

Feature-Driven Nano-QSTR Modeling for Nanoparticle Cytotoxicity Prediction using Ensemble Machine Learning

Aries P. Valeriano ^{1,*}, Charla S. Paredes ², Farley G. Bondaug ^{3,4}, Irelie P. Ebardo ^{3,4}, Penelope P. Almonte ⁴, Michelle Amor P. Sabugaa ⁴, John Riz V. Bagnol ⁵, Rey Y. Capangpangan ⁶

¹ Department of Mathematics and Statistics, University of Southeastern Philippines, Davao City, 8000, Davao del Sur, Philippines; apvaleriano@usep.edu.ph;

² Department of Biological Sciences, Mindanao State University-Iligan Institute of Technology, Iligan City, 9200, Lanao del Norte, Philippines; charla.paredes@g.msuiit.edu.ph;

³ Department of Science and Technology - Science Education Institute, Taguig City, 1631, Metro Manila, Philippines; farley.bondaug@msunaawan.edu.ph (F.B.); irelie.ebardo@msunaawan.edu.ph (I.E.);

⁴ Research on Environment and Nanotechnology Laboratories, Research Division, Mindanao State University at Naawan, Naawan, 9023, Misamis Oriental, Philippines; penelope.almonte@msunaawan.edu.ph (P.A.); michelleamor.sabugaa@msunaawan.edu.ph (M.A.S.);

⁵ Laboratoire de Mathématiques Appliquées de Compiègne, University of Technology of Compiègne, Compiègne, 60200, France; john-riz.bagnol@utc.fr;

⁶ Department of Physical Sciences and Mathematics, Mindanao State University at Naawan, Naawan, 9023, Misamis Oriental, Philippines; rey.capangpangan@msunaawan.edu.ph

* Correspondence: apvaleriano@usep.edu.ph

Received: 7.02.2026; Accepted: 17.05.2026; Published: 30.09.2026

Abstract: The rapid growth of engineered nanoparticles in medicine, electronics, and industry has increased the need for reliable toxicity assessment. In response, this study develops nano-QSTR regression models using Extreme Gradient Boosting (XGBoost) and Random Forest (RF), with Ridge regression included as a baseline model, to predict nanoparticle cytotoxicity measured as percent cell viability. The models incorporate features related to nanoparticle properties, biological characteristics, exposure conditions, synthesis methods, and surface coatings. A dataset of 4,856 samples was used, with 3,887 for training and 969 for testing. During repeated cross-validation (rcv) on the training data, XGBoost achieved $R^2_{rcv}=81.84\%\pm 1.79$, $RMSE_{rcv}=12.73\%\pm 0.60$, and $MAE_{rcv}=8.15\%\pm 0.29$, and RF reached $R^2_{rcv}=76.42\%\pm 2.46$, $RMSE_{rcv}=14.79\%\pm 0.90$, and $MAE_{rcv}=10.01\%\pm 0.40$. Whereas, Ridge performed substantially worse ($R^2_{rcv}=33.74\%\pm 2.53$, $RMSE_{rcv}=24.33\%\pm 0.61$, $MAE_{rcv}=18.42\%\pm 0.45$). External validation confirmed these trends: XGBoost achieved $R^2_{test}=80.25\%$, $RMSE_{test}=13.09\%$, and $MAE_{test}=9.01\%$, RF reached $R^2_{test}=75.59\%$, $RMSE_{test}=14.87\%$, $MAE_{test}=10.32\%$, and Ridge obtained $R^2_{test}=33.02\%$, $RMSE_{test}=24.13\%$, $MAE_{test}=18.27\%$. Feature importance analysis from the ensemble models identified primarily nanoparticle concentration, size, diameter, zeta potential, exposure time, number of cells, and synthesis method as key drivers of cytotoxicity. These results demonstrate that ensemble models can reliably predict cytotoxicity and highlight factors critical for early safety assessment.

Keywords: ensemble machine learning; nano-QSTR; cytotoxicity.

© 2026 by the authors. This article is an open-access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited. The authors retain copyright of their work, and no permission is required from the authors or the publisher to reuse or distribute this article, as long as proper attribution is given to the original source.

1. Introduction

Nanotechnology has enabled the creation of engineered nanoparticles with unique physicochemical properties, which are increasingly applied in medicine, electronics, and industrial processes [1, 2]. However, these materials can interact with biological systems in complex ways, potentially reducing cell viability or inducing cytotoxic effects [3, 4]. Experimental evaluation of nanoparticle toxicity is often costly, time-consuming, and ethically challenging, limiting the ability to screen the rapidly growing variety of nanoparticles [5, 6]. Computational approaches, such as quantitative structure–activity relationship (QSAR) or quantitative structure–toxicity relationship (QSTR) modeling, offer a practical solution by linking measurable nanoparticle and exposure features to biological responses. These *in silico* methods can support rapid toxicity assessment and guide the design of safer nanoparticles [6, 7].

Numerous studies have applied nano-QSAR or nano-QSTR models to predict nanoparticle cytotoxicity or effects on cell viability. Early work by Puzyn *et al.* showed that nano-QSAR models could predict the cytotoxicity of metal oxide nanoparticles using descriptors that capture size, composition, and surface properties [5, 7]. Researchers have also developed nano-QSAR models to understand the cytotoxic properties of TiO₂-based nanoparticles using SMILES-based descriptors and structural features [8].

Recent machine learning-based nano-QSAR studies demonstrate the growing role of advanced algorithms in nanosafety assessment. For example, Random Forest and AdaBoost have been used to predict cytotoxic responses of nano-TiO₂ and heavy metal mixtures in human cells, achieving improved accuracy when combined with data clustering [9]. Meneses *et al.* developed nano-QSTR models using tree-based ensemble methods, including Random Forest, Extra Trees, and Gradient Boosting, showing that nonlinear learners can effectively capture complex structure–toxicity relationships [10]. Similarly, Park *et al.* (2025) evaluated a multi-target nano-QSAR framework for continuous cytotoxicity prediction using XGBoost, Random Forest, Support Vector Regression, Extra Trees, Gradient Boosting, CatBoost, Transformer, and Multilayer Perceptron, providing a comprehensive example of ensemble methods applied to regression tasks [11]. Despite their predictive power, these studies relied on relatively limited feature sets focused on specific physicochemical or structural properties, which may restrict model generalizability across diverse nanoparticles and exposure conditions.

In contrast, Ahmadi *et al.* applied several machine learning classifiers, including Decision Tree, Random Forest, Support Vector Machine, Naïve Bayes, and Artificial Neural Network, to predict nanoparticle toxicity using physicochemical properties, exposure conditions, and cellular response data. Their study formulated toxicity prediction as a classification task based on cell death responses. While these models effectively distinguished toxicity outcomes, categorical endpoints limit their ability to capture continuous variation in cytotoxicity [12]. Valeriano *et al.* further demonstrated high predictive performance using regularized regression models, such as Ridge, Lasso, and Elastic Net, consistently identifying cell tissue, material, synthesis method, cell type, coat or functional group, and cell source as key features. However, these models primarily addressed classification tasks using binary-coded cell viability rather than continuous percentage cell viability [13].

Collectively, previous nano-QSAR and nano-QSTR studies show three main limitations. First, many rely on a limited set of features. Second, several treat toxicity predictions as a classification problem (toxic vs. non-toxic) rather than predicting the exact

percentage of cell viability. Third, even when ensemble methods are used, researchers often do not compare their performance with simpler regression models to clearly demonstrate their added value.

To address these gaps, this study develops nano-QSTR regression models that predict continuous cytotoxicity based on percentage cell viability. The framework integrates nanoparticle physicochemical properties, exposure conditions, biological descriptors, and synthesis and coating information. We implemented Extreme Gradient Boosting (XGBoost) and Random Forest (RF) as ensemble methods and compared them directly with Ridge regression as a baseline to assess whether more complex approaches improve predictive performance.

XGBoost and Random Forest (RF) were selected because they are widely used ensemble methods that perform well on structured tabular data. Random Forest is a bagging-based ensemble, which means it builds many decision trees using random subsets of the data and features, then combines their predictions. This helps reduce overfitting, capture complex patterns, and make the model more stable [14].

XGBoost is a boosting-based ensemble, which builds trees sequentially, with each tree focusing on the errors of the previous ones. This process allows boosting to gradually improve predictions by combining the strengths of multiple weak learners. XGBoost adds regularization and optimized tree construction to improve both accuracy and computational efficiency [15]. Comparative studies show that XGBoost provides a good balance of training speed and predictive performance compared with other gradient boosting methods, such as LightGBM and CatBoost [16].

By using both RF and XGBoost, this study compares two different ensemble strategies, bagging and boosting, within the same nano-QSTR framework, demonstrating how these approaches capture patterns in the data differently. In addition, Ridge regression is used as a simple linear baseline model that addresses multicollinearity without removing any features [17]. Valeriano *et al.* have shown that this model performs well when cytotoxicity is treated as a binary outcome; however, its performance may decline when predicting continuous cytotoxicity. This provides a clear reference point for evaluating the advantages of nonlinear ensemble models [13].

To ensure reliable results, model development follows OECD guidelines for QSAR or QSTR studies, including clear endpoint definition, unambiguous algorithms, appropriate validation, applicability domain assessment, and mechanistic interpretation, if possible [18].

2. Materials and Methods

2.1. Data description.

This study utilized a structured nanotoxicity dataset adapted from Valeriano *et al.* [13], originally compiled by Gul *et al.* [19], which provides experimentally measured cytotoxicity in terms of cell viability (%) as the target variable for engineered nanoparticles. It contained $n = 4683$ observations, and each observation included 25 features categorized as follows: physicochemical properties: diameter (nm), size in medium (nm), size in water (nm), zeta in medium (mV), zeta in water (mV), surface charge, shape, aspect ratio, polydispersity index (PDI), core material, and nanoparticle type. Exposure-related features: concentration ($\mu\text{g/ml}$) and exposure time (hr). Biological features (including assay-related): number of cells (cells per well), cell type, cell tissue origin, subject (human or animal), cell source, cell morphology, cell

age, cell line or primary cell, cell viability test, and test indicator. Synthesis and coating information: coat functional group and synthesis method. After removing features with a high percentage of missing values and observations with cell viability values less than 0, a total of 21 features and 4856 observations were retained for modeling.

2.2. Software and packages.

The results of this study were generated using R software [20]. Data wrangling and visualization were performed mainly using functions from the tidyverse and ggplot2 packages, respectively [21]. Missing values were handled using the kNN() function from the VIM package [22]. Feature selection was conducted with the Boruta package using its Boruta() and TentativeRoughFix() functions [23], while machine learning models were implemented using the caret package for training, hyperparameter tuning, and model evaluation [24]. Specifically, the train() function from caret was used with the argument method set to "xgbTree", "rf", and "glmnet" for training XGBoost, Random Forest, and Ridge regression models, respectively. Standardization and dummy encoding were performed using recipe(), step_center(), step_dummy(), prep(), and bake() functions from the recipes package [25]. Principal component analysis and Hotelling's T^2 for applicability domain assessment were carried out using base R functions, such as prcomp().

2.3. Data preparation, model development, and evaluation.

2.3.1. Data splitting.

Prior to splitting, observations with cell viability < 0 were removed. The cleaned dataset ($n = 4856$) was then divided into an 80% training set ($n = 3887$) and a 20% testing set ($n = 969$) using stratified random sampling to preserve the distribution of cytotoxicity values across both subsets. We used the training set for model fitting and hyperparameter tuning, and held out the testing set for external validation to provide an unbiased assessment of predictive performance. An 80:20 split was selected to balance model estimation and generalization assessment. This split allows the majority of the data to be used for model learning while retaining a sufficiently large independent test set for reliable performance evaluation. Allocating substantially more than 80% of the data to training may weaken the reliability of performance evaluation due to a limited test set, whereas allocating more than 40% to testing may reduce the stability of parameter estimation. Given the dataset size ($n=4856$) and the use of repeated cross-validation within the training set, this proportion supports robust model estimation while maintaining adequate data for independent evaluation of predictive generalization [26, 27].

2.3.2. Data preprocessing.

Data preprocessing was conducted to prepare the dataset for machine learning modeling. The preprocessing consisted of several steps, namely missing value handling, categorical feature handling, scaling, and feature selection. The first step, missing value handling, involved excluding features with a high proportion of missing values (i.e., $> 50\%$, corresponding to about 1,944 or more observations) to reduce noise and enhance model reliability. Moreover, missing values in the remaining features were imputed using k-nearest neighbors (kNN) with $k = 5$ [22]. Distances between observations in the kNN algorithm were calculated using a variation of Gower's distance, suitable for mixed data (i.e., both categorical

and numeric features). Each feature was weighted according to its importance. Numeric values were imputed as a weighted mean, while categorical values were imputed as the most common category among the neighbors. The choice of $k=5$ follows the default value that balances the use of too few neighbors, which can make imputation unstable, and too many neighbors, which can oversmooth the data. This provides reliable imputation while preserving local patterns in the nanotoxicity data [28]. The second step, categorical feature handling, involved grouping rare categories (categories $<1.5\%$ of the total observations or $n<58$) within categorical variables into a single category labeled “Others” to reduce sparsity. This threshold was chosen to prevent unstable parameter estimates and overfitting from very small groups while retaining meaningful variability in the data. Moreover, all categorical features were then converted to binary numeric variables using dummy encoding to make them compatible with machine learning algorithms. The third step, scaling, involved standardizing numeric features using z-score transformation to ensure comparability across features measured on different scales. The fourth and final step, feature selection, used the Boruta algorithm, a random forest-based method that identifies relevant predictors by comparing each feature's importance to shadow (randomly permuted) versions of the same feature [23]. Features confirmed as important by Boruta were retained for modeling, while features deemed uninformative were removed.

All preprocessing steps, including kNN imputation, rare category lumping, dummy encoding, standardization, and feature selection using the Boruta algorithm, were applied exclusively to the training set. The same transformations were then applied to the testing set, with missing values imputed independently within the test data using the same kNN procedure ($k=5$ weighted Gower distance for mixed features). That is, categorical levels in the test set were aligned with those in the training set, and numeric features were standardized using the training set means and standard deviations. Independent imputation of the test set was necessary because the VIM kNN algorithm for mixed data cannot directly use the training set as a reference. Likewise, the features selected by the Boruta algorithm in the training set were retained for the testing set. By processing the training and testing sets in this way, data leakage is avoided, and model evaluation reflects reliable predictive performance on unseen data.

2.3.3. Model development.

Two ensemble machine learning methods were used to develop nano-QSTR models: XGBoost and RF. These methods were chosen for their ability to handle complex datasets and nonlinear relationships. In addition, Ridge regression was included as a baseline model to provide a regularized linear benchmark for comparison with the ensemble approaches.

All three models were trained on the processed training dataset, and hyperparameter tuning used a grid search strategy, systematically testing multiple combinations of hyperparameters to identify the set that produced the best predictive performance. Grid search was combined with 5-fold cross-validation repeated five times, in which the training data were split into five equal parts: four parts were used to fit the model and one part to validate it. This process was repeated five times with different random splits to reduce variability in performance estimates and provide a more reliable evaluation of model accuracy. Combining grid search and repeated cross-validation ensures the selected hyperparameters are robust, reduces the risk of overfitting, and provides realistic performance estimates on unseen data [27].

XGBoost is a gradient boosting method that builds decision trees sequentially while optimizing a regularized objective function to reduce overfitting [15]. Hyperparameters, which

govern model structure and learning behavior, were tuned to optimize performance using the following grid. The number of boosting iterations (*nrounds*) was varied from 100 to 1,000 in increments of 50, resulting in $nrounds \in \{100, 150, 200, 250, \dots, 1000\}$. The learning rate (*eta*) was varied across the values 0.025, 0.05, 0.1, and 0.3, or $eta \in \{0.025, 0.05, 0.1, 0.3\}$. The maximum tree depth (*max_depth*) was varied across values from 2 to 6, or $max_depth \in \{2, 3, 4, 5, 6\}$. The remaining hyperparameters were fixed at $gamma = 0$, $subsample = 1$, $colsample_bytree = 1$, and $min_child_weight = 1$.

The range of *nrounds* was selected to allow sufficient model learning while limiting the risk of overfitting. The learning-rate values provided a range from highly stable learning at $eta = 0.025$ to faster convergence at $eta = 0.3$, with the intermediate values ($eta \in \{0.5, 0.1\}$) providing a balance between learning speed and stability. The selected *max_depth* values allowed the model to range from shallow trees ($max_depth = 2$), which may reduce overfitting but potentially underfit the data, to deeper trees ($max_depth = 6$), which are capable of capturing more complex patterns but carry a greater risk of overfitting. The intermediate values ($max_depth \in \{3, 4, 5\}$) may provide a balance between model flexibility and generalization by capturing moderate patterns while limiting overfitting. The remaining hyperparameters were fixed at standard values to constrain the tuning space and ensure stable training. Specifically, *gamma* controls the minimum loss reduction required to make a split and was fixed at 0 to allow all potential splits and simplify tuning. The *subsample* hyperparameter was set to 1 to ensure that all training observations were used in each boosting round, thereby avoiding additional randomness. The *colsample_bytree* hyperparameter was fixed at 1, meaning that all features were used when constructing each tree. The *min_child_weight* hyperparameter sets the minimum sum of instance weights required in a child node and was fixed at 1 to avoid overly restrictive splitting while allowing overfitting to be controlled through other tuned hyperparameters.

Random Forest (RF) builds multiple decision trees using bootstrap samples of the data and random subsets of features at each split [14], which reduces the effects of noise and correlated predictors while allowing the model to capture complex patterns. Hyperparameter tuning focused on the number of features considered at each split (*mtry*), using the values 2, 64, and 127. The number of trees (*ntree*) was fixed at 500, and the minimum node size (*nodesize*) at 5. The $mtry \in \{2, 64, 127\}$ values represent few, moderate, and many features, respectively, to balance randomness and model accuracy, while *ntree* and *nodesize* were held constant. Setting *ntree* to 500 ensures a sufficiently large forest for stable predictions without high computational cost. The *nodesize* parameter controls the minimum number of observations in terminal nodes and was fixed at 5, which is the default for regression tasks, to balance bias and variance.

Ridge regression extends ordinary least squares regression by adding L2 regularization to reduce multicollinearity and prevent overfitting, particularly when features are highly correlated [17, 27]. In Ridge regression, the *lambda* hyperparameter controls the strength of regularization. Smaller *lambda* values result in minimal shrinkage and allow the model to fit the data more closely, whereas larger *lambda* values apply stronger shrinkage, reducing variance but potentially increasing bias. To thoroughly evaluate the effect of regularization, *lambda* was systematically tuned over a wide range from 0.1 to 10 in increments of 0.05, that is, $lambda \in \{0.1, 0.15, 0.2, \dots, 10\}$. This extended range ensures that both weak and strong regularization levels were examined, allowing the model to identify the optimal balance between bias and variance.

All hyperparameter ranges for XGBoost, RF, and Ridge were chosen to balance model accuracy and generalization while considering computational efficiency. Model performance was evaluated using root mean squared error (*RMSE*), mean absolute error (*MAE*), and coefficient of determination (R^2). Let y_i denote observed cytotoxicity and $\hat{y}_i$ denote predicted values for n observations. The evaluation metrics are defined as follows:

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

and

$$R^2 = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2} \quad (3)$$

where $\bar{y}$ is the mean observed cell viability in the corresponding dataset partition. Performance metrics were computed for each fold and repetition during cross-validation and then averaged to obtain stable estimates of predictive performance. The optimal hyperparameters were selected based on the best mean cross-validated metrics. For external validation, the final tuned models were evaluated once using the independent testing dataset [27].

2.3.4. External validation.

External validation was carried out using an independent testing set to assess how well the models generalize to unseen nanoparticle samples. Model behavior was first examined using plots of predicted versus true cytotoxicity and prediction error distributions to visually evaluate agreement with experimental measurements and identify differences in error patterns among XGBoost, RF, and Ridge. To test for statistically significant differences in prediction errors among models, we applied the Kruskal–Wallis test, followed by Bonferroni-adjusted pairwise comparisons. The Kruskal–Wallis test is a nonparametric alternative to one-way ANOVA that compares the distributions of three or more independent groups without assuming normality [29]. The null hypothesis (H_0) states that the distribution (or median) of prediction errors is the same across all three models. The alternative hypothesis (H_1) states that at least one model differs. H_0 was rejected when the p -value was less than $\alpha = 0.05$ level of significance. If the overall test was significant, we conducted Bonferroni-adjusted pairwise comparisons to determine which specific model pairs differed [30]. For each pairwise comparison, H_0 states that the two models have equal prediction error distributions, while H_1 states that the distributions differ. The Bonferroni correction adjusts the significance level by dividing α by the number of comparisons to control the family-wise error rate. A pairwise difference was considered significant when the adjusted p -value was less than $\alpha = 0.05$. This nonparametric approach was preferred over ANOVA because prediction errors may violate normality and homogeneity of variance assumptions. These analyses provide insight into prediction stability, bias, and performance across different cytotoxicity ranges. After this qualitative assessment, standard external validation metrics were calculated to allow a direct quantitative comparison of predictive accuracy between the two nano-QSTR models. Moreover, we compared the external-validation performance metrics with those from model training by computing their differences to assess overfitting. Overfitting occurs when a trained model fails to generalize well to new data. It is indicated when performance metrics during model training are substantially higher than those obtained during external validation.

2.4. Feature importance.

Feature importance analysis was performed to improve the interpretability of the ensemble machine learning models. For XGBoost and RF, importance scores were derived from the internal structure of the tree-based models.

In the XGBoost model, feature importance was based on the gain metric, which measures how much each feature contributes to reducing prediction error when it is used to split decision trees. Features that result in larger reductions in error across the ensemble receive higher importance scores.

For the RF model, variable importance was estimated using a permutation-based approach. For each tree in the forest, the mean squared error (*MSE*) is first calculated on the out-of-bag (OOB) portion of the data. Then, the values of each feature are randomly permuted, and the *MSE* is recalculated on the same OOB data. The increase in *MSE* caused by permuting the feature is averaged across all trees and normalized by the standard error. Features that cause a larger increase in *MSE* when permuted are considered more important. If the standard error is zero for a feature, the normalization step is skipped.

Finally, for Ridge, feature importance was assessed based on the magnitude of the standardized model coefficients. Larger absolute coefficients indicate a stronger influence of the corresponding feature on cytotoxicity.

For all models, importance values were scaled relative to the most influential feature, which was assigned a value of 100. All other predictors were expressed as percentages of this maximum importance. This normalization allows direct comparison of feature importance within each model and improves the clarity of interpretation. The top-ranked features were selected to present the key factors influencing cytotoxicity prediction [14, 15, 24].

2.4. Applicability domain assessment: PCA bounding box and Hotelling's T^2

The applicability domain (AD) of the developed nano-QSTR models was assessed using a principal component analysis (PCA)-based approach, combining a bounding box method with Hotelling's T^2 statistic. PCA was performed on the preprocessed training feature matrix, and the principal components were ranked according to the proportion of explained variance. The first two components (*PC1* and *PC2*), which together captured the largest variance, were initially used to define a bounding box, where the minimum and maximum scores for each component among the training samples set the AD limits:

$$PC_{\min}(j) = \min PC_i(j), PC_{\max}(j) = \max PC_i(j) \quad (4)$$

where $PC_i(j)$ denotes the score of training sample i on principal component j . A testing sample was considered inside the AD if:

$$PC_{\min}(j) \leq PC_{\text{test}}(j) \leq PC_{\max}(j) \text{ for all retained PCs.} \quad (5)$$

While the bounding box provides a straightforward range-based check, it does not quantify the multivariate distance of a sample from the central distribution of training data. To address this, we applied Hotelling's T^2 statistic, which measures the squared multivariate distance of a sample in the PCA space, accounting for the variance along each component. For sample i , the T^2 distance is defined as: $T_i^2 = \sum_{j=1}^a \frac{t_{ij}^2}{\lambda_j}$, where t_{ij} is the score of sample i on principal component j , λ_j is the eigenvalue of that component, and a is the number of retained PCs. A critical threshold for T^2 at a chosen confidence level (commonly $\alpha = 0.95$) can be

computed based on the F -distribution: $T_{\text{crit}}^2 = \frac{a(n-1)}{n-a} F_{a,n-a,\alpha}$, where n is the number of training samples, and $F_{a,n-a,\alpha}$ is the critical F value. Samples with T^2 values exceeding this threshold are considered statistically distant from the training distribution and therefore outside the AD. By combining the PCA bounding box and Hotelling's T^2 , this approach ensures that test samples are both within the range of observed feature values and not multivariately distant from the training set. This dual assessment provides a robust applicability domain suitable for nonlinear nano-QSTR models and is consistent with established projection-based AD methodologies in QSAR/QSTR studies [31-39].

3. Results and Discussion

3.1. Data preparation for model training.

Initial cleaning removed observations with biologically implausible cytotoxicity values (cell viability < 0), these are observations 1107, 1339, 1718, 1742, 1750, 2771, and 4542, reducing the dataset slightly from 4863 to 4856 samples, indicating minimal data quality issues. The dataset was then split, resulting in 3,887 samples for training and 969 samples for the testing set. Features with more than 50% missing values in the training set ($n > 1944$), such as aspect ratio, PDI, zeta in medium, and size in medium, were excluded, reducing the number of features from 25 to 21. This step reduced the number of features from 25 to 21, as shown in Table 1. Remaining missing values were imputed using k-nearest neighbors. For categorical features, rare categories representing less than 1.5% ($n < 58$) of observations were lumped into an "Others" group. Numeric features were standardized to have a mean of 0 and variance of 1, while categorical variables were dummy encoded, expanding the feature set from 21 to 130 features. Boruta feature selection was then applied, removing three non-informative binary features (Cell Type: L929, Cell Tissue: Subcutaneous Connective Tissue, and Cell Morphology: Neuronal), resulting in a final set of 127 features. As a result, these steps improved data quality, reduced noise and sparsity, and produced a robust feature set for model development while ensuring consistent preprocessing across training and testing sets.

Table 1. Features with missing values.

No.	Features	No. of missing values	Percentage (%)
1	Aspect Ratio	3606	92.77
2	PDI	2625	67.53
3	Zeta Potential in Medium (mV)	2420	62.26
4	Particle Size in Medium (nm)	2079	53.49
5	Particle Size in Water (nm)	1445	37.18
6	Zeta Potential in Water (mV)	1387	35.68
7	Particle Diameter (nm)	968	24.9
8	Surface Charge	959	24.67
9	Number of Cells (cells/well)	711	18.29
10	Concentration ($\mu\text{g/mL}$)	141	3.63
11	Coating Functional Group	63	1.62
12	Cell Age	4	0.1

The remaining missing values were imputed using the k-nearest neighbors algorithm with $k = 5$. For categorical features, rare categories representing less than 1.5% of the observations, corresponding to fewer than 58 samples, were grouped into an "Others" category to reduce sparsity. Categorical features were then converted into dummy variables, which expanded the feature space from 21 to 130 features. Numerical features were standardized to have a mean of 0 and a variance of 1 to ensure comparability across features and to improve

model stability. Boruta feature selection was then applied, removing three non-informative binary features (Cell Type: L929, Cell Tissue: Subcutaneous Connective Tissue, and Cell Morphology: Neuronal), resulting in a final set of 127 features. As a result, these steps improved data quality, reduced noise and sparsity, and produced a robust feature set for model development, while ensuring that preprocessing was applied consistently to both training and testing sets.

3.2. Model training with hyperparameter tuning and repeated 5-fold cross-validation.

Figure 1 shows the results of repeated cross-validation (rcv) for the XGBoost model. The model performed best with a maximum tree depth (max_depth) of 6, a learning rate (eta) of 0.1, and 1,000 boosting iterations ($nrounds$). Under these settings, the model achieved an R^2_{rcv} of 81.84%, an $RMSE_{rcv}$ of 12.73%, and a MAE_{rcv} of 8.15%. This means that the model can explain about 81.84% of the variation in cytotoxicity, showing strong predictive ability for this dataset. Given that cytotoxicity values range from 0 to 234.63%, with a median of 87.44% and a mean of 77.42%, the R^2_{rcv} , $RMSE_{rcv}$, and MAE_{rcv} values are relatively small. Most predictions are therefore close to the observed values, and the model errors are minor compared to the data range. The choice of a moderately deep tree ($max_depth = 6$) allows the model to capture complex patterns without overfitting. The learning rate ($eta = 0.1$) ensures stable learning in small steps, while the high number of boosting rounds ($nrounds = 1,000$) lets the model gradually refine its predictions. These hyperparameters enable XGBoost to model the relationship between nanoparticle features and cytotoxicity effectively, producing accurate and reliable predictions for the dataset.

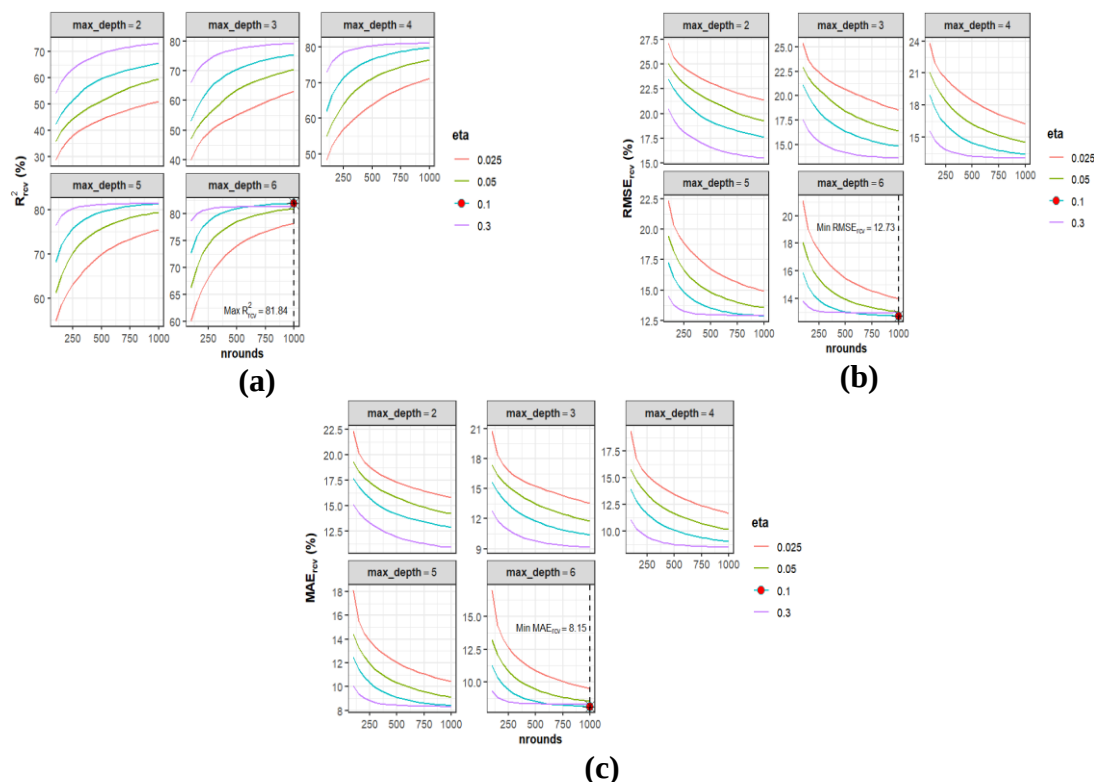


Figure 1. XGBoost performance across various hyperparameters: (a) R^2_{rcv} vs $nrounds$ by max_depth and eta ; (b) $RMSE_{rcv}$ vs $nrounds$ by max_depth and eta ; (c) MAE_{rcv} vs $nrounds$ by max_depth and eta .

Figure 2 presents the repeated cross-validated performance of the RF model across different $mtry$ values. The highest R^2_{rcv} (76.42%) and the lowest $RMSE_{rcv}$ (14.79%) were

obtained when $mtry = 64$. Although the lowest MAE_{rcv} (9.99%) occurred at $mtry = 127$, the R^2_{rcv} and $RMSE_{rcv}$ results indicate that $mtry = 64$ provides a better balance between predictive accuracy and error and was therefore selected as the optimal setting. At this value, the MAE_{rcv} was 10.01%. Using $mtry = 64$ allows the model to consider a moderate number of features at each split, helping to capture important patterns in the training data while avoiding excessive randomness.

When compared to XGBoost, the RF model shows lower R^2_{rcv} and higher $RMSE_{rc}$ and MAE_{rcv} during training, indicating a weaker fit to the training data. This suggests XGBoost better captures the complex relationships between nanoparticle features and cytotoxicity in this dataset. However, these results are based only on training performance, and external validation is still required to evaluate how well both models perform on unseen data.

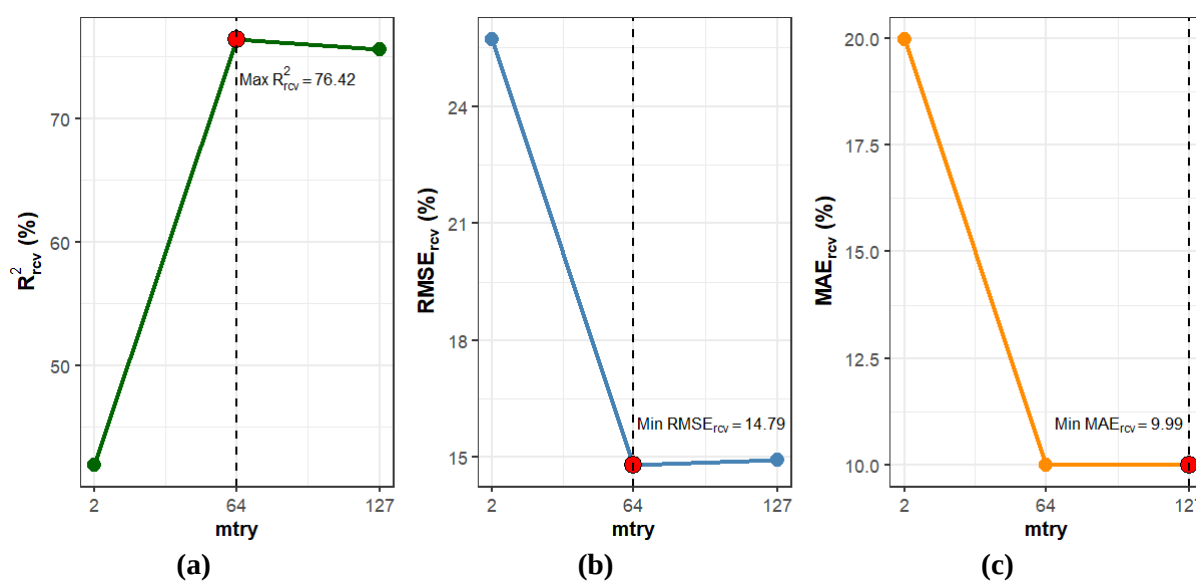


Figure 2. RF performance for different $mtry$ hyperparameter values: (a) R^2_{rcv} vs $mtry$; (b) $RMSE_{rcv}$ vs $mtry$; (c) MAE_{rcv} vs $mtry$.

Figure 3 shows the repeated 5-fold cross-validated performance of the Ridge model for different values of the regularization parameter $lambda$. The best results were obtained at $lambda = 0.41$, where the model achieved the highest R^2_{rcv} of 33.74%, the lowest $RMSE_{rcv}$ of 24.33%, and the lowest MAE_{rcv} of 18.42%. As $lambda$ increased, R^2_{rcv} gradually decreased, while $RMSE_{rcv}$ and MAE_{rcv} increased.

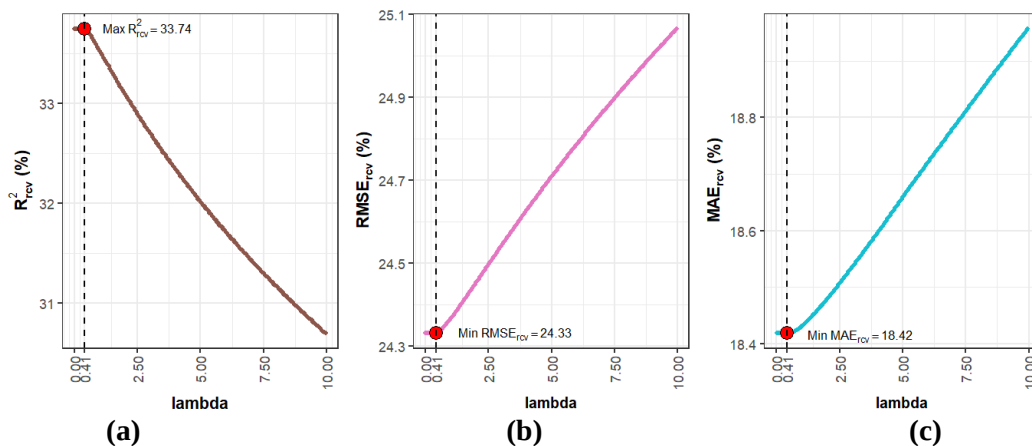


Figure 3. Ridge performance for different $lambda$ hyperparameter values (a) R^2_{rcv} vs $lambda$; (b) $RMSE_{rcv}$ vs $lambda$; (c) MAE_{rcv} vs $lambda$.

This means that using a larger *lambda* reduces the model's ability to fit the data well. Compared with XGBoost and RF, the Ridge model shows much lower R and higher $RMSE_{rcv}$, and MAE_{rcv} . XGBoost achieved an R^2_{rcv} of 81.84%, and RF achieved 76.42%. Ridge explained only about 33.74% of the variation in cytotoxicity. In addition, the prediction errors of Ridge were almost twice as large as those of XGBoost. These results show that Ridge is less accurate for this dataset.

Table 2 reports the mean and standard deviation (SD) of the performance metrics obtained from five repetitions of 5-fold cross-validation. The mean values correspond to the optimal performance observed in Figures 1–3, reflecting each model's best-tuned hyperparameters. Smaller standard deviation values indicate greater stability, as they imply that the model's performance remains consistent across the cross-validation folds. For XGBoost, the standard deviations are relatively small (R^2_{rcv} : $\pm 1.79\%$, $RMSE_{rcv}$: $\pm 0.60\%$, MAE_{rcv} : $\pm 0.29\%$), indicating consistent performance across folds. The RF model shows slightly higher variability (R^2_{rcv} : $\pm 2.46\%$, $RMSE_{rcv}$: $\pm 0.90\%$, MAE_{rcv} : $\pm 0.40\%$), but the results remain stable overall. Ridge also exhibits comparable variability (R^2_{rcv} : $\pm 2.53\%$, $RMSE_{rcv}$: $\pm 0.61\%$, MAE_{rcv} : $\pm 0.45\%$). The relatively small standard deviations across all models suggest that the repeated cross-validation procedure produced stable, reliable performance estimates. Therefore, the differences observed between XGBoost, RF, and Ridge are consistent across folds and reflect true differences in predictive performance rather than random variation in the data splits.

Table 2. Training performance of the optimal nano-QSTR models with uncertainty (Mean $\pm$ SD).

Model	$R^2_{rcv} \pm SD$ (%)	$RMSE_{rcv} \pm SD$ (%)	$MAE_{rcv} \pm SD$ (%)
XGBoost	81.84 $\pm$ 1.79	12.73 $\pm$ 0.60	8.15 $\pm$ 0.29
RF	76.42 $\pm$ 2.46	14.79 $\pm$ 0.90	10.01 $\pm$ 0.40
Ridge	33.74 $\pm$ 2.53	24.33 $\pm$ 0.61	18.42 $\pm$ 0.45

3.3. Model performance and prediction error analysis on the testing set.

Prediction errors from the testing set show how well the models estimate nanoparticle cytotoxicity. As shown in Figure 4a, the median prediction errors for XGBoost and RF are close to 0 relative to the dashed reference line, indicating that neither model systematically over- nor underestimates cytotoxicity. Ridge regression, included as a simpler baseline model, shows a median prediction error slightly below 0, suggesting a mild tendency to underestimate cytotoxicity. Moreover, its error distribution is considerably wider, with more extreme deviations, suggesting less precise predictions across the tested nanoparticle samples. Differences in the spread of prediction errors are notable. XGBoost exhibits the narrowest error distribution, implying its stability. RF has a wider range of errors, including several beyond $\pm 50\%$, indicating variable performance for certain feature combinations. Ridge regression shows the largest spread and the most extreme outliers, reflecting the limitations of a linear approach in modeling the complex, nonlinear relationships in nanoparticle cytotoxicity. The agreement between true and predicted cytotoxicity/cell viability values further illustrates these differences, as shown in Figure 4b. Predictions from XGBoost and RF generally follow the diagonal reference line, showing good overall agreement with experimental measurements. At low cytotoxicity levels (cell viability $> 100\%$), the spread of predictions increases for all models, with Ridge exhibiting the greatest scatter, RF moderate, and XGBoost the least. In the mid-range of cytotoxicity ($50\% \leq$ cell viability $\leq 100\%$), all models track the observed values reasonably well, although Ridge shows slightly larger deviations. At high cytotoxicity levels (cell viability $< 50\%$), XGBoost predictions remain the tightest around the true values, followed

by RF, while Ridge predictions show greater variability. Statistical testing confirms these observations. A Kruskal–Wallis test comparing prediction errors across models revealed a significant overall difference ($p\text{-value} = 0.001 < 0.05$). Bonferroni-adjusted post-hoc comparisons showed that Ridge errors were significantly larger than those of RF ($\text{adjusted } p\text{-value}=0.012 < 0.05$) and XGBoost ($\text{adjusted } p\text{-value}= 0.001 < 0.05$). In contrast, the difference between RF and XGBoost errors was not statistically significant ($\text{adjusted } p\text{-value}= 0.487 > 0.05$), indicating that although XGBoost visually appears slightly closer to the true values, its predictive performance is statistically comparable to that of RF. Together, these results demonstrate that both XGBoost and RF outperform Ridge in predicting nanoparticle cytotoxicity.

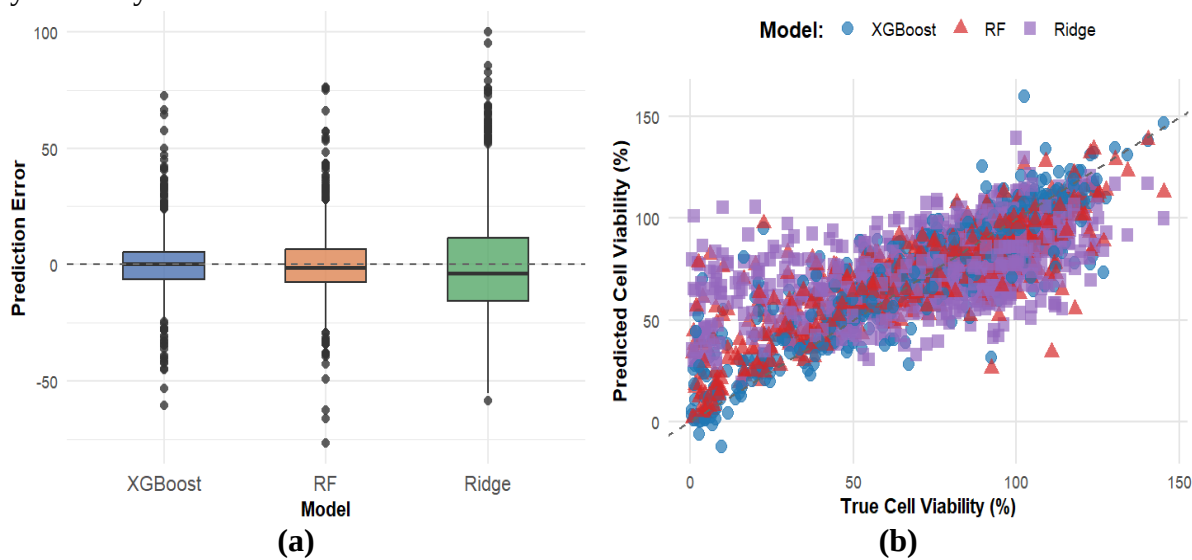


Figure 4. Comparison of XGBoost, RF, and Ridge model predictions on the testing set: (a) Distribution of prediction errors; (b) True vs predicted cell viability.

The external validation metrics presented in Table 3 support the findings observed in Figure 4. XGBoost shows slightly stronger predictive performance, with an R^2_{test} of 80.25%, $RMSE_{test}$ of 13.09%, and MAE_{test} of 9.01% compared with RF, which achieved an R^2_{test} of 75.59%, $RMSE_{test}$ of 14.87%, and MAE_{test} of 10.32%. These results indicate that XGBoost captures a greater proportion of the variation in nanoparticle cytotoxicity and yields lower average prediction errors on unseen samples, consistent with its narrower error distribution and more stable behavior across different cytotoxicity levels. In contrast, Ridge regression shows substantially lower predictive performance, with an R^2_{test} of 33.02%, $RMSE_{test}$ of 24.13%, and MAE_{test} of 18.27%. The relatively large errors and reduced explained variance suggest that the linear structure assumed by Ridge regression may be insufficient to capture the complex nonlinear relationships between nanoparticle descriptors and cytotoxicity, which tree-based ensemble methods such as XGBoost and RF model more effectively.

Table 3. Testing performance of the optimal nano-QSTR models.

Model	R^2_{test} (%)	$RMSE_{test}$ (%)	MAE_{test} (%)
XGBoost	80.25	13.09	9.01
RF	75.59	14.87	10.32
Ridge	33.02	24.13	18.27

Recalling the repeated cross-validation results during model training, XGBoost attained a R^2_{rcv} of 81.84%, $RMSE_{rcv}$ of 12.73%, and MAE_{rcv} of 8.15%, with only a small decrease observed during external validation. Similarly, RF achieved cross-validated performance of

R^2_{rcv} = 76.42%, $RMSE_{rcv}$ = 14.79%, and MAE_{rcv} = 10.01%, with testing results closely matching these values. Conversely, Ridge regression showed substantially lower cross-validated performance, with R^2_{rcv} = 33.74%, $RMSE_{rcv}$ = 24.33%, and MAE_{rcv} = 18.42%, indicating weaker predictive capability during training compared with the ensemble-based models. Table 4 quantifies the differences (Δ) between training and testing performance: for XGBoost R^2 decreased by 1.59%, $RMSE$ increased by 0.36%, and MAE increased by 0.86%; for RF, the corresponding differences were 0.83%, 0.08%, and 0.33%; while for Ridge regression, R^2 decreased by 0.72%, $RMSE$ increased by 0.20%, and MAE increased by 0.15%. These small discrepancies between cross-validated and external validation metrics indicate no overfitting and demonstrate that the models generalize well to unseen data.

Table 4. Differences in training and testing performance of the optimal nano-QSTR models.

Model	$\Delta R^2 = R^2_{rcv} - R^2_{test}$ (%)	$\Delta RMSE = RMSE_{rcv} - RMSE_{test}$ (%)	$\Delta MAE = MAE_{rcv} - MAE_{test}$ (%)
XGBoost	1.59	-0.36	-0.86
RF	0.83	-0.08	-0.33
Ridge	0.72	-0.2	-0.15

3.4. Feature importance analysis.

Figure 5 shows the top 10 most important features among the 127 descriptors for both the XGBoost and RF models. For both models, concentration is the most influential feature, followed by particle size in water, zeta potential in water, particle diameter, exposure time, number of cells, and synthesis method (commercial). These features consistently rank highest, indicating their strong predictive power for cytotoxicity.

Concentration is expected to be highly important because cell responses to nanoparticles are dose-dependent. Higher concentrations increase particle–cell interactions, leading to stress and damage that reduce viability [2, 3]. Exposure time also affects toxicity, as longer exposure allows more particles to enter and accumulate in cells [4].

Smaller particles, reflected by particle size and diameter, have higher surface area, are more easily taken up by cells, and tend to be more reactive and potentially harmful than larger particles [40]. Zeta potential reflects surface charge, which affects particle stability in suspension and interactions with cell membranes, influencing cellular uptake and toxicity [41]. The number of cells modifies the effective dose per cell, which can change observed viability even at the same concentration [42]. Synthesis method (commercial) also matters, as production processes can alter surface quality, impurities, and coatings, affecting cell responses [43].

Differences between the two models appear mainly in the lower-ranked features (ranks 8–10). XGBoost emphasizes cell tissue (bone marrow), nanoparticle shape (others), and nanoparticle type (inorganic), showing greater sensitivity to tissue context and particle structure. Bone marrow is biologically sensitive because it contains fast-dividing cells vulnerable to nanoparticles that enter circulation and affect blood cell production [3, 2]. Non-spherical particles interact differently with membranes, affecting uptake and toxicity [44], and inorganic particles can generate reactive oxygen species or release ions, driving toxicity [2].

RF, on the other hand, gives more weight to synthesis method (others), material (Au), and coat functional group (none), reflecting stronger sensitivity to particle composition and surface properties. Synthesis method influences size, shape, defects, and purity, affecting stability and biological interactions. Gold nanoparticles are chemically stable, but their effects depend on size and coating, with small or uncoated particles showing higher uptake and toxicity

[45, 46]. Uncoated surfaces promote protein binding, forming a protein corona that affects cellular recognition and uptake, whereas coatings can reduce nonspecific interactions and improve biocompatibility [47]. These results show that both models capture biologically meaningful relationships between dose, particle properties, exposure, and cell response, supporting the reliability and interpretability of the ensemble Nano-QSTR models.

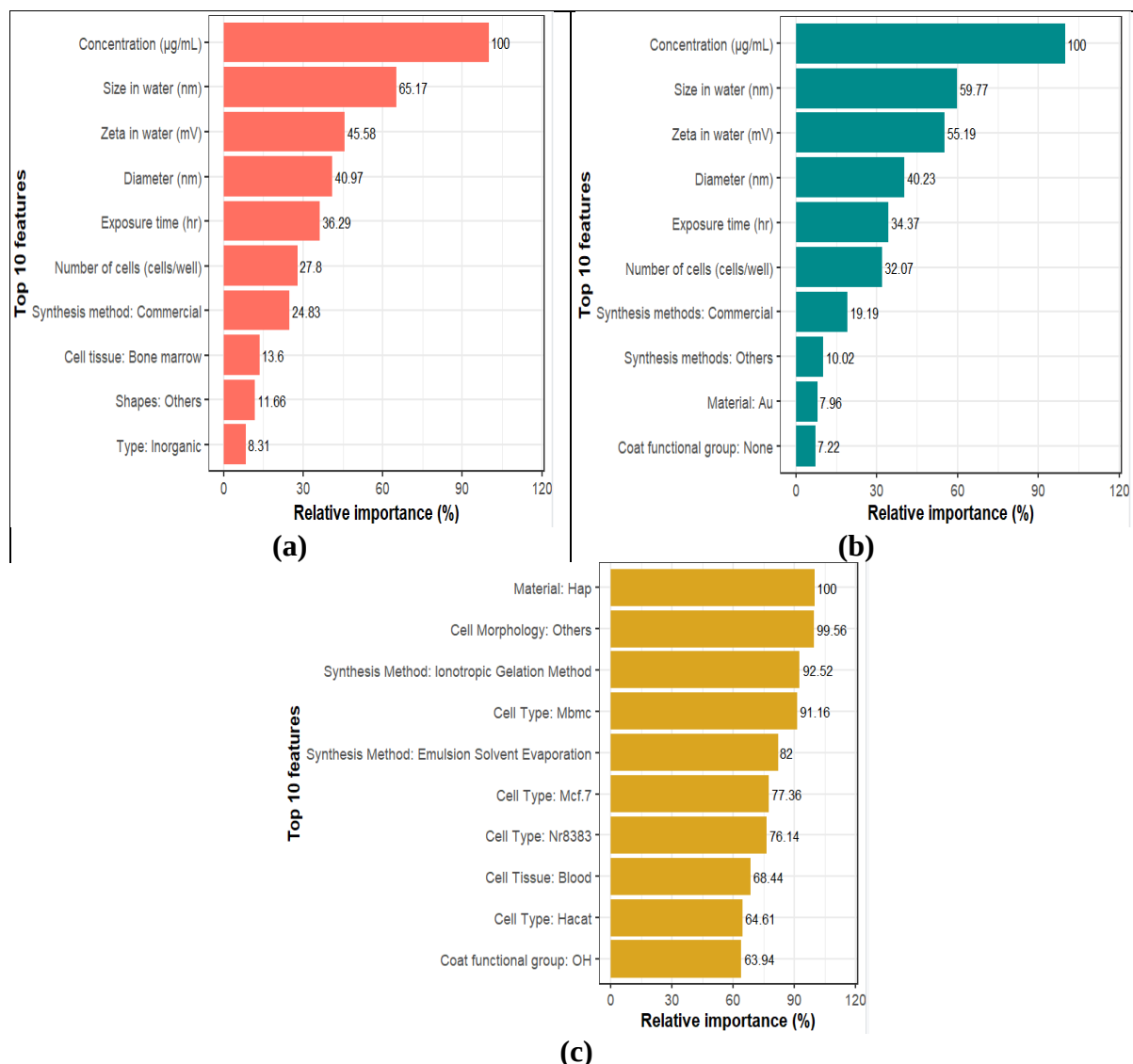


Figure 5. Top 10 most important features: (a) XGBoost; (b) RF; (c) Ridge.

In contrast, the Ridge regression model (Figure 4c), which represents a simpler linear benchmark, identified a different set of influential features. The top features include material (Hap), cell morphology (others), synthesis methods such as ionotropic gelation and emulsion solvent evaporation, several specific cell types (e.g., Mbmcc, MCF.7, NR8383), and coat functional group (OH). These results indicate that the linear Ridge model primarily captures associations related to experimental cell context rather than the physicochemical dose-related features emphasized by the tree-based models. This difference may suggest that Ridge is less capable of modeling complex nonlinear interactions among dose, particle properties, and biological conditions when cytotoxicity is expressed as a continuous numeric response (e.g., percent cell viability) rather than a binary toxicity outcome. Consequently, while Ridge provides a useful baseline regression benchmark, the ensemble models capture broader and

more biologically consistent relationships, which explains their superior predictive performance.

3.5. Applicability domain assessment.

Figure 6 presents the PCA plot of the training and testing sets based on the features used in the Nano-QSTR models. The x-axis ($PC1$) and y-axis ($PC2$) represent the first two principal components, which capture the largest sources of variation in the dataset. We performed principal component analysis on the preprocessed training data only, and projected the test samples onto the resulting PCA space. Training samples are shown in dark blue, while testing samples are shown in yellow.

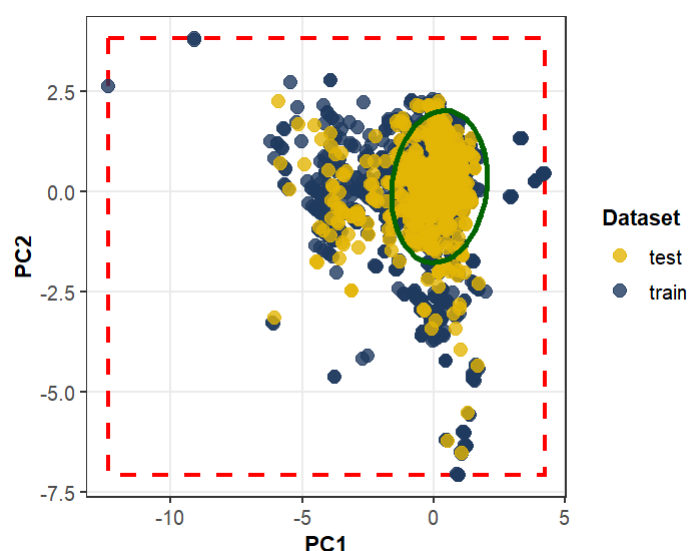


Figure 6. Applicability domain defined by PCA bounding box and Hotelling's T^2 .

The applicability domain (AD) was defined using quantitative criteria derived from the PCA representation of the training data. First, we constructed a PCA bounding box using the minimum and maximum principal component scores of the training set. The red dashed rectangle in Figure 5 represents this bounding box and defines the limits of the feature space covered by the training samples. Based on the training data, the minimum and maximum scores were -12.39 and 4.23 for $PC1$, and -7.07 and 3.82 for $PC2$. A testing sample is considered to fall within this region if its projected scores satisfy $-12.39 \leq PC1_{\text{test}} \leq 4.23$ and $-7.07 \leq PC2_{\text{test}} \leq 3.82$. In addition, Hotelling's T^2 was used as a multivariate distance measure to identify observations that deviate substantially from the central distribution of the training data. The green ellipse in Figure 5 represents the 95% confidence region defined by Hotelling's T^2 with a critical value of $T^2_{\text{crit}} = 6$. Because this statistic measures distance from the multivariate center, the resulting ellipse is concentrated around the densest region of the training samples and is therefore more restrictive than the PCA bounding box. Most testing samples fall within the PCA bounding box, indicating that they lie within the feature space covered by the training data. However, some observations appear outside the Hotelling's T^2 ellipse, suggesting that they are relatively distant from the central training distribution even though they remain within the overall domain. Predictions for samples near the domain boundaries may therefore be associated with greater uncertainty.

3.6. Compliance with the OECD (Q)SAR validation principles.

The Nano-QSTR models were developed following the five validation principles of the Organization for Economic Co-operation and Development (OECD). In line with principle 1 (a defined endpoint), the endpoint was clearly defined as *in vitro* cytotoxicity, expressed as percentage cell viability, and values below 0 were removed to ensure consistency. In accordance with principle 2 (an unambiguous algorithm), the modeling methods- XGBoost, Random Forest, and Ridge regression- are clearly described, including data preprocessing, hyperparameter ranges, repeated 5-fold cross-validation, and the selected optimal settings to ensure transparency and reproducibility. For principle 3 (a defined applicability domain), a PCA-based bounding box built from the preprocessed training data was used to verify whether testing samples were within the model's range. Most testing samples were inside this domain, supporting reliable predictions. Following principle 4 (goodness-of-fit, robustness, and predictivity), model performance was evaluated using repeated cross-validation and an independent testing set with R^2 , $RMSE$, and MAE . The small differences between training and testing results show good generalization without evidence of overfitting. Finally, addressing principle 5 (mechanistic interpretation), feature importance analysis based on ensemble models identified meaningful features such as concentration, particle size, zeta potential, and exposure time, which agree with known mechanisms of nanoparticle toxicity.

3.7. Study limitations and potential sources of bias.

Despite the strong predictive performance of the Nano-QSTR models, some limitations should be noted. The dataset includes 4856 observations from multiple studies covering physicochemical properties, exposure-related features, biological features, and synthesis and coating information. While these factors were included as model features, some unexplained variability between studies may remain, such as minor differences in reagents, instruments, or laboratory procedures. Some features, including particle size in medium and zeta potential in medium, were excluded due to a high proportion of missing values. Although this improves model robustness, it may reduce mechanistic completeness. Rare categorical levels were grouped into an "Others" category to reduce overfitting, which may mask small but meaningful effects. In addition, some features may be correlated, particularly among features describing nanoparticle size or surface properties, which may influence how feature importance is distributed among related features in the machine learning models. Moreover, the applicability domain analysis also shows that a few testing samples lie near the boundary of the training space, making predictions for these samples less certain.

4. Conclusions

Nano-QSTR regression models were developed to predict nanoparticle cytotoxicity as continuous cell viability (%), integrating physicochemical properties, exposure conditions, biological descriptors, and synthesis/coating information. Extreme Gradient Boosting (XGBoost) and Random Forest (RF) were implemented as ensemble models and compared with Ridge regression as a baseline. During repeated cross-validation, XGBoost achieved $R^2_{rcv}=81.84\% \pm 1.79\%$, $RMSE_{rcv}=12.73\% \pm 0.60\%$, $MAE_{rcv}= 8.15\% \pm 0.29\%$, while RF achieved $R^2_{rcv}=76.42\% \pm 2.46\%$, $RMSE_{rcv}=14.79\% \pm 0.90\%$, $MAE_{rcv}= 10.01\% \pm 0.40\%$. Ridge regression, used as a baseline, showed substantially lower performance with $R^2_{rcv}=33.74\% \pm 2.53\%$, $RMSE_{rcv}=24.33\% \pm 0.61\%$, $MAE_{rcv}= 18.42\% \pm 0.45\%$, confirming that the ensemble

models strongly outperform the simpler baseline. External validation results were consistent, with XGBoost achieving $R^2_{test} = 80.25\%$, $RMSE_{test} = 13.09\%$, $MAE_{test} = 9.01\%$, RF achieving $R^2_{test} = 75.59\%$, $RMSE_{test} = 14.87\%$, $MAE_{test} = 10.32\%$, and Ridge reaching $R^2_{test} = 33.02\%$, $RMSE_{test} = 24.13\%$, $MAE_{test} = 18.27\%$, further demonstrating the superior predictive performance of the ensemble models. The small differences between training and testing performance, for example, XGBoost: $\Delta R^2 = 1.59\%$, $\Delta RMSE = -0.36\%$, $\Delta MAE = -0.86\%$, indicate that the models did not overfit and they generalize well to unseen data.

Feature importance analysis, based on XGBoost and RF, determined nanoparticle concentration, particle size and diameter, zeta potential, exposure time, number of cells, and synthesis method as the most influential features, reflecting the combined effects of physicochemical, biological, and exposure-related characteristics. Applicability domain analysis showed that most testing samples fall within the training feature space, supporting the robustness of predictions. Despite the strengths of the models built, some limitations remain. The dataset was compiled from multiple studies, and residual variability from experimental differences may remain. Certain features, such as particle size and zeta potential in medium, were excluded due to high missingness, and a few samples near the boundaries of the applicability domain may have less certain predictions. Future work could incorporate additional surface chemistry and biological features, such as cellular uptake, reactive oxygen species generation, and protein corona formation, and include multiple toxicity endpoints beyond cytotoxicity. The proposed nano-QSTR models provide reliable, quantitative predictions of nanoparticle cytotoxicity, with XGBoost delivering the highest accuracy, and can complement experimental testing in early-stage nanomaterial safety evaluation and design.

Supplementary materials

Supplementary materials are available for download [here](#).

Author Contributions

Conceptualization, A.V. and C.P.; methodology, A.V. and C.P.; software, A.V. and C.P.; validation, A.V. and C.P.; formal analysis, A.V. and C.P.; investigation, A.V. and C.P.; resources, R.C.; data curation, J.R.B.; writing—original draft preparation, A.V. and C.P.; writing—review and editing, I.E. and P.A.; visualization, F.B.; supervision, I.E. and R.C.; project administration, M.A.S; funding acquisition, R.C. All authors have read and agreed to the published version of the manuscript.

Institutional Review Board Statement

Not applicable.

Informed Consent Statement

Not applicable.

Data Availability Statement

The data presented in this study are openly available at <https://bit.ly/nanodata>.

Funding

This research was funded by the Department of Science and Technology–Philippine Council for Industry, Energy, and Emerging Technology Research and Development (DOST-PCIEERD), grant number [10185]. The funders had no role in the design of the study; in the collection, analysis, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

Acknowledgments

This work was supported by the Department of Science and Technology–Philippine Council for Industry, Energy, and Emerging Technology Research and Development (DOST-PCIEERD) through Grant No. 10185.

Conflicts of Interest

The authors declare no conflict of interest.

Artificial Intelligence Disclosure

Artificial intelligence-based tools were used to assist with grammar checking and readability improvement during manuscript preparation. All scientific content, analysis, and conclusions were developed by the authors.

Abbreviations

The following abbreviations are used in this manuscript:

Abbreviaton	Definition
AD	Applicability Domain
kNN	k-Nearest Neighbors
MAE	Mean Absolute Error
MSE	Mean Squared Error
OOB	Out of Bag
PC	Principal Component
PCA	Principal Component Analysis
PDI	Polydispersity Index
QSAR	Quantitative Structure–Activity Relationship
QSTR	Quantitative Structure–Toxicity Relationship
RF	Random Forest
RMSE	Root Mean Squared Error
SD	Standard Deviation
XGBoost	Extreme Gradient Boosting

References

1. Roco, M. C. The long view of nanotechnology development: The National Nanotechnology Initiative at 10 years. *J. Nanopart. Res.* **2011**, *13*, 427–445, <https://doi.org/10.1007/s11051-010-0192-z>.
2. Nel, A.; Xia, T.; Mädler, L.; Li, N. Toxic potential of materials at the nanolevel. *Science* **2006**, *311*, 622–627, <https://doi.org/10.1126/science.1114397>
3. Oberdörster, G.; Oberdörster, E.; Oberdörster, J. Nanotoxicology: An emerging discipline evolving from studies of ultrafine particles. *Environ. Health Perspect.* **2005**, *113*, 823–839, <https://doi.org/10.1289/ehp.7339>.
4. Foldbjerg, R.; Jiang, X.; Miclăuș, T.; Chen, C.; Autrup, H.; Beer, C. Silver nanoparticles—Wolves in sheep’s clothing? *Toxicol. Res.* **2015**, *4*, 563–575, <https://doi.org/10.1039/C4TX00110A>.

5. Puzyn, T.; Rasulev, B.; Gajewicz, A.; Hu, X.; Dasari, T. P.; Michalkova, A.; Hwang, H.-M.; Toropov, A.; Leszczynska, D.; Leszczynski, J. Using nano-QSAR to predict the cytotoxicity of metal oxide nanoparticles. *Nat. Nanotechnol.* **2011**, *6*, 175–178, <https://doi.org/10.1038/nnano.2011.10>.
6. Li, J.; Wang, C.; Yue, L.; Chen, F.; Cao, X.; Wang, Z. Nano-QSAR modeling for predicting the cytotoxicity of metallic and metal oxide nanoparticles: A review. *Ecotoxicol. Environ. Saf.* **2022**, *243*, 113955, <https://doi.org/10.1016/j.ecoenv.2022.113955>.
7. Pan, Y.; Li, T.; Cheng, J.; Telesca, D.; Zink, J. I.; Jiang, J. Nano-QSAR modeling for predicting the cytotoxicity of metal oxide nanoparticles using novel descriptors. *RSC Adv.* **2016**, *6*, 25766–25775, <https://doi.org/10.1039/C6RA01298A>.
8. Qi, R.; Pan, Y.; Cao, J.; Yuan, B.; Wang, Y.; Jiang, J. Toward comprehension of the cytotoxicity of heterogeneous TiO₂-based engineered nanoparticles: A nano-QSAR approach. *Environ. Sci.: Nano* **2021**, *8*, 927–936, <https://doi.org/10.1039/D0EN01266A>.
9. Sang, L.; Wang, Y.; Zong, C.; Wang, P.; Zhang, H.; Guo, D.; Yuan, B.; Pan, Y. Machine learning for evaluating the cytotoxicity of mixtures of nano-TiO₂ and heavy metals. *Molecules* **2022**, *27*, 6125, <https://doi.org/10.3390/molecules27186125>.
10. Meneses, J.; González Durruthy, M.; Fernandez de Gortari, E.; Toropova, A. P.; Toropov, A. A.; Alfaro Moreno, E. A nano-QSTR model to predict nano cytotoxicity: An approach using human lung cells data. *Part. Fibre Toxicol.* **2023**, *20*, 21, <https://doi.org/10.1186/s12989-023-00530-0>.
11. Park, J.; Rashid, S.; Copsey, H.; Tran, L.; Zabeo, A.; Hristozov, D.; Gakis, G. P.; Charitidis, C.; Yoon, S.; Shin, H. K. NanoToxRadar: A multitarget nano-QSAR model for predicting the cytotoxicity values of multicomponent nanoparticles. *ACS Nanoscience Au* **2025**, *5*, 344–352, <https://doi.org/10.1021/acsnanoscienceau.5c00035>.
12. Ahmadi, M.; Ayyoubzadeh, S. M.; Ghorbani-Bidkorpeh, F. Toxicity prediction of nanoparticles using machine learning approaches. *Toxicology* **2024**, *501*, 153697. <https://doi.org/10.1016/j.tox.2023.153697>.
13. Valeriano, A.; Bondaug, F.; Ebarido, I.; Almonte, P.; Sabugaa, M.A.; Bagnol, J.R.; Latayada, M.J.; Macalalag, J.M.; Paradero, B.D.; Mayes, M.; Balanay, M.; Alguno, A.; Capangpangan, R. Predicting cytotoxicity of engineered nanoparticles using regularized regression models: An in silico approach. *SAR QSAR Environ. Res.* **2023**, *34*, 591–604, <https://doi.org/10.1080/1062936X.2023.2242785>.
14. Breiman, L. Random forests. *Mach. Learn.* **2001**, *45*, 5–32, <https://doi.org/10.1023/A:1010933404324>.
15. Chen, T.; Guestrin, C. XGBoost: A scalable tree boosting system. In Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discov. Data Min., San Francisco, CA, USA, **2016**; 785–794, <https://doi.org/10.1145/2939672.2939785>.
16. Bentéjac, C.; Csörgő, A.; Martínez-Muñoz, G. A comparative analysis of gradient boosting algorithms. *Artif. Intell. Rev.* **2021**, *54*, 1937–1967, <https://doi.org/10.1007/s10462-020-09896-5>.
17. Hoerl, A.E.; Kennard, R.W. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics* **1970**, *12*, 55–67, <https://doi.org/10.1080/00401706.1970.10488634>.
18. OECD. Guidance document on the validation of (quantitative) structure–activity relationship [(Q)SAR] models (OECD Series on Testing and Assessment No. 69). OECD Publishing, **2014**, <https://doi.org/10.1787/9789264085442-en>.
19. Gul, G.; Yildirim, R.; Ileri Ercan, N. Cytotoxicity analysis of nanoparticles by association rule mining. *Environ. Sci.: Nano* **2021**, *8*, 937–949, <https://doi.org/10.1039/D0EN01240H>.
20. R Core Team. R: A language and environment for statistical computing [Software], Version 4.3.1, R Foundation for Statistical Computing: Vienna, Austria, **2023**, <https://www.R-project.org>.
21. Wickham, H. ggplot2: Elegant Graphics for Data Analysis [Software], Springer-Verlag: New York, NY, USA, **2016**, <https://ggplot2.tidyverse.org>.
22. Kowarik, A.; Templ, M. Imputation with the R Package VIM. *J. Stat. Softw.* **2016**, *74*, 1–16, <https://doi.org/10.18637/jss.v074.i07>.
23. Kurs, M.B.; Rudnicki, W.R. Feature selection with the Boruta package. *J. Stat. Softw.* **2010**, *36*, 1–13, <https://doi.org/10.18637/jss.v036.i11>.
24. Kuhn, M. Building predictive models in R using the caret package. *J. Stat. Softw.* **2008**, *28*, 1–26, <https://doi.org/10.18637/jss.v028.i05>.
25. Kuhn, M.; Wickham, H.; Hvitfeldt, E. recipes: Preprocessing and Feature Engineering Steps for Modeling [Software], R package version 1.3.1, **2025**, <https://CRAN.R-project.org/package=recipes>.

26. Gholamy, A.; Kreinovich, V.; Kosheleva, O. Why 70/30 or 80/20 relation between training and testing sets: A pedagogical explanation. *Int. J. Intell. Technol. Appl. Stat.* **2018**, *11*, 105–110, [https://doi.org/10.6148/IJITAS.201806_11\(2\).0003](https://doi.org/10.6148/IJITAS.201806_11(2).0003).
27. Boehmke, B.; Greenwell, B. M. *Hands-on machine learning with R*; CRC Press: Boca Raton, FL, USA, **2019**, <https://doi.org/10.1201/9780367816377>.
28. Troyanskaya, O.; Cantor, M.; Sherlock, G.; Brown, P.; Hastie, T.; Tibshirani, R.; Botstein, D.; Altman, R.B. Missing value estimation methods for DNA microarrays. *Bioinformatics* **2001**, *17*, 520–525, <https://doi.org/10.1093/bioinformatics/17.6.520>.
29. Kruskal, W.H.; Wallis, W.A. Use of ranks in one-criterion variance analysis. *J. Am. Stat. Assoc.* **1952**, *47*, 583–621, <https://doi.org/10.1080/01621459.1952.10483441>.
30. Dunn, O.J. Multiple comparisons using rank sums. *Technometrics* **1964**, *6*, 241–252, <https://doi.org/10.1080/00401706.1964.10490181>.
31. Lacasa, L.; Pardo, A.; Arbelo, P.; Sánchez-Domínguez, M.; Bascones, N.; Yeste, P.; Martínez-Cava, A.; Rubio, G.; Gómez, I.; Valero, E.; de Vicente, J. Towards certification: A complete statistical validation pipeline for supervised learning in industry. *Expert Syst. Appl.* **2025**, *277*, 127169, <https://doi.org/10.1016/j.eswa.2025.127169>.
32. Sahigara, F.; Mansouri, K.; Ballabio, D.; Mauri, A.; Consonni, V.; Todeschini, R. Comparison of different approaches to define the applicability domain of QSAR models. *Molecules* **2012**, *17*, 4791–4810, <https://doi.org/10.3390/molecules17054791>.
33. Roy, K.; Kar, S.; Das, R.N. *A Primer on QSAR/QSPR Modeling: Fundamental Concepts*; Springer: Cham, Switzerland, **2015**, <https://doi.org/10.1007/978-3-319-17281-1>.
34. Nithya, N.; Vijayachitra, S. Fault detection and analysis using statistical data for continuous stirred tank reactor (CSTR). *Int. J. ChemTech Res.* **2018**, *11*, 266–274, <https://doi.org/10.20902/IJCTR.2018.110232>.
35. Jaworska, J.; Nikolova-Jeliazkova, N.; Aldenberg, T. QSAR applicability domain estimation by projection of the training set in descriptor space: A review. *Altern. Lab. Anim.* **2005**, *33*, 445–459, <https://doi.org/10.1177/026119290503300508>.
36. Eriksson, L.; Jaworska, J.; Worth, A.P.; Cronin, M.T.; McDowell, R.M.; Gramatica, P. Methods for reliability and uncertainty assessment and for applicability evaluations of classification-and regression-based QSARs. *Environmental health perspectives* **2003**, *111*, 1361, <https://doi.org/10.1289/ehp.5758>.
37. Netzeva, T.I.; Worth, A.P.; Aldenberg, T.; Benigni, R.; Cronin, M.T.; Gramatica, P.; Jaworska, J.S.; Kahn, S.; Klopman, G.; Marchant, C.A. Current status of methods for defining the applicability domain of (quantitative) structure-activity relationships: The report and recommendations of ecvam workshop 52. *Alternatives to Laboratory Animals* **2005**, *33*, 155–173, <https://doi.org/10.1177/026119290503300209>.
38. Phanus-umporn, C.; Shoombuatong, W.; Prachayasittikul, V.; Anuwongcharoen, N.; Nantasenamat, C. Privileged substructures for anti-sickling activity via cheminformatic analysis. *RSC Adv.* **2018**, *8*, 5920–5935, <https://doi.org/10.1039/C7RA12079F>.
39. Tropsha, A. Best practices for QSAR model development, validation, and exploitation. *Mol. Inf.* **2010**, *29*, 476–488, <https://doi.org/10.1002/minf.201000061>.
40. Huang, Y.-W.; Cambre, M.; Lee, H.-J. The toxicity of nanoparticles depends on multiple molecular and physicochemical mechanisms. *Int. J. Mol. Sci.* **2017**, *18*, 2702, <https://doi.org/10.3390/ijms18122702>.
41. Midekessa, G.; Godakumara, K.; Ord, J.; Viil, J.; Lättেকivi, F.; Dissanayake, K.; Kopanchuk, S.; Rinken, A.; Andronowska, A.; Bhattacharjee, S.; Rinken, T.; Fazeli, A. Zeta potential of extracellular vesicles: toward understanding the attributes that determine colloidal stability. *ACS Omega* **2020**, *5*, 16701–16710, <https://doi.org/10.1021/acsomega.0c01582>.
42. Drasler, B.; Sayre, P.; Steinhäuser, K. G.; Petri-Fink, A.; Rothen-Rutishauser, B. In vitro approaches to assess the hazard of nanomaterials. *NanoImpact* **2017**, *8*, 99–116, <https://doi.org/10.1016/j.impact.2017.08.002>.
43. Warheit, D.B.; Sayes, C.M.; Reed, K.L.; Swain, K.A. Health effects related to nanoparticle exposures: Environmental, health and safety considerations for assessing hazards and risks. In *Pharmaceutical Sciences Encyclopedia*; Wiley: New York, NY, USA, **2007**, <https://doi.org/10.1016/j.pharmthera.2008.07.001>.
44. Champion, J.A.; Mitragotri, S. Shape-induced inhibition of phagocytosis of polymer particles. *Proc. Natl. Acad. Sci. U.S.A.* **2009**, *26*, 244–249, <https://doi.org/10.1073/pnas.0600555103>.
45. Alkilany, A.M.; Murphy, C.J. Toxicity and cellular uptake of gold nanoparticles: What we have learned so far? *J. Nanopart. Res.* **2010**, *12*, 2313–2333, <https://doi.org/10.1007/s11051-010-9911-8>.
46. Connor, E.E.; Mwamuka, J.; Gole, A.; Murphy, C. J.; Wyatt, M. D. Gold nanoparticles are taken up by human cells but do not cause acute cytotoxicity. *Small* **2005**, *1*, 325–327, <https://doi.org/10.1002/sml.200400093>.

47. Simon, J.; Müller, L.K.; Kokkinopoulou, M.; Lieberwirth, I.; Morsbach, S.; Landfester, K.; Mailänder, V. Exploiting the biomolecular corona: pre coating of nanoparticles enables controlled cellular interactions. *Nanoscale* **2018**, *10*, 10731–10739, <https://doi.org/10.1039/C8NR03331E>.

Publisher's Note & Disclaimer

The statements, opinions, and data presented in this publication are solely those of the individual author(s) and contributor(s) and do not necessarily reflect the views of the publisher and/or the editor(s). The publisher and/or the editor(s) disclaim any responsibility for the accuracy, completeness, or reliability of the content. Neither the publisher nor the editor(s) assume any legal liability for any errors, omissions, or consequences arising from the use of the information presented in this publication. Furthermore, the publisher and/or the editor(s) disclaim any liability for any injury, damage, or loss to persons or property that may result from the use of any ideas, methods, instructions, or products mentioned in the content. Readers are encouraged to independently verify any information before relying on it, and the publisher assumes no responsibility for any consequences arising from the use of materials contained in this publication.