig, D.; Allott, M. Soil failure and force prediction for soil engaging discs. *Soil Use and Management* **1987**, *3*, 106–114. https://doi.org/10.1111/j.1475-2743.1987.tb00719.x.
24. R. L. Kushwaha.; J. Shen. Finite Element Analysis of the Dynamic Interaction Between Soil and Tillage Tool. *Transactions of the ASAE* **1995**, *38*, 1315–1319. https://doi.org/10.13031/2013.27953.
25. Upadhyaya.; S. K..; U. A. Rosa.; and D. Wulfsohn. Application of the Finite Element Method in Agricultural Soil Mechanics. In *Advances in Soil Dynamics Volume 2*, 1 ed.; American Society of Agricultural and Biological Engineers: St. Joseph, MI, 2002; pp. 117–153. https://doi.org/10.13031/2013.9451.
26. Itzhak Shmulevich.; Dror Rubinstein.; Zvika Asaf. Chapter 5. Discrete Element Modeling of Soil-Machine Interactions. In *Advances in Soil Dynamics Volume 3*, 1 ed.; American Society of Agricultural and Biological Engineers: St. Joseph, MI, 2009; pp. 399–433. https://doi.org/10.13031/2013.26879.
27. Jude Liu.; Radhey Lal Kushwaha. Two-decade Achievements in Modeling of Soil - Tool Interactions. In Proceedings of the 2008 Providence, Rhode Island, June 29 - July 2, 2008. American Society of Agricultural and Biological Engineers, 2008. https://doi.org/10.13031/2013.24973.
28. Taheri, S.; Sandu, C.; Taheri, S.; Pinto, E.; Gorsich, D. A technical survey on Terramechanics models for tire–terrain interaction used in modeling and simulation of wheeled vehicles. *Journal of Terramechanics* **2015**, *57*, 1–22. https://doi.org/10.1016/j.jterra.2014.08.003.
29. Ghosh, S.; Konar, A., An Overview of Computational Intelligence Algorithms. In *Call Admission Control in Mobile Cellular Networks*; Springer Berlin Heidelberg: Berlin, Heidelberg, 2013; pp. 63–94. https://doi.org/10.1007/978-3-642-30997-7_2.
30. Vasant, P. *Handbook of research on novel soft computing intelligent algorithms: Theory and practical applications*; IGI Global, 2013.
31. Xing, B.; Gao, W.J. *Innovative computational intelligence: a rough guide to 134 clever algorithms*; Vol. 62, Springer, 2014.
32. Ibrahim, D. An overview of soft computing. *Procedia Computer Science* **2016**, *102*, 34–38.
33. Ding, S.; Li, H.; Su, C.; Yu, J.; Jin, F. Evolutionary artificial neural networks: a review. *Artificial Intelligence Review* **2013**, *39*, 251–260.
34. Stanley, K.O.; Clune, J.; Lehman, J.; Miikkulainen, R. Designing neural networks through neuroevolution. *Nature Machine Intelligence* **2019**, *1*, 24–35.
35. Elbes, M.; Alzubi, S.; Kanan, T.; Al-Fuqaha, A.; Hawashin, B. A survey on particle swarm optimization with emphasis on engineering and network applications. *Evolutionary Intelligence* **2019**, *12*, 113–129.
36. Karaboga, D.; Kaya, E. Adaptive network based fuzzy inference system (ANFIS) training approaches: a comprehensive survey. *Artificial Intelligence Review* **2019**, *52*, 2263–2293.
37. Badgujara, C.; Das, S.; Flippo, D.; Welch, S.M.; Martinez-Figueroa, D. A Deep Neural Network-Based Approach to Predict the Traction, Mobility, and Energy Consumption of Autonomous Ground Vehicle on Sloping Terrain Field. *Computers and Electronics in Agriculture (Under Review)* **2022**.
38. Scholz, M. Validation of nonlinear PCA. *Neural processing letters* **2012**, *36*, 21–30.
39. Stone, J.V. Independent component analysis: an introduction. *Trends in cognitive sciences* **2002**, *6*, 59–64.

40. Cherkassky, V.; Ma, Y. Comparison of loss functions for linear regression. In Proceedings of the 2004 IEEE International Joint Conference on Neural Networks (IEEE Cat. No.04CH37541), 2004, Vol. 1, pp. 395–400. https://doi.org/10.1109/IJCNN.2004.1379938.
41. Čížek, P.; Sadıkoğlu, S. Robust nonparametric regression: A review. *Wiley Interdisciplinary Reviews: Computational Statistics* **2020**, *12*, e1492.
42. Huang, S.; Wu, Q. Robust pairwise learning with Huber loss. *Journal of Complexity* **2021**, *66*, 101570.
43. Vapnik, V.; Levin, E.; Le Cun, Y. Measuring the VC-dimension of a learning machine. *Neural computation* **1994**, *6*, 851–876.
44. Zou, H.; Hastie, T. Regularization and variable selection via the elastic net. *Journal of the royal statistical society: series B (statistical methodology)* **2005**, *67*, 301–320.
45. Wilamowski, B.M.; Yu, H. Improved computation for Levenberg–Marquardt training. *IEEE transactions on neural networks* **2010**, *21*, 930–937.
46. Rodriguez, J.D.; Perez, A.; Lozano, J.A. Sensitivity analysis of k-fold cross validation in prediction error estimation. *IEEE transactions on pattern analysis and machine intelligence* **2009**, *32*, 569–575.
47. Das, S.; Koduru, P.; Gui, M.; Cochran, M.; Wareing, A.; Welch, S.M.; Babin, B.R. Adding local search to particle swarm optimization. In Proceedings of the 2006 IEEE International Conference on Evolutionary Computation. IEEE, 2006, pp. 428–433.
48. Taghavifar, H.; Mardani, A. Energy loss optimization of run-off-road wheels applying imperialist competitive algorithm. *Information Processing in Agriculture* **2014**, *1*, 57–65. https://doi.org/10.1016/j.inpa.2014.06.001.
49. Taghavifar, H.; Mardani, A. Evaluating the effect of tire parameters on required drawbar pull energy model using adaptive neuro-fuzzy inference system. *Energy* **2015**, *85*, 586–593. https://doi.org/10.1016/j.energy.2015.03.072.
50. Rumelhart, D.E.; Hinton, G.E.; Williams, R.J. Learning representations by back-propagating errors. *nature* **1986**, *323*, 533–536.
51. Sapna, S.; Tamilarasi, A.; Kumar, M.P.; et al. Backpropagation learning algorithm based on Levenberg Marquardt Algorithm. *Comp Sci Inform Technol (CS and IT)* **2012**, *2*, 393–398.
52. Abu-Mostafa, Y.S.; Magdon-Ismail, M.; Lin, H.T. *Learning from data*; Vol. 4, AMLBook New York, 2012.
53. Ghosh, J.; Nag, A. An overview of radial basis function networks. *Radial Basis Function Networks 2* **2001**, pp. 1–36.
54. Ruß, G. Data mining of agricultural yield data: A comparison of regression models. In Proceedings of the Industrial Conference on Data Mining. Springer, 2009, pp. 24–37.
55. da Silva Jr, E.M.; Maia, R.D.; Cabacinha, C.D. Bee-inspired RBF network for volume estimation of individual trees. *Computers and Electronics in Agriculture* **2018**, *152*, 401–408.
56. Zhang, D.; Zang, G.; Li, J.; Ma, K.; Liu, H. Prediction of soybean price in China using QR-RBF neural network model. *Computers and Electronics in Agriculture* **2018**, *154*, 10–17.
57. Ashraf, T.; Khan, Y.N. Weed density classification in rice crop using computer vision. *Computers and Electronics in Agriculture* **2020**, *175*, 105590.
58. Eide, Å.J.; Lindblad, T.; Paillet, G. Radial-basis-function networks. In *Intelligent Systems*; CRC Press, 2018; pp. 7–1.
59. Bock, H.H. Clustering methods: a history of k-means algorithms. *Selected contributions in data analysis and classification* **2007**, pp. 161–172.
60. Smola, A.J.; Schölkopf, B. A tutorial on support vector regression. *Statistics and computing* **2004**, *14*, 199–222.
61. Pisner, D.A.; Schnyer, D.M. Support vector machine. In *Machine learning*; Elsevier, 2020; pp. 101–121.
62. Cervantes, J.; Garcia-Lamont, F.; Rodríguez-Mazahua, L.; Lopez, A. A comprehensive survey on support vector machine classification: Applications, challenges and trends. *Neurocomputing* **2020**, *408*, 189–215.
63. Mucherino, A.; Papajorgji, P.; Pardalos, P.M. A survey of data mining techniques applied to agriculture. *Operational Research* **2009**, *9*, 121–140.
64. Mehdizadeh, S.; Behmanesh, J.; Khalili, K. Using MARS, SVM, GEP and empirical equations for estimation of monthly mean reference evapotranspiration. *Computers and electronics in agriculture* **2017**, *139*, 103–114.
65. Patrício, D.I.; Rieder, R. Computer vision and artificial intelligence in precision agriculture for grain crops: A systematic review. *Computers and electronics in agriculture* **2018**, *153*, 69–81.
66. Kok, Z.H.; Shariff, A.R.M.; Alfatni, M.S.M.; Khairunniza-Bejo, S. Support vector machine in precision agriculture: a review. *Computers and Electronics in Agriculture* **2021**, *191*, 106546.
67. Burges, C.J. A tutorial on support vector machines for pattern recognition. *Data mining and knowledge discovery* **1998**, *2*, 121–167.
68. Hindi, H. A tutorial on convex optimization. In Proceedings of the Proceedings of the 2004 American Control Conference. IEEE, 2004, Vol. 4, pp. 3252–3265.
69. Hindi, H. A tutorial on convex optimization II: duality and interior point methods. In Proceedings of the 2006 American Control Conference. IEEE, 2006, pp. 11–pp.
70. Chapelle, O. Training a support vector machine in the primal. *Neural computation* **2007**, *19*, 1155–1178.
71. Liang, Z.; Li, Y. Incremental support vector machine learning in the primal and applications. *Neurocomputing* **2009**, *72*, 2249–2258.
72. Wu, J.; Wang, Y.G. Iterative Learning in Support Vector Regression With Heterogeneous Variances. *IEEE Transactions on Emerging Topics in Computational Intelligence* **2022**, pp. 1–10. https://doi.org/10.1109/TETCI.2022.3182725.
73. Zimmermann, H.J. Fuzzy set theory. *Wiley interdisciplinary reviews: computational statistics* **2010**, *2*, 317–332.
74. Iancu, I. A Mamdani type fuzzy logic controller. *Fuzzy logic-controls, concepts, theories and applications* **2012**, *15*, 325–350.

75. Guerra, T.M.; Kruszewski, A.; Lauber, J. Discrete Tagaki–Sugeno models for control: Where are we? *Annual Reviews in control* **2009**, *33*, 37–47.
76. Nguyen, A.T.; Taniguchi, T.; Eciolaza, L.; Campos, V.; Palhares, R.; Sugeno, M. Fuzzy control systems: Past, present and future. *IEEE Computational Intelligence Magazine* **2019**, *14*, 56–68.
77. Nakanishi, H.; Turksen, I.; Sugeno, M. A review and comparison of six reasoning methods. *Fuzzy sets and systems* **1993**, *57*, 257–294.
78. Ying, H.; Ding, Y.; Li, S.; Shao, S. Comparison of necessary conditions for typical Takagi-Sugeno and Mamdani fuzzy systems as universal approximators. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans* **1999**, *29*, 508–514.
79. Huang, Y.; Lan, Y.; Thomson, S.J.; Fang, A.; Hoffmann, W.C.; Lacey, R.E. Development of soft computing and applications in agricultural and biological engineering. *Computers and electronics in agriculture* **2010**, *71*, 107–127.
80. Touati, F.; Al-Hitmi, M.; Benhmed, K.; Tabish, R. A fuzzy logic based irrigation system enhanced with wireless data logging applied to the state of Qatar. *Computers and electronics in agriculture* **2013**, *98*, 233–241.
81. Zareiforoush, H.; Minaei, S.; Alizadeh, M.R.; Banakar, A.; Samani, B.H. Design, development and performance evaluation of an automatic control system for rice whitening machine based on computer vision and fuzzy logic. *Computers and Electronics in Agriculture* **2016**, *124*, 14–22.
82. Kisi, O.; Sanikhani, H.; Zounemat-Kermani, M.; Niazi, F. Long-term monthly evapotranspiration modeling by several data-driven methods without climatic data. *Computers and Electronics in Agriculture* **2015**, *115*, 66–77.
83. Valdés-Vela, M.; Abrisqueta, I.; Conejero, W.; Vera, J.; Ruiz-Sánchez, M.C. Soft computing applied to stem water potential estimation: A fuzzy rule based approach. *Computers and Electronics in Agriculture* **2015**, *115*, 150–160.
84. Malik, A.; Kumar, A.; Piri, J. Daily suspended sediment concentration simulation using hydrological data of Pranhita River Basin, India. *Computers and Electronics in Agriculture* **2017**, *138*, 20–28.
85. Shafaei, S.; Loghavi, M.; Kamgar, S. Appraisal of Takagi-Sugeno-Kang type of adaptive neuro-fuzzy inference system for draft force prediction of chisel plow implement. *Computers and Electronics in Agriculture* **2017**, *142*, 406–415.
86. Shiri, J.; Keshavarzi, A.; Kisi, O.; Iturraran-Viveros, U.; Bagherzadeh, A.; Mousavi, R.; Karimi, S. Modeling soil cation exchange capacity using soil parameters: Assessing the heuristic models. *Computers and Electronics in Agriculture* **2017**, *135*, 242–251.
87. Jang, J.S.; Sun, C.T. Neuro-fuzzy modeling and control. *Proceedings of the IEEE* **1995**, *83*, 378–406. https://doi.org/10.1109/5.364486.
88. Babuška, R.; Verbruggen, H. Neuro-fuzzy methods for nonlinear system identification. *Annual reviews in control* **2003**, *27*, 73–85.
89. Shihabudheen, K.; Pillai, G.N. Recent advances in neuro-fuzzy system: A survey. *Knowledge-Based Systems* **2018**, *152*, 136–162.
90. de Campos Souza, P.V. Fuzzy neural networks and neuro-fuzzy networks: A review the main techniques and applications used in the literature. *Applied soft computing* **2020**, *92*, 106275.
91. Wu, W.; Li, L.; Yang, J.; Liu, Y. A modified gradient-based neuro-fuzzy learning algorithm and its convergence. *Information Sciences* **2010**, *180*, 1630–1642.
92. Wang, L.; Zhang, H. An adaptive fuzzy hierarchical control for maintaining solar greenhouse temperature. *Computers and electronics in agriculture* **2018**, *155*, 251–256.
93. Shafaei, S.; Loghavi, M.; Kamgar, S. An extensive validation of computer simulation frameworks for neural prognostication of tractor tractive efficiency. *Computers and Electronics in Agriculture* **2018**, *155*, 283–297.
94. Petković, B.; Petković, D.; Kuzman, B.; Milovančević, M.; Wakil, K.; Ho, L.S.; Jermsittiparsert, K. Neuro-fuzzy estimation of reference crop evapotranspiration by neuro fuzzy logic based on weather conditions. *Computers and Electronics in Agriculture* **2020**, *173*, 105358.
95. Wiktorowicz, K. RFIS: regression-based fuzzy inference system. *Neural Computing and Applications* **2022**, pp. 1–22.
96. Cheng, C.B.; Cheng, C.J.; Lee, E. Neuro-fuzzy and genetic algorithm in multiple response optimization. *Computers & Mathematics with Applications* **2002**, *44*, 1503–1514.
97. Shihabudheen, K.; Mahesh, M.; Pillai, G.N. Particle swarm optimization based extreme learning neuro-fuzzy system for regression and classification. *Expert Systems with Applications* **2018**, *92*, 474–484.
98. Castellano, G.; Castiello, C.; Fanelli, A.M.; Jain, L. Evolutionary neuro-fuzzy systems and applications. In *Advances in evolutionary computing for system design*; Springer, 2007; pp. 11–45.
99. Aghelpour, P.; Bahrami-Pichaghchi, H.; Kisi, O. Comparison of three different bio-inspired algorithms to improve ability of neuro fuzzy approach in prediction of agricultural drought, based on three different indexes. *Computers and electronics in agriculture* **2020**, *170*, 105279.
100. Huang, Y.; Lan, Y.; Thomson, S.J.; Fang, A.; Hoffmann, W.C.; Lacey, R.E. Development of soft computing and applications in agricultural and biological engineering. *Computers and Electronics in Agriculture* **2010**, *71*, 107–127. https://doi.org/10.1016/j.compag.2010.01.001.
101. Hassan, A.; Tohmaz, A. Performance of Skidder Tires in Swamps — Comparison between Statistical and Neural Network Models. *Transactions of the ASAE* **1995**, *38*, 1545–1551.
102. Taghavifar, H.; Mardani, A.; Hosseinloo, A.H. Appraisal of artificial neural network-genetic algorithm based model for prediction of the power provided by the agricultural tractors. *Energy* **2015**, *93*, 1704–1710. https://doi.org/10.1016/j.energy.2015.10.066.

103. Shafaei, S.; Loghavi, M.; Kamgar, S. Benchmark of an intelligent fuzzy calculator for admissible estimation of drawbar pull supplied by mechanical front wheel drive tractor. *Artificial Intelligence in Agriculture* **2020**, *4*, 209–218. https://doi.org/10.1016/j.aiia.2020.10.001.
104. Cutini, M.; Costa, C.; Brambilla, M.; Bisaglia, C. Relationship between the 3D Footprint of an Agricultural Tire and Drawbar Pull Using an Artificial Neural Network. *Applied Engineering in Agriculture* **2022**, *38*, 293–301. https://doi.org/10.13031/aea.13851.
105. ASABE. General Terminology for Traction of Agricultural Traction and Transport Devices and Vehicles. American National Standard ANSI/ASAE S296.5 DEC2003 (R2018), 2018.
106. Carman, K.; Taner, A. Prediction of Tire Tractive Performance by Using Artificial Neural Networks. *Mathematical and Computational Applications* **2012**, *17*, 182–192. https://doi.org/10.3390/mca17030182.
107. Taghavifar, H.; Mardani, A. On the modeling of energy efficiency indices of agricultural tractor driving wheels applying adaptive neuro-fuzzy inference system. *Journal of Terramechanics* **2014**, *56*, 37–47. https://doi.org/10.1016/j.jterra.2014.08.002.
108. Taghavifar, H.; Mardani, A. Applying a supervised ANN (artificial neural network) approach to the prognostication of driven wheel energy efficiency indices. *Energy* **2014**, *68*, 651–657. https://doi.org/10.1016/j.energy.2014.01.048.
109. Taghavifar, H.; Mardani, A. Use of artificial neural networks for estimation of agricultural wheel traction force in soil bin. *Neural Computing and Applications* **2014**, *24*, 1249–1258. https://doi.org/10.1007/s00521-013-1360-8.
110. Ekinci, S.; Carman, K.; Kahramanlı, H. Investigation and modeling of the tractive performance of radial tires using off-road vehicles. *Energy* **2015**, *93*, 1953–1963. https://doi.org/10.1016/j.energy.2015.10.070.
111. Pentoś, K.; Pieczarka, K. Applying an artificial neural network approach to the analysis of tractive properties in changing soil conditions. *Soil and Tillage Research* **2017**, *165*, 113–120. https://doi.org/10.1016/j.still.2016.08.005.
112. Pentoś, K.; Pieczarka, K.; Lejman, K. Application of Soft Computing Techniques for the Analysis of Tractive Properties of a Low-Power Agricultural Tractor under Various Soil Conditions. *Complexity* **2020**, *2020*, 1–11. https://doi.org/10.1155/2020/7607545.
113. Taghavifar, H.; Mardani, A.; Karim-Maslak, H.; Kalbkhani, H. Artificial Neural Network estimation of wheel rolling resistance in clay loam soil. *Applied Soft Computing* **2013**, *13*, 3544–3551. https://doi.org/10.1016/j.asoc.2013.03.017.
114. Taghavifar, H.; Mardani, A. A knowledge-based Mamdani fuzzy logic prediction of the motion resistance coefficient in a soil bin facility for clay loam soil. *Neural Computing and Applications* **2013**, *23*, 293–302. https://doi.org/10.1007/s00521-013-1400-4.
115. Taghavifar, H.; Mardani, A. A comparative trend in forecasting ability of artificial neural networks and regressive support vector machine methodologies for energy dissipation modeling of off-road vehicles. *Energy* **2014**, *66*, 569–576. https://doi.org/10.1016/j.energy.2014.01.022.
116. Almaliki, S.; Alimardani, R.; Omid, M. Artificial Neural Network Based Modeling of Tractor Performance at Different Field Conditions. *Agricultural Engineering International: CIGR Journal* **2016**, *18*, 262–274.
117. Shafaei, S.; Loghavi, M.; Kamgar, S. An extensive validation of computer simulation frameworks for neural prognostication of tractor tractive efficiency. *Computers and Electronics in Agriculture* **2018**, *155*, 283–297. https://doi.org/10.1016/j.compag.2018.10.027.
118. Shafaei, S.; Loghavi, M.; Kamgar, S. Feasibility of implementation of intelligent simulation configurations based on data mining methodologies for prediction of tractor wheel slip. *Information Processing in Agriculture* **2019**, *6*, 183–199. https://doi.org/10.1016/j.inpa.2018.10.004.
119. Küçüksariyildiz, H.; Çarman, K.; Sabanci, K. Prediction of Specific Fuel Consumption of 60 HP 2WD Tractor Using Artificial Neural Networks. *International Journal of Automotive Science And Technology* **2021**. https://doi.org/10.30939/ijastech..1010318.
120. Badgujar, C.; Flippo, D.; Welch, S. Artificial neural network to predict traction performance of autonomous ground vehicle on a sloped soil bin and uncertainty analysis. *Computers and Electronics in Agriculture* **2022**, *196*, 106867. https://doi.org/10.1016/j.compag.2022.106867.
121. Choi, Y.S.; Lee, K.S.; Park, W.Y. Application of a Neural Network to Dynamic Draft Model. *Agricultural and Biosystems Engineering* **2000**, *1*, 67–72.
122. ASABE. Agricultural Machinery Management Data. Technical Report ASAE D497.4 MAR99, American Society of Agricultural and Biological Engineers, 2000.
123. Al-Janobi, A.; Al-Hamed, S.; Aboukarima, A.; Almajhadi, Y. MODELING OF DRAFT AND ENERGY REQUIREMENTS OF A MOLDBOARD PLOW USING ARTIFICIAL NEURAL NETWORKS BASED ON TWO NOVEL VARIABLES. *Engenharia Agrícola* **2020**, *40*, 363–373. https://doi.org/10.1590/1809-4430-eng.agric.v40n3p363-373/2020.
124. Shafaei, S.; Loghavi, M.; Kamgar, S.; Raoufat, M. Potential assessment of neuro-fuzzy strategy in prognostication of draft parameters of primary tillage implement. *Annals of Agrarian Science* **2018**, *16*, 257–266. https://doi.org/10.1016/j.aasci.2018.04.001.
125. Çarman, K.; Çıtıl, E.; Taner, A. Artificial Neural Network Model for Predicting Specific Draft Force and Fuel Consumption Requirement of a Mouldboard Plough. *Selcuk Journal of Agriculture and Food Sciences* **2019**, *33*, 241–247. https://doi.org/10.15316/SJAFS.2019.183.
126. Al-Hamed, S.A.; Wahby, M.F.; Al-Saqer, S.M.; Aboukarima, A.M.; Ahmed, A.S. Artificial neural network model for predicting draft and energy requirements of a disk plow. *The Journal of Animal & Plant Sciences* **2013**, *23*, 1714–1724.
127. Shafaei, S.M.; Loghavi, M.; Kamgar, S. A comparative study between mathematical models and the ANN data mining technique in draft force prediction of disk plow implement in clay loam soil. *Agricultural Engineering International: CIGR Journal* **2018**, *20*, 71–79.

128. Aboukarima, A.E.W.M.; Saad, A.F. Assessment of Different Indices Depicting Soil Texture for Predicting Chisel Plow Draft Using Neural Networks. *Alexandria Science Exchange Journal* **2006**, *27*.
129. Shafaei, S.; Loghavi, M.; Kamgar, S. Appraisal of Takagi-Sugeno-Kang type of adaptive neuro-fuzzy inference system for draft force prediction of chisel plow implement. *Computers and Electronics in Agriculture* **2017**, *142*, 406–415. https://doi.org/10.1016/j.compag.2017.09.023.
130. Marey, S.; Aboukarima, A.; Almajhadi, Y. PREDICTING THE PERFORMANCE PARAMETERS OF CHISEL PLOW USING NEURAL NETWORK MODEL. *Engenharia Agrícola* **2020**, *40*, 719–731. https://doi.org/10.1590/1809-4430-eng.agric.v40n6p719-731/2020.
131. DeJong-Hughes, J. Tillage implements, 2021.
132. Alimardani, R.; Abbaspour-Gilandeh, Y.; Khalilian, A.; Keyhani, A.; Sadati, S.H. Prediction of draft force and energy of subsoiling operation using ANN model. *Journal of Food, Agriculture and Environment* **2009**, *7*, 537–542.
133. Bergtold, J.; Sailus, M.; Jackson, T. *Conservation Tillage Systems in the Southeast: Production, Profitability and Stewardship*; 2020.
134. Askari, M.; Abbaspour-Gilandeh, Y. Assessment of adaptive neuro-fuzzy inference system and response surface methodology approaches in draft force prediction of subsoiling tines. *Soil and Tillage Research* **2019**, *194*, 104338. https://doi.org/10.1016/j.still.2019.104338.
135. Abbaspour-Gilandeh, M.; Shahgoli, G.; Abbaspour-Gilandeh, Y.; Herrera-Miranda, M.A.; Hernández-Hernández, J.L.; Herrera-Miranda, I. Measuring and Comparing Forces Acting on Moldboard Plow and Para-Plow with Wing to Replace Moldboard Plow with Para-Plow for Tillage and Modeling It Using Adaptive Neuro-Fuzzy Interface System (ANFIS). *Agriculture* **2020**, *10*, 633. https://doi.org/10.3390/agriculture10120633.
136. Abbaspour-Gilandeh, Y.; Sedghi, R. Predicting soil fragmentation during tillage operation using fuzzy logic approach. *Journal of Terramechanics* **2015**, *57*, 61–69. https://doi.org/10.1016/j.jterra.2014.12.002.
137. Marakoğlu, T.; Çarman, K. Fuzzy knowledge-based model for prediction of soil loosening and draft efficiency in tillage. *Journal of Terramechanics* **2010**, *47*, 173–178. https://doi.org/10.1016/j.jterra.2009.10.001.
138. Abbaspour-Gilandeh, Y.; Fazeli, M.; Roshanianfard, A.; Hernández-Hernández, M.; Gallardo-Bernal, I.; Hernández-Hernández, J.L. Prediction of Draft Force of a Chisel Cultivator Using Artificial Neural Networks and Its Comparison with Regression Model. *Agronomy* **2020**, *10*, 451. https://doi.org/10.3390/agronomy10040451.
139. Zhang, Z.X.; Kushwaha, R. Applications of neural networks to simulate soil-tool interaction and soil behavior. *Canadian Agricultural Engineering* **1999**, *41*, 119–125.
140. Mohammadi, A. Modeling of Draft Force Variation in a Winged Share Tillage Tool Using Fuzzy Table Look-Up Scheme. *Agricultural Engineering International: CIGR Journal* **2012**, *14*, 262–268.
141. Akbarnia, A.; Mohammadi, A.; Alimardani, R.; Farhani, F. Simulation of draft force of winged share tillage tool using artificial neural network model. *Agricultural Engineering International: CIGR Journal* **2014**, *16*, 57–65.
142. Usaborisut, P.; Prasertkan, K. Specific energy requirements and soil pulverization of a combined tillage implement. *Heliyon* **2019**, *5*, e02757. https://doi.org/10.1016/j.heliyon.2019.e02757.
143. Upadhyay, G.; Raheman, H. Comparative assessment of energy requirement and tillage effectiveness of combined (active-passive) and conventional offset disc harrows. *Biosystems Engineering* **2020**, *198*, 266–279. https://doi.org/10.1016/j.biosystemseng.2020.08.014.
144. Shafaei, S.; Loghavi, M.; Kamgar, S. Prognostication of energy indices of tractor-implement utilizing soft computing techniques. *Information Processing in Agriculture* **2019**, *6*, 132–149. https://doi.org/10.1016/j.inpa.2018.08.001.
145. Rahman, A.; Kushwaha, R.L.; Ashrafizadeh, S.R.; Panigrahi, S. Prediction of Energy Requirement of a Tillage Tool in a Soil Bin using Artificial Neural Network. In Proceedings of the 2011 Louisville, Kentucky, August 7 - August 10, 2011. American Society of Agricultural and Biological Engineers, 2011. https://doi.org/10.13031/2013.37744.
146. Saleh, B.; Aly, A. Artificial Neural Network Model for Evaluation of the Ploughing Process Performance. *International Journal of Control Automation and Systems* **2013**, *2*, 1–11.
147. Shafaei, S.; Loghavi, M.; Kamgar, S. On the neurocomputing based intelligent simulation of tractor fuel efficiency parameters. *Information Processing in Agriculture* **2018**, *5*, 205–223. https://doi.org/10.1016/j.inpa.2018.02.003.
148. Shafaei, S.M.; Loghavi, M.; Kamgar, S. On the Reliability of Intelligent Fuzzy System for Multivariate Pattern Scrutinization of Power Consumption Efficiency of Mechanical Front Wheel Drive Tractor. *Journal of Biosystems Engineering* **2021**, *46*, 1–15. https://doi.org/10.1007/s42853-020-00083-6.
149. Carman, K. Prediction of soil compaction under pneumatic tires a using fuzzy logic approach. *Journal of Terramechanics* **2008**, *45*, 103–108. https://doi.org/10.1016/j.jterra.2008.10.001.
150. Taghavifar, H.; Mardani, A.; Taghavifar, L. A hybridized artificial neural network and imperialist competitive algorithm optimization approach for prediction of soil compaction in soil bin facility. *Measurement* **2013**, *46*, 2288–2299. https://doi.org/10.1016/j.measurement.2013.04.077.
151. Taghavifar, H.; Mardani, A. Fuzzy logic system based prediction effort: A case study on the effects of tire parameters on contact area and contact pressure. *Applied Soft Computing* **2014**, *14*, 390–396. https://doi.org/10.1016/j.asoc.2013.10.005.
152. Taghavifar, H.; Mardani, A. Wavelet neural network applied for prognostication of contact pressure between soil and driving wheel. *Information Processing in Agriculture* **2014**, *1*, 51–56. https://doi.org/10.1016/j.inpa.2014.05.002.

153. Taghavifar, H. A supervised artificial neural network representational model based prediction of contact pressure and bulk density. *Journal of Advances in Vehicle Engineering* **2015**, *1*, 14–21.
154. Chen, X.W.; Lin, X. Big data deep learning: challenges and perspectives. *IEEE access* **2014**, *2*, 514–525.
155. Hoi, S.C.; Sahoo, D.; Lu, J.; Zhao, P. Online learning: A comprehensive survey. *Neurocomputing* **2021**, *459*, 249–289.
156. Wagner, C.; Smith, M.; Wallace, K.; Pourabdollah, A. Generating uncertain fuzzy logic rules from surveys: capturing subjective relationships between variables from human experts. In Proceedings of the 2015 IEEE International Conference on Systems, Man, and Cybernetics. IEEE, 2015, pp. 2033–2038.
157. Evans, R.; Grefenstette, E. Learning explanatory rules from noisy data. *Journal of Artificial Intelligence Research* **2018**, *61*, 1–64.
158. Mashwani, W.K. Comprehensive survey of the hybrid evolutionary algorithms. *International Journal of Applied Evolutionary Computation (IJAEC)* **2013**, *4*, 1–19.
159. Weiss, K.; Khoshgoftaar, T.M.; Wang, D. A survey of transfer learning. *Journal of Big data* **2016**, *3*, 1–40.
160. Zhuang, F.; Qi, Z.; Duan, K.; Xi, D.; Zhu, Y.; Zhu, H.; Xiong, H.; He, Q. A comprehensive survey on transfer learning. *Proceedings of the IEEE* **2020**, *109*, 43–76.
161. Abdella, M.; Marwala, T. The use of genetic algorithms and neural networks to approximate missing data in database. In Proceedings of the IEEE 3rd International Conference on Computational Cybernetics, 2005. ICCC 2005. IEEE, 2005, pp. 207–212.
162. Amiri, M.; Jensen, R. Missing data imputation using fuzzy-rough methods. *Neurocomputing* **2016**, *205*, 152–164.
163. Capuano, N.; Chiclana, F.; Fujita, H.; Herrera-Viedma, E.; Loia, V. Fuzzy group decision making with incomplete information guided by social influence. *IEEE Transactions on Fuzzy Systems* **2017**, *26*, 1704–1718.
164. Olden, J.D.; Jackson, D.A. Illuminating the "black box": a randomization approach for understanding variable contributions in artificial neural networks. *Ecological Modelling* **2002**, *154*, 135–150. https://doi.org/10.1016/S0304-3800(02)00064-9.
165. Sheu, Y.h. Illuminating the Black Box: Interpreting Deep Neural Network Models for Psychiatric Research. *Frontiers in Psychiatry* **2020**, *11*, 551299. https://doi.org/10.3389/fpsyt.2020.551299.
166. LeCun, Y.; Bengio, Y.; Hinton, G. Deep learning. *nature* **2015**, *521*, 436–444.
167. Rusk, N. Deep learning. *Nature Methods* **2016**, *13*, 35–35.
168. Liu, W.; Wang, Z.; Liu, X.; Zeng, N.; Liu, Y.; Alsaadi, F.E. A survey of deep neural network architectures and their applications. *Neurocomputing* **2017**, *234*, 11–26.
169. Kamilaris, A.; Prenafeta-Boldú, F.X. Deep learning in agriculture: A survey. *Computers and electronics in agriculture* **2018**, *147*, 70–90.
170. Khan, S.; Tufail, M.; Khan, M.T.; Khan, Z.A.; Iqbal, J.; Wasim, A. Real-time recognition of spraying area for UAV sprayers using a deep learning approach. *Plos one* **2021**, *16*, e0249436.
171. Saleem, M.H.; Potgieter, J.; Arif, K.M. Automation in agriculture by machine and deep learning techniques: A review of recent developments. *Precision Agriculture* **2021**, *22*, 2053–2091.
172. Hu, K.; Coleman, G.; Zeng, S.; Wang, Z.; Walsh, M. Graph weeds net: A graph-based deep learning method for weed recognition. *Computers and electronics in agriculture* **2020**, *174*, 105520.
173. Godara, S.; Toshniwal, D. Deep Learning-based query-count forecasting using farmers' helpline data. *Computers and Electronics in Agriculture* **2022**, *196*, 106875.
174. Altalak, M.; Alajmi, A.; Rizg, A.; et al. Smart Agriculture Applications Using Deep Learning Technologies: A Survey. *Applied Sciences* **2022**, *12*, 5919.
175. Hryniowski, A.; Wong, A. DeepLABNet: end-to-end learning of deep radial basis networks with fully learnable basis functions. *arXiv preprint arXiv:1911.09257* **2019**.
176. Li, Y.; Zhang, T. Deep neural mapping support vector machines. *Neural Networks* **2017**, *93*, 185–194.
177. Zhang, Y.; Ishibuchi, H.; Wang, S. Deep Takagi–Sugeno–Kang fuzzy classifier with shared linguistic fuzzy rules. *IEEE Transactions on Fuzzy Systems* **2017**, *26*, 1535–1549.
178. Das, R.; Sen, S.; Maulik, U. A survey on fuzzy deep neural networks. *ACM Computing Surveys (CSUR)* **2020**, *53*, 1–25.
179. Yu, Y.; Si, X.; Hu, C.; Zhang, J. A review of recurrent neural networks: LSTM cells and network architectures. *Neural computation* **2019**, *31*, 1235–1270.
180. Niu, Z.; Zhong, G.; Yu, H. A review on the attention mechanism of deep learning. *Neurocomputing* **2021**, *452*, 48–62.
181. Wang, Y.; Wang, H.; Peng, Z. Rice diseases detection and classification using attention based neural network and bayesian optimization. *Expert Systems with Applications* **2021**, *178*, 114770.
182. Navada, A.; Ansari, A.N.; Patil, S.; Sonkamble, B.A. Overview of use of decision tree algorithms in machine learning. In Proceedings of the 2011 IEEE control and system graduate research colloquium. IEEE, 2011, pp. 37–42.
183. Loh, W.Y. Classification and regression trees. *Wiley interdisciplinary reviews: data mining and knowledge discovery* **2011**, *1*, 14–23.
184. Chen, X.; Wang, B.; Gao, Y. Symmetric Binary Tree Based Co-occurrence Texture Pattern Mining for Fine-grained Plant Leaf Image Retrieval. *Pattern Recognition* **2022**, *129*, 108769.
185. Saggi, M.K.; Jain, S. Reference evapotranspiration estimation and modeling of the Punjab Northern India using deep learning. *Computers and Electronics in Agriculture* **2019**, *156*, 387–398.
186. Zhang, L.; Traore, S.; Ge, J.; Li, Y.; Wang, S.; Zhu, G.; Cui, Y.; Fipps, G. Using boosted tree regression and artificial neural networks to forecast upland rice yield under climate change in Sahel. *Computers and Electronics in Agriculture* **2019**, *166*, 105031.
187. Segal, M.R. Machine learning benchmarks and random forest regression **2004**.
188. Cutler, A.; Cutler, D.R.; Stevens, J.R. Random forests. In *Ensemble machine learning*; Springer, 2012; pp. 157–175.

189. da Silva Júnior, J.C.; Medeiros, V.; Garrozi, C.; Montenegro, A.; Gonçalves, G.E. Random forest techniques for spatial interpolation of evapotranspiration data from Brazilian's Northeast. *Computers and electronics in agriculture* **2019**, *166*, 105017.
190. Zhang, Y.; Sui, B.; Shen, H.; Ouyang, L. Mapping stocks of soil total nitrogen using remote sensing data: A comparison of random forest models with different predictors. *Computers and Electronics in Agriculture* **2019**, *160*, 23–30.
191. Amirruddin, A.D.; Muharam, F.M.; Ismail, M.H.; Ismail, M.F.; Tan, N.P.; Karam, D.S. Hyperspectral remote sensing for assessment of chlorophyll sufficiency levels in mature oil palm (Elaeis guineensis) based on frond numbers: Analysis of decision tree and random forest. *Computers and electronics in agriculture* **2020**, *169*, 105221.
192. Karimi, S.; Shiri, J.; Marti, P. Supplanting missing climatic inputs in classical and random forest models for estimating reference evapotranspiration in humid coastal areas of Iran. *Computers and Electronics in Agriculture* **2020**, *176*, 105633.
193. Obsie, E.Y.; Qu, H.; Drummond, F. Wild blueberry yield prediction using a combination of computer simulation and machine learning algorithms. *Computers and Electronics in Agriculture* **2020**, *178*, 105778.
194. Ramos, A.P.M.; Osco, L.P.; Furuya, D.E.G.; Gonçalves, W.N.; Santana, D.C.; Teodoro, L.P.R.; da Silva Junior, C.A.; Capristo-Silva, G.F.; Li, J.; Baio, F.H.R.; et al. A random forest ranking approach to predict yield in maize with uav-based vegetation spectral indices. *Computers and Electronics in Agriculture* **2020**, *178*, 105791.
195. Rastgou, M.; Bayat, H.; Mansoorizadeh, M.; Gregory, A.S. Estimating the soil water retention curve: Comparison of multiple nonlinear regression approach and random forest data mining technique. *Computers and Electronics in Agriculture* **2020**, *174*, 105502.
196. dos Santos Luciano, A.C.; Picoli, M.C.A.; Duft, D.G.; Rocha, J.V.; Leal, M.R.L.V.; Le Maire, G. Empirical model for forecasting sugarcane yield on a local scale in Brazil using Landsat imagery and random forest algorithm. *Computers and Electronics in Agriculture* **2021**, *184*, 106063.
197. Mariano, C.; Monica, B. A random forest-based algorithm for data-intensive spatial interpolation in crop yield mapping. *Computers and Electronics in Agriculture* **2021**, *184*, 106094.
198. Dhaliwal, J.K.; Panday, D.; Saha, D.; Lee, J.; Jagadamma, S.; Schaeffer, S.; Mengistu, A. Predicting and interpreting cotton yield and its determinants under long-term conservation management practices using machine learning. *Computers and Electronics in Agriculture* **2022**, *199*, 107107.
199. Yoo, B.H.; Kim, K.S.; Park, J.Y.; Moon, K.H.; Ahn, J.J.; Fleisher, D.H. Spatial portability of random forest models to estimate site-specific air temperature for prediction of emergence dates of the Asian Corn Borer in North Korea. *Computers and Electronics in Agriculture* **2022**, *199*, 107113.
200. Elavarasan, D.; Vincent, D.R.; Sharma, V.; Zomaya, A.Y.; Srinivasan, K. Forecasting yield by integrating agrarian factors and machine learning models: A survey. *Computers and Electronics in Agriculture* **2018**, *155*, 257–282.
201. Huang, G.B.; Zhou, H.; Ding, X.; Zhang, R. Extreme Learning Machine for Regression and Multiclass Classification. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* **2012**, *42*, 513–529. https://doi.org/10.1109/TSMCB.2011.2168604.
202. Ding, S.; Zhao, H.; Zhang, Y.; Xu, X.; Nie, R. Extreme learning machine: algorithm, theory and applications. *Artificial Intelligence Review* **2015**, *44*, 103–115.
203. Huang, G.; Huang, G.B.; Song, S.; You, K. Trends in extreme learning machines: A review. *Neural Networks* **2015**, *61*, 32–48.
204. Mohammadi, K.; Shamshirband, S.; Motamedi, S.; Petković, D.; Hashim, R.; Gocic, M. Extreme learning machine based prediction of daily dew point temperature. *Computers and Electronics in Agriculture* **2015**, *117*, 214–225.
205. Gocic, M.; Petković, D.; Shamshirband, S.; Kamsin, A. Comparative analysis of reference evapotranspiration equations modelling by extreme learning machine. *Computers and Electronics in Agriculture* **2016**, *127*, 56–63.
206. Patil, A.P.; Deka, P.C. An extreme learning machine approach for modeling evapotranspiration using extrinsic inputs. *Computers and Electronics in Agriculture* **2016**, *121*, 385–392.
207. Feng, Y.; Peng, Y.; Cui, N.; Gong, D.; Zhang, K. Modeling reference evapotranspiration using extreme learning machine and generalized regression neural network only with temperature data. *Computers and Electronics in Agriculture* **2017**, *136*, 71–78.
208. Sadgrove, E.J.; Falzon, G.; Miron, D.; Lamb, D. Fast object detection in pastoral landscapes using a colour feature extreme learning machine. *Computers and electronics in agriculture* **2017**, *139*, 204–212.
209. Ali, M.; Deo, R.C.; Downs, N.J.; Maraseni, T. Multi-stage committee based extreme learning machine model incorporating the influence of climate parameters and seasonality on drought forecasting. *Computers and electronics in agriculture* **2018**, *152*, 149–165.
210. Shi, P.; Li, G.; Yuan, Y.; Huang, G.; Kuang, L. Prediction of dissolved oxygen content in aquaculture using Clustering-based Softplus Extreme Learning Machine. *Computers and electronics in agriculture* **2019**, *157*, 329–338.
211. Gong, D.; Hao, W.; Gao, L.; Feng, Y.; Cui, N. Extreme learning machine for reference crop evapotranspiration estimation: Model optimization and spatiotemporal assessment across different climates in China. *Computers and Electronics in Agriculture* **2021**, *187*, 106294.
212. Nahvi, B.; Habibi, J.; Mohammadi, K.; Shamshirband, S.; Al Razgan, O.S. Using self-adaptive evolutionary algorithm to improve the performance of an extreme learning machine for estimating soil temperature. *Computers and Electronics in Agriculture* **2016**, *124*, 150–160.
213. Wu, L.; Huang, G.; Fan, J.; Ma, X.; Zhou, H.; Zeng, W. Hybrid extreme learning machine with meta-heuristic algorithms for monthly pan evaporation prediction. *Computers and electronics in agriculture* **2020**, *168*, 105115.
214. Zhu, B.; Feng, Y.; Gong, D.; Jiang, S.; Zhao, L.; Cui, N. Hybrid particle swarm optimization with extreme learning machine for daily reference evapotranspiration prediction from limited climatic data. *Computers and Electronics in Agriculture* **2020**, *173*, 105430.

215. Yu, W.; Zhuang, F.; He, Q.; Shi, Z. Learning deep representations via extreme learning machines. *Neurocomputing* **2015**, *149*, 308–315.
216. Tissera, M.D.; McDonnell, M.D. Deep extreme learning machines: supervised autoencoding architecture for classification. *Neurocomputing* **2016**, *174*, 42–49.
217. Abdelghafour, F.; Rosu, R.; Keresztes, B.; Germain, C.; Da Costa, J.P. A Bayesian framework for joint structure and colour based pixel-wise classification of grapevine proximal images. *Computers and Electronics in Agriculture* **2019**, *158*, 345–357.
218. Khanal, A.R.; Mishra, A.K.; Lambert, D.M.; Paudel, K.P. Modeling post adoption decision in precision agriculture: A Bayesian approach. *Computers and Electronics in Agriculture* **2019**, *162*, 466–474.
219. Tetteh, G.O.; Gocht, A.; Conrad, C. Optimal parameters for delineating agricultural parcels from satellite images based on supervised Bayesian optimization. *Computers and Electronics in Agriculture* **2020**, *178*, 105696.
220. Fang, Y.; Xu, L.; Chen, Y.; Zhou, W.; Wong, A.; Clausi, D.A. A Bayesian Deep Image Prior Downscaling Approach for High-Resolution Soil Moisture Estimation. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* **2022**, *15*, 4571–4582.
221. Koller, D.; Friedman, N. *Probabilistic graphical models: principles and techniques*; MIT press, 2009.
222. Hrycej, T. Gibbs sampling in Bayesian networks. *Artificial Intelligence* **1990**, *46*, 351–363.
223. Chapman, R.; Cook, S.; Donough, C.; Lim, Y.L.; Ho, P.V.V.; Lo, K.W.; Oberthür, T. Using Bayesian networks to predict future yield functions with data from commercial oil palm plantations: A proof of concept analysis. *Computers and electronics in agriculture* **2018**, *151*, 338–348.
224. Kocian, A.; Massa, D.; Cannazzaro, S.; Incrocci, L.; Di Lonardo, S.; Milazzo, P.; Chessa, S. Dynamic Bayesian network for crop growth prediction in greenhouses. *Computers and electronics in agriculture* **2020**, *169*, 105167.
225. Bilmes, J.A.; et al. A gentle tutorial of the EM algorithm and its application to parameter estimation for Gaussian mixture and hidden Markov models. *International computer science institute* **1998**, *4*, 126.
226. Lu, J. A survey on Bayesian inference for Gaussian mixture model. *arXiv preprint arXiv:2108.11753* **2021**.
227. Mouret, F.; Albughdadi, M.; Duthoit, S.; Kouamé, D.; Rieu, G.; Tourneret, J.Y. Reconstruction of Sentinel-2 derived time series using robust Gaussian mixture models—Application to the detection of anomalous crop development. *Computers and Electronics in Agriculture* **2022**, *198*, 106983.
228. Zhu, C.; Ding, J.; Zhang, Z.; Wang, J.; Wang, Z.; Chen, X.; Wang, J. SPAD monitoring of saline vegetation based on Gaussian mixture model and UAV hyperspectral image feature classification. *Computers and Electronics in Agriculture* **2022**, *200*, 107236.
229. Quinonero-Candela, J.; Rasmussen, C.E. A unifying view of sparse approximate Gaussian process regression. *The Journal of Machine Learning Research* **2005**, *6*, 1939–1959.
230. Wilson, A.G.; Knowles, D.A.; Ghahramani, Z. Gaussian process regression networks. *arXiv preprint arXiv:1110.4411* **2011**.
231. Smola, A.; Bartlett, P. Sparse greedy Gaussian process regression. *Advances in neural information processing systems* **2000**, *13*.
232. Azadbakht, M.; Ashourloo, D.; Aghighi, H.; Radiom, S.; Alimohammadi, A. Wheat leaf rust detection at canopy scale under different LAI levels using machine learning techniques. *Computers and Electronics in Agriculture* **2019**, *156*, 119–128.
233. Shabani, S.; Samadianfard, S.; Sattari, M.T.; Shamshirband, S.; Mosavi, A.; Kmet, T.; Várkonyi-Kóczy, A.R. Modeling daily pan evaporation in humid climates using gaussian process regression. *arXiv preprint arXiv:1908.04267* **2019**.
234. Nieto, P.G.; García-Gonzalo, E.; Puig-Bargués, J.; Solé-Torres, C.; Duran-Ros, M.; Arbat, G. A new predictive model for the outlet turbidity in micro-irrigation sand filters fed with effluents using Gaussian process regression. *Computers and electronics in agriculture* **2020**, *170*, 105292.
235. Rastgou, M.; Bayat, H.; Mansoorizadeh, M.; Gregory, A.S. Prediction of soil hydraulic properties by Gaussian process regression algorithm in arid and semiarid zones in Iran. *Soil and Tillage Research* **2021**, *210*, 104980.
236. Nguyen, L.; Nguyen, D.K.; Nghiem, T.X.; Nguyen, T. Least square and Gaussian process for image based microalgal density estimation. *Computers and Electronics in Agriculture* **2022**, *193*, 106678.
237. Zhang, C.; Ma, Y. *Ensemble machine learning: methods and applications*; Springer, 2012.
238. Sagi, O.; Rokach, L. Ensemble learning: A survey. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* **2018**, *8*, e1249.
239. Zhou, Z.H. Ensemble learning. In *Machine learning*; Springer, 2021; pp. 181–210.
240. Chaudhary, A.; Kolhe, S.; Kamal, R. A hybrid ensemble for classification in multiclass datasets: An application to oilseed disease dataset. *Computers and Electronics in Agriculture* **2016**, *124*, 65–72.
241. Haagsma, M.; Page, G.F.; Johnson, J.S.; Still, C.; Waring, K.M.; Sniezko, R.A.; Selker, J.S. Model selection and timing of acquisition date impacts classification accuracy: A case study using hyperspectral imaging to detect white pine blister rust over time. *Computers and Electronics in Agriculture* **2021**, *191*, 106555.
242. Kar, S.; Purbey, V.K.; Suradhaniwar, S.; Korbu, L.B.; Kholová, J.; Durbha, S.S.; Adinarayana, J.; Vadez, V. An ensemble machine learning approach for determination of the optimum sampling time for evapotranspiration assessment from high-throughput phenotyping data. *Computers and Electronics in Agriculture* **2021**, *182*, 105992.
243. Chaudhary, A.; Thakur, R.; Kolhe, S.; Kamal, R. A particle swarm optimization based ensemble for vegetable crop disease recognition. *Computers and Electronics in Agriculture* **2020**, *178*, 105747.
244. Chia, M.Y.; Huang, Y.F.; Koo, C.H. Improving reference evapotranspiration estimation using novel inter-model ensemble approaches. *Computers and Electronics in Agriculture* **2021**, *187*, 106227.

245. Wu, T.; Zhang, W.; Jiao, X.; Guo, W.; Hamoud, Y.A. Evaluation of stacking and blending ensemble learning methods for estimating daily reference evapotranspiration. *Computers and Electronics in Agriculture* **2021**, *184*, 106039.
246. Koyama, K.; Lyu, S. Soft-labeling approach along with an ensemble of models for predicting subjective freshness of spinach leaves. *Computers and Electronics in Agriculture* **2022**, *193*, 106633.
247. Xu, C.; Ding, J.; Qiao, Y.; Zhang, L. Tomato disease and pest diagnosis method based on the Stacking of prescription data. *Computers and Electronics in Agriculture* **2022**, *197*, 106997.
248. Aiken, V.C.F.; Dórea, J.R.R.; Acedo, J.S.; de Sousa, F.G.; Dias, F.G.; de Magalhães Rosa, G.J. Record linkage for farm-level data analytics: comparison of deterministic, stochastic and machine learning methods. *Computers and Electronics in Agriculture* **2019**, *163*, 104857.
249. Weber, V.A.M.; de Lima Weber, F.; da Silva Oliveira, A.; Astolfi, G.; Menezes, G.V.; de Andrade Porto, J.V.; Rezende, F.P.C.; de Moraes, P.H.; Matsubara, E.T.; Mateus, R.G.; et al. Cattle weight estimation using active contour models and regression trees Bagging. *Computers and Electronics in Agriculture* **2020**, *179*, 105804.
250. Genedy, R.A.; Ogejo, J.A. Using machine learning techniques to predict liquid dairy manure temperature during storage. *Computers and Electronics in Agriculture* **2021**, *187*, 106234.
251. Mohammed, S.; Elbeltagi, A.; Bashir, B.; Alsafadi, K.; Alsilibe, F.; Alsalman, A.; Zeraatpisheh, M.; Széles, A.; Harsányi, E. A comparative analysis of data mining techniques for agricultural and hydrological drought prediction in the eastern Mediterranean. *Computers and Electronics in Agriculture* **2022**, *197*, 106925.
252. Ayan, E.; Erbay, H.; Varçın, F. Crop pest classification with a genetic algorithm-based weighted ensemble of deep convolutional neural networks. *Computers and Electronics in Agriculture* **2020**, *179*, 105809.
253. Barbosa, A.; Hovakimyan, N.; Martin, N.F. Risk-averse optimization of crop inputs using a deep ensemble of convolutional neural networks. *Computers and Electronics in Agriculture* **2020**, *178*, 105785.
254. e Lucas, P.d.O.; Alves, M.A.; e Silva, P.C.d.L.; Guimarães, F.G. Reference evapotranspiration time series forecasting with ensemble of convolutional neural networks. *Computers and electronics in agriculture* **2020**, *177*, 105700.
255. Gonzalo-Martín, C.; García-Pedrero, A.; Lillo-Saavedra, M. Improving deep learning sorghum head detection through test time augmentation. *Computers and Electronics in Agriculture* **2021**, *186*, 106179.
256. Gu, Z.; Zhu, T.; Jiao, X.; Xu, J.; Qi, Z. Neural network soil moisture model for irrigation scheduling. *Computers and Electronics in Agriculture* **2021**, *180*, 105801.
257. Khanramaki, M.; Asli-Ardeh, E.A.; Kozegar, E. Citrus pests classification using an ensemble of deep learning models. *Computers and Electronics in Agriculture* **2021**, *186*, 106192.
258. Li, Q.; Jia, W.; Sun, M.; Hou, S.; Zheng, Y. A novel green apple segmentation algorithm based on ensemble U-Net under complex orchard environment. *Computers and Electronics in Agriculture* **2021**, *180*, 105900.
259. Gonzalez, R.; Iagnemma, K. Slippage estimation and compensation for planetary exploration rovers. State of the art and future challenges. *Journal of Field Robotics* **2018**, *35*, 564–577. https://doi.org/10.1002/rob.21761.
260. Gonzalez, R.; Chandler, S.; Apostolopoulos, D. Characterization of machine learning algorithms for slippage estimation in planetary exploration rovers. *Journal of Terramechanics* **2019**, *82*, 23–34. https://doi.org/10.1016/j.jterra.2018.12.001.