*metals*

MDPI

*Article*

# A Study on Prediction of Size and Morphology of Ag Nanoparticles Using Machine Learning Models for Biomedical Applications

**Athira Prasad, Tuhin Subhra Santra and Rengaswamy Jayaganthan ***

Department of Engineering Design, Indian Institute of Technology Madras, Chennai 600036, India
* Correspondence: edjay@iitm.ac.in

**Abstract:** The synthesis of silver nanoparticles (AgNPs) holds significant promise for various applications in fields ranging from medicine to electronics. Accurately predicting the particle size during synthesis is crucial for optimizing the properties and performance of these nanoparticles. In this study, we compare the efficacy of tree-based models compared with the existing models, for predicting the particle size in silver nanoparticle synthesis. The study investigates the influence of input features, such as reaction parameters, precursor concentrations, etc., on the predictive performance of each model type. Overall, this study contributes to the understanding of modeling techniques for nanoparticle synthesis and underscores the importance of selecting appropriate methodologies for accurate particle size prediction, thereby facilitating the optimization of synthesis processes and enhancing the effectiveness of silver nanoparticle-based applications.

**Keywords:** silver nanoparticle; machine learning; particle synthesis; particle size prediction; tree-based modelling



**Citation:** Prasad, A.; Santra, T.S.; Jayaganthan, R. A Study on Prediction of Size and Morphology of Ag Nanoparticles Using Machine Learning Models for Biomedical Applications. *Metals* **2024**, *14*, 539. https://doi.org/10.3390/met14050539

Academic Editor: Leonid M. Kustov

Received: 2 March 2024
Revised: 18 April 2024
Accepted: 26 April 2024
Published: 2 May 2024



## 1. Introduction

The characteristics of a nanomaterial and its functional properties for biomedical applications are governed by various parameters pertaining to its synthesis comprising starting precursor materials besides temperature, pressure, and design of the lab scale reactors. It is mostly a trial-and-error approach that is followed to synthesize nanomaterials with the desired properties. This is an expensive, time-consuming, and low-efficiency procedure. With the growing demand for nanomaterials for various applications, it is critical to quickly and precisely predict the characteristics based on synthesis factors [1].

Computer algorithms are used in machine learning (ML), a branch of artificial intelligence, to create mathematical models that can carry out particular tasks like clustering, dimensionality reduction, and prediction directly from collected data, such as graphs, images, or numerical data, as opposed to from established physical laws. These ML models are especially helpful in situations where interrelationships between input experimental variables and output results are complex, lacking detailed mechanistic understanding governed by fundamental physical laws. Applications of ML include chemical recognition, materials design and discovery, synthesis reaction prediction, and nanoparticle size prediction. Depending on how they learn, machine learning algorithms can be divided into three categories: semi-supervised learning techniques (generative models), unsupervised learning (clustering, association rules, etc.), and supervised learning (decision tree, boosting, support vector machine (SVM), etc.) [2,3].

Nonlinear machine learning models, which involve interactions between input circumstances and outputs properties that are not linear, can be categorized into two types of algorithms: instance-based and model-based. By using a model that is defined by a collection of variables that are inferred through training, model-based techniques create predictions. Deep learning and tree-based algorithms are two examples of model-based

techniques that have been extensively used for the prediction of results in nanoparticle synthesis. Supervised learning is essentially an attempt to investigate an unknown function in which there is a relationship amongst the variables that are input and an unknown target for the output. Labelled data are used in supervised learning techniques to build a machine learning model that predicts the association between desired attributes and characteristics. We must identify both dependent and independent variables before selecting an algorithm. Next, the correlation between both the independent and dependent variables ought to be ascertained. Unlike supervised algorithms used for predictions, unsupervised methods are utilized for statistical analysis and visualization and do not require training with previous experiments. A type of unsupervised algorithms known as generative machine learning models seeks to produce new data points that resemble those in a selected dataset [4].

Reaction conditions, such as reaction temperature, reaction time, chemical reagent concentration ratio, reaction precursors, reaction ligands, solution reagents, microreactor channel structure, external stimuli, and so forth, are typically chosen as independent variables in nanoparticle preparation techniques, while the dimension, shape, and electronic and optical characteristics of the nanoparticles are typically chosen as dependent variables. The gathering and preparation of the dataset is a crucial step before developing a machine learning model [3]. The trial-and-error approach used to collect experimental data for optimizing the synthetic procedure for nanoparticle synthesis is time-consuming and not economical [1]. Therefore, with the physics-informed ML models, it is possible to optimize and predict the feasible synthetic procedures for producing nanoparticles. Alternatively, for the training of machine learning models, reputable datasets from published articles, on nanoparticle synthesis, are another source. In addition, online search engines could be used to find pertinent datasets from open-source shared information and repositories. Typically, a dataset is separated into two parts: the test set and the training set. The majority of the dataset is used for training the suitable machine learning model, which is subsequently assessed using the test dataset. Typically, two datasets with an 8:2 or 7:3 ratio are created from the data for training and testing the ML models. The superior performance of the machine learning approach cannot be simply assessed using the training set alone [1–4]. Any computational technique that helps narrow down the design space by forecasting desired process variables prior to synthesis would be beneficial to reduce the number of steps involved to produce nanoparticles by chemical routes. One crucial physio-chemical characteristic of nanoparticles that can influence their use in nanomedicine is size. For example, it has been discovered that a nanoparticle's size has a significant role in in vivo experiments performed for various therapeutic applications. Since the size of the nanoparticle influences its permeability and retention, size optimization is also crucial for the design and development of nanoparticles used to treat a range of tumors [5]. Particle size and particle density index (PDI), two crucial metrics for evaluating a drug-loaded nanoparticle formulation, depend on a number of factors such as composition, the duration of sonication, and extrusion temperature. For achieving an ideal particle size with a narrow size distribution, empirical methods are often employed to adjust these independent parameters through iterative trial-and-error methodology [6].

Various studies have considered the best model to use for predicting size of the nanoparticles. Silver is one of the most commonly studied nanoparticles, synthesized either through a chemical or green synthesis route. Several studies have been carried out using silver nanoparticles for various fields like biomedical applications [7], cosmetics [8], electronic field [9], etc. Determining the size of the particle is important, for example in the antibacterial activity of the particle [10,11], central metabolism of wheat seedlings [12], etc. In order to predict the size of silver nanoparticles (AgNPs) made using a green approach, Shabanzadeh et al. [13] proposed an artificial neural network (ANN) model. A number of variables that affect the size of the nanoparticles are taken into consideration by the ANN model, including temperature, starch concentration, and NaOH volumes. The Levenberg–Marquardt (LM) back-propagation algorithm was used to train the network. The coefficient of determination value for the test data was 0.9787, for the best predictive model.

The size of AgNPs in montmorillonite/starch bionanocomposites, prepared through chemical routes, was predicted using an ANN model by Shabanzadeh et al. [14]. With a log-sigmoid transfer function for hidden layers and a linear transfer function for output layers, the ideal ANN architecture was found to be 4:10:1. To ascertain AgNP's size, TEM was employed. The concentration of silver nitrate ($AgNO_3$), the reaction temperature, the starch weight percentage, and the concentration of $NaBH_4$ served as independent variables. As $AgNO_3$ concentration, starch percentage, and $NaBH_4$ concentration increase, the size of the nanoparticles decreases, according to the modelling results.

An ANN was used as computational tool by Shabanzadeh et al. [15] to model the size of AgNP in montmorillonite/chitosan bionanocomposites (BNCs). The montmorillonite (MMT) interlayer space has been utilized as an adsorbent for cationic ions, as a substrate for anchoring transition metal complexes, and for the production of material and biomaterial nanoparticles. In this study, a one-hidden-layer neural network with a single output was utilized as an ANN. AgNP size was modelled as a function of several parameters, such as d-spacing of clay layers, reaction temperature, percentage of chitosan, and concentration of silver nitrate. The ANN is trained using the back-propagation Levenberg–Marquardt (LM) algorithm. Furthermore, the AgNP particle size and distribution prepared at various experimental values showed that larger AgNP particle sizes were obtained at higher reaction temperatures and $Ag^+$ ion concentrations; however, AgNP diameters decreased as chitosan percentages increased.

To design an environmental friendly and efficient process for the preparation of AgNP in BNCs matrix through a green synthesis technique, Shabanzadeh et al. [16] used the relationships between multi-input variables, such as $AgNO_3$ molar concentration, reaction temperature, percentage of starch, and amount of MMT. MATLAB is used to put the suggested method into practice. To forecast the size of AgNPs, an ANN network consisting of a 4-10-1 feed-forward multilayer perceptron (MLP) with a linear function at the output layer and a tangent sigmoid transfer function at the hidden layers was employed in this study.

The prediction model for determining the size of AgNPs made via green synthesis was developed by Sattari and Khayati [17] using the gene expression programming (GEP) technique. The data required to build the GEP models were gathered by the researchers through 30 distinct experiments. Plant extract, reaction temperature, $AgNO_3$ concentration, and stirring duration are among the input factors taken into account by the model.

Nathanael et al. [18] proposed a technique that combines machine learning with a T-junction microfluidic system to optimize the synthesis of the AgNPs and to forecast the particle size of AgNPs synthesized in microfluidic systemsusing trisodium citrate (TC), tannic acid (TA), and silver nitrate as reducing and stabilizing agents. They have employed a decision-tree-guided design of experiments for determining the size of AgNPs. The synthesized silver nanoparticle's stability is affected by storage temperature, pH, and concentrations of trisodium citrate, which have influenced nucleation and growth rate—the nucleation constant ($k_1$) and growth constant ($k_2$). The Finke–Watzky (F–W) two step mechanism was used in an independent set of beaker experiments to derive the nucleation and growth constants. Table 1 summarizes the training features (inputs) with the range of those features that have been studied.

Shafaei et al. [19] predicted the size of AgNPs made using a green synthesis technique using a hybrid artificial neural network particle swarm optimization approach. In order to attain the smallest possible size of AgNPs, the study also focused on optimizing the practical procedure. Silver nitrate, the precursor of silver, and opium syrup, a reducing and stabilizing substance, were used to create silver nanoparticles. An experiment using a factorial D-optimal array design was used to gather the experimental data. The ratio of $AgNO_3$ to opium syrup, the feed rate at which the reducing agent is added, pH, reaction temperature, and agitation speed all affect the size of the samples. The size of the silver nanoparticle was chosen as the output and the process parameters as the inputs for the optimization process.

**Table 1.** Inputs and their investigated range, adapted from [18].

| Input Parameters and Their Investigated Range | |
|---|---|
| **Feature** | **Range Investigated** |
| Nucleation constant ($min^{-1}$) | 0.0011–0.1 |
| Growth constant ($M^{-1}min^{-1}$) | 13.66–77.97 |
| Storage temperature (°C) | 0–20 |
| Dean number/Reynolds number | 0–0.41 |
| Reynolds number | 0.0849–16.96 |

Using Vitex negundo L extract as a reducing agent and stabilizer, Shabanzadeh et al. [20] demonstrated the biosynthesis of silver nanoparticles. The additional reagents that were utilized were Muller Hinton agar, $AgNO_3$, methanol, and nutrient agar. An ANN model was used in order to predict the nanoparticle size. Thirty produced samples from experimental datasets were used in their work. The molar concentration of $AgNO_3$, weight% of Vitex negundo extract, reaction temperature, and stirring time are significant variables that can affect the size of silver nanoparticles. Using experimental data for training, the ANN model demonstrated excellent accuracy in forecasting the size of nanoparticles under various conditions. This ANN model served as a valuable tool for developing a sustainable and efficient process for producing silver nanoparticles. It was found that the concentration of $AgNO_3$, temperature of reaction, stirring time, and the quantity of Vitex negundo L extract had an impact on the size of the nanoparticles; an increase in $AgNO_3$ concentration, temperature of reaction, stirring time, produced larger nanoparticles, and an increase in Vitex negundo L extract produced smaller nanoparticles.

The use of an ANN model to predict the size of AgNPs synthesized in the interlayer space of montmorillonite was reported in the work of Shabanzadeh et al. [21]. The ANN model assisted in optimizing the design parameters and in minimizing costly experimental research. For the purpose of modelling AgNP size prediction in their study, a multilayer perception (MLP)-based feed-forward ANN that makes use of an LM-based back-propagation learning method was used. Three sets of the experimental data were randomly selected to serve as training, validation, and testing, respectively. Absolute average deviation, coefficient of determination, and root mean square error are calculated to assess the predictive performance of the ANN model. Their work focused on how various parameters, including montmorillonite d-spacing, UV–visible wavelength, reaction temperature, and $AgNO_3$ concentration, affect AgNP size. The results of the analysis indicate that the two main variables influencing the size of the nanoparticles are temperature and $AgNO_3$ concentration.

In the present study, the data were collected from four distinct experimental sets and employed conventional machine learning models. The datasets used for modelling include those from Nathanael et al. [18], Shafaei et al. [19], Shabanzadeh et al. [20], and Shabanzadeh et al. [21]. The experimental datasets used were synthesized either by chemical or green synthesis routes. The complete dataset for each experiment was split to test and train. Tree-based models such as decision tree regressor, random forest regressor, and extreme gradient boost regressor were used to assess the prediction of the size of the silver nanoparticles. The efficiency of the model was evaluated by determining the coefficient of determination ($R^2$), mean absolute error (*MAE*), and mean square error (*MSE*) for all the three models.

When the data availability is less, traditional models can, at times, provide better accuracy compared to other complicated models used. In this paper, the Section 2 details the processes involved and the various models utilized in our current study. The Section 3 provides an overview of the different performance measures considered to determine the efficiency of the models. The Section 4 assesses the effectiveness of several models and presents different plots to corroborate the findings. Moreover, it provides insight into the significance of the features in the model. The summary of the findings is included in the Section 5.

## 2. Methodology

The Python programming language was utilized in conjunction with the scikit-learn package (version 1.3.2) to perform the machine learning analysis. Various tree-based ML techniques such as decision tree (DT), random forest (RF), and extreme gradient boosting (XGBoost) were utilized to assess the predictive accuracy for estimating the size of AgNPs. In numerous applications, these ML techniques have been widely applied. Figure 1 shows the machine learning process involved to predict the nanoparticle size.

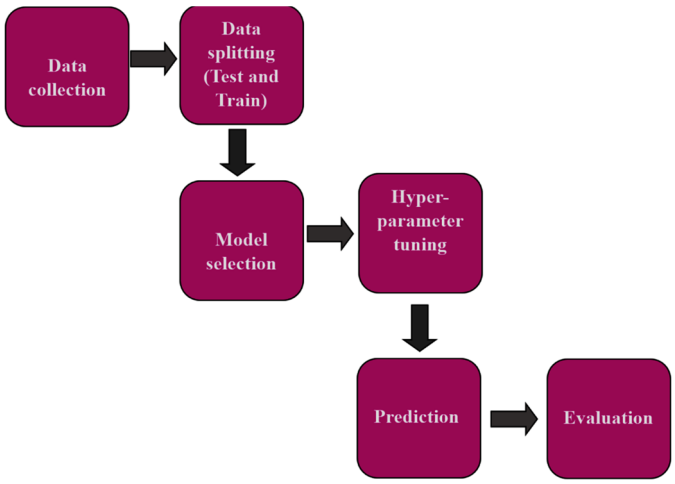


**Figure 1.** Machine learning process.

The dataset collected from the literature was first split to train and test data. This is performed in order to make sure that the model works well on unseen data. Choosing the right model is crucial in the prediction. Here, we have considered tree-based models. Hyperparameter tuning helps to determine the best parameters suitable for the model that can significantly impact the model's predictive performance. For regression models, the coefficient of determination, mean square error, and mean absolute error are the most commonly used evaluation techniques.

### *2.1. Decision Tree*

The machine learning technique of a decision tree is quite flexible and can be used for problems involving several outputs in addition to classification and regression [22]. Capable of fitting intricate datasets, these algorithms are incredibly strong [23]. For researches on medical application [24–26], road safety [27–29], to manage and analyze the data on metal nanoparticles [30], DT methodology is becoming very popular. DTs require very little data preparation, which is only one of their numerous benefits. Scikit-learn employs an optimized version of the classification and regression (CART) algorithm to build models that predict values of target variables by learning basic decision rules inferred from data features. The CART cost function for regression is given by Equation (1).

$$J(k,t_k) = \frac{m_{left}}{m} MSE_{left} + \frac{m_{right}}{m} MSE_{right} \quad (1)$$

where

$$MSE_{node} = \sum_{i \in node} (\hat{y}_{node} - y^{(i)})^2$$
$$\hat{y}_{node} = \frac{1}{m_{node}} \sum_{i \in node} y^{(i)}$$

where $m$ is the number of the samples, $m_{left}$ and $m_{right}$ represents the number of samples that fall into the left child node and right child node, respectively, after splitting the dataset based on the chosen feature and threshold. $MSE_{left}$ and $MSE_{right}$ represents the mean squared error measure for the samples in the left child node and right child node,

respectively. The number of samples that belong to the node is represented by $m_{node}$. $\hat{y}_{node}$ represents the predicted value of the node. The predicted value for the $i^{th}$ sample in the dataset is represented by $y^{(i)}$ [23].

DTs are a non-parametric supervised learning method used for regression and classification. The concept is actually very straightforward: the algorithm uses a single feature '$k$' and a threshold '$t_k$' to first divide the training set into two subsets. The pair ($k$, $t_k$) that generates the purest subsets (weighted by their size) is sought after. After splitting the training set into two successfully, it recursively divides the subsets using the same reasoning, then the sub-subsets, and so on. When it reaches the maximum depth (specified by the max_depth hyperparameter) or fails to locate a split that will lower impurity, it ceases recursing [23].

Depending on the variables' state, DTs can be seen in two different ways. The problem can be resolved within the framework of regression if the target variable is numerical; if not, it is a classification problem. The decision tree regressor class in scikit-learn is used to construct a regression tree. The CART method seeks to minimize the mean squared error (*MSE*) by dividing the training set into smaller segments. DTs are prone to overfitting in regression problems, just as they are in classification tasks. DTs are powerful, adaptable, easy to utilize, and simple to comprehend and interpret. Unless adjusting the random_state hyperparameter, significantly different models are obtained from the same training set of data as the training process is stochastic in scikit-learn [23,25].

### *2.2. Ensemble Methods*

Compared to a single estimator, ensemble technique improves generalizability and robustness by combining the predictions of many base estimators constructed using a particular learning methodology. Random forests and gradient-boosted trees are two prominent examples of ensemble methods. The term "ensemble" refers to a set of predictors, hence the names "ensemble learning" and "ensemble method" for the technique and algorithm used [23].

#### 2.2.1. Random Forest

One of the most widely used machine learning algorithms is a random forest developed by Breiman in 2001, which is simply an ensemble of DTs [23]. In 1994, Breiman developed the concept of "bagging" to lower a learning algorithm's variance while maintaining a relatively small bias. By using distinct randomized versions of the initial learning sample for every run, RF either explicitly incorporate randomization within the learning algorithm or takes advantage of it to generate an ensemble of models that are more or less strongly varied. Subsequently, a simple average is used to combine the predictions made by several models, or the vote of the majority if classification is involved. Other supervised learning algorithms, such as SVM or neural networks, are frequently not competitive with these general randomization approaches, which frequently significantly increases the accuracy of decision or regression trees [31,32]. RF is an advancement over the bagged regression tree. To build individual trees, it still relies on bootstrapped sampling. At each splitting node of the tree, it only permits the use of a random subset of features rather than all of them. It, thus, mandates variation amongst basic models. Variance of RF for a total number trees ($K$) is given as

$$\rho\sigma^2 + \frac{1-\rho}{K}\sigma^2 \tag{2}$$

where $\sigma^2$ is the variance of each individual tree, and $\rho$ is the correlation between the trees [33]. Breiman suggested this approach as an improvement on tree bagging. It employs a bootstrap replica of the learning sample, the CART algorithm avoiding pruning, with the random subspace method's modification to build a tree. Finding a random subset of size K of candidate attributes at each test node yields the best split. In order to increase predictive accuracy and reduce over-fitting, a random forest meta estimator fits many classification decision trees on different subsamples of the dataset and then averages the

results. If bootstrap = true, as is the default, the sub-sample size is managed using the max_samples argument; if not, each tree is constructed using the entire dataset [31]. The number of trees in an RF model can be found without following any set rule. Typically, when the "out-of-bag" (OBB) error rate has reached a consistent minimum error rate value, this is the point at which the number of trees is at its best [34].

#### 2.2.2. Extreme Gradient Boost

Adding additional models to the ensemble in a sequential manner is the basic notion behind boosting. A new, weak base-learner model is trained at each iteration based on the error of the entire ensemble that has been learned a priori. Gradient-boosting machines, or GBMs for short, use a learning process that fits new models one after the other to produce a response variable estimate that is more accurate. The main idea behind this technique is to build new base-learners that have the highest possible correlation with the ensemble's negative gradient of the loss function [35]. An open-source program for approximate tree learning, XGBoost, is a unique sparsity-aware algorithm proposed by Chen and Guestrin [36]. It has been demonstrated that the XGBoost method offers significant model performance while lowering calculation costs. Scalability in all circumstances is the primary reason behind XGBoost's success. XGBoost can use the least number of resources to solve issues at real-world scale [37]. The block storage structure, which supports parallel computing, was adopted. Even with a lot of data, the algorithm can continue to operate quite efficiently. The loss function is given as:

$$\mathcal{L}(\phi) = \sum_i l(\hat{y}_i, y_i) + \sum_k \Omega(f_k) \tag{3}$$

where $\Omega\,(f) = \gamma T + \frac{1}{2}\,\lambda\omega^2$

The difference between the target $y_i$ and the predicted value $\hat{y}_i$ is measured by the differentiable convex loss function, denoted by $l$ in this case. The model's complexity is penalized by the second term, Ω. In order to prevent over-fitting, the extra regularization term smoothens the final learned weights [1,36]. This algorithm, which uses the gradient-boosting method, is the creation of a decision-tree-based model with boosting. In order for the integration process of this CART model to produce an effective prediction model with good time efficiency, the XGBoost algorithm reduces the gradient to form a basic classification and regression tree (CART) model using the errors from the previous model. The XGBoost algorithm's primary boosting concept is to start with a basic, inaccurate CART and iterate with a model that assesses the prior error to obtain an accurate model [38].

## 3. Evaluation Matrix

The quality of training and the predictability of the models were validated using coefficient of determination ($R^2$), shown as Equation (4); mean absolute error (*MAE*), shown as Equation (5); and mean square error (*MSE*), shown as Equation (6); all are given below.

$$\boldsymbol{R^2} = \mathbf{1} - \frac{\sum_{i=1}^{n}(\boldsymbol{X_i} - \boldsymbol{Y_i})^2}{\sum_{i=1}^{n}(\overline{\boldsymbol{Y}} - \boldsymbol{Y_i})^2} \tag{4}$$

$$\boldsymbol{MAE} = \frac{\mathbf{1}}{\boldsymbol{n}}\sum_{i=1}^{n}|\boldsymbol{X_i} - \boldsymbol{Y_i}| \tag{5}$$

$$\boldsymbol{MSE} = \frac{\mathbf{1}}{\boldsymbol{n}}\sum_{i=1}^{n}(\boldsymbol{X_i} - \boldsymbol{Y_i})^2 \tag{6}$$

In the equations, $n$ is the number of samples within training and testing datasets, $X_i$ is the predicted $i^{\text{th}}$ value, $\boldsymbol{Y_i}$ is the actual $i^{th}$ value, and $\overline{\boldsymbol{Y}}$ is the mean of true values. An improved model fit to the data is shown by $R^2$ values that are closer to 1. Regression model accuracy describing the experimental data should be improved by smaller values of *MAE* and *MSE*. To measure the variance amongst the actual and expected outcomes for each

observation, the mean square error, or *MSE*, is employed. The model's predicted values and actual values are quantified by the mean absolute error (*MAE*), which measures the average absolute difference between them [6,39].

## 4. Results and Discussion

The first experimental dataset used for modelling is from the study reported by Nathanael et al. [18]. Using titanium citrate and tannic acid as reducing agents and silver nitrate as the precursor, they synthesized AgNPs by a chemical route, as shown in Figure 2. In order to direct the experiments, Nathanael et al. created a DT algorithm that was based on 20 samples that were repeated three times each. To improve the efficacy of the prediction models, they employed ten additional trials, including three replications that were conducted randomly or in accordance with the DT-guided design of experiments. The machine learning model considers the size of the particle, characterized by transmission electron microscopy and dynamic light scattering measurements, as the dependent variable. The independent variables are the nucleation constant, growth rate constant, storage temperature, Reynolds number, and ratio of dean number to Reynolds number. Table 2 presents performance measure of the original dataset and the decision-tree-guided design of experiments approach for AgNP size prediction using all the three models adopted in their work [18].

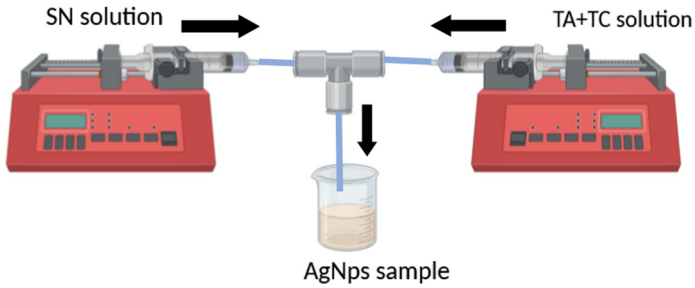


**Figure 2.** Experimental set up for synthesis of silver nanoparticles, adapted from [18].

**Table 2.** Performance measures of the existing model.

| Dataset | Model | *MSE* | *MAE* | $R^2$ Score |
|---|---|---|---|---|
| Original dataset | Random forest | 9.42 | 2.52 | 0.44 |
| | Decision tree | 9.43 | 2.46 | 0.45 |
| | XGBoost | 8.88 | 2.27 | 0.47 |
| Designed experiments | Random forest | 5.82 | 1.96 | 0.66 |
| | Decision tree | 6.41 | 2.01 | 0.62 |
| | XGBoost | 6.56 | 2.16 | 0.61 |

In the present investigation, we used the original dataset collected from the literature [18], which consisted of 20 samples that were replicated three times for a total of 60 datasets. All three models employed the same dataset, and the more accurate predictive model is obtained with appropriate hyperparameter optimization. The best performance is showcased by XGBoost model with learning_rate = 0.15, max_depth = 4, and gamma = 0.2, which results in the highest coefficient of determination of 0.973 and the lowest *MSE* and *MAE* of 1.09 and 0.73, respectively. Figure 3a compares the actual and the predicted values using the scatter plot. In the scatter plot, the points lie close to the best fit line with less deviation, indicating that it is a good model, and by plotting the probability density function for actual and predicted values, as in Figure 3b, it is observed that the peak of the actual and predicted values matches, which emphasizes the better predictability of the model. The results of the evaluation matrix for all three models are shown in Table 3.

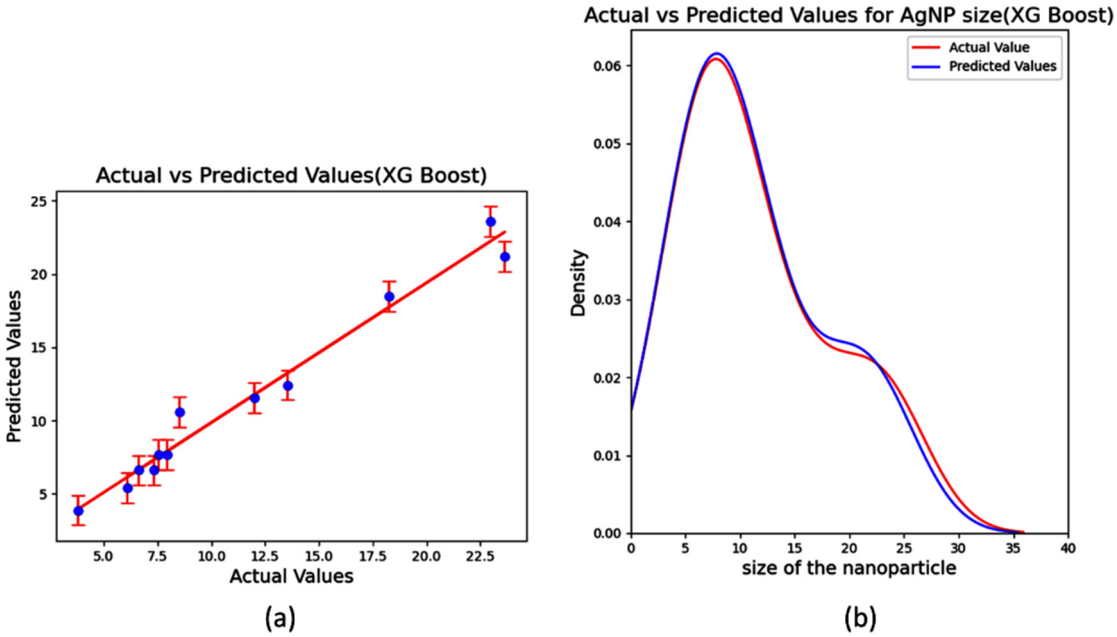


**Figure 3.** Actual versus predicted measurement of size of AgNPs for dataset 1 (**a**) Scatterplot with error bar; (**b**) Kernel density plot.

**Table 3.** Performance measures of the new model for dataset 1.

| Model | *MSE* | *MAE* | $R^2$ Score |
|---|---|---|---|
| Random forest | 1.18 | 0.75 | 0.971 |
| Decision tree | 2.28 | 1.13 | 0.95 |
| XGBoost | 1.09 | 0.73 | 0.973 |

A summary plot is useful to obtain a thorough insight of each feature value's contribution to the final forecast for each data point. The impact of each synthesis parameter is displayed in Figure 4 based on the feature value. For the model output, the synthesis parameters are organized in decreasing order of significance. This indicates that growth constant has the most influence on the outcome of the prediction, whereas the Reynolds number has the least for this model. The $R^2$ score of the model considering the most significant features such as the growth constant and nucleation constant is 0.973. If these two independent parameters are not considered, the accuracy of the model drops, giving an $R^2$ score of 0.561, showing how important these features are for predicting a size of the particle. Figure 5 shows the scatter plot with the error graph of actual values versus predicted values, avoiding the significant features in which the points lie away from the best fit line.

Tree-based models were used to assess the second batch of the 103 dataset, which was derived from Shafaei et al. [19] where they have synthesized AgNPs through the green approach. Shafaei et al. used a factorial D-optimal array design of experiments to gather experimental data in order to investigate the size of AgNPs. The AgNPs' size was examined by X-ray diffraction. For training and optimization, their study adopted particle swarm optimization (PSO) and ANN techniques. The $AgNO_3$/opium ratio, feed rate, pH, temperature, and agitation speed were among the independent characteristics that were used in the investigation. Throughout the synthesis process, these factors were altered to study its influence on the AgNPs' estimated grain size, which was determined using Scherer's equation. Since the findings from the various ANN PSO networks were inconsistent, the model was assessed using a combined function as in Equation (7), where the network with a combined function closer to zero has been proposed to perform better.

Because ANN-PSO 9's combined function value is less than 0.101, and the $R^2$ score is equal to 0.9972, it is regarded as the best network.

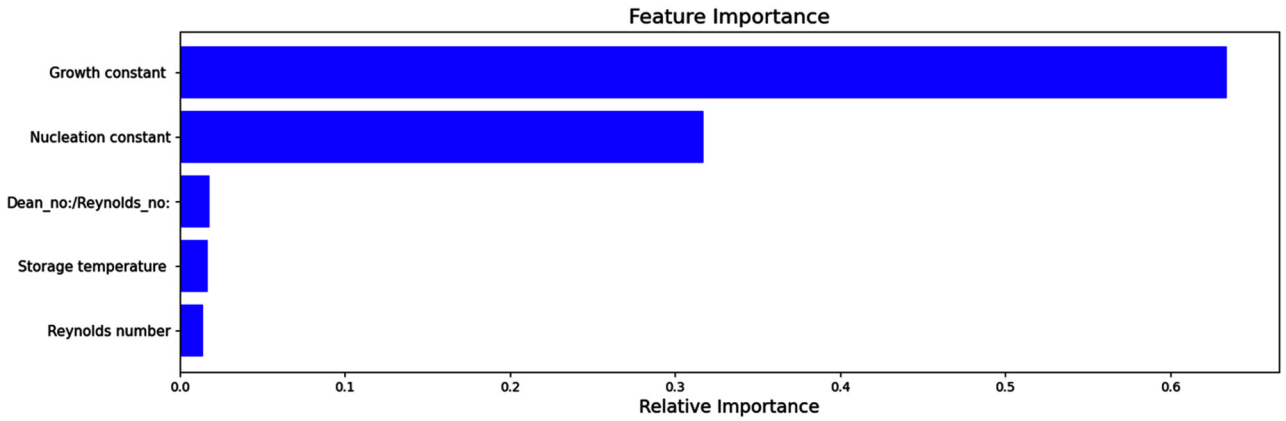


**Figure 4.** Feature importance in XGBoost model for dataset 1.

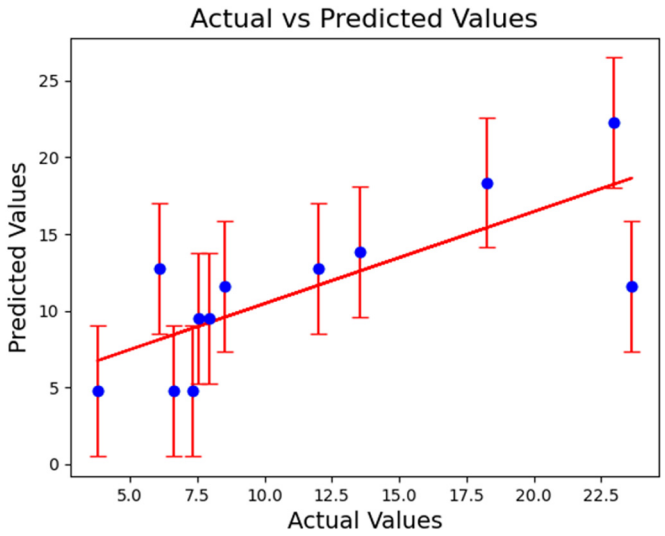


**Figure 5.** Scatter plot with error bar for actual versus predicted measurement of size of AgNPs, ignoring the most significant features: growth constant and nucleation constant.

$$\text{Combined function} = RMSE + MAPE + 1/R^2 \quad (7)$$

To simplify the modelling process, we have employed tree-based models in this instance. The models' efficiency was assessed using the $R^2$ score, *MAE*, and *MSE*. Table 4's results demonstrate that XGBoost with learning_rate = 0.05 and max_depth = 3 outperforms the other two tree models in terms of performance. Figure 6a,b display scatter plot with error graph and kernel density plot for actual values and the predicted values, showing the performance the model. It is observed that in the scatter plot, the points are more scattered and in the plot for probability density function, it is noted that there is a deviation between the peaks of the actual value and the predicted value, which shows the predicted value has a deviation from the actual value. In this case, the size of the particles was examined using XRD, indicating that it is the grain size rather than the AgNP particle size. This could be the reason for the models' lower score when compared to the datasets from the other three experiments. Figure 7 illustrates the relative importance of each feature and their role in AgNP size prediction. The relevance of the most important feature for this dataset is shown in Figure 8 by plotting the scatter plot. It is observed that the points are more scattered than

Figure 6a where all the features are taken into consideration for modelling. The accuracy of the model reduced, giving an $R^2$ score of 0.65 from 0.79 on avoiding the most significant feature contributing to the model prediction.

**Table 4.** Performance measures of the tree-based model for dataset 2.

| Model | *MSE* | *MAE* | $R^2$ Score |
|---|---|---|---|
| Random forest | 44.1 | 5.3 | 0.725 |
| Decision tree | 34.5 | 4.5 | 0.785 |
| XGBoost | 33.5 | 4.3 | 0.79 |

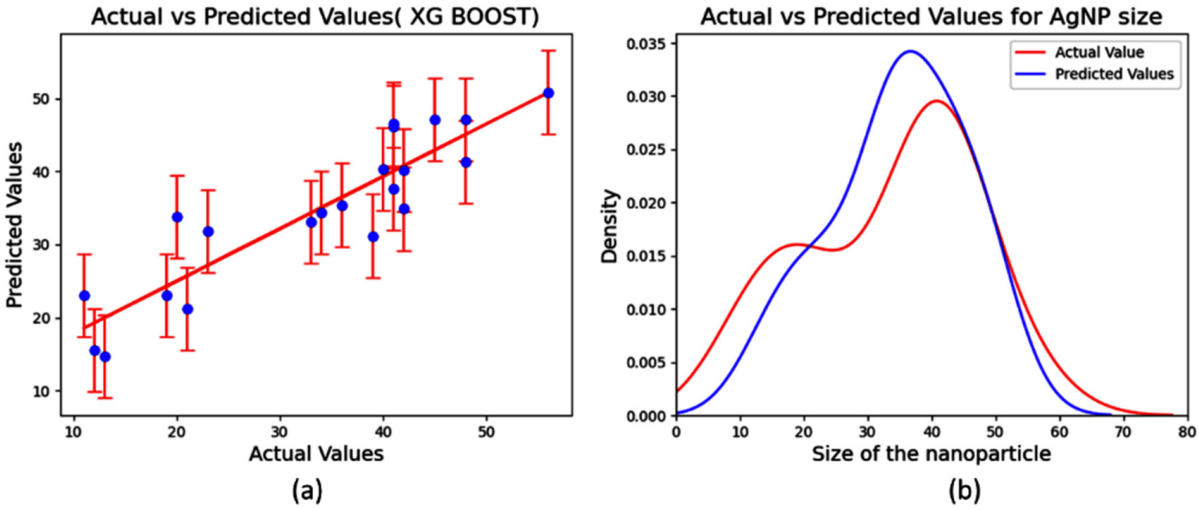


**Figure 6.** Actual versus predicted measurement of size of AgNPs for dataset 2. (**a**) Scatterplot with error bar; (**b**) Kernel density plot.

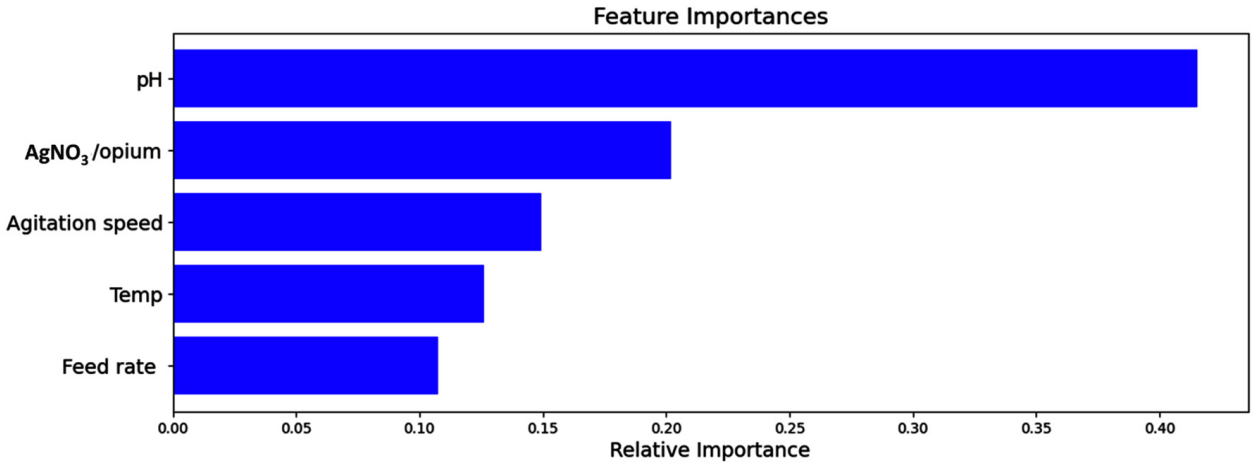


**Figure 7.** Feature importance in XGBoost model for dataset 2.

The experimental dataset 3 used for our modelling is the synthesized AgNPs by Shabanzadeh et al. [20] using Vitex negundo L. extract as a reducing agent and stabilizer; thirty prepared samples were made by this route. Using the weight percentage of Vitex negundo L. extract, reaction temperature, stirring time, and $AgNO_3$ molar concentration as input parameters, an ANN model was created to forecast the size of the nanoparticles in order to construct an efficient green nanoparticle synthesis process. The synthesized AgNPs' TEM analysis confirms the nanoparticles' size. The dataset was split into train, test,

and validation sets in this study. The correlation coefficient of the ANN model is around 0.998, with a mean square error of 0.4576.

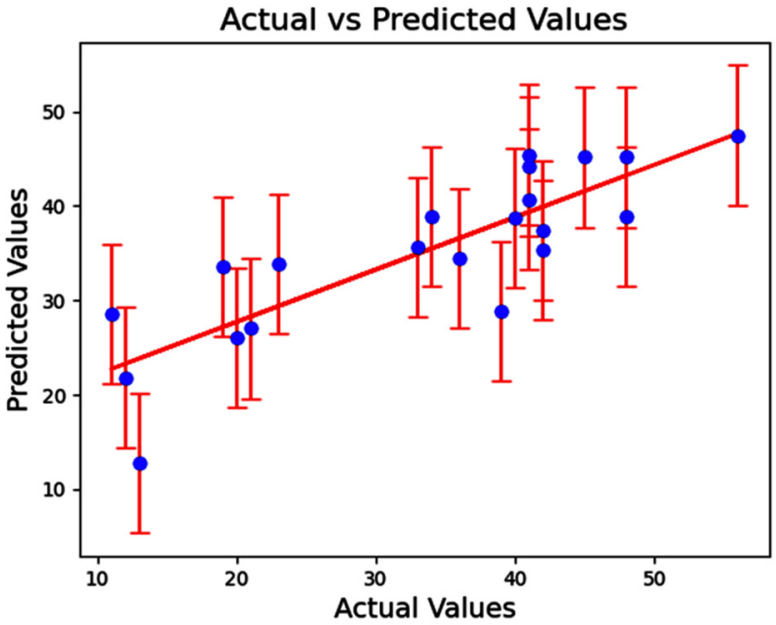


**Figure 8.** Actual versus predicted measurement of size of AgNPs avoiding the significant feature: pH.

Table 5 shows the results of the *MSE*, *MAE*, and $R^2$ score on employing the tree-based models. The dataset is split to train and test sets, with a best train score of 99.9 and test score of 99.4 exhibited by XGBoost with hyperparameter values of learning_rate = 0.15, max_depth = 3. This model has led to an $R^2$ score of 0.994, *MSE* of 0.145, and *MAE* of 0. 292. The result shows that the traditional models could also produce high precision in prediction while comparing with the complex neural networks. Figure 9a,b show the different plots of performance of the model, the scatter plot, and plot for probability density function comparing actual values and predicted values. Both plots reveal that model designed to predict the size of the particle performs well. For predicting the size of the nanoparticle in a model, each feature has its own importance and from Figure 10, we observe that plant extract contributes the most, giving a feature score of 0.750. As shown in Figure 11, we can observe more scattering of points in the plot, showing that there is a decrease in accuracy as we remove the most important feature from the model as compared to Figure 9a. The predictive ability of the model has decreased. When the most significant feature is not taken into account, the model's $R^2$ score is 0.93, which is less than the score obtained when the feature is taken into account.

**Table 5.** Performance measures of the tree-based models for dataset 3.

| Model | *MSE* | *MAE* | $R^2$ Score |
|---|---|---|---|
| Random forest | 0.41 | 0.55 | 0.980 |
| Decision tree | 0.39 | 0.52 | 0.984 |
| XGBoost | 0.145 | 0.292 | 0.994 |

The fourth set of data was obtained from Shabanzadeh et al. [21]. Here, the research group has used an ANN with four neurons in one hidden layer to model the optimum size of AgNP nanoparticles to investigate the effects of different input parameters—$AgNO_3$ concentration, temperature, wavelength, and MMT interlayer d-spacing. The dataset was split into a test, train, and validation set to determine the best model. *RMSE* and $R^2$ score values are 0.7917 and 0.955 for test set.

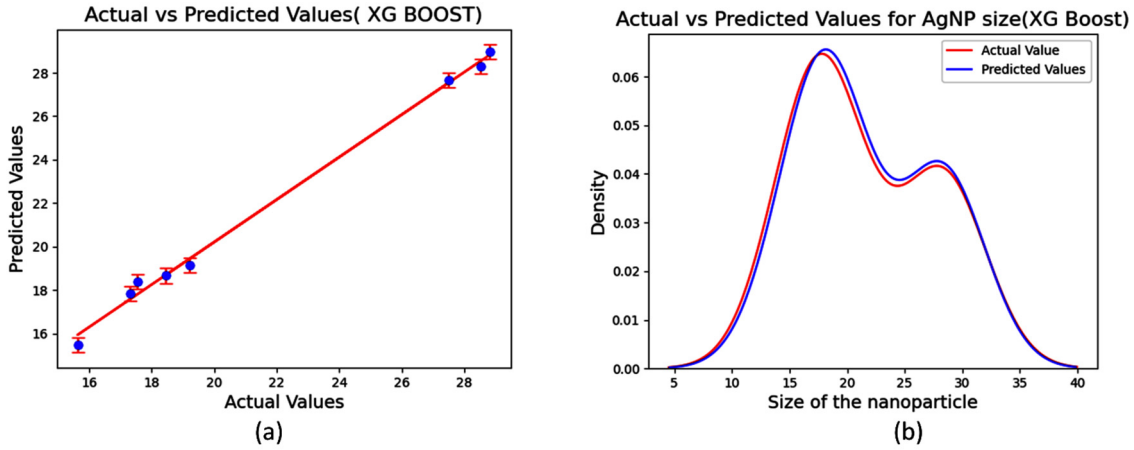


**Figure 9.** Actual versus predicted measurement of size of AgNPs for dataset 3. (**a**) Scatterplot with error bar; (**b**) Kernel density plot.

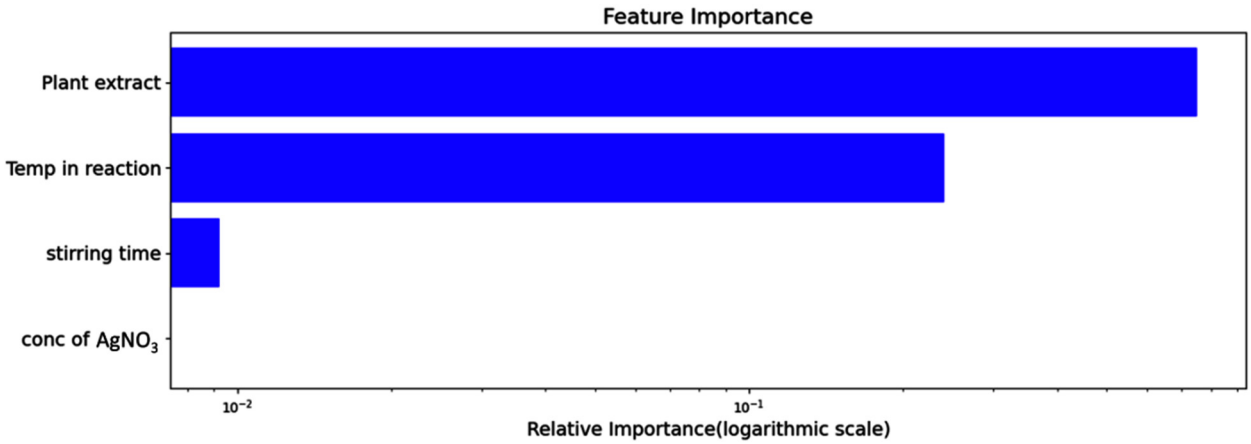


**Figure 10.** Feature importance in XGBoost model for dataset 3.

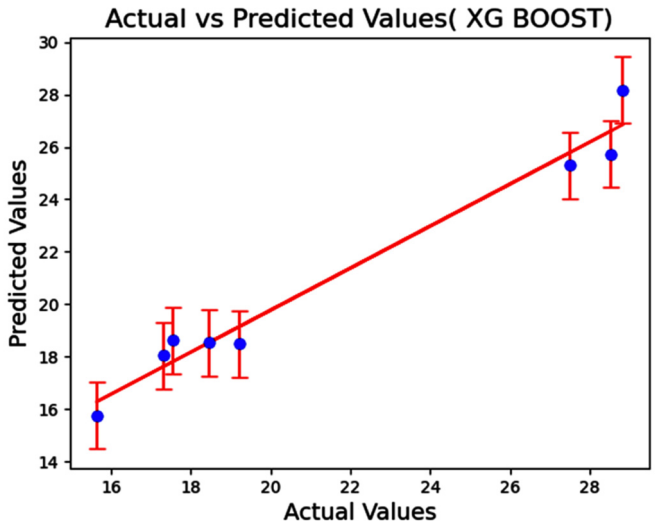


**Figure 11.** Actual versus predicted measurement of size of AgNPs avoiding the significant feature: weight percentage of plant extract.

Table 6 shows the performance measures of the tree-based models in which the DT model shows the best performance giving the coefficient of determination as nearly 1.

Figure 12a shows the plot for the actual versus predicted values. Both visualizations in Figure 12a,b offer valuable insights into the regression model's performance: the scatter plot directly compares the model's predictions to the actual values, while the probability density function plot provides a deeper understanding of error distribution. Here in the scatter plot, the points lie almost on the line and the kernel density plots overlap each other, showing the decision tree models predict extremely well. The data are split in the ratio of 80:20. The model gives the best train score of 100 and test score of 99.95. The *RMSE* value is 0. 045. The concentration of silver nitrate is the most important feature that contributes to the prediction of size of AgNPs, giving a score of 0.964. Figure 13 shows the feature importance of the model. Figure 14 illustrates the scatter plot avoiding the most significant feature of the model. The model's accuracy has declined, resulting in a lower $R^2$ score of 0.65.

**Table 6.** Performance measures of the tree-based models for dataset 4.

| Model | *MSE* | *MAE* | $R^2$ Score |
|---|---|---|---|
| Random forest | 0.0026 | 0.039 | 0.9992 |
| Decision tree | 0.0019 | 0.0316 | 0.9995 |
| XGBoost | 0.017 | 0.086 | 0.995 |

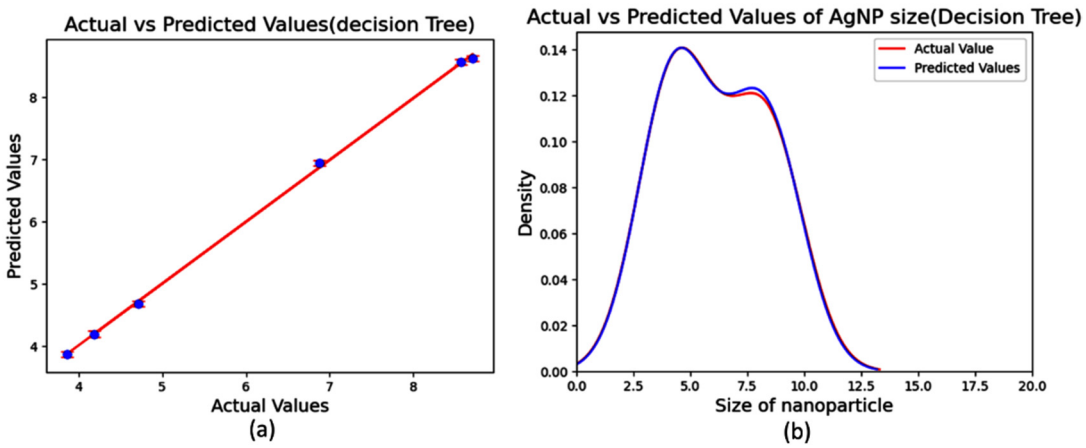


**Figure 12.** Actual versus predicted measurement of size of AgNPs for dataset 4. (**a**) Scatterplot with error bar; (**b**) Kernel density plot.

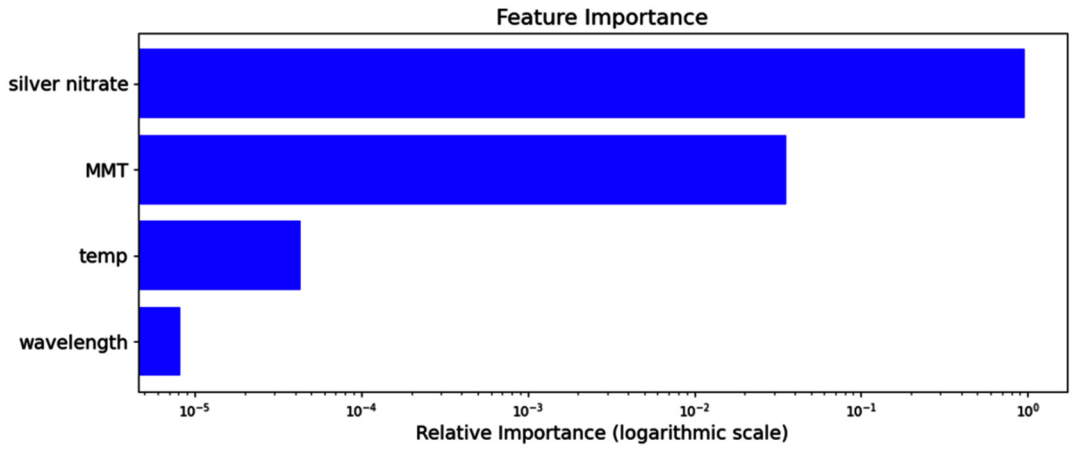


**Figure 13.** Feature importance in decision tree model for dataset 4.

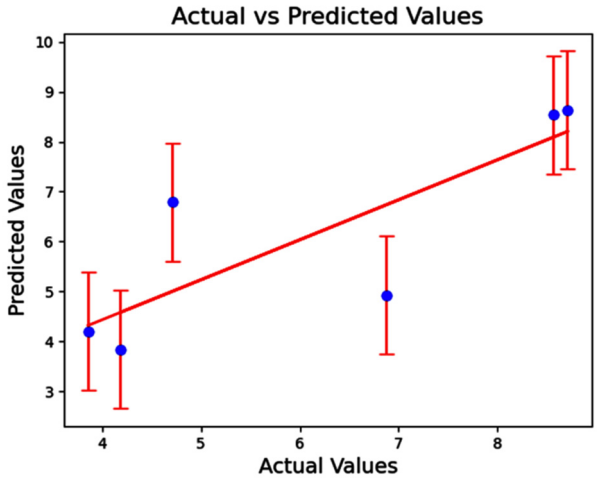


**Figure 14.** Actual versus predicted measurement of size of AgNPs avoiding the significant feature: concentration of silver nitrate.

## 5. Conclusions

In the present work, the size of the silver nanoparticles synthesized using chemical and green synthesis methods is predicted using tree-based machine learning models. Tree-based models exhibits strong predictive accuracy. In each of the reference papers considered in this work, researchers have used different independent features for synthesis. Results show that the tree-based models predict well for different synthesis parameter and, hence, could serve as substitutes for the complex models, which are computationally intensive. The optimal models for every dataset are listed below, along with the accuracy attained.

- For dataset 1 used from Nathanael et al. [18], instead of modelling their dataset directly, they used DT, RF, and XGBoost, which were assisted by a DT-based design of experiments. This allowed them to achieve the highest score across all three tree models. 0.66 was the group's highest $R^2$ score. Modelling the original dataset directly with a tree-based model, they achieved an $R^2$ score of 0.47 with XGBoost. Here, we used the original dataset for the nanoparticle synthesis, and acheived $R^2$ score of 0.973 with the XGBoost model.
- Shafaei et al. [19] used 103 datasets to obtain a model for predicting the grain size of nanoparticles, which was determined through XRD. The ANN PSO model achieved a best $R^2$ score of 0.9972. Using the tree-based models in our study, XGBoost is found to be the best model among the other two models with a $R^2$ score of 0.79.
- Shabanzadeh et al. [20] synthesized AgNPs utilizing Vitex negundo L. extract as a stabilizing and reducing agent This is dataset 3 for our models. An ANN model was constructed using thirty prepared samples. Size confirmation of the synthesized AgNPs is made by TEM image analysis. According to their analysis, the ANN model's correlation determination is approximately 0.998, with a MSE of 0.4576. The XGB model shows a better *MSE* value of 0.145 and *MAE* of 0.292 when compared to the three tree-based models in this study, and it shows an $R^2$ score of 0.994, which is compatible with the ANN model. As compared to complicated neural networks, the outcome demonstrates that tree-based models in the present work could also make a better prediction of size of nanoparticles.
- The fourth dataset for modelling was gathered from Shabanzadeh et al.'s [21] study. *RMSE* and $R^2$ values for the test set are 0.7917 and 0.955 respectively. Considering these data for tree based modelling, the DT model performs the best, with a coefficient of determination 0.999 and an *RMSE* equal to 0.045, as observed in the present work.

**Author Contributions:** Conceptualization, A.P. and R.J.; methodology, A.P. and R.J; Software, A.P.; validation, A.P. and R.J.; formal analysis, A.P. and R.J.; investigation, A.P.; resources, data curation, A.P.; writing—A.P.; writing—review and editing, A.P., R.J. and T.S.S.; visualization, A.P. and R.J.; supervision, R.J. and T.S.S. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**Data Availability Statement:** The data presented in this study are available in article.

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

## References

1. NgJin, K.; Wang, W.; Qi, G.; Peng, X.; Gao, H.; Zhu, H.; He, X.; Zou, H.; Yang, L.; Yuan, J.; et al. An Explainable Machine-Learning Approach for Revealing the Complex Synthesis Path-Property Relationships of Nanomaterials. *Nanoscale* **2023**, *15*, 15358–15367. [CrossRef] [PubMed]
2. Lv, H.; Chen, X. Intelligent Control of Nanoparticle Synthesis through Machine Learning. *Nanoscale* **2022**, *14*, 6688–6708. [CrossRef] [PubMed]
3. Lu, W.; Xiao, R.; Yang, J.; Li, H.; Zhang, W. Data Mining-Aided Materials Discovery and Optimization. *J. Mater.* **2017**, *3*, 191–201. [CrossRef]
4. Tao, H.; Wu, T.; Aldeghi, M.; Wu, T.C.; Aspuru-Guzik, A.; Kumacheva, E. Nanoparticle Synthesis Assisted by Machine Learning. *Nat. Rev. Mater.* **2021**, *6*, 701–716. [CrossRef]
5. Jones, D.E.; Ghandehari, H.; Facelli, J.C.; City, S.L.; Chemistry, P.; City, S.L.; City, S.L. A Review of the Applications of Data Mining and Machine Learningfor the Prediction of Biomedical Properties of Nanoparticles. *Comput. Methods Programs Biomed.* **2017**, *132*, 93–103. [CrossRef]
6. Hoseini, B.; Jaafari, M.R.; Golabpour, A.; Momtazi-Borojeni, A.A.; Karimi, M.; Eslami, S. Application of Ensemble Machine Learning Approach to Assess the Factors Affecting Size and Polydispersity Index of Liposomal Nanoparticles. *Sci. Rep.* **2023**, *13*, 18012. [CrossRef] [PubMed]
7. Kalantari, K.; Mostafavi, E.; Afifi, A.M.; Izadiyan, Z.; Jahangirian, H.; Rafiee-Moghaddam, R.; Webster, T.J. Wound Dressings Functionalized with Silver Nanoparticles: Promises and Pitfalls. *Nanoscale* **2020**, *12*, 2268–2291. [CrossRef] [PubMed]
8. Fytianos, G.; Rahdar, A.; Kyzas, G.Z. Nanomaterials in Cosmetics: Recent Updates. *Nanomaterials* **2020**, *10*, 979. [CrossRef]
9. Mo, L.; Guo, Z.; Yang, L.; Zhang, Q.; Fang, Y.; Xin, Z. Silver Nanoparticles Based Ink with Moderate Sintering in Flexible and Printed Electronics. *Int. J. Mol. Sci.* **2019**, *20*, 2124. [CrossRef]
10. Raza, M.A.; Kanwal, Z.; Rauf, A.; Sabri, A.N.; Riaz, S.; Naseem, S. Size- and Shape-Dependent Antibacterial Studies of Silver Nanoparticles Synthesized by Wet Chemical Routes. *Nanomaterials* **2016**, *6*, 74. [CrossRef]
11. Dong, Y.; Zhu, H.; Shen, Y.; Zhang, W.; Zhang, L. Antibacterial Activity of Silver Nanoparticles of Different Particle Size against Vibrio Natriegens. *PLoS ONE* **2019**, *14*, e0222322. [CrossRef] [PubMed]
12. Lahuta, L.B.; Szablińska-Piernik, J.; Stałanowska, K.; Głowacka, K.; Horbowicz, M. The Size-Dependent Effects of Silver Nanoparticles on Germination, Early Seedling Development and Polar Metabolite Profile of Wheat (*Triticum aestivum* L.). *Int. J. Mol. Sci.* **2022**, *23*, 13255. [CrossRef] [PubMed]
13. Shabanzadeh, P.; Norazaksenu; Shameli, K.; Ismail, F.; Mohagheghtabar, M. Application of Artificial Neural Network (ANN) for the Prediction of Size of Silver Nanoparticles Prepared by Green Method. *Dig. J. Nanomater. Biostruct.* **2013**, *8*, 541–549.
14. Shabanzadeh, P.; Yusof, R.; Shameli, K. Neural Network Modeling for Prediction Size of Silver Nanoparticles in Montmorillonite/Starch Synthesis By Chemical Reduction Method. *Dig. J. Nanomater. Biostruct.* **2014**, *9*, 1699–1711.
15. Shabanzadeh, P.; Senu, N.; Shameli, K.; Ismail, F.; Zamanian, A.; Mohagheghtabar, M. Prediction of Silver Nanoparticles' Diameter in Montmorillonite/Chitosan Bionanocomposites by Using Artificial Neural Networks. *Res. Chem. Intermed.* **2015**, *41*, 3275–3287. [CrossRef]
16. Shabanzadeh, P.; Yusof, R.; Shameli, K. Artificial Neural Network for Modeling the Size of Silver Nanoparticles' Prepared in Montmorillonite/Starch Bionanocomposites. *J. Ind. Eng. Chem.* **2015**, *24*, 42–50. [CrossRef]
17. Sattari, R.; Khayati, G.R. Prediction of the Size of Silver Nanoparticles Prepared via Green Synthesis: A Gene Expression Programming Approach. *Sci. Iran.* **2020**, *27*, 3399–3411. [CrossRef]
18. Nathanael, K.; Cheng, S.; Kovalchuk, N.M.; Arcucci, R.; Simmons, M.J.H. Optimization of Microfluidic Synthesis of Silver Nanoparticles: A Generic Approach Using Machine Learning. *Chem. Eng. Res. Des.* **2023**, *193*, 65–74. [CrossRef]
19. Shafaei, A.; Khayati, G.R. A Predictive Model on Size of Silver Nanoparticles Prepared by Green Synthesis Method Using Hybrid Artificial Neural Network-Particle Swarm Optimization Algorithm. *Measurement* **2020**, *151*, 107199. [CrossRef]
20. Shabanzadeh, P.; Yusof, R.; Shameli, K. Modeling of Biosynthesized Silver Nanoparticles in Vitex Negundo L. Extract by Artificial Neural Network. *RSC Adv.* **2015**, *5*, 87277–87285. [CrossRef]
21. Shabanzadeh, P.; Senu, N.; Shameli, K.; Tabar, M.M. Artificial Intelligence in Numerical Modeling of Silver Nanoparticles Prepared in Montmorillonite Interlayer Space. *J. Chem.* **2013**, *2013*, 305713. [CrossRef]

22. Myles, A.J.; Feudale, R.N.; Liu, Y.; Woody, N.A.; Brown, S.D. An Introduction to Decision Tree Modeling. *J. Chemom.* **2004**, *18*, 275–285. [CrossRef]
23. Géron, A. *Hands-on Machine Learning Whith Scikit-Learing, Keras and Tensorfow*; O'Reilly Media, Inc.: Sebastopol, CA, USA, 2019; ISBN 978-1-492-03264-9.
24. Moslehi, S.; Rabiei, N.; Soltanian, A.R.; Mamani, M. Application of Machine Learning Models Based on Decision Trees in Classifying the Factors Affecting Mortality of COVID-19 Patients in Hamadan, Iran. *BMC Med. Inform. Decis. Mak.* **2022**, *22*, 192. [CrossRef] [PubMed]
25. Ozcan, M.; Peker, S. A Classification and Regression Tree Algorithm for Heart Disease Modeling and Prediction. *Healthc. Anal.* **2023**, *3*, 100130. [CrossRef]
26. Shahid, S.; Javaid, A. Application of Machine Learning Decision Tree in Diagnosing Joint Pain. In Proceedings of the 2022 Medical Technologies Congress (TIPTEKNO), Antalya, Turkey, 31 October–2 November 2022; pp. 1–4. [CrossRef]
27. Abellán, J.; López, G.; De Oña, J. Analysis of Traffic Accident Severity Using Decision Rules via Decision Trees. *Expert Syst. Appl.* **2013**, *40*, 6047–6054. [CrossRef]
28. Tamir, T.S.; Xiong, G.; Li, Z.; Tao, H.; Shen, Z.; Hu, B.; Menkir, H.M. Traffic Congestion Prediction Using Decision Tree, Logistic Regression and Neural Networks. *IFAC-PapersOnLine* **2020**, *53*, 512–517. [CrossRef]
29. Xie, S.; Hu, G.; Wang, X.; Xing, C.; Liu, Y. A Decision Tree-Based Online Traffic Classification Method for QoS Routing in Data Center Networks. *Secur. Commun. Netw.* **2022**, *2022*, 9419676. [CrossRef]
30. Desai, A.S.; Ashok, A.; Edis, Z.; Bloukh, S.H.; Gaikwad, M.; Patil, R.; Pandey, B.; Bhagat, N. Meta-Analysis of Cytotoxicity Studies Using Machine Learning Models on Physical Properties of Plant Extract-Derived Silver Nanoparticles. *Int. J. Mol. Sci.* **2023**, *24*, 4220. [CrossRef]
31. Geurts, P.; Ernst, D.; Wehenkel, L. Extremely Randomized Trees. *Mach. Learn.* **2006**, *63*, 3–42. [CrossRef]
32. Liaw, A.; Wiener, M. Classification and Regression by RandomForest. *R News J.* **2002**, *2*, 18–22.
33. Zhang, Y.; Haghani, A. A Gradient Boosting Method to Improve Travel Time Prediction. *Transp. Res. Part C Emerg. Technol.* **2015**, *58*, 308–324. [CrossRef]
34. Gholizadeh, M.; Jamei, M.; Ahmadianfar, I.; Pourrajab, R. Prediction of Nanofluids Viscosity Using Random Forest (RF) Approach. *Chemom. Intell. Lab. Syst.* **2020**, *201*, 104010. [CrossRef]
35. Natekin, A.; Knoll, A. Gradient Boosting Machines, a Tutorial. *Front. Neurorobot.* **2013**, *7*, 21. [CrossRef] [PubMed]
36. Chen, T.; Guestrin, C. XGBoost: A Scalable Tree Boosting System. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 13–17 August 2016; pp. 785–794. [CrossRef]
37. Wen, H.T.; Wu, H.Y.; Liao, K.C. Using XGBoost Regression to Analyze the Importance of Input Features Applied to an Artificial Intelligence Model for the Biomass Gasification System. *Inventions* **2022**, *7*, 126. [CrossRef]
38. Fajrul Haqqi, M.; Saroji, S.; Prakoso, S. An Implementation of XGBoost Algorithm to Estimate Effective Porosity on Well Log Data. *J. Phys. Conf. Ser.* **2023**, *2498*, 012011. [CrossRef]
39. Chicco, D.; Warrens, M.J.; Jurman, G. The Coefficient of Determination R-Squared Is More Informative than SMAPE, MAE, MAPE, MSE and RMSE in Regression Analysis Evaluation. *PeerJ Comput. Sci.* **2021**, *7*, e623. [CrossRef]