Article

Modeling the Settling Velocity of a Sphere in Newtonian and non-Newtonian Fluids with Machine-learning Algorithms

Sayeed Rushd 1,*, Noor Hafsa 2, Majdi Al-Faiad 1 and Md Arifuzzaman 3

1 Department of Chemical Engineering, College of Engineering, King Faisal University, PO Box 380, Al Ahsa 31982, Saudi Arabia; malfaiad@kfu.edu.sa
2 Department of Computer Science, College of Computer Science and Information Technology, King Faisal University, PO Box 400, Al Ahsa 31982, Saudi Arabia; nhafsa@kfu.edu.sa
3 Department of Civil Engineering, College of Engineering, King Faisal University, PO Box 380, Al Ahsa 31982, Saudi Arabia; marifuzzaman@kfu.edu.sa
* Correspondence: mrushd@kfu.edu.sa

Abstract: The traditional procedure of predicting the settling velocity of a spherical particle is inconvenient as it involves iterations, complex correlations, and an unpredictable degree of uncertainty. The limitations can be addressed efficiently with artificial intelligence-based machine-learning algorithms (MLAs). The limited number of isolated studies conducted to date were constrained to specific fluid rheology, a particular MLA, and insufficient data. In the current study, the generalized application of ML was comprehensively investigated for Newtonian and three varieties of non-Newtonian fluids such as Power-law, Bingham, and Herschel Bulkley. A diverse set of nine MLAs were trained and tested using a large dataset of 967 samples. The ranges of generalized particle Reynolds number (Re<sub>o</sub>) and drag coefficient (C<sub>D</sub>) for the dataset were 10<sup>3</sup> < Re<sub>o</sub> < 10<sup>4</sup> and 10<sup>1</sup> < C<sub>D</sub> < 10<sup>4</sup>, respectively. The performances of the models were statistically evaluated using an evaluation metric of the coefficient-of-determination (R<sup>2</sup>), root-mean-square-error (RMSE), mean-squared-error (MSE), and mean-absolute-error (MAE). The support vector regression with polynomial kernel demonstrated the optimum performance with R<sup>2</sup> = 0.92, RMSE = 0.066, MSE = 0.0044, and MAE = 0.044. Its generalization capability was validated using the ten-fold-cross-validation technique, leave-one-feature-out experiment, and leave-one-data-set-out validation. The outcome of the current investigation was a generalized approach to modeling the settling velocity.

Keywords: artificial intelligence; solid particle; support vector machine; statistics; modeling; ten-fold-cross-validation; leave-one-out feature analysis

1. Introduction

Settling velocity (V<sub>s</sub>) is the constant free-falling velocity of a solid particle in a stationary liquid when the opposing gravity and drag forces acting on the particle approximately equals one another. It is a significant parameter in the industries that involve solid–liquid two-phase flows. Probably, the most important examples are the slurry transportation in minerals and coal processing, wastewater treatment, dredging, hydraulic fracturing, and drilling operations. It is also essential to the fluidized bed as applied for catalysis, adsorption, ion exchange, water softening, and food processing. The concept of V<sub>s</sub> is also vital in mixing operations. In these applications, the settling velocities of amphorous particles experiencing hindrances from the walls, adjacent particles, and the flow conditions in both Newtonian and non-Newtonian liquids are of practical interest. Interestingly, this kind of non-ideal settling velocity in complex operating conditions can be correlated to the ideal V<sub>s</sub> of a spherical particle measured with a lab-scale setup or predicted using a reliable model. That is why both measuring and predicting V<sub>s</sub> in all kinds of fluids are active research interests since the 1850s [1–25].
Correlating the settling velocity of spherical particles to the measurable properties of liquid and solid was the primary focus of the previous studies. Most of the existing correlations were developed based on the following non-dimensional groups for Newtonian fluids:

Drag coefficient:

\[
C_D = \frac{\text{gravitational force}}{\text{inertial force}} = \frac{4(\rho_s - \rho_l)d \cdot g}{3V_s^2 \rho_l}
\]

Particle Reynolds number:

\[
Re_p = \frac{\text{inertial force}}{\text{viscous force}} = \frac{d \cdot V_s \rho_l}{\mu_i}
\]

where,

\(\rho_s\): solid density (kg/m³)

\(\rho_l\): liquid density (kg/m³)

\(d\): particle diameter (m)

\(g\): gravitational acceleration (m/s²)

\(\mu\): viscosity of Newtonian liquid (Pa.s)

The hydrodynamic complexities associated with \(V_s\) are considered to be sufficiently captured with \(C_D\) and \(Re_p\). The idea of correlating \(C_D\) to \(Re_p\) was pioneered by Stokes with a single-parameter equation: \(C_D = \frac{24}{Re_p}\) [9]. The limitation of this model is that it is applicable only if \(Re_p < 0.1\). Based on the Stokes law, many empirical and semi-empirical correlations have been developed afterward. For example, the \(C_D-\text{Re}_p\) correlation proposed by Cheng was found to perform better compared to seven (7) other similar models [4]. All of these models were developed for spherical particles in Newtonian fluids. The Cheng correlation is presented with Equation (3).

\[
C_D = \frac{24}{Re_p} \left( 1 + 0.27 \cdot Re_p^{0.43} \right) + 0.47 \left[ 1 - \exp(-0.04 \cdot Re_p^{0.38}) \right], Re_p < 2 \times 10^5
\]

Although applying this kind of correlation is comparatively easier for a Newtonian fluid, a similar application to a non-Newtonian fluid is not convenient at all. Modeling complex non-Newtonian rheology is a difficult task [2,5,7,8,12,16,19–21,23–27]. Nevertheless, a \(C_D-\text{Re}_p\) correlation can be used to predict \(V_s\) in a non-Newtonian fluid by replacing \(\text{Re}_p\) with a generalized particle Reynolds number (\(\text{Re}_C\)) [6,7,12,16,19,20]. The equivalent viscosity used to define \(\text{Re}_C\) is usually measured or estimated at a shear rate \((\dot{\gamma})\) of \(\frac{V_s}{d}\). The \(\text{Re}_C\) for a Herschel Bulkley (HB) fluid is as follows [7,12]:

\[
\text{Re}_C = \frac{d^n V_s^2 - n \rho_l}{\tau_o \left( \frac{d}{V_s} \right)^{\frac{n}{K}} + K}
\]

where,

\(\tau_o\): yield stress (Pa)

\(\tau\): shear stress (Pa)

\(K\): fluid consistency index (-)

\(n\): flow behavior index (-)

The non-Newtonian rheology of an HB fluid is defined with Equation (5). It can be transformed into proper models for other rheologies such as Newtonian (\(\tau = K \dot{\gamma}\)), Bingham (\(\tau = \tau_o + K \dot{\gamma}\)), and power-law (\(\tau = K \dot{\gamma}^n\)) by considering the appropriate values of \(\tau_o\), \(K\), and \(n\). It should be mentioned that, even after transforming a \(C_D-\text{Re}_p\) correlation with \(\text{Re}_C\), the modified formula requires validation and further modification before applying to
predict the $V_s$ in a non-Newtonian fluid (see, for example, [2,7,19,20]). This is because the uncertainty associated with the values predicted using the original $C_{Dr-ReC}$ correlation can be unacceptable. The prediction process also requires repeated iterations as $V_s$ is an implicit function of both dimensionless groups. The iterative procedure is inconvenient from an engineering perspective. Many direct correlations for $V_s$ have been developed to avoid the difficulties associated with implicit $C_{Dr-ReC}$ models. For example, a total of twenty-six (26) explicit correlations were reported by Agwu et al. [1]. Explicit models are, in general, convenient to apply. However, this kind of correlation has some noteworthy limitations, such as rheology specificity, higher uncertainty, constrained range of $Re_C$, and limited capability of addressing the effect of $\tau$ [1,2,6,7,11,16,19,20].

A common objective of developing explicit and implicit models was to advance a generalized approach to predict $V_s$ accurately and conveniently in both Newtonian and non-Newtonian fluids. However, the objective is yet to be achieved. A successful generalized model is not available in the literature. This particular limitation of traditional models opens up the opportunity of applying contemporary artificial intelligence (AI) to predict $V_s$. Different AI-based methodologies were successfully implemented in solving various scientific problems [28-37]. However, only a limited number of investigations were conducted earlier to check the efficacy of the predictive AI-modules known as the machine learning algorithms (MLAs) in predicting settling velocity [13]. Rooki et al. [14] and Agwu et al. [1] applied one of the popular MLAs, artificial neural network (ANN) based on a commercial computational platform, MATLAB, to predict $V_s$ using datasets comprised of 88 and 676 samples, respectively. Only 70 data points were used for training, while 18 points were reported to be used for testing in Reference [14]. Two independent datasets consisting of 336 and 340 samples were used for modeling and validation, respectively, in Reference [1]. The 336 modeling data were divided further into training (70%), testing (15%), and validating (15%) subsets. That is, only 235 samples were used for training. Even though the performance of ANN was demonstrated to be significantly better than the traditional models, its application was not justified by comparing with other MLAs. For both of these studies, the AI-model was selected arbitrarily without statistical analysis. Besides, the proposed ANN architectures were complex and prone to overfitting.

The knowledge gaps existing in the studies performed earlier in the field of modeling $V_s$ can be summarized as follows:

- The traditional models for predicting $V_s$ are categorized as implicit and explicit models.
  - a. The semi-mechanistic $C_{Dr-ReC}$ models involve implicit correlations of $V_s$. Predicting $V_s$ using such correlations is inconvenient from an engineering perspective as it demands iterations.
  - b. Most of the existing implicit correlations were developed for Newtonian fluids. The extension of these models to non-Newtonian fluids involves higher uncertainties.
  - c. Considering the inconvenience of implicit models, many explicit correlations were proposed for $V_s$. This kind of direct model involves complex empirical correlations, which are usually rheology-specific and involve a high degree of uncertainty.
  - d. A generalized traditional model applicable to various fluid rheologies is not available in the literature to date.

- Limited efforts have been undertaken to apply AI-based MLAs to develop a generalized model for predicting $V_s$ of spheres in both Newtonian and non-Newtonian fluids. The previous ML investigations were confined to a limited set of fluid rheology, a specific MLA (ANN), and insufficient data.

In this regard, a comprehensive investigation was conducted in the current study on the AI-based prediction of $V_s$. A diverse set of MLAs including support vector vector regression
(SVR), random forest (RF), stochastic gradient boosting (SGB), Bayesian additive regression tree (BART), K-nearest neighbor (KNN) regression, multilayer perceptron (MLP), and ANN were applied. The ten-fold-cross-validation technique with an evaluation metric of a group of statistical parameters was used in identifying the most optimum AI-module. The statistical parameters were mean squared error (MSE), coefficient of determination ($R^2$), mean absolute error (MAE), and root mean square error (RMSE). A dataset of 967 data points collected from the literature was used for the current analysis. Both Newtonian and non-Newtonian fluids were used for the wet-experiments. The non-Newtonian rheology of the experimental fluids was explained with Bingham, Power-law, and Herschel-Buckley models. The leave-one-feature-out analysis was performed to analyze the feature influences on the prediction. Moreover, a leave-one-dataset-out validation was carried out to examine the generalization capability of the optimal model in the prediction of $V_s$ for different types of fluids. All dry-experiments and statistical analyses were conducted using an open-source computation platform, R. The generalized model validated as part of the current study is expected to lead to the development of a universal model to predicting $V_s$ with a decision support system (DSS) for the industry.

2. Materials and Methods

2.1. Regression Algorithms

Nine regression algorithms, ranging from parametric and non-parametric models to Bayesian regression were tested to perform the ML modeling of the settling velocity. Specifically, support vector regression with radial basis function (SVR-RBF) kernel, SVR with polynomial (SVR-polynomial) kernel, and SVR with linear (SVR-linear) kernel along with RF regression, SGB, BART, KNN regression, MLP, and ANN algorithms were selected for the current study. A brief explanation of each regression algorithm is presented below highlighting the statistical formulation of the estimated response with the input parameters [30–44].

2.1.1. SVR-Radial Basis Function

Given a training set $(X, Y) = (x_1, y_1), (x_2, y_2), ..., (x_N, y_N)$, the SVR attempts to find a hypothesis $f(x)$ such that it will be insensitive to at most ε-deviations from the actual responses in the training data and as flat as possible. The output of $f(x)$ or the predicted response, denoted as $\hat{y}$, can be described by the following expression:

$$\hat{y} = (w, \varphi(x)) + b$$  \hspace{1cm} (6)

where, $(w, \varphi(x))$ denotes the dot product of weight vector $(w)$ and mapped feature vector after applying the non-linear transformation function $\varphi(x)$. With $a_i$ coefficients as support vectors and the selection of a suitable kernel function for non-linear feature mapping, $k(x_i, x)$, the equation for $\hat{y}$ can be described as:

$$\hat{y} = \sum_{i=1}^{N} (a_i k(x_i, x)) + b$$  \hspace{1cm} (7)

For RBF Kernel, $K(x, x')$ measures the similarity between two feature vectors $x$ and $x'$ as a decaying function (squared norm) of the distance between the vectors. In this case, the more similar the features the smaller the kernel. The RBF kernel can be expressed as:

$$K(x, x') = \exp \left( -\frac{1}{2\sigma^2} \|x - x'\|^2 \right)$$  \hspace{1cm} (8)

where $\sigma$ is the radius or width of the RBF kernel and indicates the influence or reach of the features on the model. The $C$ and $\sigma$ are two tuning parameters to optimize in RBF kernel-based SVR, where $C$ is the regularization parameter of the objective function that we want to minimize based on the training dataset. The objective loss refers to the difference between actual and predicted responses.
2.1.2. SVR-Polynomial

Another popular version of non-linear SVR is the SVR-Polynomial algorithm, where a polynomial kernel function is used for learning non-linear feature interactions. Under the SVR regularized objective function framework, the polynomial kernel measures the similarity between two column vectors using a degree-d polynomial equation. Including the kernel function, the response \( \hat{y} \) can be expressed as follows:

\[
\hat{y} = \sum_{i=1}^{N} a_i (\gamma x^T x_i + c)^d + b \quad (9)
\]

where \( x \) and \( x' \) are two column vectors, \( \gamma \) is a scalar parameter, \( c \) and \( b \) are constants, and \( d \) is the kernel degree. The tuning parameters for SVR-Polynomial are degree \( (d) \), scale \( (\gamma) \), and regularization parameter \( (C) \).

2.1.3. SVR-Linear

The SVR-Linear or SVR with the linear kernel is a special case of SVR-polynomial, in which degree \( d = 0 \) and coefficient \( c = 0 \) in Equation (9). Therefore, the similarity between two input vectors is measured by simply taking the inner product between them. The linear kernel assumes a linear relationship among the features, therefore, aims to learn the feature interaction in the original input space. The estimated response \( \hat{y} \) for SVR-Linear can be expressed as:

\[
\hat{y} = \sum_{i=1}^{N} a_i x^T x_i + b \quad (10)
\]

The SVR-Linear training optimizes two tuning parameters, \( C \) and the loss used in the objective function.

2.1.4. Random Forest Regression

The RF is an ensemble method that uses a collection of regression trees and introduces randomization to choose the variable splits at each node of the tree. The algorithm starts with partitioning the training data into \( M \) subsets using bootstrap sampling. Then, a single regression tree (\( f_i \)) is fit for each subset by using a random subset of features for splitting at each node, which results in a forest of \( M \) regression trees. After fitting an RF model to the whole training data, the prediction of response \( (\hat{y}) \) for an unseen test sample (\( x' \)) can be made by averaging all the individual regression tree predictions on \( x' \):

\[
\hat{y} = \frac{1}{M} \sum_{i=1}^{M} f_i (x') \quad (11)
\]

where \( M \) is the total number of regression trees in the RF model and \( f_i (x') \) is a single regression tree output on \( x' \). The hyper-parameter for RF optimization is \( mtry \), which indicates the number of predictor variables randomly selected as candidates at each split in the tree.

2.1.5. Stochastic Gradient Boosting

In a Simple Gradient Boosting algorithm, an additive regression model is constructed by sequentially fitting a simple parameterized function as a base learner to the gradient of the loss, i.e., the error between actual and predicted response. Instead of using the full sample, a randomly selected subsample is used to minimize the loss at each iteration in the stochastic version. In the case of gradient tree boosting, the base learner could be as simple as a regression tree. The functional form of the gradient boosting based approximation of the predicted response is as follows:
\[
\hat{y} = \sum_{m=0}^{M} \beta_m f(x; a_m)
\]

where the functions \( f(x; a_m) \) are the base learners that are simple functions of \( x \) with parameters \( a = \{a_1, a_2, \ldots\} \) and \( \beta_m \) are expansion coefficients. Both \( a \) and \( \beta_m \) are jointly fit to the training data. The parameters of the base learner define the split points of the regression tree [20]. In SGB, the hyper-parameters for tuning are the number of trees \( (m) \) and the number of splits to be performed at each node, i.e., the maximum nodes per tree.

2.1.6. Bayesian Additive Regression Tree

Motivated by the ensemble methods, BART is a Bayesian approach that uses a sum-of-trees model to fit or approximate the hypothesis function that outputs the predicted response [21]. The concept is to amplify the sum-of-trees model by imposing a law that regularizes the fit by weakening the effect of the individual regression tree. In effect, each regression tree in BART becomes a weak learner that can explain a small and specific portion of the training data. Thus, BART takes advantage of the additive representation of multiple regression trees in predicting the outcome without making a single large tree unduly influential. The predicted response for a set of predictor variables \( x = (x_1, x_2, \ldots, x_n) \) could be written according to the sum-of-trees model:

\[
\hat{y} = \sum_{j=1}^{m} g(x, T_j)
\]

where \( T \) is a single binary regression tree and \( m \) is the number of regression trees. Each tree \( T \) consists of a set of interior node decision rules and a set of terminal nodes regularized by some prior parameters. In this case, the tuning parameter is \( m \) or the number of trees used in the sum-of-trees model.

2.1.7. K-Nearest Neighbor Regression

Similar to KNN-based classification, KNN regression attempts to determine the real value of a new observation by averaging the values of its \( K \)-nearest neighbors. The KNN is a non-parametric model and generally uses Euclidian distance to find the nearest neighbor cases by calculating the distance between predictors and, thereby, approximates the response by averaging the observed responses. The predicted response thus becomes:

\[
\hat{y} = \frac{1}{k} \sum_{i \in N_k(x', D)} y_i
\]

where \( k \) is the number of nearest neighbors and \( N_k(x, D) \) defines the neighborhood of the new sample \( x' \) that consists of \( K \) nearest neighbors within distance \( D \). The value of \( K \) is subject to optimization based on the training data and will greatly affect the generalization ability of the KNN model.

2.1.8. Multilayer Perceptron

The MLP is a fully connected feed-forward ANN. The architecture consists of three or more layers (one input layer, one or more hidden layers, and one output layer) of non-linearly activating nodes. Due to the fully connected architecture, each node in one layer connects with a certain weight to every other neuron to the following layer. The conventional MLP architecture with one hidden layer is shown in Figure 1. As shown in the figure, the neurons in the hidden layer apply a non-linear activation function \( \varphi \) to map the weighted inputs to the outputs of neurons, making it particularly suitable for distinguishing non-linearly separable data. The function that approximates the actual response in MLP is as follows:
\[ \hat{y} = \sum_{i=1}^{N} w_i \phi_i(x) + b_i \]  
(15)

where \( N \) is the number of neurons in the hidden layer; \( \phi_i(x) \) is the non-linear activation function applied to the input at each hidden layer neuron; \( w_i \) and \( b_i \) are the weighting coefficients and bias associated with each input, respectively. In MLP, the training is performed in a supervised learning technique called error backpropagation. The tuning parameters for MLP are the number of neurons in the hidden layer and the decay that is the learning rate for minimizing the error gradient. These parameters were used in the current study to fine-tune the model.

2.1.9. Artificial Neural Network

Similar to MLP, ANN is an interconnected architecture involving multiple neurons, organized into multiple layers. The components of ANN are generally neurons, connections with corresponding weights, and propagation function that can be either forward or backward. ANN architecture can be as shallow as MLP (three layers only) or can be very deep, commonly called Deep Neural Networks. A simple ANN involving one hidden layer uses similar equations like MLP (Equation (15)) for predicting the response variables. For a complex ANN involving several hidden layers, the equation needs to be scaled to accommodate the right number of layers. Training of an ANN involves minimizing the loss function on batches of input examples using gradient descent-like optimization algorithms until the network produces sufficiently accurate predictions. The tuning parameters for the neural network are the size of the hidden layers and the learning rate.

2.2. Evaluation Metrics

Several evaluation metrics such as \( \text{MSE} \), \( R^2 \), \( \text{RMSE} \), and \( \text{MAE} \) were used to evaluate the regression performance. Brief descriptions of these statistical parameters along with their formulations are described below.

2.2.1. Mean Squared Error

The MSE is a statistical measure that calculates the average of the squares of the errors, i.e., the average squared difference between the estimated and actual values. \( \text{MSE} \)
corresponds to the expected value of the squared error loss in the prediction. It can be described as below:

\[ MSE = \frac{1}{N} \sum_{i=1}^{N} (\hat{y}_i - y_i)^2 \]  

(16)

where \( y_1, y_2, \ldots, y_n \) are observed responses, \( \hat{y}_1, \hat{y}_2, \ldots, \hat{y}_n \) are estimated responses, and \( N \) is the total number of observations.

### 2.2.2. Coefficient of Determination

The \( R^2 \) is a statistical measure of how well a regression model fits a set of observations. It is defined as the percentage of the response variable variation that is explained by a regression model. The equation for \( R^2 \) has the following form:

\[ R^2 = \frac{\text{Explained variations}}{\text{Total variations}} \]  

(17)

The \( R^2 \) value ranges between 0 and 100%. Generally, the higher the \( R^2 \) the better the model fits the data, i.e., the model can explain most of the variances in the observed responses.

### 2.2.3. Mean Absolute Error

The MAE is a measure of the difference between two continuous variables. It is an average of absolute errors between the estimated and true responses. It can be defined with the following formula:

\[ MAE = |\hat{y} - y| \]  

(18)

where \( \hat{y} \) and \( y \) are the estimated and observed responses, respectively.

### 2.2.4. Root Mean Square Error

The RMSE is a statistical measure that calculates the standard deviation of the estimation errors. It is a standard way to calculate the error of a model in estimating real-valued responses. The formula for RMSE is as follows:

\[ RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^{N} (\hat{y}_i - y_i)^2} \]  

(19)

### 2.3. Dataset

The experimental data collected from twelve different studies were used for the current investigation. Spherical particles of different sizes were employed for the wet-experiments. The experimental details are summarized in Table 1. The system properties are presented in Table 2. A total of 967 data points was used for the current study. There were six independent or input parameters \((d, \rho_s, \rho_l, \tau_o, K, \text{and } n)\) and one dependent or output parameter \((V_s)\). A statistical description of these parameters along with the Reynolds numbers and drag coefficients is presented in Table 3.

| Reference Fluids                        | Rheological Model | Measuring Method                      | Experimental Error |
|-----------------------------------------|-------------------|---------------------------------------|--------------------|
| [2] Kaolinite-water mixtures             | Bingham           | Electrical impedance tomography (EIT)  | 2%                 |
| Water/glycerol                          |                   |                                       |                    |
| [6] Aqueous solutions of Carboxymet-hylcellulose | Newtonian Power law | Electronic stopwatch                    | 3%                 |

Table 1. Summary of the experimental details.
Table 2. Summary of datasets used to apply machine-learning algorithms (MLAs).

| Reference | Number of Data Points | Particle Diameter (mm) | Particle Density (kg/m³) | Fluid Density (kg/m³) | τ₀: 1.3–30.0 Pa | K: 0.9800–0.0074 Pa.sⁿ | n: 1 |  
|-----------|-----------------------|------------------------|--------------------------|-----------------------|-----------------|------------------------|-------|  
| [2]       | 100                   | 12.7–19.1              | 2710–7841                | 1174–1357             |                 |                        |       |  
| [6]       | 23                    | 1.22–3.16              | 2314–11,444              | 997.9                 |                 |                        |       |  
| [7]       | 532                   | 2–6                    | 2230–7700                | 1000–1030             | 1.3              | 0–3.82 Pa              | 0.7449–1 |  
| [11]      | 58                    | 6–15                   | 7638–8876                | 997–1490              |                 |                        |       |  
| [15]      | 25                    | 1.2–5                  | 7730–7949                | 1000–1044.418         |                 |                        |       |  
| [16]      | 15                    | 1.5–3.5                | 2260–2727                | 1000                  |                 |                        |       |  
| [17]      | 17                    | 3–7                    | 2500                     | 997–1000              |                 |                        |       |  
| [18]      | 8                     | 0.8–5.9                | 1170–2900                | 1000                  |                 |                        |       |  
| [19]      | 50                    | 1.09–4                 | 2510–770                | 994.0–1004.7          |                 |                        |       |  
| [20]      | 60                    | 0.71–4.0               | 2510–5900                | 996–1005              |                 |                        |       |  

Solutions of Flowzan Newtonian Herschel Bulkley Bingham High-speed camera 8%  
Aqueous solutions Floxit 5250 L Bentonite/Bicarbonate mud. Power law Electronic stopwatch 3–4%  
Bentonite/MMH mud. Xanthan gum/seawater mud Power law Electronic stopwatch 1%  
Aqueous solutions of Carboxymethyl cellulose Water/glycerol Power law Electronic stopwatch  
Aqueous solutions of Carboxymethyl cellulose A specialized lighting system  
Aqueous solutions of Carbopol Aqueous solutions of Hydrolyzed Polyacrylamide Power law Electronic stopwatch 3.5%  
Aqueous solutions of Carbopol Power law Power law Newtonian Bingham Herschel Bulkley Calculated from the bed expansion data -  
Water/glycerol Newtonian PIS 5%  
Water/glycerol Newtonian High-speed camera 5%  

Solutions of Newtonian Herschel Bulkley Bingham A specialized lighting system  
8%  

Table 3. Statistics of the input parameters.

|                        | Particle Diameter ($d_i$) | Particle Density ($\rho_i$) | Yield Stress ($\tau_0$) | Flow Consistency Index ($K$) | Flow Behavior Index ($n$) | Fluid Density ($\rho_f$) | Reynolds Number ($Re_c$) | Drag Coefficient ($Cd$) |
|------------------------|---------------------------|----------------------------|--------------------------|------------------------------|---------------------------|--------------------------|--------------------------|--------------------------|
| **Unit**               | m                         | kg/m$^3$                   | Pa                       | Pa.s$^0$                     | -                         | kg/m$^3$                 | -                        | -                        |
| **Maximum**            | 0.01910                   | 11,444                     | 30,000                   | 19.7360                     | 1.0000                    | 1490                     | 6.7 × 10$^3$             | 8.2 × 10$^4$             |
| **Minimum**            | 0.00071                   | 1170                       | 0.000                    | 0.0010                      | 0.0641                    | 994                      | 9.7 × 10$^{-4}$          | 3.1 × 10$^{-1}$          |
| **Mean**               | 0.00554                   | 4534.63                    | 2.488                    | 0.4283                      | 0.7186                    | 1063.91                  | 6.5 × 10$^2$             | 3.1 × 10$^2$             |
| **Standard Deviation** | 0.00401                   | 2370.94                    | 5.269                    | 1.7624                      | 0.2599                    | 112.42                   | 1.1 × 10$^3$             | 3.0 × 10$^3$             |

2.4. ML Modeling

The predictive power of MLA was assessed in the current study by training and testing nine regression models to estimate the settling velocity from six experimental parameters. The MLA modeling included the following steps in successive order: (i) parameter optimization and model selection, (ii) feature importance analysis, and (iii) comprehensive validation of the optimum model [33–37].

2.4.1. Parameter Optimization and Model Selection

The MLA modeling of $v_t$ prediction involved associated parameter optimization on a training set and the final model selection based on the validation using the independent test data points. At first, the datasets extracted from the literature were merged. Next, the merged dataset was split into a training (80%) and a test (20%) subset using random sampling. The training of nine ML models was carried out using the training data points (80%). The test dataset (20%), which can be considered as an independent test set, was withheld for model verification. The statistics of training and test data are presented in Table 4.

Table 4. Dataset statistics.

| Dataset | Percentage | No. Data Points |
|---------|------------|-----------------|
| Training| 80%        | 774             |
| Test    | 20%        | 193             |
| Total   | 100%       | 967             |

A grid search was integrated with a repeated $k$-fold cross-validation (CV) technique for choosing the most optimized parameters for the ML models and the final model selection [35]. Different hyper-parameters of nine ML models (see Table 5) were optimized such that average prediction error would be minimized on training data. At each fold in the CV, a model was fit on the subsets of training data. The fitted model was then evaluated on the hold-out validation dataset by calculating the RMSE of the prediction. This process was performed $k$-times with the desired number of repetitions. The model that minimized the average RMSE across the repeated CV was selected as the final model and...
its corresponding parameters were identified as the optimized ones. In a simple mathematical notation, the following optimization problem was attempted to be solved:

$$
\min_{\text{params}} \left( \sum_{i=1}^{n} \sum_{j=1}^{k} \text{RMSE}_{ij} \right)
$$

(20)

where \( n \) is the number of repetitions, \( k \) is the number of folds in CV, and \( \text{RMSE}_{ij} \) is the root mean square deviation of the predicted response from the actual response \( (V_i) \) for \( j \)-th fold in \( i \)-th iteration of the repeated CV. The algorithm used for model selection in the present study is depicted in Figure 2. Note that, the number of folds in CV \( (k) \) and repetition times \( (N) \) were considered as 10 in this study. The details of the ML models including R packages, hyper-parameter names, and their optimized values after cross-validated training are reported in Table 5.

**Algorithm: Model Selection using Repeated 10-fold Cross-Validation**

**Input** Learning model and Training set divided into equal-sized folds \( F_i \)

**Output** Final model with optimal parameters

1. **Repeat**
2. Divide the training dataset into 10 folds
3. for each fold \( F_i \)
4. Make the \( F_i \) as a validation dataset
5. Make the remaining folds as the training dataset
6. Tune the model (e.g., random forest) on the training dataset
7. Evaluate the model on the validation dataset
8. Calculate the prediction error (e.g., RMSE)
9. **End**
10. Calculate the average performance over 10 folds
11. **until** \( N = 10 \)
12. Calculate the average performance over 10 repetitions
13. Determine the Model with the best performance across the repeated CV
14. Return the Model with the optimal parameter set

**Figure 2.** Algorithmic representation of the repeated k-fold cross-validation (CV) algorithm used for parameter optimization and model selection.

**Table 5.** The ML models, hyperparameter names, and corresponding optimized values after cross-validated training (the last column includes R packages used for different ML models).

| Model                     | Hyperparameter Names | Optimized Values | R package    |
|---------------------------|----------------------|------------------|--------------|
| Random Forest             | [mtry]               | [4]              | randomForest |
| SVR—RBF Kernel           | [sigma, C]           | [0.278, 8]       | Kernlab      |
| SVR—Polynomial Kernel    | [degree, scale, C]   | [3, 0.1, 1]      | Kernlab      |
| SVR—Linear Kernel        | [Cost, Loss function]| [0.25, L1-loss]  | Kernlab      |
| Stochastic Gradient Boosting | [n.trees, interaction.depth] | [150, 3] | gbm          |
| Multilayer Perceptron    | [layer1, decay]      | [5, 0.0]         | RSNNS        |
| KNN Regression           | [no. nearest neighbors] | [5]      | Kknn         |
| Bayesian Additive Regression | [num_trees]         | [150]            | bartMachine  |
| Neural Network            | [size, decay]        | [20, 0.001]      | Neuralnet    |
2.4.2. Feature Importance Analysis and Validation

Leave-One-Feature-Out Experiment

While fitting an ML model to the training data, it triggers the question of which variables have the most predictive power or which variables have contributed most in predicting the response variable. The most important variables usually play critical roles in forecasting the response and their values have a significant impact on the predicted outcome. Less relevant features can be eliminated from the model due to their lower impacts, making the model less complex and quicker to train and predict. In practice, two different approaches are adopted to perform feature importance analysis. One is an accuracy-based measure and the other one is a model-based importance measure. In this study, an accuracy-based measure was used to investigate the contribution of different independent parameters in predicting the settling velocity as a response. Accordingly, a leave-one-feature-out experiment was performed [43]. Every single feature was systematically excluded during the model training and the corresponding $R^2$ was calculated. Similar to Reference [1], a relative feature influence (RFI) was introduced in the current study for the convenience of analyzing the feature significance. The formula used to calculate RFI is as follows:

$$RFI = \frac{R^2_{\text{incld}} - R^2_{\text{excl}}}{{\text{RFI}}} \times 100$$  (21)

The RFI indicates the relative importance of a feature in predicting the responses ($V_s$) and is derived using the $R^2$ metrics. The larger the RFI the more significant the feature is. Leave-one-dataset-out validation

A leave-one-dataset-out experiment was performed to validate the ML models [44]. To conduct this experiment, a series of data augmentation was performed. First, some datasets [15–18] were merged under one dataset named Rooki (2012). Second, a base training dataset was created by integrating two datasets collected from References [7,11]. These two datasets cover all fluid types used for the current study. It is important to mention that a good amount of base training data associated with each fluid type is needed to produce a better-generalized model. The dry experiment conducted for leave-one-dataset-out validation is demonstrated with an algorithmic flowchart in Figure 3. This validation approach comprising six experiments works through training and testing the model by excluding one dataset at a time.

| Algorithm: Leave-one-dataset-out Validation |
|--------------------------------------------|
| **Input**                                  |
| The six datasets with (X, Y) values         |
| **Output**                                 |
| Validation performances associated with six datasets |
| 1. for each dataset $D_i$                  |
| 2. Make the $D_i$ as validation dataset    |
| 3. Make the remaining Datasets as the training dataset |
| 4. Tune the model (SVR-Poly) on the training dataset |
| 5. Evaluate the model on the validation dataset |
| 6. Calculate the prediction error (e.g., RMSE) |
| 7. End                                      |
| 8. Return the validation performances associated with six datasets |

**Figure 3.** Algorithmic representation of the leave-one-data-set-out validation involving six datasets.

2.5. Computing Framework

All dry experiments and statistical analyses were performed on a computer with an Intel CORE i7 8th Gen, 2GHz processor, and 8 GB of RAM configuration. All hardware ran under a 64-bit Windows 10 operating system. As a statistical computing framework,
R (version 4.0.2) was used for the current study. The prediction time for a data matrix of 200 × 6 dimension was 10 s.

3. Results

3.1. Evaluation of Traditional Modeling Methodologies

The results of traditional implicit and explicit methods are presented in Figures 4–6 and Table 6.
Figure 4. Predictions of $V_s$ with Cheng [4] model and the associated uncertainty for spherical particles in (A) Newtonian fluids and (B) non-Newtonian fluids.

![Graph showing predictions of V_s with Cheng model](image1)

Figure 5. Predictions of $V_s$ with Ferguson and Church [45] model (fluid rheology: Newtonian; particle shape: spherical).

![Graph showing predictions of V_s with Ferguson and Church model](image2)

Figure 6. Predictions of $V_s$ with Okesanya and Kuru [19] model (fluid rheology: non-Newtonian; particle shape: spherical).

![Graph showing predictions of V_s with Okesanya and Kuru model](image3)

Table 6. The performance of explicit models on the complete dataset.

| Model                     | Fluid Rheology | MAE  | RMSE | MSE  | $R^2$ |
|---------------------------|----------------|------|------|------|-------|
| Ferguson and Church [45]  | Newtonian      | 0.030| 0.002| 0.0018| 0.965 |
Table 7. The performances of nine (9) MLA models on training and independent test data.

| Algorithm   | Train MAE | Train RMSE | Train $MSE$ | Train $R^2$ | Test MAE | Test RMSE | Test $MSE$ | Test $R^2$ |
|-------------|-----------|------------|-------------|-------------|----------|-----------|------------|-------------|
| SVR-Poly    | 0.035     | 0.054      | 0.0029      | 0.931       | 0.044    | 0.0044    | 0.921      |
| SGB         | 0.028     | 0.043      | 0.0019      | 0.955       | 0.045    | 0.0055    | 0.906      |
| SVR-RBF     | 0.021     | 0.038      | 0.0014      | 0.965       | 0.038    | 0.0055    | 0.902      |
| RF          | 0.017     | 0.029      | 0.0008      | 0.979       | 0.041    | 0.0056    | 0.901      |
| BART        | 0.038     | 0.057      | 0.0033      | 0.930       | 0.053    | 0.0067    | 0.880      |
| ANN         | 0.029     | 0.046      | 0.0021      | 0.950       | 0.047    | 0.0067    | 0.875      |
| MLP         | 0.072     | 0.103      | 0.0106      | 0.750       | 0.065    | 0.0083    | 0.851      |
| KNN-Regression | 0.027   | 0.049      | 0.0024      | 0.941       | 0.055    | 0.0098    | 0.819      |
| SVR-Linear  | 0.097     | 0.125      | 0.0156      | 0.631       | 0.092    | 0.0117    | 0.773      |

3.2. MLA Model Evaluation on the Independent Test Set

The performances of nine models in predicting settling velocities for the independent test set are shown in Table 7. Both training and test predictions were evaluated with $MAE$, $RMSE$, $MSE$, and $R^2$ metrics. The performances are also depicted in Figure 7 by plotting the measured values against the predicted values of $V_s$.
3.3. Feature Importance Analysis

The results of the leave-one-feature-out experiment using the best model, SVR-Polynomial Kernel are described in Table 8. The bar plot demonstrating the RFI of six experimental parameters is presented in Figure 8.

Table 8. The results of the leave-one-feature-out experiment using SVR-Polynomial Kernel.

| Experiment Number | Excluded Feature | MAE   | RMSE  | MSE   | R²   |
|-------------------|------------------|-------|-------|-------|------|
| 1                 | Particle Density | 0.115 | 0.169 | 0.0286| 0.525|
3.4. Leave-One-Dataset-Out Validation

The results of the leave-one-dataset-out validation are presented in Table 8. All the validation experiments were performed using the best performing SVR-polynomial kernel model. The test performances were evaluated using a similar set of evaluation metrics presented in Table 9.

| Validation Experiment Number | Dataset Name                          | Fluid Type | MAE   | RMSE  | MSE   | R²   |
|------------------------------|--------------------------------------|------------|-------|-------|-------|------|
| 1                            | Arabi and Sanders (2016) [2]          | Bingham    | 0.264 | 0.333 | 0.1109| 0.715|
| 2                            | Kelessidis (2004) [16]                | Power Law  | 0.132 | 0.112 | 0.0125| 0.913|
| 3                            | Okesanya et al. (2020) [20]          | Power Law  | 0.041 | 0.029 | 0.0008| 0.655|
| 4                            | Rooki (2012) [15–18]                 | Power Law  | 0.051 | 0.067 | 0.0045| 0.848|
| 5                            | Shahi (2014) [21]                    | Newtonian  | 0.031 | 0.037 | 0.0014| 0.664|
| 6                            | Song et al. (2017) [22]              | Newtonian  | 0.277 | 0.345 | 0.1190| 0.889|

4. Discussion

4.1. Limitations of Existing Analytical Models

As mentioned earlier, the implicit method of predicting \( V_t \) requires using a \( C_d - Re_f \) or \( C_d - Re_c \) correlation to iteratively predict \( V_t \) in a Newtonian or non-Newtonian fluid. Although many \( C_d - Re_f \) correlations are available in the literature, all of those were proposed for Newtonian fluids. Originally, Cheng [4] also proposed the correlation for Newtonian rheology. Later, the model was extended by replacing \( Re_f \) with \( Re_c \) to different non-Newtonian fluids [7,20,23]. The results of applying this correlation to the current dataset are presented in Figure 4 by plotting \( C_d \) against \( Re_f \) and \( Re_c \) for Newtonian and non-Newtonian fluids, respectively. As expected, the performance of the Cheng correlation is better for Newtonian fluids compared to the non-Newtonian fluids. Most of the measured values collapse over the model curve for Newtonian fluids. For non-Newtonian fluids, the uncertainty of prediction is considerably high. It can be less than \(-60\%\) and more than \(200\%\) for the current dataset. The values of the model coefficients determined empirically for Newtonian fluids do not necessarily account for the variation in viscosity due to shear in non-Newtonian fluids. That is why this kind of correlation is usually validated and modified before using for a non-Newtonian fluid. The associated complexities and uncertainties render predicting \( V_t \) using a \( C_d - Re_f \) correlation an inconvenient option for an engineer.

On the other hand, the explicit method involves correlation(s) that can be used to predict \( V_t \) directly based on the system properties. The following exemplary models were used for the current study.

4.1.1. Ferguson and Church (FC) Model

Probably, the most accurate explicit model for \( V_t \) in Newtonian fluids was developed by Ferguson and Church [45]. Agwu et al. [1] found its performance as comparable to ANN. The FC model (Equation (22)) applies to both laminar and turbulent conditions.
\[ V_s = \frac{gRd^2}{18(\mu_l/\rho_l) + (0.75CGd^3)^{0.5}} \quad (22) \]

In Equation (22), \( R = \frac{\rho_s - \rho_l}{\rho_l} \) and \( C (-) \) is a dimensionless constant. The empirical value of \( C \) is 0.54 for a spherical particle.

### 4.1.2. Okesanya and Kuru (OK) Model

It is one of the latest semi-mechanistic models. A series of steps involving conditional calculations are required to be followed to calculate \( V_s \) with the OK model. The most important equations are presented as follows.

\[
\bar{\tau} = \frac{dg(\rho_s - \rho_f)}{6} \quad (23)
\]

\[
V^* = \sqrt{\frac{\bar{\tau}}{\rho_f}} \quad (24)
\]

\[
Re^* = \frac{d\rho_f V^*(\bar{\tau} - \tau_o)}{K\bar{\tau}} \quad (25)
\]

\[
Re^*_G = \frac{\frac{d^n V^{2-n} \rho_f}{\tau_o \left( \frac{d}{\rho_f} \right)^n + K}} \quad (26)
\]

\[
\varepsilon = \frac{\tau_o}{\bar{\tau}} \quad (27)
\]

\[
\alpha = 1 + \varepsilon \quad (28)
\]

\[
Re^*_G = 0.7015Re^*_G^{1.3692}Re^*^{0.2598} - 3.2294e^{0.26674} \quad (if \quad \alpha < 1.3) \quad (29)
\]

\[
Re^*_G = (4.1 \times 10^{-9})Re^*_G^{0.8154}Re^*^{0.7579} - 18.673e^{0.7519} \quad (if \quad \alpha > 1.3) \quad (30)
\]

\[
\frac{V_s}{V^*} = 5.0286\left[1 - e^{(-0.064Re^*_G)}\right] \quad (if \quad Re^*_G < 15) \quad (31)
\]

\[
\frac{V_s}{V^*} = e^{\left(\frac{1.4962 - 3.741}{Re^*_G - 4.816}\right)} \quad (if \quad Re^*_G > 15) \quad (32)
\]

In Equations (23)–(32),

- \( \bar{\tau} \): surficial shear stress (Pa)
- \( V^* \): shear velocity (m/s)
- \( Re^* \): shear Reynolds number (-)
- \( Re^*_G \): shear generalized Reynolds number (-)
- \( \varepsilon \): relative characteristics shear stress (-)
- \( \alpha \): shape factor (-)

Compared to the OK model, the FC model is simple and easy to apply. Its performance in predicting \( V_s \) of a spherical particle in a Newtonian fluid is creditable. The agreement between the predicted and measured values is within an acceptable margin of error (Figure 5 and Table 7). However, the model cannot be applied to non-Newtonian fluids in its current form since it requires a constant value for \( \mu_l \). On the other hand, the OK model involves a complex numerical scheme. Although mathematical complexity is not uncommon in modeling a non-Newtonian system, this model is afflicted with a significant limitation. Its application is limited to three rheological models: (i) Herschel Bulkley, (ii) Bingham, and (iii) Casson model. The OK model does not apply to other rheological models, such as power-law and Newton’s law. As shown in Figure 6 and Table 7, the model yields result for the Herschel Bulkley fluids with a practical error limit. However, it fails
to do so for Bingham fluids. Thus, similar to other existing models, both of these explicit models are rheology-specific and prone to yielding significant errors.

4.2. Analysis of Current AI Model

A wide range of ML models was selected for modeling the \( V_s \) of spherical particles in the present study to develop a generalized model applicable to both Newtonian and non-Newtonian fluids. Parametric regression models, such as SVR, MLP, NN, as well as non-parametric regression models such as KNN, including a comparatively latest approach called Bayesian regression were considered. Among the Bagging and Boosting algorithms, RF and SGB were selected. Table 6 shows the results of ML modeling of the selected regression models through repeated cross-validated training. Among all models, the best model was the SVR-Polynomial kernel \( R^2 = 0.921 \) with the optimized values of the parameters, such as degree-d of the kernel as three (3), input scaling parameter as 0.1, and the regularization parameter as 1 (see Table 5). As modeling \( V_s \) associates non-linear feature interactions, non-linear ML models were found to have superior performances. In the case of SVR, both RBF and Polynomial kernel map the input space to higher dimensional feature space, and, subsequently, the problem becomes linearly separable into that feature space. The SVR-Polynomial kernel demonstrated better performance than its RBF kernel counterpart in the case of independent test data prediction (see Table 7 and Figure 7). It should be noted that the SVR-Poly yielded comparable values of \( R^2 \) for both training and test datasets \( R^2 = 0.931 \) for training dataset; \( R^2 = 0.921 \) for test dataset), which demonstrates the robustness of the model. The second-best model was the SGB \( R^2 = 0.906 \), which is the stochastic version of the general gradient boosting algorithm. The SVR–RBF along with the RF ranked as the third-best model with an identical \( R^2 \) value of 0.901 for the test dataset. It is important to observe that both bagging (RF) and boosting (SGB) algorithms with stochastic components were found to perform well either by choosing the best possible random set of predictors (RF) or observations (SGB) for splitting at each node of the regression tree and iteration of minimizing the objective loss function. Both models were able to capture the non-linearity of the data and accurately estimate the actual response.

The SVR-Linear had the least prediction accuracy \( R^2 = 0.773 \) among the ML models evaluated for the current study. Given the superior performances of other non-linear kernels in SVR, it is apparent that the original input features are non-linearly separable, and, therefore, SVR-Linear inevitably led to an underfitting of the data. The KNN, as a non-parametric regression method, achieved the second-lowest accuracy with \( R^2 = 0.819 \). The ANN and MLP achieved the next lower accuracies and exhibited comparable performances with \( R^2 = 0.871 \) and \( R^2 = 0.851 \), respectively. This is most likely because both ANN and MLP architectures have similar model complexity with a single hidden layer and different numbers of hidden neurons (20 for ANN and 5 for MLP). The BART could achieve a moderate correlation \( R^2 = 0.88 \) with the imposition of regularization on each tree while fitting to a small portion of the training data. It helped to achieve a fairly bias-free prediction when several trees were fitted to the whole set of observations.

The comparative performance analysis of the ML models on independent test data demonstrated all parametric regression models except SVR-Linear to perform better than non-parametric regression models, e.g., KNN-regression. Among the parametric models, the SVR with non-linear kernels was the top performer, while the regression tree with both stochastic and Bayesian components was in the second-best position. The ANN family of regressor models showed ordinary performances that were comparable to the existing ANN models such as [1] and [14]. The current study proves that MATLAB-based ANN may not provide the best accuracy. Another interesting observation is that the training performances of MLP and SVR-linear are lower than the corresponding independent test performances in terms of all metrics. In the case of MLP, there could be a local minimum that led to the model’s lower performance during training, as this is a usual problem in an NN-like architecture. In the case of SVR-Linear, this is a clear sign of underfitting. In both cases, the models could produce a better prediction for test data points.
One of the important aspects of the current study is a comprehensive feature importance analysis. A leave-one-feature-out strategy was used with six experiments for the purpose. It involved training and testing the model by excluding one feature at a time. The SVR-Polynomial kernel model was used for these experiments because of its superior performance over other ML models on an independent test set. A new set of $R^2$ values were obtained from the leave-one-feature-out experiments. Then, the $R^2$ values obtained before and after excluding each feature were compared. A relative feature importance indicator was used by taking into account the differences observed in $R^2$ values in the presence or absence of a specific feature. The RFI indicates the influence of a feature on the model in relation to the $R^2$ metric. A decreased ranking-order of the features can be obtained based on their RFI values. As shown in Figure 8, the Particle Density and the Particle Diameter are identified as the two most significant features having respective RFI values of 39.6 and 24.8, followed by a decreasing rank-order of Yield Stress, Flow Consistency Index, Flow Behavior Index, and Liquid density. Specifically, the ordering of the features from the highest to lowest ranks is as follows: $\rho > d_i > \tau_s > K > n > \rho_l$ (Figure 8). A similar trend of feature influence was reported in a previous ANN-study [1]. It can be explained with the physics of the settling process. The motion of a particle in a stationary fluid is triggered by the gravitational force ($F_g$). When it is released in the fluid, an acceleration of $g (-9.81 \text{ m/s}^2)$ acts on it. Afterward, the motion of the particle in the fluid actuates a drag force ($F_d$) on itself. Its incremental velocity also intensifies $F_d$. When $F_d$ nearly matches $F_g$, an equilibrium condition persists and the particle continues moving at a constant velocity known as the settling velocity. Thus, the opposing forces ensue $V_s$. The primary force in this process is $F_g$, which is strongly dependent on a spherical particle’s density and size or diameter. The dependency is conceivably stronger on $\rho_l$ than $d_i$. On the other hand, $F_d$ is the secondary force stemmed from the particle motion. It is known to be a strong function of Newtonian fluid viscosity and weakly dependent on $\rho_l$. For the non-Newtonian fluids considered in the current study, the rheology was modeled with yield stress, flow consistency index, and flow behavior index. The most important among these parameters is $\tau_s$ as $F_d$ cannot be actuated until $F_g$ overcomes it. The viscosity of the non-Newtonian fluid is denoted with $K$ while its shear-thinning behavior is signified with $n$. Understandably, $K$ plays a more significant role than $n$ in the settling process. Thus, the feature-importance, as shown in Figure 8, is in sync with the physics of the particle settling phenomenon. It justifies the utility of the MLA in predicting $V_s$.

A prediction model can be considered reliable if it generalizes well on completely unseen data that were not used during the training. The leave-one-dataset-out validation (LOOV) examines the generalizability of an MLA. Furthermore, it checks whether training produces a reliable model that could be tested by any other set of experimentally measured values for different types of fluids to predict the corresponding $V_s$. Although the LOOV experiment performs an objective analysis of the ML-model performance, it is largely absent in the previous ML studies of $V_s$ modeling. Most of the previous studies evaluated their proposed models based on test data that come from the same training distribution. This type of in-distribution evaluation does not inform about the performance of the model when applied to an out-distribution data or a completely new dataset. On the other hand, leave-one-out validation is one of the preferred methods to comprehensively evaluate the generalized performance of ML models [34,35,44]. Agwu et al. [1] attempted to conduct a similar validation. However, their effort was constrained by the number of datasets as they used only two independent datasets. In the current study, the LOOV was used to investigate the generalizability of the MLA that was identified as the optimum choice to model the $V_s$ of a sphere in both Newtonian and non-Newtonian fluids. The dry experiments were conducted by training the SVR-Polynomial kernel model on six datasets. The SVR-Polynomial was selected for the LOOV analysis because of its superior performance over the other ML models in predicting $V_s$ for independent test data. Then, the generalization ability of the model was assessed with four evaluation metrics.
for each dataset. The SVR-Polynomial kernel model achieves moderate to high performances with the $R^2$ values between 0.65 and 0.91 with an average of 0.78. An analogous analysis was reported in Reference [1]. They applied a total of eighteen (18) explicit models along with their ANN-model to predict the $V_\ell$ for a specific dataset. A statistical comparison of the predictions produced $R^2$ values in the range of 0.23–0.94 with an average of 0.49. An objective comparison of these independent analyses suggests the current model is capable of producing reliable results for both Newtonian and non-Newtonian fluids within an acceptable error margin.

The open-source statistical computing framework, R, instead of commercial MATLAB, was used in the present study for conducting all ML experiments and performing the statistical analysis. The majority of the previous ANN-based studies utilized MATLAB for developing the prediction models. Compared to MATLAB, R has several advantages: (1) R is completely free of charge, as it is open-source software; (2) it provides numerous packages and built-in functions for ML modeling, fine-tuning the hyper-parameters, data visualization and pre-processing, post-prediction statistical analyses, plotting of result graphs and trends, etc.; (3) R can be used with any operating system; (4) it can be used to develop a web-based DSS that would allow a user to estimate $V_\ell$ for a given set of process conditions. Furthermore, R is easy to program, well-documented, and evolving continually due to the freelance contributions from the scientific community. Therefore, R is considered a more economic, flexible, and fast-evolving scientific and statistical computing framework compared to MATLAB. It is the industrial choice for data analysis.

As mentioned earlier, the best performing MLA in the current study was SVR-Poly. The upshots of this model are demonstrated in Figure 9 and discussed as follows:

(a) The SVR-Poly can predict $V_\ell$ of spherical particles in Newtonian and different varieties of non-Newtonian fluids. Whereas, the FC applies to only Newtonian fluids, and the OK could be applied to two non-Newtonian fluids.

(b) Prediction uncertainties associated with SVR-Poly are comparable for the Newtonian and non-Newtonian rheologies. That is, the ML model is insensitive to fluid rheology. Unlike the traditional models, it is capable of predicting $V_\ell$ without a bias to the fluid properties.

(c) Prediction accuracy of SVR-Poly is significantly better than the traditional explicit models such as FC and OK models.

![Figure 9](image-url)  
**Figure 9.** Comparative performance of the ML model, SVR-Poly, and the traditional explicit models, Ferguson and Church (FC) and Okesanya and Kuru (OK) on the test data.
5. Conclusions

In the current study, we investigate an ML-based methodology for predicting the settling velocity of spherical particles in the Newtonian fluid and the non-Newtonian fluids following different varieties of rheology such as Power-law, Bingham Plastic, and Herschel Bulkley models. This is because the traditional analytical approach of modeling the velocity, including both explicit and implicit models, cannot provide a generalized solution. Among nine tested ML models, SVR with polynomial kernel provided the best performance in terms of the statistical metrics on an independent test dataset \( R^2 = 0.92; \text{RMSE} = 0.066; \text{MSE} = 0.0044; \text{MAE} = 0.044 \). A ten-fold-cross-validation technique was used to ascertain the most suitable model. A comprehensive feature analysis using leave-one-feature-out experiments assured the strong correlation among the parameters selected for modeling and the experimental observations. The correlations were quantified using a relative feature influence indicator. The order of significance for the input parameters was as follows: particle density > particle diameters > yield stress > flow consistency index > flow behavior index > liquid density. Additionally, leave-one-dataset-out validation experiments were conducted to test the reliability of the model for unseen experimental datasets. The model confirmed a reasonable performance in estimating the settling velocity for completely unseen data. It could predict more than 80% of the test data within an error limit of ±20%. The present study provides strong evidence that ML is very effective and accurate as a general predicting tool for the settling velocity of spheres in both Newtonian and non-Newtonian fluids. It can be used as a reliable method in industrial-scale design and operation due to its economical requirements of time and computational resources.

While the current study demonstrates the generalized capability of MLAs in predicting settling velocity, further research is necessary to develop a universal ML model independent of particle shape, particle proximity, fluid rheology, wall proximity, and flow conditions. The works currently underway in this regard include applying the ML models in predicting the hindered settling velocity and developing a smart decision support system.

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

Funding: This research was funded by the Deanship for Research and Innovation, Ministry of Education, Saudi Arabia, project number IFT20081.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data presented in this study are openly available in the following references: [2,6,7,11,15–22].

Acknowledgments: The authors extend their appreciation to the Deanship for Research and Innovation, Ministry of Education, Saudi Arabia. The authors are also grateful to the Deanship of Scientific Research, the College of Computer Science and Information Technology, and the College of Engineering at King Faisal University, Saudi Arabia.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

References

1. Agwu, O.E.; Akpabio, J.U.; Dosunmu, A. Artificial Neural network model for predicting drill cuttings settling velocity. *Petroleum* 2019, in press, doi:10.1016/j.petlm.2019.12.003.
29. Golshan, A.; Ghodsiyeh, D.; Gohari, S.; Amran, A.; Baharudin, B.T. Computational intelligence in optimization of machining operation parameters of ST-37 steel. Appl. Mech. Mater. 2012, 248, 456–461, doi:10.4028/www.scientific.net/AMM.248.456.

30. Clarke, S.M.; Griebisch, J.H.; Simpson, T.W. Analysis of support vector regression for approximation of complex engineering analyses. J. Mech. Des. 2005, 127, 1077–1087, doi:10.1115/1.1897403.

31. Ladicky, L.U.; Jeong, S.; Solenthaler, B.; Pollefeys, M.; Gross, M. Data-driven fluid simulations using regression forests. ACM TOG 2015, 34, 1–9, doi:10.1145/2816795.2818129.

32. Patri, A.; Patnaik, Y. Random forest and stochastic gradient tree boosting based approach for the prediction of airfoil self-noise. Procedia Comput. Sci. 2015, 46, 109–121, doi:10.1016/j.procs.2015.02.001.

33. Wang, Z.; Liu, Q.; Liu, Y. Mapping landslide susceptibility using machine learning algorithms and GIS: A case study in Shexian County, Anhui Province, China. Symmetry 2020, 12, 1954, doi:10.3390/sym12121954.

34. Chen, C.; Deng, J.; Yu, X.; Wu, F.; Men, K.; Yang, Q.; Zhu, Y.; Liu, X.; Jiang, Q. Identification of novel inhibitors of DDR1 against idiopathic pulmonary fibrosis by integrative transcriptome meta-analysis, computational and experimental screening. Mol. Biosyst. 2016, 12, 1540–1551, doi:10.1039/c5mb00911a.

35. Menden, K.; Marouf, M.; Oller, S.; Dalmia, A.; Magruder, D.S.; Kloiber, K.; Heutink, P.; Bonn, S. Deep learning–based cell composition analysis from tissue expression profiles. Sci. Adv. 2020, 6, eaba2619, doi:10.1126/sciadv.aba2619.

36. Achirul Nanda, M.; Boro Seminar, K.; Nandika, D.; Maddu, A. A Comparison study of Kernel functions in the support vector machine and its application for Termite detection. Information 2018, 9, 5, doi:10.3390/info9010005.

37. Gutiérrez-Esparza, G.O.; Infante Vázquez, O.; Vallejo, M.; Hernández-Torruco, J. Prediction of metabolic syndrome in a Mexican population applying machine learning algorithms. Symmetry 2020, 12, 581, doi:10.3390/sym12040581.

38. Hill, J.; Linero, A.; Murray, J. Bayesian additive regression trees: A review and look forward. Annu. Rev. Stat. Appl. 2020, 7, 251–278, doi:10.1146/annurev-statistics-031219-041110.

39. Pereira Härter, F.; Fraga de Campos Velho, H. Multilayer perceptron neural network in a data assimilation scenario. Eng. Appl. Comput. Fluid Mech. 2010, 4, 237–245, doi:10.1080/19942060.2010.11015313.

40. Friedman, J.H. Stochastic gradient boosting. CSDA 2002, 38, 367–378, doi:10.1016/S0167-9473(01)00065-2.

41. Chipman, H.A.; George, E.I.; McCulloch, R.E. BART: Bayesian additive regression trees. Ann. Appl. Stat. 2010, 4, 266–298, doi:10.1214/09-AOAS285.

42. Rodriguez, J.D.; Perez, A.; Lozano, J.A. Sensitivity analysis of k-fold cross validation in prediction error estimation. IEEE PAMI 2009, 32, 569–575, doi:10.1109/TPAMI.2009.187.

43. Cawley, G.C.; Talbot, N.L. Efficient leave-one-out cross-validation of kernel fisher discriminant classifiers. Pattern Recognit. 2003, 36, 2585–2592, doi:10.1016/S0031-3203(03)00136-5.

44. Fushiki, T. Estimation of prediction error by using K-fold cross-validation. Stat. Comput. 2011, 21, 137–146, doi:10.1007/s11222-009-9153-8.

45. Ferguson, R.I. and Church, M. A simple universal equation for grain settling velocity. J. Sediment. Res. 2004, 74, 933–937, doi:10.1306/051204740933.