

UNIVERSIDADE DE BRASÍLIA

INSTITUTO DE CIÊNCIAS BIOLÓGICAS
PROGRAMA DE PÓS-GRADUAÇÃO EM NANOCIÊNCIA E NANOBIOTECNOLOGIA

RODRIGO DA FONSECA SILVEIRA

Development of software for predictive modeling of polymeric nanoparticles of pharmaceutical interest

Brasília

2026

UNIVERSIDADE DE BRASÍLIA

INSTITUTO DE CIÊNCIAS BIOLÓGICAS
PROGRAMA DE PÓS-GRADUAÇÃO EM NANOCIÊNCIA E NANOBIOTECNOLOGIA

RODRIGO DA FONSECA SILVEIRA

Desenvolvimento de software para modelagem preditiva de nanopartículas poliméricas de interesse farmacêutico

Tese de doutorado apresentada ao Programa de Pós- Graduação em Nanociência e Nanobiotecnologia da Universidade de Brasília como parte dos requisitos para obtenção do título de doutor.

Orientador: Prof. Dr. Marcílio Sérgio Soares da Cunha Filho

Coorientador: Dr. Idejan Padilha Gross

Brasília

2026

RODRIGO DA FONSECA SILVEIRA

Development of software for predictive modeling of polymeric nanoparticles of pharmaceutical interest

Tese de doutorado apresentada ao Programa de Pós- Graduação em Nanociência e Nanobiotecnologia da Universidade de Brasília como parte dos requisitos para obtenção do título de doutor.

BANCA EXAMINADORA

Marcílio Cunha Filho
Universidade de Brasília – presidente

João Paulo Longo
Universidade de Brasília

Nádia Felix Silva
Universidade Federal de Goiás

Sávio Salvarino Teles
Universidade Federal de Goiás

Guilherme Martins Gelfuso
Universidade de Brasília - suplente

RESUMO

A formulação de nanopartículas poliméricas é um processo complexo e multivariado, tradicionalmente guiado por abordagens empíricas de "tentativa e erro" que dificultam o escalonamento industrial. Embora a ciência de dados e o aprendizado de máquina (machine learning - ML) ofereçam alternativas promissoras para a implementação do Quality by Design (QbD), a literatura atual carece de validações robustas, baseando-se predominantemente em pequenos conjuntos de dados e modelos restritos a um único fármaco.

Neste contexto, esta tese estruturada em formato multipaper apresenta o desenvolvimento, a consolidação e a implementação tecnológica da plataforma NANO-AI. A pesquisa iniciou com a construção de uma prova de conceito focada no fármaco modelo isoniazida, evoluindo posteriormente para um pipeline otimizado de duas etapas: uma classificação binária para predição da viabilidade de formação das nanopartículas, seguida por modelos de regressão para estimar os atributos críticos de qualidade (tamanho hidrodinâmico, índice de polidispersidade, potencial zeta e eficiência de encapsulação).

Utilizando um banco de dados composto por 925 formulações contendo nove princípios ativos físico-quimicamente diversos, o modelo classificador (Extreme Gradient Boosting - XGBoost) foi o responsável pela identificação de sistemas viáveis. Na etapa de regressão, os algoritmos Deep Learning e XGBoost alcançaram coeficientes de determinação (R^2) superiores a 0,95 para o tamanho, potencial zeta e eficiência de encapsulação. A eficiência preditiva do framework foi atestada por meio de experimentos interlaboratoriais e pela generalização zero-shot em três fármacos inéditos (cafeína, nifurtimox e itraconazol), e demonstrou satisfatória capacidade de extrapolação preditiva para esses fármacos não utilizados na fase de treinamento dos modelos.

Por fim, a plataforma se materializou na implementação de uma arquitetura de software completa, integrando extração automatizada de dados em Python, armazenamento em bancos de dados relacionais PostgreSQL, algoritmos em linguagem R e um simulador de formulações interativo hospedado no Microsoft Power BI, alimentado por mais de 720.000 formulações virtuais simuladas.

Palavras-chave: nanoprecipitação; nanocápsula polimérica; mineração de dados; ciência de dados.

ABSTRACT

The formulation of polymeric nanoparticles is a complex and multivariate process traditionally guided by empirical trial-and-error approaches, which hinder industrial scale-up. Although data science and machine learning (ML) provide promising alternatives for the implementation of Quality by Design (QbD), the current literature lacks robust validation studies, relying predominantly on small datasets and models restricted to a single drug.

In this context, this multipaper-format thesis presents the development, consolidation, and technological implementation of the NANO-AI platform. The research began with the construction of a proof of concept focused on the model drug isoniazid and subsequently evolved into an optimized two-stage pipeline: a binary classification model for predicting nanoparticle formation feasibility, followed by regression models to estimate critical quality attributes (hydrodynamic size, polydispersity index, zeta potential, and encapsulation efficiency).

Using a database composed of 925 formulations containing nine physicochemically diverse active pharmaceutical ingredients, the classification model (Extreme Gradient Boosting – XGBoost) was responsible for identifying viable systems. In the regression stage, the Deep Learning and XGBoost algorithms achieved coefficients of determination (R^2) greater than 0.95 for size, zeta potential, and encapsulation efficiency. The predictive performance of the framework was validated through interlaboratory experiments and zero-shot generalization involving three previously unseen drugs (caffeine, nifurtimox, and itraconazole), demonstrating satisfactory predictive extrapolation capability for compounds not included in the model training phase.

Finally, the platform was materialized through the implementation of a complete software architecture integrating automated data extraction in Python, storage in PostgreSQL relational databases, algorithms developed in the R programming language, and an interactive formulation simulator hosted on Microsoft Power BI, supported by more than 720,000 simulated virtual formulations.

Keywords: nanoprecipitation; polymeric nanocapsule; data mining; data science; machine learning.

FIGURES LIST

Chapter II

Figure 1. Cross-validation strategies for small datasets. (A) Scheme in which only one random split of the dataset was made. (B) A scheme where parts of the dataset are split n-times randomly and the model is trained over and over, with the final accuracy being an average of these executions. The blank portion is the testing data, and the gray portions are the training data, randomly changed every round

Figure 2. Authors' training areas related to the selected articles involving ANN in producing polymeric nanoparticles for pharmaceutical purposes.

Chapter III

Figure 1. Machine learning pipeline based on 3 steps. Step 1 with a binary classification algorithm; Step 2 with a multiclassification algorithm; and Step 3 with a regression algorithm to predict the expected size of the nanoparticles.

Figure 2. Cluster of size created by histogram analysis of the production and formulation variables used to synthesize isoniazid polymeric nanoparticles.

Figure 3. Representative images of nanoparticle precipitation observed at different stages of the production process: (a) during rotary evaporation (Id 44); (b) during the dripping of the organic phase into the aqueous phase (Id 61); and (c) after the production process (Id 74).

Figure 4. Improvement of classification model performance across experimental rounds. Metrics such as accuracy, precision, LogLoss, and R^2 were tracked for each machine learning algorithm in Step 1 (binary classification) and Step 2 (multiclassification) of the pipeline.

Figure 5. Size distribution in 3 rounds of experimental formulations for isoniazid nanoparticles by nanoprecipitation.

Figure 6. Evolution of classification performance metrics—accuracy, precision, LogLoss, and R^2—for five machine learning algorithms across three experimental rounds.

Chapter IV

Figure 1 Schematic representation of the experimental design space, including formulation variables, process parameters, and molecular descriptors used to construct the dataset.

Figure 2. Schematic representation of the two-step machine learning framework (NANO-AI) for nanoparticle design, integrating feasibility screening and property prediction.

Figure 3. Comparative performance of classification models for nanoparticle formation prediction (Step 1). Model performance is evaluated using (A) Area Under the ROC Curve (AUC), (B) F1-score, (C) Precision, and (D) Recall.

Figure 4. Regression performance of machine learning models for nanoparticle property prediction (Step 2). R^2 values and error metrics (RMSE and MAE) are shown for size, PDI, encapsulation efficiency, and zeta potential.

Figure 5. Mean prediction error (±SD) of the NANO-AI framework across operators and laboratories for nanoparticle size, PDI, zeta potential and encapsulation efficiency.

Figure 6. Zero-shot validation of the NANO-AI framework for nanoparticle property prediction using unseen APIs (caffeine, nifurtimox, and itraconazole). Prediction errors (%) are shown as a function of experimentally measured values for (A) mean hydrodynamic diameter, (B) polydispersity index (PDI), (C) zeta potential, and (D) encapsulation efficiency (EE). Each point represents an individual formulation.

Chapter V

Figure 1 The computational architecture of the NANO-AI platform is organized into five distinct functional layers. The first layer comprises the primary experimental data source, consisting of researcher-generated spreadsheets documenting nanoparticle formulations. The second layer constitutes the Python-based ETL (Extract, Transform, Load) pipeline, encompassing four principal scripts responsible for data transformation, database ingestion, record retrieval, and the generation of synthetic datasets. The third layer utilizes a PostgreSQL relational database system, which structures and stores the processed data within interconnected tables. The fourth layer serves as the predictive engine, housing the trained classification and regression models. Finally, the fifth layer is the visualization and deployment interface, utilizing Power BI to host the NANO-AI formulation simulator dashboard.

Figure 2. Illustration of the distribution of the 1,006 experimental records across the investigated drug set. The core developmental datasets were generated following the protocol established for isoniazid (Chapter 3), with each drug undergoing 30 experimental rounds to yield 90 or more unique formulations. Although caffeine, nifurtimox, and itraconazole were utilized for external validation, the NANO-AI platform employs an active learning framework; consequently, these validation data were reincorporated into the relational database to further optimize model accuracy as part of a continuous refinement cycle.

Figure 3. The NANO-AI simulator interface provides a dual-panel configuration for real-time formulation design. The lateral control panel features a suite of interactive filters that allow for the dynamic modulation of nanoparticle attributes, including polymer matrix selection, the target active pharmaceutical ingredient (API), and specific numerical intervals for hydrodynamic size, encapsulation efficiency, polydispersity index (PDI), and zeta potential. Furthermore, users can calibrate a statistical probability threshold for nanoparticle formation. Adjustments to these parameters trigger instantaneous updates in the primary results table, which enumerates the corresponding formulation variables.

TABLES LIST

Chapter II

Table 1. Comparison and steps of data science methods knowledge discovery in databases, cross-industry standard process for data mining and SEMMA.

Table 2. Studies of polymeric nanoparticle production using data science models.

Table 3. Analysis of selected articles using a checklist of the good data analysis practices for machine learning.

Chapter III

Table 1. Nanoparticle size as the output variable from the ML pipeline in Step 2, along with the corresponding experimental results obtained in the 2nd round. Predicted nanoparticle size (nm) using three machine learning models (Random Forest, XGBoost, and ANN) in 2nd round, compared with available experimental data. The final predicted range corresponds to the most frequent size range predicted across the three models. The result evaluation indicates whether the formulation was considered successful. n.a. = not accessed.

Table 2. Final predictions. The first column shows the value of nanoparticle size measured in the laboratory. The next 3 columns indicate the classes predicted in step 2 (Random Forest, XGBoost and Deep Learning, respectively), while the fourth column represents the value predicted by the Deep Learning regression algorithm of step 3. Column 5 represents the final result of the pipeline and indicates the regression value predicted in step 3 if it lies between the range predicted in step 2.

ABBREVIATIONS LIST

ANN	Artificial Neural Networks
ML	Machine Learning
CRISP-DM	Cross-Industry Standard Process for Data Mining
KDD	Knowledge Discovery in Databases
SEMMA	Sampling, exploration, modification, modeling, and evaluation
DBSAN	Density-based spatial clustering of applications with noise
BIRCH	Balanced iterative reducing and clustering using hierarchies
FF	FeedForward
MLP	Multilayer perceptron
P	Perceptron
DCN	Deep Convolutional Network
ReLU	Rectified linear unit
SQL	Structured Query Language
PVA	Polyvinyl alcohol
EE	Encapsulation Efficiency
UV	Ultraviolet
API	Active Pharmaceutical Ingredient
AUC	Area Under the Curve
RMSE	Root Mean Square Error
MAE	Mean Absolute Error
QbD	Quality by Design
TRL	Technology Readiness Level

SUMMARY

1. Introduction 14

1.1 Background and the Empirical Formulation Problem 14

1.2 The Role of Data Science and Methodological Gaps in the Literature 15

1.3 Development of the Predictive Framework 17

1.4 Consolidation, Generalization and the NANO-AI Platform 19

1.5 Technological Implementation and Software Architecture 21

2. The role of artificial intelligence and data science in nanoparticles development: a review 23

2.1 Background 24

2.2 Data science 26

2.2.1 Methodologies and Analysis Steps 26

2.2.2 Basics considerations on ANN 28

2.2.3. State-of-the-art polymeric nanoparticles assisted by data science 30

2.2.4 Output features used in ANN 31

2.2.5 Computer-aided design of experiments 33

2.2.6 Computer languages, frameworks, and programming for data science 35

2.2.7 Number of experiments 36

2.2.8 Useful data manipulation techniques 37

2.2.9 Authors academic background 40

2.3 Conclusion 44

2.4 References 45

3. Multistep machine learning pipeline for polymeric nanoparticle design 52

3.1 Introduction 53

3.2 Experimental 55

3.2.1 Materials 55

3.2.2 Production of nanoparticles 55

3.2.3 Characterization of nanoparticles 56
2.2.4. Algorithms and hyperparameters 56
2.2.5. Pipeline for ML models 58
2.2.6. Step 1: Binary classification model 59
2.2.7. Step 2: Multiclassification model 60
2.2.8. Step 3: Outcome refinement 62
3.3 Results and Discussion 63
3.3.1 First experimental round 63
3.3.2 Second experimental round 65
3.3.3 Third experimental round 68
3.3.4 Data distribution and model learning progress 71
3.3.5 Algorithm performance evolution across rounds 73
3.3.6 Step 3: Outcome refinement 75
3.4 Conclusions 78
3.5 References 78

4. A Machine Learning Framework for Polymeric Nanoparticle Design with Zero-Shot Generalization Across Active Pharmaceutical Ingredients 85

4.1 Introduction 86
4.2 Materials and Methods 88
4.2.1 Materials 88
4.2.2 Nanoparticle preparation 88
4.2.3 Physicochemical characterization 88
4.2.4 Dataset construction and Experimental design 89
4.2.5 Feature Engineering and Input Variables 91
4.2.6 ML model development 92
4.2.7 Two-Step predictive framework 93
4.2.8 Model evaluation and interpretation 95
4.2.9 External validation strategy 97
4.3 Results and Discussion 98
4.3.1 Step 1 (Binary classification): Model selection and Feasibility prediction performance 98
4.3.2 Step 2 (Regression): Prediction of nanoparticle properties 101
4.3.3 Robustness under interlaboratory and Operator variability 107
4.3.4 Generalization to Unseen APIs 109
4.4 Conclusion 111

4.5 References.....111

5. Software Architecture and Technological Implementation of the NANO-AI Platform.....118

5.1 Data Acquisition, Ingestion, and Structuring (ETL Pipeline).....118

5.2 Machine Learning Model Training.....120

5.3 Deployment and the Interactive Power BI Simulator.....124

5.4 Interactive Formulation Simulator and Predictive Dashboard.....125

1 Introduction

1.1 Background and the Empirical Formulation Problem

Nanotechnology has emerged as one of the most promising platforms for technological innovation in healthcare, with polymeric nanoparticles standing out due to their versatility in protecting active pharmaceutical ingredients, modulating drug release profiles, and enabling targeted tissue delivery. However, the formulation of polymeric nanoparticles constitutes a highly complex and multivariate system. Subtle variations in process parameters (such as stirring conditions and solvent addition rates) or formulation variables (including polymer concentration and the intrinsic physicochemical properties of the drugs) directly impact critical quality attributes, namely particle size, polydispersity index (PDI), zeta potential, and encapsulation efficiency. Traditionally, the development of these nanocarriers has been primarily driven by empirical "trial-and-error" approaches or classical statistical models, such as response surface methodologies. These conventional strategies frequently struggle to capture the nonlinear interactions and the high dimensionality inherent to the nanoprecipitation process, leading to limited experimental reproducibility, high operational costs, and significant barriers to successful industrial scale-up.

Indeed, despite the substantial volume of academic research, the successful transition of nanomedicines from the laboratory to industrial scales remains severely restricted. This translational gap is largely attributed to the high experimental variability and the acute sensitivity of the nanoprecipitation process to slight parametric fluctuations. When relying on conventional "trial-and-error" methods, researchers are forced to navigate a high-dimensional search space characterized by complex, interdependent variables, such as polymer features, solvent/antisolvent ratios, and the intrinsic physicochemical properties of the active ingredients. While classical statistical tools, like

factorial designs and response surface methodologies, provide some structured experimental guidance, they are fundamentally limited to exploring localized regions of the formulation space. Because these conventional methods generally assume linearity or rely on low-order polynomial models, they inherently fail to map the high-order, non-linear interactions governing colloidal systems. Consequently, traditional empirical strategies are not only inefficient and highly resource-consuming but also lead to formulation instability and poor interlaboratory reproducibility, highlighting the urgent need for more advanced, data-driven predictive frameworks.

Consequently, the persistent reliance on empirical methodologies severely hinders the implementation of Quality by Design (QbD) principles, which are increasingly demanded in modern pharmaceutical development. A QbD approach requires a comprehensive understanding of the formulation design space and the ability to accurately predict how critical material attributes and process parameters influence the final nanocarrier. However, comprehensively mapping this high-dimensional space exclusively through physical experimentation is practically unfeasible, creating a significant bottleneck in the translation of nanomedicines from bench-scale research to clinical and industrial applications. Therefore, a fundamental paradigm shift is essential: the transition from a resource-intensive, trial-and-error experimental model to a rational, predictive formulation strategy capable of deciphering the complex physical chemistry of nanoprecipitation and ensuring interlaboratory reproducibility.

1.2 The Role of Data Science and Methodological Gaps in the Literature

To overcome the limitations inherent to traditional empirical approaches and classical experimental designs, the integration of data science, specifically machine learning (ML), has emerged as a transformative strategy in pharmaceutical

nanotechnology. Unlike conventional statistical tools, such as response surface methodologies, ML algorithms excel at deciphering the high-dimensional and nonlinear interactions that govern nanoprecipitation processes. By effectively recognizing complex patterns from multivariate datasets without requiring prior mechanistic assumptions, these advanced computational models offer a robust framework to optimize formulation variables. Consequently, ML not only accelerates the rational design of polymeric nanoparticles but also strategically aligns their development with Quality by Design (QbD) principles, enabling more accurate predictions of critical quality attributes while significantly reducing the experimental workload.

Despite the clear theoretical advantages of machine learning, a critical analysis of the current state-of-the-art, detailed in Chapter 2, reveals that its application in pharmaceutical nanotechnology remains constrained by significant methodological shortcomings. Most computational studies in this field operate with small, imbalanced datasets that frequently fail to meet the minimum sample size requirements necessary for complex algorithms. Furthermore, there is a recurrent lack of robust data preprocessing techniques, such as K-fold cross-validation and appropriate data balancing methods, coupled with a widespread reliance on inflexible proprietary software rather than versatile programming languages like R or Python. From a formulation perspective, existing predictive models are overwhelmingly restricted to single active pharmaceutical ingredients (APIs) and predominantly focus on estimating particle size, routinely neglecting other essential critical quality attributes like encapsulation efficiency and zeta potential. Consequently, these methodological fragilities systematically result in overfitted models that exhibit a limited capacity to generalize predictions when exposed to novel molecules or real-world interlaboratory variability, thereby restricting their practical application in industrial environments.

Consequently, addressing these critical literature gaps is essential for the advancement of computational nanomedicine. There is a pressing need to develop comprehensive machine learning workflows that strictly adhere to reinforce data science protocols, incorporating essential steps such as appropriate data balancing techniques, rigorous cross-validation strategies, and the systematic evaluation of multiple complementary algorithms. Furthermore, predictive models must evolve beyond the isolated prediction of particle size to encompass a holistic set of critical quality attributes, such as zeta potential and encapsulation efficiency, to truly inform rational formulation design. By overcoming these methodological deficiencies, it becomes possible to construct reliable, scalable predictive frameworks capable of navigating the multivariate complexity of polymeric nanoparticles. This paradigm shift, from fragmented, overfitted computational exercises to rigorously validated, data-driven platforms, constitutes the central motivation of this thesis, establishing the foundation for the development of the multistep predictive framework detailed in the subsequent section.

1.3 Development of the Predictive Framework

In direct response to the methodological gaps identified in the literature, Chapter 3 details the preliminary development of a modular, multistep machine learning pipeline designed to rationalize the formulation of polymeric nanoparticles. Using isoniazid as a challenging model drug, this initial proof-of-concept established a structured predictive workflow organized into three sequential and iterative stages: (1) a binary classification step to predict the feasibility of nanoparticle formation, (2) a multiclass classification step to categorize particle size ranges, and (3) a final regression step to refine the exact hydrodynamic diameter estimation. To systematically overcome the inherent challenges of small and imbalanced experimental datasets, the algorithms

were progressively retrained across consecutive experimental rounds, leveraging synthetic virtual simulations to strategically guide the selection of new experiments and optimize data distribution. This dynamic approach successfully mitigated data scarcity and enhanced prediction reliability, demonstrating that Artificial Neural Networks (ANN) could consistently achieve high predictive performance, with coefficients of determination (R^2) exceeding 0.9 across both classification and regression tasks.

A key innovation of this multistep strategy was the integration of intermediate classification outputs as functional constraints during the final regression phase, which significantly enhanced overall model robustness and prevented physically unrealistic estimations. Consequently, the finalized pipeline demonstrated satisfactory predictive accuracy across a broad hydrodynamic size range—spanning from approximately 75 to 768 nm—while maintaining maximum absolute errors below 40 nm. Most importantly, while isoniazid provided a stringent benchmark due to its challenging formulation characteristics, the insights gained from this initial framework extended far beyond a single compound. This preliminary study successfully validated the hypothesis that a modular, data-driven architecture could systematically reduce experimental workload and support Quality by Design (QbD) initiatives. Ultimately, this proof-of-concept established the methodological foundation necessary to scale the predictive system, paving the way for its extrapolation to a highly heterogeneous chemical space involving multiple active pharmaceutical ingredients.

Furthermore, a crucial methodological advancement introduced in this framework was the implementation of an active, iterative experimental design to systematically resolve dataset imbalances. Initial exploratory rounds revealed a pronounced data gap within specific hydrodynamic size intervals, particularly between 300 and 800 nm. To overcome this limitation without resorting to exhaustive empirical

trials, the provisional algorithms were utilized to simulate tens of thousands of virtual formulations, intelligently guiding the selection of subsequent experiments to specifically populate these underrepresented regions. This targeted data acquisition, combined with an ensemble voting mechanism to enhance classification confidence, allowed the models to progressively refine their accuracy across three experimental cycles. Ultimately, this proof-of-concept not only validated the hypothesis that a modular, data-driven architecture can effectively support Quality by Design (QbD) initiatives by minimizing material waste and experimental workload, but it also established the solid computational foundation required for system scale-up. Having successfully navigated the stringent benchmark provided by isoniazid, the pipeline was primed for extrapolation to a highly heterogeneous chemical space involving multiple active pharmaceutical ingredients, which constitutes the primary focus of the subsequent section.

1.4 Consolidation, Generalization and the NANO-AI Platform

Building upon the foundational proof-of-concept established with a single model drug, Chapter 4 details the consolidation and expansion of the initial predictive pipeline into a generalized, two-step machine learning platform termed NANO-AI. To overcome the limitations of domain-restricted models and accurately map a highly heterogeneous chemical space, the experimental database was systematically expanded to encompass 925 formulations comprising nine physicochemically diverse active pharmaceutical ingredients (APIs). This substantial increase in dataset volume and complexity enabled the structural simplification of the previous multistep workflow into an optimized two-stage architecture. Specifically, the consolidated framework integrates an initial binary classification layer to reliably screen formulation feasibility, followed directly by regression models designed to estimate critical physicochemical properties,

including particle size, polydispersity index, zeta potential, and encapsulation efficiency. Ultimately, the NANO-AI platform aims to provide a scalable computational tool that extends predictive capabilities beyond its original training domain, introducing the potential for zero-shot generalization across novel, unseen drugs while reinforcing the application of Quality by Design (QbD) principles in the rational development of nanomedicines.

Beyond theoretical performance, ensuring the practical applicability of the NANO-AI platform required solid evaluation under real-world experimental variability. Through comprehensive interlaboratory assessments involving different operators and distinct research facilities, the predictive models demonstrated reliable results, confirming that the algorithms effectively capture fundamental physicochemical trends rather than laboratory-specific artifacts. Furthermore, the successful zero-shot validation corroborated the framework's ability to extrapolate learned relationships to novel molecules, providing reliable directional guidance for key critical quality attributes—even in the absence of prior empirical data. Transitioning from a purely computational exercise to a highly functional technological product, the system was implemented within an interactive, web-accessible interface that allows users to virtually simulate and filter optimal formulation conditions based on their specific developmental needs. By enabling researchers to computationally navigate the complex multidimensional design space prior to physical execution, the NANO-AI platform potentially reduces material waste and experimental workload. Ultimately, this consolidation elevates the research to an advanced level of technological readiness, establishing the definitive scientific and technological contribution of this doctoral thesis.

1.5 Technological Implementation and Software Architecture

To bridge the gap between advanced computational modeling and practical laboratory application, the final stage of this research, detailed in Chapter 5, focuses on the technological implementation of the NANO-AI platform as a fully functional software system. Recognizing that predictive algorithms must be accessible to formulation scientists without advanced programming expertise, the validated machine learning models were integrated into a automated digital architecture. The system's backend employs Python scripts to manage data ingestion from cloud-based experimental records, structuring the information within a PostgreSQL relational database. Subsequently, the predictive models, coded in R, process this standardized dataset to simulate formulation outcomes and estimate critical nanoparticle characteristics. Finally, the resulting predictions are deployed through an interactive, web-accessible dashboard developed in Power BI. This user-friendly interface acts as an formulation simulator, allowing researchers to dynamically filter and select optimal conditions based on their specific developmental needs, such as drug type, polymer, and target particle size. By enabling users to computationally navigate the design space prior to physical execution, this software implementation fundamentally prevents material waste and reduces experimental time, thereby consolidating the applied nature of this thesis and elevating the technology to an advanced level of operational readiness.

Furthermore, this project addresses a gap in the field, as no comprehensive, data-driven formulation simulator for polymeric nanoparticles currently exists. By transitioning from isolated computational scripts to a centralized, multi-user environment—integrated with automated data ingestion from cloud-based records and established data governance—the NANO-AI system functions as a technological asset

for industrial transfer, currently reaching Technology Readiness Level (TRL) 8. To detail the development of this platform, this thesis is structured as follows: Chapter 2 reviews the application of machine learning in nanomedicine; Chapter 3 describes the initial predictive framework for a single model drug; Chapter 4 details the platform's consolidation and its zero-shot generalization capacity across multiple active pharmaceutical ingredients; and Chapter 5 examines the software architecture, database management, and the deployment of the NANO-AI simulator.

2 The role of artificial intelligence and data science in nanoparticles development: a review*

* article published in Nanomedicine, DOI: https://doi.org/10.1080/17435889.2024.2359355

Abstract

Artificial intelligence has revolutionized many sectors with unparalleled predictive capabilities supported by machine learning (ML). So far, this tool has not been able to provide the same level of development in pharmaceutical nanotechnology. This review discusses the current data science methodologies related to polymeric drug-loaded nanoparticle production from an innovative multidisciplinary perspective while considering the strictest data science practices. Several methodological and data interpretation flaws were identified by analyzing the few qualified ML studies. Most issues lie in following appropriate analysis steps, such as cross-validation, balancing data, or testing alternative models. Thus, better-planned studies following the recommended data science analysis steps along with adequate numbers of experiments would change the current landscape, allowing the exploration of the full potential of ML.

Keywords

polymeric nanoparticle; data mining; data science; artificial neural network; machine learning; quality by design.

2.1 Background

Over time, data science has progressed fast, uncovering new knowledge hidden in data and contemporizing humanity's significant achievements. Algorithms and models for classifying or predicting information have gained prominence, particularly those based on the machine learning (ML) approach. In this context, the rise of artificial neural networks (ANN) has revolutionized the field, especially due to the bio-inspired architecture that allows the capture of highly complex relationships and its growing capacity to learn and adapt with the progressive increase in available knowledge [1]. Thus, data science has supported decision-making in many fields, from finance to education to pharmaceuticals [2].

The modern framework of data sciences has an essentially interdisciplinary nature, not only because of its potential application in all areas of knowledge but also because it requires analytical steps from very specialized domains. For example, computer science is essential to select, prepare, and process datasets using programming codes. Statistics, in turn, creates models based on data distribution, probability, and resampling techniques. Additionally, mathematical science is indispensable in recognizing models associated with time series, such as ANN, or algebraic concepts, such as vectors and matrices. Finally, design and marketing are necessary to present the results intuitively and naturally for the project's target audience, known as "data storytelling" [2]. In this intricate association of expertise, the result is invariably compromised if any step fails, making the conclusions obtained useless or even dangerously wrong.

Nanotechnology, coincidentally, presents similar development milestones and has an equally disruptive character. The possibility of manipulating matter in nanometric dimensions has demonstrated remarkable potential in energy, environment, and health. However, nanotechnology is still far from having its full potential explored on an

industrial scale [3]. Even though most scientific studies in nanotechnology have entirely empirical developments, data science has been used to obtain predictive models related to specific properties of nanoparticles and their production process [4–6].

Indeed, computer-aided design of experiments has become a valuable addition to the systematic quality-by-design approach in pharmaceutical drug development [7]. In fact, the basis of quality-by-design lies in describing the product's desired quality attributes using computational tools such as molecular docking and surface response methods [8,9].

However, these tools alone seem pretty simplistic, considering the sophisticated processes available for data science. Response surface uses more straightforward methods, such as linear regression and polynomial regression, to model the data and cannot capture the complexity and diversity of some real-world phenomena [8]. In addition, it can be sensitive to outliers, noise, and measurement errors in the data. In the case of molecular docking, inaccurate consideration of ligand flexibility or binding energy due to the lack of reliable Information limits its scope of action to preliminary research phases [7].

On the other hand, ML models that can generalize different complex processes could fill these gaps and make more accurate predictions, mainly when supported by balanced input samples during model calibration. The ability of ML to recognize patterns and optimize processes makes it a valuable tool to support the development of nanomedicines that align with the overall goals of quality-by-design.

In fact, ML has been used to support the development of different types of drug products, including those tailored to assess *in-vivo*/*in-vitro* correlations [10], dissolution profiles [11,12], and skin permeation characteristics [13–15]. ML tools have also been explored in obtaining drug-loaded nanoparticles, particularly for the prediction of the

particle size [4-6,16–24], which holds great importance in the drug biodistribution [25,26] and drug delivery [27]. Furthermore, ML models have been created to predict the zeta potential [4,19,21] and encapsulation efficiency [21,24,28], which can be crucial for stability and drug targeting. Despite the recognized benefits of this approach, ML is far from being a reality accessible for the pharmaceutical industry in nanostructured systems.

In this scenario, this review proposes, for the first time, to perform a critical analysis of the data science methodologies and protocols used in the predictive models for drug-loaded nanoparticle properties obtained from ML. Specifically, this study focuses on the evaluation of the ML state-of-the-art related to polymeric nanoparticles, which are one of the most common nanoparticles used for therapeutic purposes, identifying if the proper steps for building a reliable model were followed and how the chosen models can be used to solve real-world problems outside the laboratory. Finally, opportunities for improvements in data science in this field are discussed.

2.2 Data science

2.2.1 Methodologies and Analysis Steps

The development of sophisticated methodologies for data science over time has helped to organize the critical steps in extracting new knowledge from datasets [29]. One of the most renowned processes is the cross-industry standard process for data mining (CRISP-DM) [30], which combines the best practices to make data mining as productive and efficient as possible. This tool has been used in financial data, human resources, production, and customer habits, mainly focused on business understanding and deployment aspects.

The knowledge discovery in databases (KDD), in turn, is another recognized method for developing strategies to discover knowledge from data, focusing on data

mining steps that have been used to identify medical problems [31], in the field of psychology for text mining [32], or even for classifying energy quality in feeders [33]. Another approach, the data mining process known as SEMMA, involves five key stages – sampling, exploration, modification, modeling, and evaluation. Such a system has proven helpful in identifying the best techniques for handling large amounts of data [34], improving probabilistic sound predictions [35], or building a predictive model for loan defaults [36]. Despite their particularities, they all have similar steps and common goals [37], as in Table 1.

The first phase considered crucial in data analysis by CRISP-DM is business understanding, in which the target product or research objective is outlined. This step is part of identifying success criteria, assessing the resources needed, evaluating the associated risks, developing a project plan, and analyzing the related costs and benefits [29].

The next step, common to the different methodologies (understanding/data selection/sampling phase), is focused on gathering essential information [37]. The focus is on describing, exploring, and understanding the data, ensuring quality, and preventing the issue of imbalanced data [29].

In the data preparation/transformation/modification phase, various techniques are applied to eliminate data imbalances, e.g., oversampling or undersampling methods [38]. Oversampling involves supplementing the original data set with artificially generated data from the existing data source [29]. Conversely, undersampling reduces the data in the original dataset to achieve a balanced representation. Then, the most appropriate modeling techniques are chosen based on the previously analyzed input characteristics and the desired target solution. Models can vary based on the specific problem [37].

The results achieved are carefully examined in the

evaluation/interpretation/assessment phase, and their compliance with the success criteria defined in the business understanding phase is assessed. In this phase, whether to continue with the implementation step or go through the previous phases again if the results do not meet the expected results. In this situation, all necessary steps are repeated until the success criteria are met and the desired results are achieved [30]. This evaluation process is crucial for determining the effectiveness and validity of the developed model.

Finally, the implementation plan is executed to put the developed model into practice, foreseen explicitly in the CRISP-DM methodology called Deployment. As a part of this process, it is crucial to establish a robust monitoring and maintenance plan to ensure the continued effectiveness and usefulness of the deployed model in the long term [29,37].

Table 1. Comparison and steps of data science methods knowledge discovery in databases, cross-industry standard process for data mining and SEMMA.

Knowledge discovery in database	CRISP-DM	SEMMA	Points to be addressed
—	Business understanding	—	What does the business need?
Data selection and integration	Data understanding	Sample	What data are available or needed? Are they clean?
Data cleaning and preprocessing		Explore	
Transformation	Data preparation	Modify	How to organize the data for modeling?
Data mining	Modeling	Model	What modeling techniques could be used?
Pattern evaluation/Interpretation	Evaluation	Assessment	Which model best meets the business goals?
—	Deployment	—	How can stakeholders access the results in the real world?

2.2.2 Basics considerations on ANN

The functioning of biological neural networks inspires ANN and usually consists of three layers level: the input, the hidden, and the output layers [1,39]. The input layers of the neural networks receive the available database that can be composed, for example, of experimental and/or theoretical results. The hidden layers perform the required

calculations and transformations on the input data. Finally, the output layer generates the results or predictions based on the processed inputs.

The multilayer perceptron is an extended version of the feedforward neural network. The term feedforward is derived from the unidirectional flow of Information within the network, where data moves forward from the input layers through the hidden layers to the output layer [1,39,40]. The training of neurons in a multilayer perceptron is usually performed using the backpropagation learning algorithm, which facilitates the adjustment of weights and biases and improves the precision and accuracy of predictions. It serves as a fundamental mathematical mechanism in data mining and ML. Multilayer perceptron can approximate any continuous function and solve problems that are not linearly separable. An ANN is distinguished by its ability to perform better on large amounts of data, unlike traditional algorithms [40,41].

ANN also has hyperparameters that must be specified before training the network [42,43]. These hyperparameters include the activation function, the number of nodes in the hidden layer, and the number of epochs. The activation function of a neuron in an ANN determines whether it should be activated based on the inputs it receives [42,44,45]. Different activation functions can be used depending on the nature of the problem.

The number of neurons in the hidden layer of an ANN is a crucial hyperparameter. Typically, the number of neurons in the hidden layer can be chosen between the size of the input layer and the size of the output layer [39,43]. If too few neurons are used, the model may not have enough capacity to capture the underlying patterns in the data, resulting in inadequate fitting. On the other hand, too many neurons can lead to overfitting, where the model memorizes the training data instead of learning generalizable patterns [39,42].

The epoch refers to a complete cycle over the entire training dataset during ANN

training. Determining an appropriate number of epochs is critical to achieving optimal performance.

A reasonable starting point for the number of epochs is often three times the number of columns or features in the dataset [39,42]. Selecting the correct hyperparameters for an ANN requires experimentation and fine-tuning to find the optimal configuration that provides the best performance and generalization ability for the problem [39,40]. It demands in-depth knowledge of the nature of the data analyzed and is carried out in a non-automated way.

2.2.3 State-of-the-art polymeric nanoparticles assisted by data science

A literature review was performed to identify studies involving the intersection of polymeric nanoparticles and data science. Web of Science, Scopus, and Google Scholar were used as databases. The search criteria involved combining keywords such as "polymeric nanoparticles" with relevant data science terms, including "machine learning," "prediction," "artificial intelligence," "neural network," and "deep learning". No time restrictions were used in order to be obtained all possible results. Only articles that specifically addressed the production of polymeric nanoparticles for therapeutic application were included. Accordingly, specific exclusion criteria were applied, eliminating studies, for example, that produced metallic nanoparticles or focused on environmental applications. Table 2 provides a comprehensive overview of data science studies that specifically address the production of polymeric nanoparticles and fulfill pre-established criteria.

Table 2. Studies of polymeric nanoparticle production using data science models.

Year	Input features	Output features	Program package	Data size	Ref.
2005	Energy interaction polymer/drug, polymer, hydration energy	Encapsulation efficiency	NeuralWare	18	[28]
2011	Polymer viscosity, drug content, S/W ratio, mixing rate	Particle size	MATLAB	51	[6]
2011	Polymer concentration, surfactant concentration, storage temperature	Particle size	MATLAB	49	[16]
2014	Solution flow rate, washing time, ultrasonic power, molar ratio of CO_2 and solvent, agitation time	Particle size	NeuralWare	10	[17]
2017	Needle diameter, voltage, polymer/enzyme concentration	Particle size	INForm	30	[18]
2017	Viscosity, contact angle, polymer concentration, S:W ratio, interfacial Tension	Particle size, Polydispersity index	Visual Gene Developer	24	[5]
2018	Acetone concentration, polymer concentration, total volume, solvent/antisolvent ratio, agitation type, agitation intensity, temperature, mixing time, flow type	Particle size, Polydispersity index, Zeta potential	FormRules	189	[4]
2021	Type of stabilizer, stabilizer concentration. solvent/antisolvent ratio, solvent volume, mixing time, flow rate, acetone concentration, polymer molecular weight, polymer concentration	Particle size, Polydispersity index, Zeta potential	FormRules	299	[19]
2014	Polymer molecular weight, polymer and drug ratio, number of blocks	Particle size	MATLAB	27	[20]
2015	$CaCl_2$ concentration, homogenizer speed, agar concentration, polymer concentration	Particle size, Polydispersity index, Zeta potential, Encapsulation efficiency	MATLAB	60	[21]
2016	Polymer concentration, thiamine pyrophosphate concentration, polymer/thiamine pyrophosphate mass ratio	Particle size	MATLAB	45	[22]
2012	Polymer concentration, buffer pH, sonication amplitude, sonication Time	Particle size	INForm	52	[23]
2017	Polymer concentration, albumin concentration, stirring time	Particle size, encapsulation efficiency, cytotoxicity	INForm	30	[24]

2.2.4 Output features used in ANN

The particle size is one of the most essential characteristics to be monitored in the production of nanoparticles, and it can be assessed easily using, for example, dynamic light scattering. This characteristic is decisive for the therapeutic use of nanostructured systems, as it influences their pharmacokinetics and stability [17,46,47]. Indeed, studies have reported a significant correlation between size and drug absorption and have highlighted that particle size is an essential factor for their biodistribution profile [6].

Unsurprisingly, this output is present in 12 of the 13 studies selected for this analysis (Table 2). At the same time, it is a challenging parameter to control since even the exact reproduction of nanoparticle-obtaining protocols can sometimes result in different particle sizes, showing that this parameter is influenced by multiple non-linear factors [48].

The encapsulation efficiency is another crucial parameter in nanostructured drug delivery systems. It measures the percentage of drugs encapsulated/entrapped within the nanoparticles. Such property is governed by the interplay of several physicochemical and processing parameters. In fact, drug properties, including size, charge, and lipophilicity, influence their interaction with the polymer matrix. Notwithstanding, polymer characteristics, such as molecular weight, hydrophobicity, and functional groups, also directly impact drug-polymer interactions and loading capacity.

Moreover, the preparation method plays a significant role in determining the encapsulation efficiency [49]. In this context, factors such as solvent selection, mixing ratios, and processing parameters (e.g., sonication, emulsification, and stirring method) may significantly affect the encapsulation efficiency. Additionally, external factors such as temperature, pH, and stabilizers can also affect drug stability and interactions within the nanoenvironment, thereby impacting encapsulation yields.

Contrary to its relevance for nanostructured medicine, only 3 among the 13 selected studies (Table 2) added encapsulation efficiency as an output to the model. The laborious determination of this assay is what limits its modeling. The drug content required for the calculations is usually determined by complex analytical methods employing high-performance liquid or gas chromatography [50,51]. A model capable of making predictions regarding this parameter would have to be built with many drugs interspersing formulation and production variables, making the experiment execution difficult.

Zeta potential, which refers to the electrical potential at the surface of the hydrodynamic shear surrounding the colloidal particles, is another critical nanoparticle characteristic. These systems' stability during the nanostructured medicine's shelf life is highly dependent on this parameter [46]. Despite its easy experimental determination,

only a few studies (three among 13) included this parameter in the model (Table 2).

Controlling parameters within the desired ranges and reproducing results, especially when there is a change in production scale, are common issues in nanoparticle production since numerous factors strongly influence such features. This characteristic raises the need for multiple experiments. However, it is also the primary indication that ANN's use of data science prediction algorithms can radically leverage the development of nanoparticles based on reliable models [28].

2.2.5 Computer-aided design of experiments

Initial database searches returned 278 results involving this review's topics of interest. However, most of them use the surface response approach and a variant of it, the Box-Behnken experimental design – a surface response experiment that works with a fractional experimental design and estimates first and second-order coefficients [52]. This method proves advantageous when the safe operating range of the process is known, which is hardly the case when it comes to polymeric nanoparticles.

The surface response methodology is a valuable tool in experimental design, offering distinct advantages in optimizing processes and understanding the intricate relationships between input variables and desired outcomes. This methodology facilitates the exploration of the design space by systematically varying factors and analyzing their impact on the response of interest. Nevertheless, the approach allows for identifying optimal conditions improving efficiency and cost-effectiveness in pharmaceutical and chemical processes. One of the main advantages of such a method lies in its ability to provide a mathematical model with a relatively small number of experiments. However, surface response can produce biased results when factors vary widely or in the presence of outliers, especially when classical experimental principles such as repetition and

control variables are not strictly followed [21,52]. Furthermore, the response surface uses more straightforward methods, such as linear regression and polynomial regression, to model the data, potentially limiting the accuracy of predictions in highly nonlinear systems inherent in the complexity and diversity of some real-world phenomena [21,53,54] .

Another very common computational approach that has been used is molecular docking. Such a theoretical approach consists of a computational method generally based on molecular mechanics, which involves describing the polyatomic system using classical physics, often parametrized by quantum-mechanical semiempirical and ab initio theoretical calculations [55]. In this context, molecular docking is often used to simulate the interaction of two molecules, which represents an interesting strategy to simulate the interaction between different nanoparticles and drugs, testing the best combination of materials to maximize the drug loading capacity, for example [56]. However, molecular docking faces challenges, especially in accurately predicting complicated intermolecular interaction [56], and the complexity of the experimental conditions involved in the nanoparticle's synthesis restricts its viability.

ANN, for instance, can learn from data without requiring prior knowledge or assumptions about the process. It can also handle nonlinear and high-dimensional data, which appears to be the case with nanoparticles in general [42]. High dimensional data are naturally a feature of nanoparticle production, which depends on countless factors such as stabilizer concentration, solvent volume, mixing time, polymer molecular weight, polymer concentration, temperature, active ingredient content, and storage temperature [5,16,17,19,22]. The studies that compared the performance of ANN to the surface response method demonstrated their superiority for almost all outcomes of their models, even though using an insufficient number of experiments [21,53,57].

Traditional computational methods are practical because they require fewer experiments based on simpler equations to model the data [54,57]. Also, they are more accessible for laypeople once there are plenty of proprietary software options compared to ANN, which requires programming and more non-automated operations to make the tuned model work. Also, data science algorithms require extensive upfront experimentation for calibration and computer programming. Therefore, most of the nanotechnology research using computer-aid relies on traditional computational methods over ML. Despite experimental feasibility, such models are limited and have little application in the real world. Therefore, this review focused on the 13 quality studies on polymer nanoparticle fabrication using ML.

2.2.6 Computer languages, frameworks, and programming for data science

Computer programming is central to data science, providing tools – in code, software, or model applications – to apply knowledge to real-world problems and facilitate their solution. Studying data science requires knowledge of computer programming, using at least one programming language directly or through specialized software [2].

Using a programming language directly lets the programmer choose all the steps and variables in a model. It also allows the latest packages to be incorporated without the disadvantage of being tied to a specific software version, the update of which depends on the manufacturer[2,58]. Studies highlight Python, R, and SQL as the most advantageous programming languages for data science[59–61]. Although Java and C++ are used somewhat, Python and R are characterized by graphical analysis and the implementation of ML models, making them particularly efficient for data cleansing and integration tasks [59–61].

Five authors used MATLAB software to run their ML models in the studies reviewed, while most used proprietary software. This pattern strongly indicates that these authors are less familiar with the state-of-the-art tools commonly used in data science.

Although MATLAB is recognized as a fast and powerful language for data science, offering an excellent graphics package, it is considered less packaged in terms of ML algorithms and more complex to use than Python and R [53]. The Neuralware software, used in two studies, was last updated in 2011 and offers only feedforward neural networks trained with backpropagation for prediction and classification problems and Kohonen self-organizing maps for clustering problems (https://www.neuralware.com/index.php/products). The latest version of the INform software used in three studies was released in 2014 (https://www.inform-software.com/en/expertise/downloads). The newest version of Visual Gene Developer, used in one of the articles, dates from 2019 and was developed to optimize synthetic genes offering only ANN algorithms (https://www.visualgenedeveloper.net/Download.html). FormRules utilized by two articles was last updated in 2016 and only works with neuro-fuzzy systems with a limited input variable (https://www.bioz.com/result/formrules). Thus, considering how quickly data analysis procedures evolve, proprietary software does not meet the necessary flexibility in ML and is not updated enough in the dynamic landscape of data science.

2.2.7 Number of experiments

The amount of data required to train an ML model is influenced by several factors, the most important of which is the number of parameters the algorithm must learn. A widely recognized benchmark for assessing the adequacy of the dataset is the application of the "10 rules of thumb" [62]. This guideline specifies that the minimum amount of

input data should be ten times greater than the number of degrees of freedom of the model [62]. Some studies suggest that a data set of more than 100 instances is generally sufficient for most models that have complexity higher than linear [63,64].

It is important to note that following these recommendations does not guarantee optimal model accuracy. However, they are valuable initial benchmarks in most cases. Hence, less than half of the selected studies complied with such requirements regarding sample size (Table 2), which is quite surprising considering that most employed ANN exclusively. If an insufficient number of experiments are utilized, the model may lack the capacity to effectively capture the intricate underlying patterns within the data, leading to suboptimal fitting and compromised generalization performance.

The use of a limited sample amount may be due to the technical challenges involved in nanoparticle production and physicochemical characterization. The comprehensive characterization of colloidal dispersions involves laborious processes, as discussed in previous section. This multi-layered characterization can extend over several months with enormous associated costs. Consequently, conducting many experiments is possibly the biggest challenge, depending on the team resources available and the deadline for completing the research [65,66].

2.2.8 Useful data manipulation techniques

Some data manipulation procedures, generally unavailable in proprietary software, can be applied in ML to improve the dataset's quality. Strategies such as data equalization techniques, cross-validation methods, and fine-tuning of hyperparameters are particularly interesting for producing nanoparticles as they can reduce the number of experiments necessary to build a model with adequate accuracy.

Cross-validation can be a valuable strategy for small datasets. This technique facilitates a more objective and precise evaluation of a ML model performance by dividing the available data into distinct training and test sets. Such a technique is recommended explicitly for small datasets as it helps to utilize the available data better and avoid problems of overfitting or selection bias [41]. This verification seems particularly important when the validation approach gains heightened significance in domains such as nanoparticle synthesis development, where the experimental details could be of great significance. A one-time, randomized split training/testing approach was used in all articles examined (Table 2). However, training the models with data proportions different from the test data for small data sets is advisable. A solution is provided by K-fold cross-validation, where the data set is randomly split into n subsets, and the models are trained and tested n times, mitigating the effects of chance and improving generalization ability (Fig. 1).

Another essential analysis relates to the problem of unbalanced data, which often occurs in scenarios with small datasets and can significantly affect the performance of ML models [38]. A dataset is considered unbalanced if there is a noticeable disproportion in the number of examples in one or more classes compared to the others. In such cases, the effectiveness of ML algorithms' inaccurate classifications and regressions is impaired, as the majority classes are favored while the minority classes have a lower recognition rate. In situations with multiple variables and continuous values, such as the particle size, even a slight imbalance in the class distribution on the input layer can lead to accuracy errors. Therefore, it is essential to perform a comprehensive equilibrium analysis at the beginning of the study to mitigate such problems [41]. Furthermore, when dealing with dependent variables that collectively influence the same outcome, it becomes crucial to address not only the individual balance of each variable in isolation but also to carefully

consider and balance the intercorrelation among the data.

Unbalanced data issues can be addressed using under- and oversampling techniques. In the case of small data sets, oversampling can be a valuable strategy. Different oversampling methods exist, including oversampling with noise [67]. In this approach, a fixed, randomly generated number is added or subtracted to the mean of the unbalanced class. Another effective oversampling strategy is the synthetic minority over-sampling technique (SMOTE), which produces synthetic data points based on the K-nearest neighbors' algorithm [67]. These approaches can help overcome the challenges associated with imbalanced data and improve the performance of machine learning models.

SMOTE technique is also recommended when dealing with small sample sizes, primarily when the input data in each range is known for its excellent accuracy regarding the output data. By using a K-Means algorithm or selecting artificial data with some deviation from the actual values (noise) — obtained through a careful curation process — these accurate output values can serve as input data. Their precision, coupled with a small margin of error when outstanding accuracy is present in each range, helps improve the model's overall accuracy. It is important to note that this technique should be implemented with new tests to validate its effectiveness before considering its consolidation into the model [67].

No data-balancing analyses were performed in any of the studies reviewed. One study only cited such a step [28], where it was executed automatically using NeuralWare software without providing a detailed description of the procedure. All other studies randomly divided the data into train and test sets. In one study, for example [51], experiments were used to train the model, only four experiments were used for testing, and an accuracy of 94% was obtained, probably due to overfitting [6]. The same problem

was identified in another study [16], in which 40 and nine experiments were used, respectively, and an accuracy of 97% was achieved.

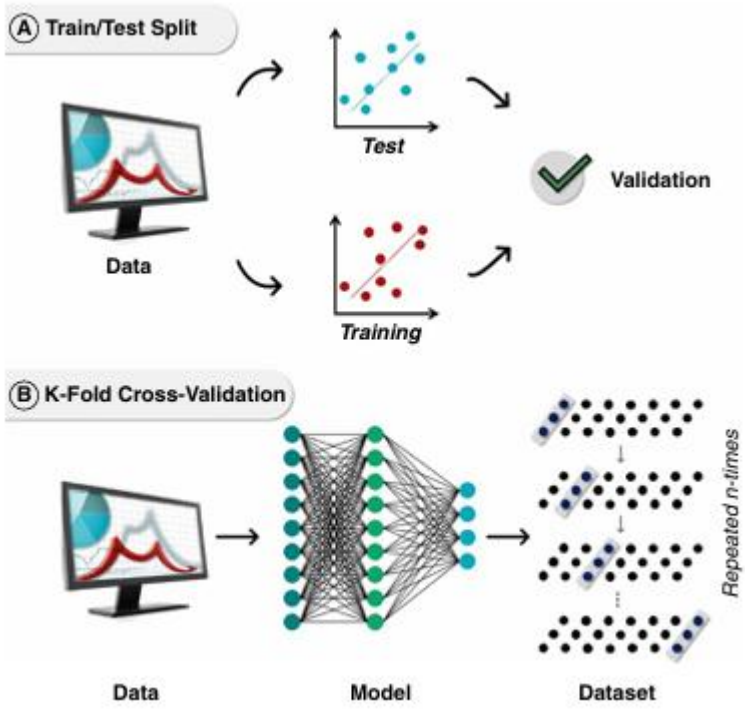


Figure 1. Cross-validation strategies for small datasets. **(A)** Scheme in which only one random split of the dataset was made. **(B)** A scheme where parts of the dataset are split n-times randomly and the model is trained over and over, with the final accuracy being an average of these executions. The blank portion is the testing data, and the gray portions are the training data, randomly changed every round.

2.2.9 Authors academic background

There was a noticeable diversity of academic background among the authors when evaluating the training areas of all the authors involved in the selected publications, of whom only 4% have academic training in computer science. Most authors (42%) have a degree in pharmacy, followed by 13% in chemical engineering and nanomedicine (Fig. 2). The papers that use MATLAB have authors with degrees in mathematics and engineering. Using a programming language is crucial to properly follow the steps established in data science for ANN, as already discussed.

The lack of knowledge about the methods and techniques of data science appears

to be an important factor that prevented the papers from achieving better results. Indeed, using a programming language is essential to properly follow the steps established in data science for ANN, as already discussed. The fact that none of the papers used languages such as R, SQL, and Python strongly suggests that the authors are not fully integrated into the ML and data science environment and possibly have never deployed and made their researched models available for external validation.

Furthermore, they need to perform the data analysis and preparation steps adequately. In these steps, the input variables are thoroughly examined. A correlation analysis must be performed to determine which of them could be discarded. If there is an imbalance in the data, it should be corrected not to compromise the model's accuracy. In addition, outliers must be identified and removed from the database. Frequently, converting continuous variables to discrete variables can reduce the complexity of the model, and it is a recommended procedure. Sometimes, an alternative to reduce complexity is to group one or more variables [2,37].

In the field of ML, it is imperative to recognize that there is no single model that is suitable for solving all problems [1,39,68]. The choice of the optimal model depends on factors such as the size and structure of the input dataset. Only three articles among the analyzed tested alternative models. Trying multiple models is interesting because each model can have different characteristics regarding complexity, accuracy, interpretability, and generalization [41]. When the dataset is small, these differences can be more significant and dependent on the training and test data. As the name suggests, data science is about experimenting and testing hypotheses, which should involve using multiple models to compare results. In classification, a simpler model, such as a decision tree, may better match the nature of the data set being analyzed than a more complex model, such as a neural network [41].

For example, in the case of nanoparticles, where multiple dimensions of variables are involved, a neural network could provide a more accurate result. Nevertheless, in the case of simple classification, which is concerned with whether nanoparticles form in a particular method, the use of simpler models may be more appropriate.

In practically all the studies, the experiments were just collected, some hyperparameters were tested, the model was run, and good accuracy values were obtained, which is usually very difficult to achieve, especially with small amounts of data and the lack of data balance analysis or cross-validation to compensate for the few experiments. Besides this weakness, some studies have not deployed the model, or the authors have validated it with few experiments using their dataset and still obtained good accuracy values, which could probably be a case of overfitting [5,6,18,20]. A good practice in data science is that another team should test the final model, which did not happen in any of the cases analyzed [30].

The pragmatic identification of the steps to be followed in using ML based on the checklist in Table 3 can serve as a basis for future studies in which ANN is used to produce polymeric nanoparticles.

2.2.10 Opportunities for improvements

An analysis of the articles regarding the implementation of essential steps in ML is presented in Table 3.

As a positive aspect, all studies considered using different values for the ANN hyperparameters, an easily adjustable variable in software such as MATLAB. Thus, all either tested a different number of neurons in the hidden layer, used more than one activation function, or tested different values for the epochs. Some studies used the surface response method to compare the results with the ANN model and discussed the

relative importance weights of the factors of the input variables and their significance [5,6,17,21,22].

However, all analyzed studies must include several basic data analysis steps discussed in the previous topics. There are no defined success criteria for building the model [2,37]. Determining the requirements for the model and the level of accuracy that should be achieved from the beginning is important in deciding whether to add a new cycle to the process, optimize the adjustment of hyperparameters, or even conduct new experiments [37].

This background analysis may suggest future research studies integrated into the ML and data science environment could profit from incorporating data scientists as team members.

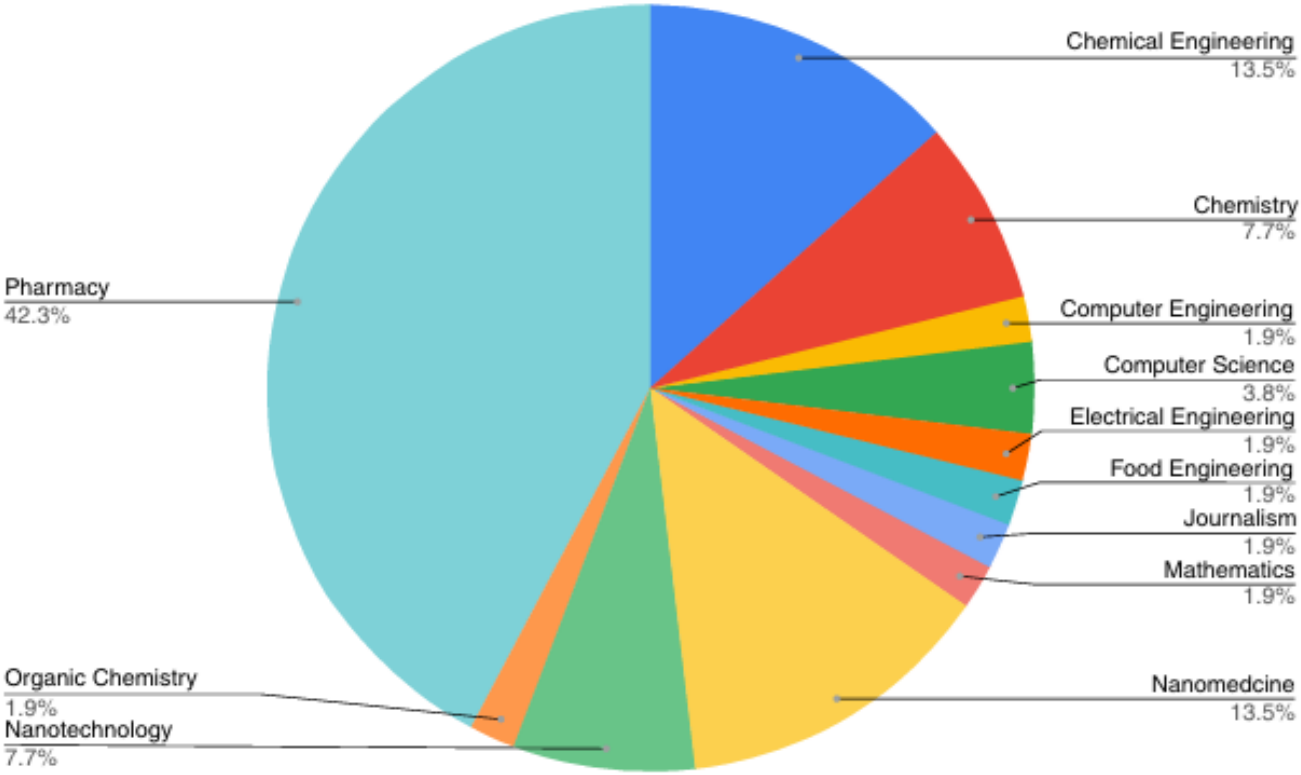


Figure 2. Authors' training areas related to the selected articles involving ANN in producing polymeric nanoparticles for pharmaceutical purposes.

Table 3. Analysis of selected articles using a checklist of the good data analysis practices for machine learning.

Practices/references	[28]	[6]	[16]	[17]	[18]	[5]	[4]	[19]	[20]	[21]	[22]	[23]	[24]
Appropriate data size	X	C	C	X	X	X	C	C	X	X	X	C	X
Steps of data analysis, preparation, and transformation	X	X	X	X	X	X	X	X	X	X	X	X	X
Cross-validation	X	X	X	X	X	X	X	X	X	X	X	X	X
Balance data	X	X	X	X	X	X	X	X	X	X	X	X	X
Wide input variables	X	X	X	C	X	C	C	C	X	X	X	X	X
Appropriate model deployment	X	X	X	X	X	X	X	X	X	X	X	X	X
Use of several models	C	X	X	X	X	X	X	X	X	C	C	X	X
Wide output variables	X	X	X	X	C	X	X	X	X	C	X	X	X
Impact of factors by ANN	X	C	X	C	X	C	X	X	C	X	C	X	C
Use of different values for ANN hyperparameters	C	C	C	C	C	C	C	C	C	C	C	C	C
Realistic accuracy determination	C	X	X	X	C	X	X	X	X	X	X	C	X
Appropriate language used for data science	X	C	C	X	X	X	X	X	C	C	C	X	X

2.3 Conclusion

This review addresses the question of how nanotechnology and data science can be combined to enhance the production of polymer drug-loaded nanoparticles. Nanomedicines are still far from realizing their full potential on an industrial scale, as repeatable production patterns are a constant challenge. While most scientific studies in the field of nanotechnology are based on empirical developments, the ability of ML algorithms to recognize patterns and optimize processes is a valuable tool to support the development of nanomedicines that align with the overall goals of quality-by-design.

In any data science project, ensuring usability for potential users is based on reliable predictive capability, which depends entirely on the quality of the model's training data. Quality means not only the balance between input and output variables but also the accuracy of the observed value. In this way, the analyzed articles in this review showed a big gap for improvement in projects dealing with nanotechnology and data science mainly due to methodological shortcomings. Indeed, only 38% of such studies have addressed appropriate data size, leading to prediction errors and biased results. Additionally, none of them clearly commented on the steps of data analysis, preparation and transformation, or cross-validation and data balancing, which can greatly compromise the predictive ability of the models. Also, no studies commented on the

possibility of overfitting, as they arrived at high metric validation scores despite too few experiments and variables and only a few articles evaluated different model possibilities in order to find the best fit for their data. Finally, no model was actually used and tested by external users with their own data set to prove the prediction results in real problems.

2.4 References

[1] Alzubi J, Nayyar A, Kumar A. Machine learning from theory to algorithms: an overview. J Phys Conf Ser. 2018. p. 12012.

[2] Dhar V. Data science and prediction. Commun ACM. 2013;56:64–73.

[3] Bastogne T. Quality-by-design of nanopharmaceuticals–a state of the art. Nanomedicine. 2017;13:2151–2157.

[4] Jara MO, Catalan-Figueroa J, Landin M, et al. Finding key nanoprecipitation variables for achieving uniform polymeric nanoparticles using neurofuzzy logic technology. Drug Deliv Transl Res. 2018;8:1797–1806.

[5] Youshia J, Ali ME, Lamprecht A. Artificial neural network based particle size prediction of polymeric nanoparticles. European Journal of Pharmaceutics and Biopharmaceutics. 2017;119:333–342.

[6] Asadi H, Rostamizadeh K, Salari D, et al. Preparation of biodegradable nanoparticles of tri-block PLA–PEG–PLA copolymer and determination of factors controlling the particle size using artificial neural network. J Microencapsul. 2011;28:406–416.

[7] Stanzione F, Giangreco I, Cole JC. Chapter four-use of molecular docking computational tools in drug discovery. Prog Med Chem. :273–343.

[8] Zeeshan M, Ali H, Ain QU, et al. A holistic QBD approach to design galactose conjugated PLGA polymer and nanoparticles to catch macrophages during intestinal inflammation. Materials Science and Engineering: C. 2021;126:112183.

[9] Vemula SK, Daravath B, Repka M. Quality by design (QbD) approach to develop fast-dissolving tablets using melt-dispersion paired with surface-adsorption method: formulation and

pharmacokinetics of flurbiprofen melt-dispersion granules. Drug Deliv Transl Res. 2023;13:3204–3222.

[10] de Matas M, Shao Q, Richardson CH, et al. Evaluation of in vitro in vivo correlations for dry powder inhaler delivery using artificial neural networks. European Journal of Pharmaceutical Sciences. 2008;33:80–90.

[11] Peh KK, Lim CP, Quek SS, et al. Use of artificial neural networks to predict drug dissolution profiles and evaluation of network performance using similarity factor. Pharm Res. 2000;17:1384–1389.

[12] Leane MM, Cumming I, Corrigan OI. The use of artificial neural networks for the selection of the most appropriate formulation and processing variables in order to predict the in vitro dissolution of sustained release minitablets. AAPS PharmSciTech. 2003;4:129–140.

[13] Yamashita F, Hashida M. Mechanistic and empirical modeling of skin permeation of drugs. Adv Drug Deliv Rev. 2003;55:1185–1199.

[14] CHEN L, LIAN G, HAN L. Prediction of human skin permeability using artificial neural network (ANN) modeling. Acta Pharmacol Sin. 2007;28:591–600.

[15] Ita K, Roshanaei S. Artificial intelligence for skin permeability prediction: deep learning. J Drug Target. 2024;32:334–346.

[16] Furtuna R, Curteanu S, Racles C. NSGA-II-RJG applied to multi-objective optimization of polymeric nanoparticles synthesis with silicone surfactants. Open Chem. 2011;9:1080–1095.

[17] Zabihi F, Xin N, Jia J, et al. High yield and high loading preparation of curcumin–PLGA nanoparticles using a modified supercritical antisolvent technique. Ind Eng Chem Res. 2014;53:6569–6574.

[18] Yaghoobi N, Majidi RF, ali Faramarzi M, et al. Preparation, optimization and activity evaluation of PLGA/streptokinase nanoparticles using electrospray. Adv Pharm Bull. 2017;7:131.

[19] Jara MO, Landin M, Morales JO. Screening of critical variables in fabricating polycaprolactone nanoparticles using Neuro Fuzzy Logic. Int J Pharm. 2021;601:120558. **

[20] Shalaby KS, Soliman ME, Casettari L, et al. Determination of factors controlling the particle size and entrapment efficiency of noscapine in PEG/PLA nanoparticles using artificial

neural networks. Int J Nanomedicine. 2014;9:4953.

[21] Zaki MR, Varshosaz J, Fathi M. Preparation of agar nanospheres: Comparison of response surface and artificial neural network modeling by a genetic algorithm approach. Carbohydr Polym. 2015;122:314–320. *

[22] Hashad RA, Ishak RAH, Fahmy S, et al. Chitosan-tripolyphosphate nanoparticles: Optimization of formulation parameters for improving process yield at a novel pH using artificial neural networks. Int J Biol Macromol. 2016;86:50–58. *

[23] Esmaeilzadeh-Gharedaghi E, Faramarzi MA, Amini MA, et al. Effects of processing parameters on particle size of ultrasound prepared chitosan nanoparticles: An Artificial Neural Networks Study. Pharm Dev Technol. 2012;17:638–647.

[24] Baharifar H, Amani A. Size, loading efficiency, and cytotoxicity of albumin-loaded chitosan nanoparticles: an artificial neural networks study. J Pharm Sci. 2017;106:411–417.

[25] Kumar M, Kulkarni P, Liu S, et al. Nanoparticle biodistribution coefficients: A quantitative approach for understanding the tissue distribution of nanoparticles. Adv Drug Deliv Rev. 2023;194:114708.

[26] Gaumet M, Vargas A, Gurny R, et al. Nanoparticles for drug delivery: The need for precision in reporting particle size parameters. European Journal of Pharmaceutics and Biopharmaceutics. 2008;69:1–9.

[27] Cho K, Wang X, Nie S, et al. Therapeutic Nanoparticles for Drug Delivery in Cancer. Clinical Cancer Research. 2008;14:1310–1316.

[28] Devarajan P V, Sonavane GS, Doble M. Computer-aided molecular modeling: a predictive approach in the design of nanoparticulate drug delivery system. J Biomed Nanotechnol. 2005;1:375–383. *

[29] Shafique U, Qaiser H. A comparative study of data mining process models (KDD, CRISP-DM and SEMMA). International Journal of Innovation and Scientific Research. 2014;12:217–222.

[30] Martinez-Plumed F, Contreras-Ochando L, Ferri C, et al. CRISP-DM twenty years later: From data mining processes to data science trajectories. IEEE Trans Knowl Data Eng.

2019;33:3048–3061.

[31] Steiner MTA, Soma NY, Shimizu T, et al. Study of a medical problem using kdd, with emphasis on exploratory data analysis. Gestão & Produção. 2006;13:325–337.

[32] Mariñelarena-Dondena L, Errecalde ML, Castro Solano A. Extracción de conocimiento con técnicas de mineria de textos aplicadas a la psicologia. Rev Argent Cienc Comport. 2017;9:0.

[33] Góes ART, Steiner MTA, Neto PJS. Classification of Power Quality Considering Voltage Sags occurred in Feeders. International Conference on Neural Computation Theory and Applications. 2013. p. 433–442.

[34] Firas O. A combination of SEMMA & CRISP-DM models for effectively handling big data using formal concept analysis based knowledge discovery: A data mining approach. World Journal of Advanced Engineering Technology and Sciences. 2023;8:9–14.

[35] Holdaway KR. Data mining methodologies enhance probabilistic well forecasting. SPE Middle East Intelligent Oil and Gas Symposium. 2013. p. D031S014R004.

[36] Tariq HI, Sohail A, Aslam U, et al. Loan default prediction model using sample, explore, modify, model, and assess (SEMMA). J Comput Theor Nanosci. 2019;16:3489–3503.

[37] Azevedo A, Santos MF. KDD, SEMMA and CRISP-DM: a parallel overview. IADS-DM. 2008;

[38] Domingues I, Amorim JP, Abreu PH, et al. Evaluation of Oversampling Data Balancing Techniques in the Context of Ordinal Classification. 2018 International Joint Conference on Neural Networks (IJCNN). IEEE; 2018. p. 1–8.

[39] Delashmit WH, Manry MT, others. Recent developments in multilayer perceptron neural networks. Proceedings of the seventh annual memphis area engineering and science conference, MAESC. 2005. p. 1–15.

[40] Alom MZ, Taha TM, Yakopcic C, et al. A state-of-the-art survey on deep learning theory and architectures. Electronics (Basel). 2019;8:292.

[41] Vabalas A, Gowen E, Poliakoff E, et al. Machine learning algorithm validation with a limited sample size. PLoS One. 2019;14:e0224365.

[42] Kadhim ZS, Abdullah HS, Ghathwan KI. Artificial Neural Network Hyperparameters

Optimization: A Survey. International Journal of Online & Biomedical Engineering. 2022;18.

[43] Pannakkong W, Thiwa-Anont K, Singthong K, et al. Hyperparameter tuning of machine learning algorithms using response surface methodology: a case study of ANN, SVM, and DBN. Math Probl Eng. 2022;2022:1–17.

[44] Lau MM, Lim KH. Review of adaptive activation function in deep neural network. 2018 IEEE-EMBS Conference on Biomedical Engineering and Sciences (IECBES). 2018. p. 686–690.

[45] Apicella A, Donnarumma F, Isgro F, et al. A survey on modern trainable activation functions. Neural Networks. 2021;138:14–32.

[46] Lima AL, Gratieri T, Cunha-Filho M, et al. Polymeric nanocapsules: A review on design and production methods for pharmaceutical purpose. Methods. 2022;199:54–66. *

[47] Pires FQ, Gross IP, Sa-Barreto LL, et al. In-situ formation of nanoparticles from drug-loaded 3D polymeric matrices. European Journal of Pharmaceutical Sciences. 2023;188:106517.

[48] Rocha JL, Pires FQ, Gross IP, et al. Propranolol-loaded nanostructured lipid carriers for topical treatment of infantile hemangioma. J Drug Deliv Sci Technol. 2023;80:104099.

[49] Oliveira ACS, Oliveira PM, Cunha-Filho M, et al. Latanoprost loaded in polymeric nanocapsules for effective topical treatment of alopecia. AAPS PharmSciTech. 2020;21:1–7.

[50] Angelo T, Pires FQ, Gelfuso GM, et al. Development and validation of a selective HPLC-UV method for thymol determination in skin permeation experiments. Journal of Chromatography B. 2016;1022:81–86.

[51] Kasbaum FE, de Carvalho DM, de Jesus Rodrigues L, et al. Development of Lipid Polymer Hybrid Drug Delivery Systems Prepared by Hot-Melt Extrusion. AAPS PharmSciTech. 2023;24:156.

[52] Ferreira SLC, Bruns RE, Ferreira HS, et al. Box-Behnken design: An alternative for the optimization of analytical methods. Anal Chim Acta. 2007;597:179–186.

[53] Bingöl D, Hercan M, Elevli S, et al. Comparison of the results of response surface methodology and artificial neural network for the biosorption of lead using black cumin. Bioresour Technol. 2012;112:111–115.

[54] Samuel OD, Okwu MO. Comparison of Response Surface Methodology (RSM) and

Artificial Neural Network (ANN) in modelling of waste coconut oil ethyl esters production. Energy Sources, Part A: Recovery, Utilization, and Environmental Effects. 2019;41:1049–1061.

[55] Prieto-Martinez FD, Arciniega M, Medina-Franco JL. Acoplamiento molecular: Avances recientes y retos. TIP Revista Especializada en Ciencias Quimico-Biológicas. 2019;21:65–87.

[56] Ghasemi JB, Abdolmaleki A, Shiri F. Molecular docking challenges and limitations. Pharmaceutical sciences: breakthroughs in research and practice. IGI Global; 2017. p. 770–794.

[57] Gomes HM, Awruch AM. Comparison of response surface and neural network with other methods for structural reliability analysis. Structural safety. 2004;26:49–67.

[58] Raschka S, Patterson J, Nolet C. Machine learning in python: Main developments and technology trends in data science, machine learning, and artificial intelligence. Information. 2020;11:193.

[59] Zhou X, Ordonez C. Programming Languages in Data Science: a Comparison from a Database Angle. 2021 IEEE International Conference on Big Data (Big Data). 2021. p. 3147–3154.

[60] Cagnoni S, Cozzini L, Lombardo G, et al. Emotion-based analysis of programming languages on Stack Overflow. ICT Express. 2020;6:238–242.

[61] Ozgur C, Colliau T, Rogers G, et al. MatLab vs. Python vs. R. Journal of data Science. 2017;15:355–371.

[62] RVSPK R, Priyanath HMS, Megama RGN. Methods and rules-of-thumb in the determination of minimum sample size when applying structural equation modelling: a review. J Soc Sci Res. 2020;15:102–109.

[63] Boomsma A. Nonconvergence, improper solutions, and starting values in lisrel maximum likelihood estimation. Psychometrika. 1985;50:229–242.

[64] Kline RB. Methodology in the social sciences. Principles and practice of structural equation modeling. 2005;2.

[65] Nassiri Koopaei N, Abdollahi M. Opportunities and obstacles to the development of nanopharmaceuticals for human use. DARU Journal of Pharmaceutical Sciences. Springer; 2016. p. 1–6.

[66] Awasthi R, Roseblade A, Hansbro PM, et al. Nanoparticles in cancer treatment: opportunities and obstacles. Curr Drug Targets. 2018;19:1696–1709.

[67] Chawla N V, Bowyer KW, Hall LO, et al. SMOTE: synthetic minority over-sampling technique. Journal of artificial intelligence research. 2002;16:321–357.

[68] Mahesh B. Machine learning algorithms-a review. International Journal of Science and Research (IJSR)Internet.. 2020;9:381–386.

3 Multistep machine learning pipeline for polymeric nanoparticle design*

* article published in AAPS PHARMSCITECH, DOI: https://doi.org/10.1208/s12249-025-03248-8

Abstract

Nanoparticles offer therapeutic potential, but production challenges limit their use. Although data science has already been applied to nanotechnology, recent machine learning studies in this field still face methodological and interpretative problems. This study aims to present an efficient pipeline machine learning method for the synthesis of nanoparticles loaded with a model drug. Several algorithms such as Extreme Gradient Boosting (XGBoost), Random Forest, Artificial Neural Networks and Naive Bayes are used to ensure not only that precipitation does not occur, but also that the prediction feature values have the lowest possible error. The final model achieved the targeted values of the validation metrics and the predicted values compared to the experimental ones had errors that were considered perfectly acceptable and, more importantly, much more appropriate than random experiments. The final pipeline result proved to be reliable at low values, such as 75 nm, and at high values, such as 768 nm, passing through intermediate ranges of 200, 300 and 500 nm. In other words, the final pipeline model met the goal of providing a good prediction for recommending future formulations in a wide range of naoparticle sizes.

Keywords: polymeric nanoparticle; data mining; data science; artificial neural network; machine learning; quality by design.

3.1 Introduction

Data science has experienced unprecedented growth, transforming industries by leveraging machine learning (ML) to extract insights from complex datasets and support decision-making processes, particularly in fields as pharmaceutical development [1,2]. The effectiveness of this approach lies in its multidisciplinary nature, combining computer science for data acquisition and processing, statistics for model development, and mathematics to analyze intricate relationships, such as those modeled by artificial neural networks [2,3].

In parallel, nanotechnology has similarly revolutionized multiple domains, enabling innovative advances in materials science, medicine, and drug delivery [4,5,6]. Integrating artificial intelligence techniques into nanotechnology roadmaps may facilitate predictive design and accelerate discoveries in areas such as innovative materials, quantum computing, and environmental diagnostics [7,8,9].

Within this interdisciplinary landscape, the production of nanoparticles has attracted significant interest due to its vast potential in drug delivery applications [10]. Nanoparticles loading active pharmaceutical ingredients present several advantages, including enhanced interaction with biological tissues, improved pharmacokinetics, high stability, controlled drug release, and targeted delivery [11,12,13]. However, the reproducible production of nanoparticles with specific physicochemical properties, such as particle size, zeta potential, and entrapping efficiency, remains a substantial challenge. Traditional production methods often depend on empirical approaches that are both labor-intensive and resource-consuming [14].

Against this backdrop, data-driven methodologies are emerging as powerful tools to support experimental nanotechnology research. By employing ML algorithms, researchers can develop predictive models to achieve a particular nanoparticle characteristic and optimize formulation and processing parameters [15,16,17].

In pharmaceutical development, computational tools are already central to the Quality by Design paradigm [17,18]. Techniques, such as response surface methodologies, are commonly

used to establish design spaces and identify critical quality attributes [19]. However, traditional statistical models, including linear and polynomial regressions, often fall short of capturing the nonlinear, multivariate, and dynamic nature of nanoparticle systems. These limitations become predominantly evident in the later stages of development, where complexity increases, and experimental noise is prevalent [20,21,22,23].

In this scenario, ML models provide a more suitable alternative, especially Deep Learning and ensemble approaches. Their ability to recognize complex, nonlinear patterns in high-dimensional datasets allows for more accurate and robust predictions [24,25]. Furthermore, many ML algorithms include built-in mechanisms for feature selection and hyperparameter optimization, reducing the need for extensive human intervention. When trained with well-curated and balanced datasets, these models can generalize to unseen conditions, significantly enhancing process understanding and development efficiency [26,27].

Despite the theoretical potential, the application of ML in nanotechnology has often been limited by suboptimal methodologies. Many existing studies report high performance using limited datasets and validations closely resembling training conditions, resulting in overfitted models with poor generalization capability [28,29,30]. In complex tasks such as nanoparticle production, a single ML model is unlikely to capture all relevant patterns across the diverse formulation and process space [31].

To address these challenges, we propose a modular, real-world ML pipeline designed for the predictive modeling of drug-loaded polymeric nanoparticle production. This study explores multiple algorithms to accurately predict nanoparticle formation and properties, including Extreme Gradient Boosting (XGBoost), Random Forest, ANNs, Naive Bayes, and Generalized Linear Models (GLMs). Our strategy involves a sequential, multi-model approach, encompassing classification (e.g., nanoparticle formation and size range prediction) and regression tasks (e.g., precise size estimation), thus supporting rational experimental planning and accelerating the design of advanced nanomedicines.

3.2 Experimental

3.2.1 Materials

The model drug isoniazid was obtained from Fluka Analytical (Lot MKBQ8553V, Seelze, Germany). Polysorbate 80 (Tween® 80) was purchased from Dinâmica Química Contemporânea Ltda (São Paulo, Brazil). Eudragit® E100 and Eudragit® L100 were kindly provided by Evonik Industries (Essen, Germany). Poly(vinyl alcohol) (PVA 40-88) was acquired from Merck (Darmstadt, Germany). Methanol (HPLC grade) was purchased from Supelco (Darmstadt, Germany). All other reagents and solvents were of analytical grade and used as received without further purification.

3.2.2 Production of nanoparticles

Isoniazid was selected as the model drug in this study due to its clinical relevance as a first-line agent for tuberculosis, a disease of major global health concern. Despite its high aqueous solubility, isoniazid presents formulation challenges such as short half-life, chemical instability, and limited suitability for controlled-release systems [30]. These features make it a particularly demanding compound for nanoprecipitation, thereby providing a stringent benchmark to evaluate the predictive capability of the proposed ML pipeline.

Nanoparticles were prepared using the nanoprecipitation technique [32,33]. Briefly, the polymer (Eudragit® E100 or Eudragit® L100) and isoniazid were dissolved in methanol under magnetic stirring. Separately, an aqueous phase containing a stabilizer (polyvinyl alcohol or Tween 80) was prepared under continuous stirring. The organic phase was then added into the aqueous phase. Subsequently, the methanol was removed by evaporation under reduced pressure in a rotary evaporator (IKA RV 3, Staufen, Germany). The resulting suspension was immediately analyzed.

All formulations were prepared using a fixed amount of isoniazid (20 mg). The formulation variables included varying concentrations of the polymers Eudragit® L100 and Eudragit® E100 (ranging from 0 to 90 mg), as well as the surfactants Tween® 80 and poly(vinyl

alcohol) (PVA, 40–88) (ranging from 0 to 50 mg). These materials were selected based on their proven biocompatibility and extensive use in developing polymeric nanoparticles for drug delivery applications [34,35,36,37,38]. The ratio between the aqueous and organic phases varied from 5:15 to 15:5 (v/v), maintaining a total preparation volume of 20 mL. Additional experimental variables included the mode of organic phase addition (dripping or direct pouring), the mixing method—either high-shear homogenization using an Ultra-Turrax® (IKA, Staufen, Germany) or conventional magnetic stirring—and the processing temperature (25 °C or 50 °C). The same trained operator prepared all formulations to minimize procedural variability.

3.2.3 Characterization of nanoparticles

Nanoparticles were characterized immediately after production for hydrodynamic diameter, polydispersity index (PDI), and zeta potential. Hydrodynamic diameter and PDI were assessed via dynamic light scattering (DLS) using disposable polystyrene cuvettes with a 1.0 cm optical path length. Zeta potential measurements were performed using electrophoretic light scattering in folded capillary cells (Malvern Instruments, UK). All analyses were conducted with a Zetasizer Nano ZS (Malvern Panalytical, Malvern, UK) at 25 °C. Before measurement, samples were diluted up to fivefold in ultrapure water to minimize multiple scattering and optimize signal quality. Each sample was analyzed in triplicate, with each replicate comprising the average of ten instrument runs. Results were expressed as mean ± standard deviation.

3.2.4 *Algorithms and hyperparameters*

This study employed multiple ML algorithms to model and predict nanoparticle production outcomes, including Random Forest, XGBoost, ANNs with multiple activation functions, Naive Bayes, and GLMs [39]. Each model underwent hyperparameter optimization to enhance performance, using appropriate evaluation metrics as selection criteria.

For the Random Forest algorithm, parameters such as ntrees (number of trees), max_depth (maximum tree depth), sample_rate (sampling rate), mtries (number of features considered for splitting), min_rows (minimum observations per node), and histogram_type

(histogram-based split selection) were adjusted to balance accuracy, model complexity, and generalization capability [40].

In the XGBoost model, we similarly tuned n_trees, max_depth, sample_rate, and min_rows, while also incorporating learn_rate to regulate convergence speed and reduce overfitting, reg_lambda, and reg_alpha for L2 and L1 regularization, respectively, and col_sample_rate to increase diversity among trees. Additionally, cross-validation was performed using the kfold parameter to validate the model's generalization ability [40].

For the GLM, we explored alpha and lambda value combinations to control regularization strength. When alpha was set to 1, the model applied L1 (Lasso) regularization, favoring sparsity by driving some coefficients to zero. Conversely, alpha = 0 enforced L2 (Ridge) regularization, shrinks coefficients without eliminating them [41].

In the Naive Bayes classifier, the alpha parameter was used for Laplace smoothing to mitigate the issue of zero probabilities in unseen categories. At the same time, min_sdev and eps_sdev were adjusted to maintain numerical stability and prevent computational errors [42].

For the ANN models, we implemented L1 and L2 regularization to avoid overfitting, while input_dropout_ratio and hidden_dropout_ratios were used to randomly deactivate neurons during training, thus improving generalization. The learning rate and its decay were controlled through rate and rate_annealing, and training momentum was fine-tuned using momentum_start and momentum_stable. Early stopping was applied using stopping_rounds, stopping_metric, and stopping_tolerance to halt training once improvements plateaued. The number of epochs determined how many complete passes were made through the dataset. Finally, activation functions such as Tanh, ReLU, and MaxOut were applied to introduce nonlinearity and enable the network to capture complex relationships in the data [43].

All models were developed using code in the R programming language (Free Software Foundation, version 4.3.3, Boston, MA, USA) and executed in the RStudio development environment. The H2O package (version 3.44.0.3, Mountain View, CA, USA) for R was utilized, as it natively contains the Random Forest, XGBoost, ANNs, Naive Bayes, and GLM algorithms, as well as the model validation metrics. The best hyperparameters were selected using the "grid"

function from the H2O package, in which multiple values are tested across various models until the most optimized values for the metrics are achieved.

3.2.5 Pipeline for ML models

The proposed pipeline for polymeric nanoparticles production is structured around the sequential development of ML models designed to guide and refine experimental design. As illustrated in Figure 1, the workflow consists of three main stages. In Step 1, a binary classification algorithm is employed to predict the likelihood of nanoparticle formation under given formulation conditions. Step 2 involves a multiclassification model, categorizing the resulting nanoparticles into predefined size ranges. Finally, Step 3 utilizes a rule-based regression model to estimate the expected particle size within each size class.

This pipeline is iterative: Steps 1 and 2 are repeated across successive rounds of experimentation. With each new round, additional experimental data generated from newly designed formulations informed by the provisional models are incorporated to retrain the algorithms. This cycle continues until the multiclassification model in Step 2 achieves a coefficient of determination (R^2) greater than 0.9, indicating satisfactory predictive performance and enabling progression to the regression stage (Step 3).

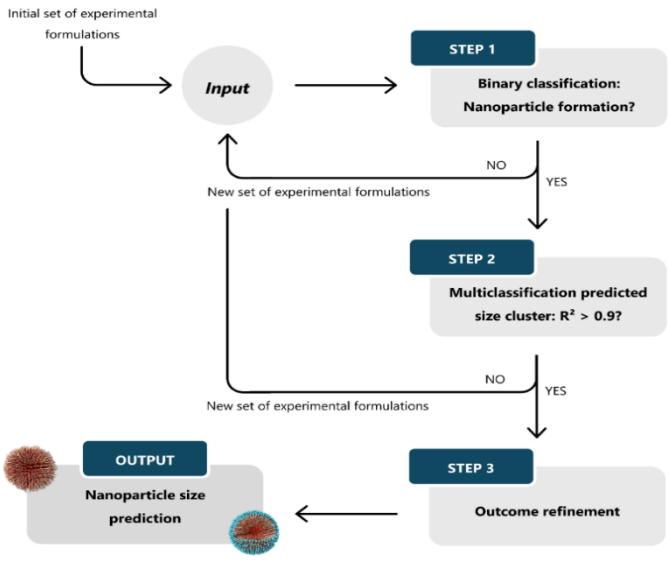


Figure 1. Machine learning pipeline based on 3 steps. Step 1 with a binary classification algorithm; Step 2 with a multiclassification algorithm; and Step 3 with a regression algorithm to predict the expected size of the nanoparticles.

3.2.6 Step 1: Binary classification model

The first step of the ML pipeline focuses on predicting the occurrence of nanoparticle formation, which depends on the convergence of multiple formulation and process parameters. This is achieved by constructing a binary classification model in which the output variable indicates whether the formulation leads to nanoparticle precipitation.

Several algorithms were trained and evaluated using K-fold cross-validation with K values of 3 or 5 to develop this model. Data resampling techniques were employed to equalize the number of positive and negative outcomes to address class imbalance. Additionally, exploratory correlation analyses were conducted to assess potential multicollinearity among input variables and to refine feature selection when needed. Correlation analysis is essential when building linear models, such as GLMs and neural networks, as multicollinearity among predictors can destabilize coefficient estimates and impair the interpretation of individual variable contributions [44].

The performance of each classifier was assessed based on accuracy and precision metrics. Emphasis was placed on precision, as false positives (i.e., incorrectly predicting nanoparticle formation) are more detrimental than false negatives in this context. A false positive could result in unnecessary experimental effort and material loss, whereas a false negative merely omits a potentially viable formulation from further testing [45].

To increase model reliability, the final prediction of nanoparticle formation was only considered valid when the two best-performing algorithms (in terms of accuracy) returned concordant predictions. Formulations with consistent positive predictions were then advanced to Step 2. If the number of formulations meeting this criterion was insufficient, an additional round of experimental data collection was conducted to expand and improve the training dataset (Figure 1).

3.2.7 Step 2: Multiclassification model

One of the major challenges in developing ML models for nanoparticle systems is the limited availability of high-quality experimental data. Nanoparticle production is both costly and labor-intensive, even at a laboratory scale [46]. Accordingly, a core objective of this pipeline is to propose a robust and realistic protocol that can be implemented with a feasible number of experiments while still delivering accurate and generalizable predictions.

In this context, Step 2 of the pipeline involves the construction of a multiclassification model to categorize particle sizes into defined intervals. This strategy addresses data scarcity more effectively than direct regression, especially in the early stages of experimentation. Classification models are generally more robust to experimental variability and noise, allowing for reliable categorization even when individual data points deviate slightly from expected trends.

After filtering the formulations identified as capable of forming stable colloidal dispersions in Step 1, these are forwarded to Step 2. This step incorporates K-fold cross-validation, correlation analysis, and feature grouping to enhance predictive performance. The same algorithms were employed, except for Naive Bayes, which is typically more suitable for binary classification tasks.

Histogram-based clustering was performed to define particle size intervals, grouping continuous size values into discrete classes based on their frequency distribution. This approach allows the model, particularly tree-based algorithms, to better capture the underlying structure of the data, enhancing classification accuracy. Additionally, this method reduces dimensionality, mitigates the influence of noise, and improves computational efficiency by simplifying the decision space at each node.

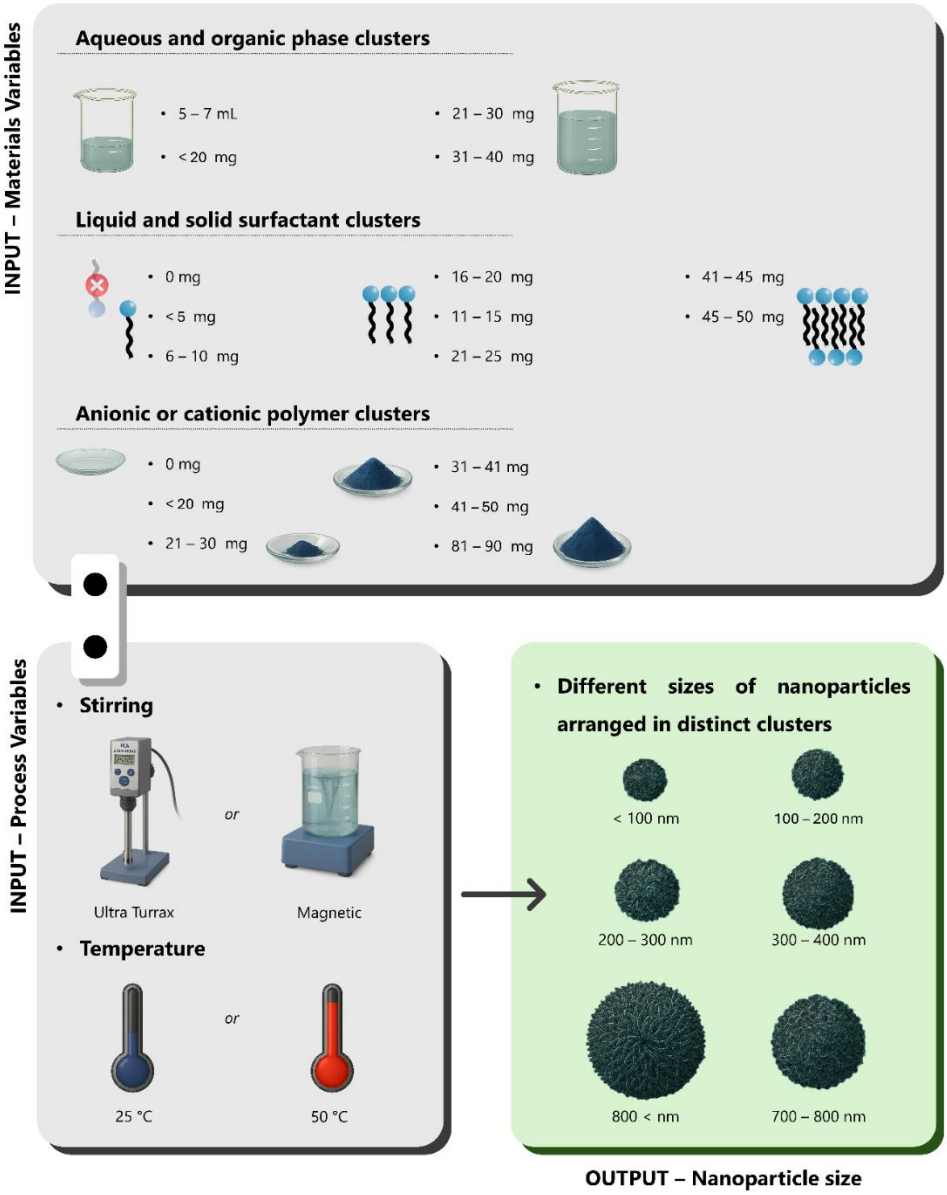


Figure 2. Cluster of size created by histogram analysis of the production and formulation variables used to synthesize isoniazid polymeric nanoparticles.

All algorithms in this step were validated using Logarithmic Loss (LogLoss) and the R^2 as complementary performance metrics [45]. LogLoss evaluates the quality of the model's probabilistic predictions, penalizing both incorrect classifications and overconfident predictions. The R^2 value in multiclassification is computed by assigning numerical representations to the classes and calculating the proportion of variance in these values that the model can explain. This metric reflects the model's ability to capture and represent variability in the classification outcomes [47,48]. Combining these metrics offers a robust framework for evaluating complex

multiclassification models, balancing precision and reliability [45,49].

According to the proposed protocol, the model is considered sufficiently accurate for progression only when it achieves an R^2 value greater than 0.9. Until this criterion is met, iterative rounds of experimentation are required, repeating Steps 1 and 2 to refine the dataset and improve model generalization (Figure 1).

A synthetic dataset was generated by simulating all plausible combinations of formulation variables within their predefined boundaries to support the planning of new experiments. This approach allowed exploration of the entire input space and enabled the identification of underrepresented or ambiguous regions in the model. In the case study, after the initial 30 experimental formulations, the simulation yielded 42,643 virtual formulations, which were used to guide the selection of the next experimental batch.

The selection process prioritized formulations that filled gaps in underrepresented size classes, thus promoting data balance and model robustness. For classification, a consensus voting mechanism was employed among the three best-performing algorithms selected from Random Forest, Linear Regression, Naive Bayes, XGBoost, and Artificial Neural Networks. A predicted size class was only accepted if at least two of the three models agreed, ensuring higher confidence in classification outcomes. This ensemble-like strategy mirrors the internal logic of Random Forest voting and contributes to greater prediction stability.

3.2.8 Step 3: Outcome refinement

In this final phase of the pipeline, the predictive model transitions from classification to regression to estimate the exact particle size of the synthesized nanoparticles. Unlike Step 2, which operates on grouped data, Step 3 uses the original, ungrouped experimental dataset to preserve the full resolution of the continuous outcome variable. This approach avoids the sparsity and bias introduced by discretization, particularly in cases where certain size ranges are underrepresented or missing, which are common issues in small experimental datasets [50].

Again, cross-validation and correlation analyses were applied to evaluate input variable consistency and mitigate overfitting. The regression models were assessed using three complementary performance metrics: the R^2, mean absolute error (MAE), and root mean square error (RMSE) [45]. MAE provides a robust measure of average prediction error and is less sensitive to outliers, while RMSE penalizes larger deviations more heavily, highlighting models prone to occasional significant errors. Together, these metrics offer a comprehensive view of the model's accuracy and stability in predicting continuous outcomes.

The algorithms employed in this step included Random Forest, Linear Regression, XGBoost, and ANNs. The Naive Bayes algorithm was excluded due to its inapplicability to regression problems. Importantly, the size range predicted in Step 2 was used as a reference constraint in a conflict between the regression output and the previous classification result. In such cases, preference was given to the output of Step 2, as classification models trained on grouped data typically exhibit greater robustness in low-data scenarios. When a mismatch occurred, the mean value of the predicted size range from Step 2 was adopted as the final predicted value for that formulation. Finally, 9 new formulations were simulated after training Steps 1, 2, and 3 of the final ML pipeline for the final test validation.

3.3 Results and Discussion

3.3.1 First experimental round

Initially, 30 distinct formulations were prepared in the laboratory, labeled as "Id 1" to "Id 30". Ideally, data science approaches benefit from large datasets that comprehensively represent all strata of the target population to improve generalizability. However, given the vast combinatorial space of nanoparticle formulations, this assumption is not feasible.

Considering the selected preparation method of nanoprecipitation and the diversity of input variables evaluated (input dimension = 7), the initial screening of 30 formulations was chosen. These formulations were designed with randomly varied inputs to feed the first predictive models in Step 1 of the ML pipeline. The goal was to assess model behavior in classifying nanoparticle formation and subsequently use these preliminary models to guide formulation

planning in future rounds. Emphasis was placed on generating balanced data with respect to the output variable nanoparticle size.

Among the initial set of 30 formulations, 14 successfully yielded nanoparticles, while 16 failed, resulting in a reasonably balanced binary dataset. The failure in nanosystem formation was indicated by the appearance of solid deposits either during or after the preparation procedure. The precipitation may reflect various underlying issues: nanoparticle aggregation, chemical incompatibilities in the composition, or insufficient solubility of the polymer or drug [15,16]. Precipitation was most frequently observed during the solvent removal step via rotary evaporation (62.5%), followed by the phase addition step (18.8%), as illustrated in Figure 3. Methanol elimination proved to be a critical step. The abrupt shift in solubility during solvent removal likely accelerated nucleation and uncontrolled particle growth, resulting in precipitation.

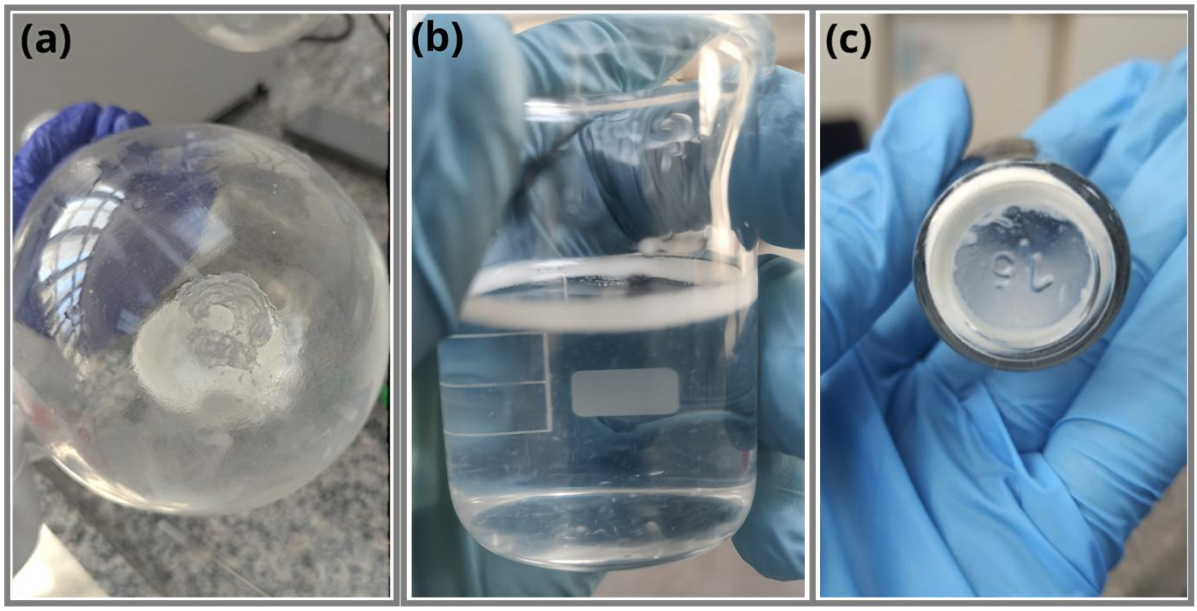


Figure 3. Representative images of nanoparticle precipitation observed at different stages of the production process: (a) during rotary evaporation (Id 44); (b) during the dripping of the organic phase into the aqueous phase (Id 61); and (c) after the production process (Id 74).

The particle size distribution obtained from these 14 successful formulations showed a marked imbalance, particularly a data gap between 300 nm and 800 nm. This lack of data within an important size interval emphasizes the need for targeted formulations in subsequent rounds to ensure adequate representation across all classes.

Still, within the first modeling round, a feature correlation analysis was conducted to

eliminate highly collinear variables. This step aimed to reduce model dimensionality and improve generalization. In the isoniazid case study, strong correlations were observed between the aqueous and organic phase volumes, as well as between the quantities of solid and liquid surfactants.

Based on the refined dataset, binary classification models (Step 1) were trained to predict nanoparticle formation. Multiple algorithm–hyperparameter combinations were tested to optimize accuracy. Deep Learning (accuracy = 0.64) and Random Forest (accuracy = 0.56) were the best-performing models.

In Step 2, which focused on predicting particle size classes, all models yielded R^2 values below 0.8, likely due to the limited number of successful formulations (n = 14). As anticipated, an additional experimental round was necessary to enhance data diversity and model performance.

3.3.2 Second experimental round

Although the experimental dataset was still limited, the provisional models developed in Steps 1 and 2 were employed to simulate and guide a second round of experiments. This strategy was considered more robust than the random selection of new formulations.

Accordingly, the two most accurate algorithms from Step 1 were selected: Deep Learning predicted 40,209 formulations as likely to succeed, while Random Forest identified 40,697 successes. The intersection of both predictions yielded 38,872 formulations consistently classified as successful candidates. These were subsequently submitted to Step 2 for particle size classification.

From this pool, 30 formulations predicted to result in particle sizes between 300 and 800 nm were randomly selected to address the data imbalance observed in the First Round. As anticipated, not all selected formulations achieved the predicted target sizes, reflecting the inherent uncertainty of ML models trained on small datasets (Table 2). Nevertheless, this approach effectively guided the experimental design to target underrepresented regions in the output space. Most new formulations successfully fell within the intended size range, which is especially notable given that this interval was sparsely represented in the initial training set.

Table 1 reveals that Step 2 correctly predicted the particle size class for 14 of 30 new formulations (47%). Notably, the voting strategy among top-performing models proved beneficial. For instance, while the deep learning algorithm predicted a 500–600 nm range for formulation ID 43, the other two models predicted 700–800 nm, corresponding to the experimentally validated result. This ensemble approach improved classification confidence in ambiguous cases.

Table 1. Nanoparticle size as the output variable from the ML pipeline in Step 2, along with the corresponding experimental results obtained in the 2nd round. Predicted nanoparticle size (nm) using three machine learning models (Random Forest, XGBoost, and ANN) in 2nd round, compared with available experimental data. The final predicted range corresponds to the most frequent size range predicted across the three models. The result evaluation indicates whether the formulation was considered successful. n.a. = not accessed.

ID	Nanoparticle size (nm)					Result evaluation
	Experimental	Random Forest	XGBoost	ANN	Predicted	
31	n.a.	300-400	300-400	300-400	300-400	FAIL
32	n.a.	400-500	400-500	300-400	400-500	FAIL
33	n.a.	400-500	400-500	400-500	400-500	FAIL
34	n.a.	800<	800<	600-700	800<	FAIL
35	200-300	200-300	200-300	200-300	200-300	SUCCESS
36	300-400	< 100	300-400	300-400	300-400	SUCCESS
37	100-200	300-400	300-400	300-400	300-400	FAIL
38	< 100	300-400	300-400	300-400	300-400	FAIL
40	800<	700-800	800<	800<	800<	SUCCESS
41	700-800	800<	700-800	700-800	700-800	SUCCESS
42	n.a.	800<	800<	800<	800<	FAIL
43	700-800	700-800	700-800	500-600	700-800	SUCCESS
44	< 100	300-400	300-400	300-400	300-400	FAIL

45	n.a.	400-500	400-500	400-500	400-500	FAIL
46	n.a.	400-500	400-500	400-500	400-500	FAIL
47	500-600	500-600	500-600	500-600	500-600	SUCCESS
48	500-600	800<	500-600	500-600	500-600	SUCCESS
49	600-700	600-700	600-700	600-700	600-700	SUCCESS
50	100-200	500-600	300-400	300-400	300-400	FAIL
51	400-500	400-500	400-500	400-500	400-500	SUCCESS
52	800<	800<	800<	800<	800<	SUCCESS
53	200-300	200-300	200-300	200-300	200-300	SUCCESS
54	300-400	300-400	300-400	500-600	300-400	SUCCESS
55	n.a.	800<	500-600	500-600	500-600	FAIL
56	200-300	800<	200-300	200-300	200-300	SUCCESS
57	200-300	800<	200-300	200-300	200-300	SUCCESS
58	< 100	200-300	200-300	200-300	200-300	FAIL
59	100-200	200-300	200-300	200-300	200-300	FAIL
60	n.a.	200-300	200-300	200-300	200-300	FAIL

Following this round, the new data (IDs 31 to 60) were integrated with the original dataset (IDs 1 to 30), resulting in 60 formulations. With this expanded dataset, both Step 1 (binary classification) and Step 2 (multiclassification) models were retrained to improve predictive performance. As expected, including new formulations led to substantial gains in accuracy and precision for both algorithms.

Unlike the first round, where the multiclassifier model relied solely on 14 successful formulations, the updated model was now trained on 35 positive cases: 14 from the initial round and 21 from the second round. As a result, improved R^2 values above 0.8 were achieved. However, the predefined threshold of $R^2 > 0.9$ was not yet met, justifying the need for an additional experimental cycle (third round) to further enhance model generalization.

3.3.3 Third experimental round

A new set of 30 simulated-aided formulations was selected for the third experimental round. This round's dual purpose was to further reduce class imbalance and evaluate the models' predictive capacity across the entire target particle size spectrum, ranging from 100 nm to 700 nm.

The predictive performance of the ML pipeline improved notably in this cycle (Table 2). In Step 1 (binary classification), 80% of the predictions were accurate, as only six out of thirty formulations failed to yield nanoparticles. In Step 2 (multiclassification), the overall accuracy reached 70%. When analyzing only the successful formulations (i.e., those in which nanoparticles were actually formed), the prediction accuracy rose to 83.3%.

These results reinforce the iterative pipeline's effectiveness: model performance improved accordingly as the training dataset grew and became more balanced. Moreover, the observed accuracy rates, especially in complex, multivariate classification tasks, demonstrate the practical utility of the simulation-aided strategy for guiding nanoparticle formulation [51].

Table. 2. Nanoparticle size as the output variable from the ML pipeline in Step 2, along with the corresponding experimental results obtained in the second round. Predicted nanoparticle size (nm) using three machine learning models (Random Forest, XGBoost, and ANN) in 3rd round, compared with available experimental data. The final predicted range corresponds to the most frequent size range predicted across the three models. The result evaluation indicates whether the formulation was considered successful. n.a. = not accessed.

ID	Nanoparticle size (nm)					Result evaluation
	Experimental	Random Forest	XGBoost	ANN	Predicted	
31	n.a.	300-400	300-400	300-400	300-400	FAIL
32	n.a.	400-500	400-500	300-400	400-500	FAIL
33	n.a.	400-500	400-500	400-500	400-500	FAIL
34	n.a.	800<	800<	600-700	800<	FAIL
35	200-300	200-300	200-300	200-300	200-300	SUCCESS
36	300-400	< 100	300-400	300-400	300-400	SUCCESS
37	100-200	300-400	300-400	300-400	300-400	FAIL
38	< 100	300-400	300-400	300-400	300-400	FAIL
40	800<	700-800	800<	800<	800<	SUCCESS
41	700-800	800<	700-800	700-800	700-800	SUCCESS
42	n.a.	800<	800<	800<	800<	FAIL
43	700-800	700-800	700-800	500-600	700-800	SUCCESS
44	< 100	300-400	300-400	300-400	300-400	FAIL
45	n.a.	400-500	400-500	400-500	400-500	FAIL
46	n.a.	400-500	400-500	400-500	400-500	FAIL
47	500-600	500-600	500-600	500-600	500-600	SUCCESS
48	500-600	800<	500-600	500-600	500-600	SUCCESS
49	600-700	600-700	600-700	600-700	600-700	SUCCESS
50	100-200	500-600	300-400	300-400	300-400	FAIL
51	400-500	400-500	400-500	400-500	400-500	SUCCESS
52	800<	800<	800<	800<	800<	SUCCESS
53	200-300	200-300	200-300	200-300	200-300	SUCCESS
54	300-400	300-400	300-400	500-600	300-400	SUCCESS

55	n.a.	800<	500-600	500-600	500-600	FAIL
56	200-300	800<	200-300	200-300	200-300	SUCCESS
57	200-300	800<	200-300	200-300	200-300	SUCCESS
58	< 100	200-300	200-300	200-300	200-300	FAIL
59	100-200	200-300	200-300	200-300	200-300	FAIL
60	n.a.	200-300	200-300	200-300	200-300	FAIL

Following the completion of Round 3, the ML algorithms for Steps 1 and 2 were retrained with the expanded dataset. Step 1 models were now trained using data from 90 formulations (IDs 1 to 90), while Step 2 models incorporated all 59 cases in which nanoparticles were successfully formed.

As illustrated in Figure 4, this iteration led to substantial improvements in model performance. Validation metrics showed clear gains over previous rounds, reflecting the benefits of a more diverse and representative training set. Notably, the coefficient of determination (R^2) surpassed the predefined threshold of 0.9 in three algorithms, satisfying the criteria for progression to Step 3, the regression phase of the pipeline.

This result confirms the effectiveness of the iterative strategy adopted, which relied on simulation-aided formulation design, model consensus voting, and continuous retraining with real experimental data [52]. With reliable classification models now in place, the pipeline is ready to generate precise particle size predictions through regression modeling, completing the data-driven framework for rational nanoparticle formulation.

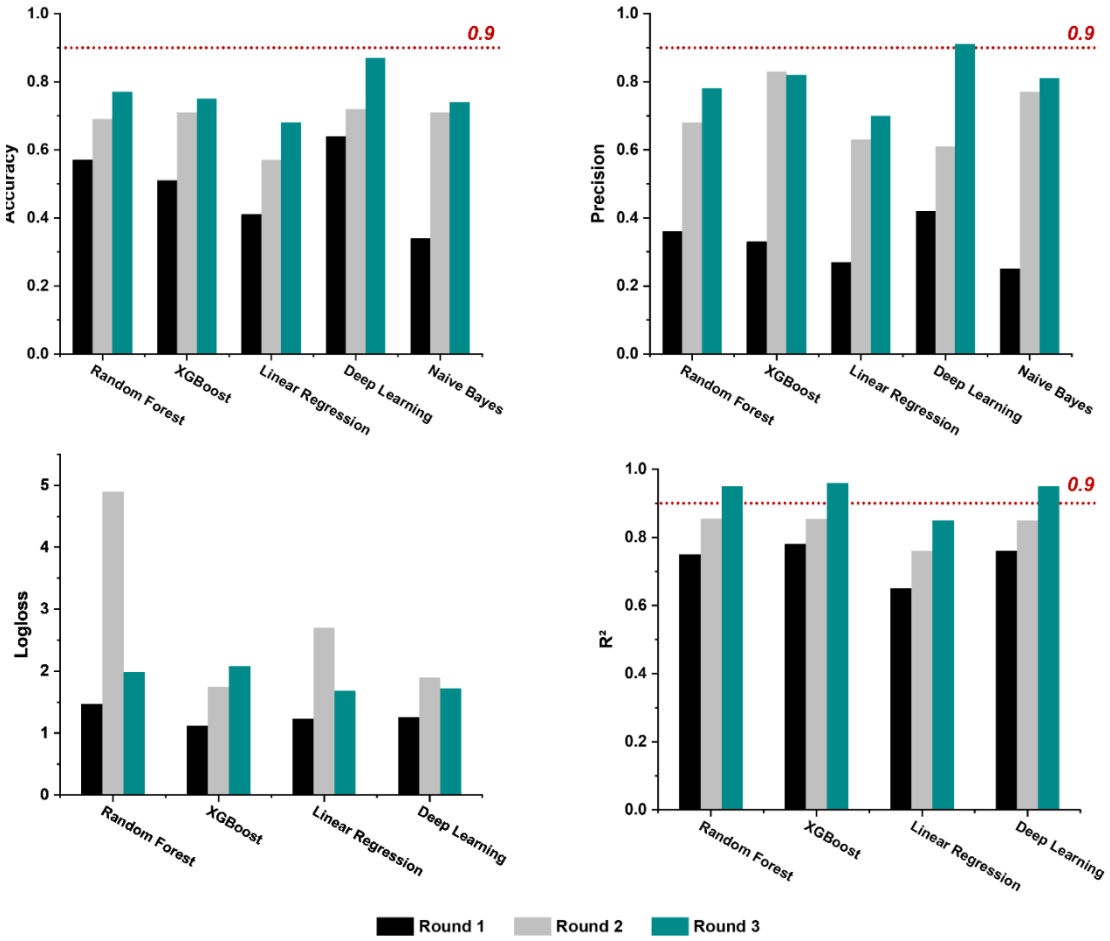


Figure 4. Improvement of classification model performance across experimental rounds. Metrics such as accuracy, precision, LogLoss, and R^2 were tracked for each machine learning algorithm in Step 1 (binary classification) and Step 2 (multiclassification) of the pipeline.

3.3.4 Data distribution and model learning progress

At the beginning of the study, 30 formulations were produced using randomly varied input parameters, generating nanoparticles with sizes ranging from <100 nm to >800 nm. Although this diversity allowed the initial model to identify key patterns in nanoparticle formation, a pronounced gap remained in the 300–800 nm range (Figure 5). To overcome this limitation, additional formulations were specifically designed.

Despite being trained on only 30 experimental data points, the first model successfully captured relevant trends. Its predictive capacity was then expanded through synthetic data generated with an R algorithm, which systematically explored all possible combinations of phase volumes, surfactants, polymers, stirring and dripping methods, and processing temperatures. This

procedure yielded 42,643 synthetic formulations that were incorporated into the Step 1 model, enabling predictions across the entire 100–800 nm range. Based on these predictions, 30 candidate formulations were selected, synthesized, and experimentally validated to fill the critical 300–800 nm gap (Table 1). The same iterative strategy was applied in Round 2, ultimately producing nanoparticles across the full size spectrum and establishing the dataset necessary to train the regression model in Step 3.

Figure 5 illustrates the progressive balancing of nanoparticle size distribution. After Round 1 (black), a clear underrepresentation was observed in the 300–800 nm interval. This gap was substantially reduced in Round 2 (gray) with the inclusion of simulation-guided formulations, leading to a more uniform distribution. Round 3 (green) further refined the dataset, extending coverage to both larger and underrepresented smaller size ranges. This systematic rebalancing was crucial for improving the accuracy and generalizability of the classification models in Steps 1 and 2, and ultimately for enabling the reliable transition to regression modeling in Step 3.

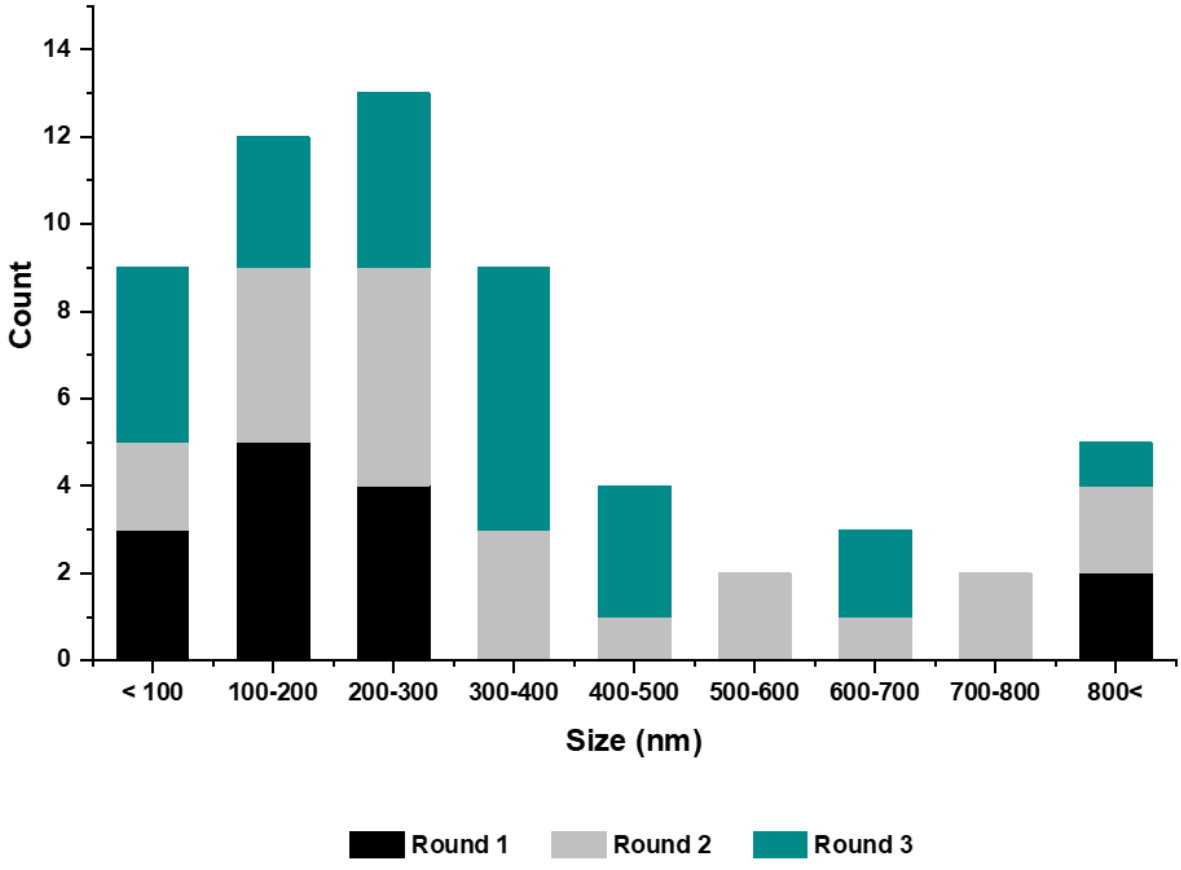


Figure 5. Size distribution in 3 rounds of experimental formulations for isoniazid nanoparticles by nanoprecipitation.

3.3.5 Algorithm performance evolution across rounds

Figure 6 presents the progressive improvement in model performance for Step 1 (binary classification) as the number of experimental formulations increased from 30 to 90. Deep Learning emerged as the most accurate algorithm, while the GLM consistently delivered the lowest performance. These results are explained by the intrinsic characteristics of each model and the complexity of the relationships present in the data. The success of nanoparticle formation is governed by nonlinear interactions among multiple formulation and process variables. GLMs, by design, assume a linear relationship between input features and the outcome, even when link functions are applied. This limitation reduces their capacity to capture the intricate patterns necessary for accurate classification, which explains their modest improvement despite a larger dataset [53].

In contrast, Deep Learning models, which are highly nonlinear and capable of modeling complex feature interactions, benefited substantially from the increased volume of training data. While neural networks typically suffer from high variance when trained on small datasets, adding more observations helped the model converge to more generalizable patterns, leading to significant improvements in accuracy and precision. Random Forest and XGBoost, which also capture nonlinear dependencies, likewise showed enhanced performance with additional data. However, the more flexible structure of Deep Learning appears to extract greater benefit from the expanded information, likely due to its higher modeling capacity [40].

From a numerical standpoint, Random Forest's accuracy increased from 0.568 in the first round to 0.770 in the 3rd, while XGBoost improved from 0.510 to 0.750. GLM rose from 0.400 to 0.680, Naive Bayes from 0.340 to 0.740, and Deep Learning advanced from 0.640 to 0.870. These findings confirm that more complex models, such as neural

networks, continue to improve as more data becomes available, whereas simpler models tend to plateau.

Figure 6 also includes the evolution of LogLoss and R^2 values for Step 2 (multiclassification). LogLoss, which penalizes incorrect and overconfident predictions, is expected to decrease as model calibration improves. Interestingly, some algorithms showed an initial increase in LogLoss during Round 2, likely reflecting poor generalization from models trained on the limited dataset of Round 1. This phenomenon is consistent with a large data gap in the 300–800 nm range at the outset, making it difficult for early models to assign accurate probabilities to those underrepresented classes.

The subsequent addition of 21 new formulations in Rounds 2 and 3 filled this critical gap, providing the necessary examples for the models to better learn and generalize, which led to a decrease in LogLoss by Round 3. In parallel, all algorithms exhibited an increase in the coefficient of determination (R^2), even though this metric is traditionally associated with regression. R^2 reflects the model's ability to explain variance in predicted outcomes in classification settings using numerical class encodings. The improved R^2 values suggest that the models developed a better structural understanding of the data, even if some probabilistic predictions remained imprecise. Ultimately, these findings reinforce the importance of employing multiple complementary metrics, such as accuracy, LogLoss, and R^2, to comprehensively evaluate model performance and generalizability [45].

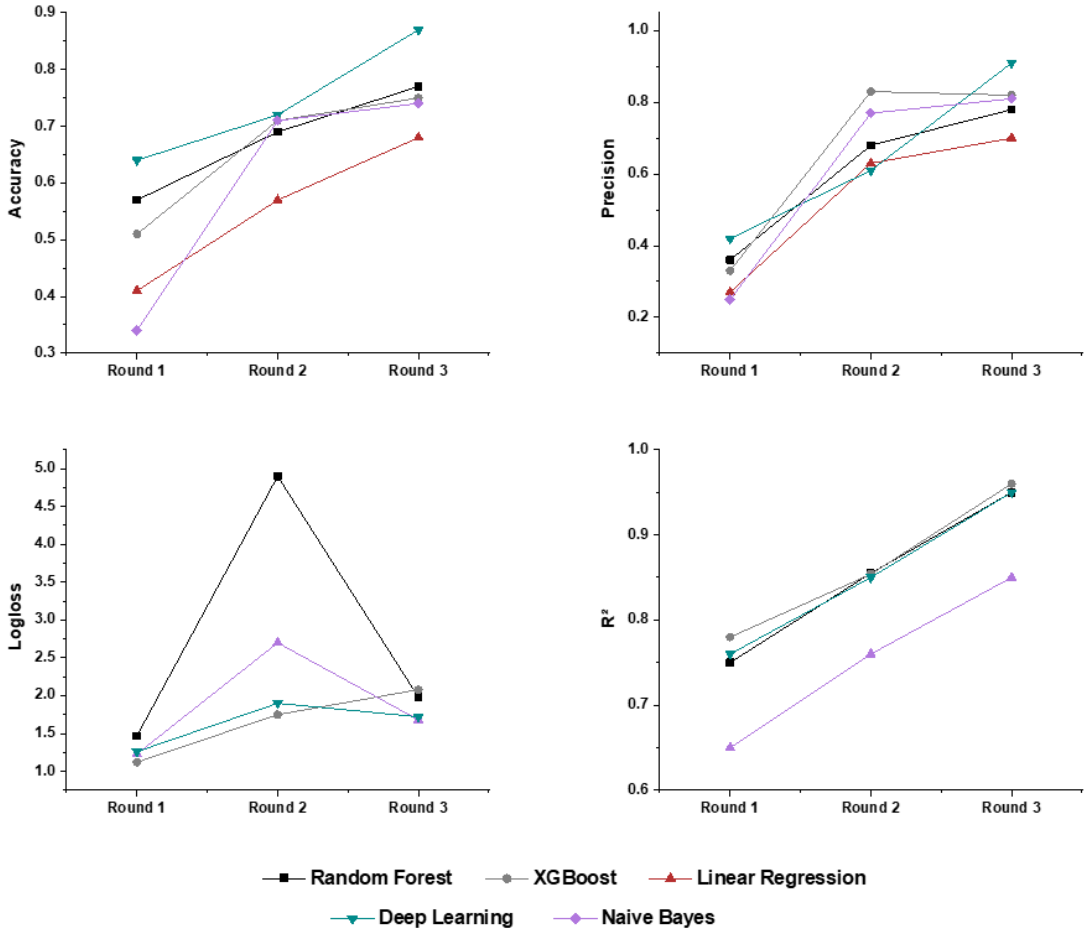


Figure 6. Evolution of classification performance metrics—accuracy, precision, LogLoss, and R^2—for five machine learning algorithms across three experimental rounds.

3.3.6 Step 3: Outcome refinement

The final step of the ML pipeline (Step 3) consists of a regression model designed to refine the multiclass predictions from Step 2 by estimating the exact numerical value of nanoparticle size. As with previous stages, multiple combinations of algorithm hyperparameters were tested to identify the model configuration with the highest predictive accuracy. This step was trained exclusively on the 59 successful formulations, ensuring the model learned from reliable and representative data.

Among the evaluated algorithms, the Deep Learning model yielded the best overall performance based on R^2, RMSE, and MAE values. Consequently, the predictions generated by

this model were adopted as the final regression outputs.

Since Step 2 operates on size intervals and demonstrated higher confidence due to a broader learning scope, its classification output was used as a benchmark to validate the regression predictions. If the predicted value from Step 3 fell outside the range indicated by Step 2, the final output was adjusted to the mean of the predicted size class. This conservative approach ensured consistency and leveraged the strength of classification robustness.

Validation of the complete ML pipeline using a set of new formulations is presented in Table 3. Notably, the corresponding experimental formulations failed in cases where the regression output deviated from the classification range, suggesting that the classification bounds served as a reliable quality filter. When comparing predicted versus actual particle sizes, the largest observed absolute error occurred at a true size of 75.27 nm, with an expected value of 113.99 nm. This deviation of approximately 39 nm is generally acceptable in most nanoparticle design applications, especially considering the inherent experimental variability in nanoparticulate systems.

The final regression model demonstrated high accuracy across the full spectrum of sizes, including lower values (e.g., 75 nm), higher ranges (e.g., 768 nm), and intermediate zones around 200, 300, and 500 nm. These findings confirm that the developed pipeline reliably predicts nanoparticle size across a broad and practically relevant range, offering a powerful tool for guiding future formulation development.

Table 3. Final predictions. The first column shows the value of nanoparticle size measured in the laboratory. The next 3 columns indicate the classes predicted in step 2 (Random Forest, XGBoost and Deep Learning, respectively), while the fourth column represents the value predicted by the Deep Learning regression algorithm of step 3. Column 5 represents the final result of the pipeline and indicates the regression value predicted in step 3 if it lies between the range predicted in step 2.

Experimental outcome	Nanoparticle size (nm)					Error (%)
	Random Forest	XGBoost	ANN	Regression (DeepL)	Predicted outcome	
768.12	600-700	600-700	700-800	603.10	603.11	21,48

313.21	200-300	200-300	200-300	282.15	282.15	9,91
75.27	100-200	100-200	100-200	113.99	113.99	52
504.12	400-500	200-300	400-500	424.92	424.92	15,8
preciptates	300-400	300-400	300-400	454.84	350	--
289.48	200-300	200-300	200-300	208.94	208.94	28,02
388.93	200-300	300-400	300-400	354.84	354.84	8,76
preciptates	200-300	200-300	200-300	140.01	250	--
268.17	200-300	200-300	200-300	218.01	218.01	18,65

The final pipeline yielded robust and accurate predictions across a wide size range with acceptable error margins and strong generalization capability. The Deep Learning algorithm emerged as the most effective across all tasks, benefiting significantly from increased training data. These findings confirm the utility of data-driven approaches not only for improving experimental efficiency but also for enabling rational design in nanoparticle-based drug delivery systems.

Importantly, the insights gained with isoniazid extend beyond this compound alone. The methodology established here can be extrapolated to other active pharmaceutical ingredients, including those with similarly unfavorable physicochemical properties. In this sense, isoniazid serves as a representative case study supporting the broader applicability of data-driven design approaches to diverse nanomedicine development scenarios.

3.4 Conclusions

This study demonstrated the successful development and validation of a modular ML pipeline for the predictive modeling of nanoparticle production using isoniazid-loaded polymeric nanoparticles as a case study. The pipeline effectively learned complex, nonlinear relationships between formulation parameters and nanoparticle size by integrating classification and regression

models across three iterative steps. The strategy of using classification outputs to guide regression, combined with ensemble voting and synthetic data simulation, proved effective in overcoming data scarcity and distribution imbalance, common challenges in early-stage nanomedicine development. These findings confirm the utility of data-driven approaches not only for improving experimental efficiency but also for enabling rational design in nanoparticle-based drug delivery systems. The proposed methodology provides a scalable and adaptable framework that can be applied to other nanotechnological systems, accelerating formulation development and supporting Quality by Design initiatives in pharmaceutical research.

3.5 References

[1] Suriyaamporn P, Pamornpathomkul B, Patrojanasophon P, et al. The artificial intelligence-powered new era in pharmaceutical research and development: a review. AAPS PharmSciTech.2024;25:188.

[2] Vidiyala N, Sunkishala P, Parupathi P, et al. The role of artificial intelligence in drug discovery and pharmaceutical development: a paradigm shift in the history of pharmaceutical industries. AAPS PharmSciTech. 2025;26:133.

[3] Fountzilas E, Pearce T, Baysal MA, et al. Convergence of evolving artificial intelligence and machine learning techniques in precision oncology. NPJ Digit Med. 2025;8:75.

[4] Taha TB, Barzinjy AA, Hussain FHS, Nurtayeva T. Nanotechnology and computer science: trends and advances. Memories. 2022;2:100011.

[5] Benderski K, Lammers T, Sofias AM. Analysis of multi-drug cancer nanomedicine. Nat Nanotechnol. 2025. https://doi.org/10.1038/s41565-025-01932-1.

[6] Ma X, Tian Y, Yang R, et al. Nanotechnology in healthcare, and its safety and environmental risks. J Nanobiotechnol. 2024;22:715.

[https://doi.org/10.1186/s12951-024-02901-x.

[7] Olawade DB, Ige AO, Olaremu AG, Ijiwade JO, Adeola AO. The synergy of artificial intelligence and nanotechnology towards advancing innovation and sustainability: a mini-review. Nano Trends. 2024;8:100052

[8] Metwally AA, Nayel AA, Hathout RM. In silico prediction of siRNA ionizable-lipid nanoparticles in vivo efficacy: machine learning modeling based on formulation and molecular descriptors. Front Mol Biosci. 2022;9: 1042720.

[9] Bastogne T. Quality-by-design of nanopharmaceuticals—a state of the art. Nanomedicine. 2017;13(7):2151–7.10. Lima AL, Gratieri T, Cunha-Filho M, Gelfuso GM. Polymeric nanocapsules: a review on design and production methods for pharmaceutical purpose. Methods. 2022;199:54–66.

[11] Gross IP, Lima AL, Sousa EC, Souza MS, Cunha-Filho M, Silva ICRD, et al. Antimicrobial and acaricide sanitizer tablets produced by wet granulation of spray-dried soap and clove oil-loaded microemulsion. PLoS ONE. 2024;19(11):e0313517.

[12] Mengarda AC, Iles B, Rodrigues VC, Lima AL, Machado VP, Reatgui WS, et al. Praziquantel nanoparticle formulation for the treatment of schistosomiasis. ACS Appl Nano Mater. 2025;8(8):3985–97.

[13] Pereira MN, Ushirobira CY, Cunha-Filho MS, Gelfuso GM, Gratieri T. Nanotechnology advances for hair loss. Ther Deliv. 2018;9(8):593–603.

[14] Gross IP, Lima AL, Sa-Barreto LL, Gelfuso GM, Cunha-Filho M. Recent advances in cutaneous drug delivery by iontophoresis. Expert Opin Drug Deliv. 2025;22(6):857–74.

[15] Jara MO, Catalan-Figueroa J, Landin M, Morales JO. Finding key nanoprecipitation variables for achieving uniform polymeric nanoparticles using neurofuzzy logic technology. Drug Deliv Transl Res. 2018;8:1797–806.

[16] Youshia J, Ali ME, Lamprecht A. Artificial neural network based particle size prediction of polymeric nanoparticles. Eur J Pharm Biopharm. 2017;119:333–42.

[17] Abd-Algaleel SA, Abdel-Bar HM, Metwally AA, Hathout RM. Evolution of the computational pharmaceutics approaches in the modeling and prediction of drug payload in lipid and polymeric nanocarriers. Pharmaceuticals (Basel). 2021;14(7):645.

[18] Maximiano FP, Costa GHY, de Sá Barreto LCL, Bahia MT, Cunha-Filho MS. Development of effervescent tablets containing benznidazole complexed with cyclodextrin. J Pharm Pharmacol. 2011;63(6):786–93.

[19] Pires FQ, Angelo T, Silva JK, Sá-Barreto LC, Lima EM, Gelfuso GM, et al. Use of mixture design in drug-excipient compatibility determinations: thymol nanoparticles case study. J Pharm Biomed Anal. 2017;137:196–203.

[20] Ansari SR, Suárez-López YDC, Thersleff T, Häggström L, Ericsson T, Katsaros I, et al. Pharmaceutical quality by design approach to develop high-performance nanoparticles for magnetic hyperthermia. ACS Nano. 2024;18(23):15284.

[21] Malaquias LF, Schulte HL, Chaker JA, Karan K, Durig T, Marreto RN, et al. Hot melt extrudates formulated using design space: one simple process for both palatability and dissolution rate improve

[ment. J Pharm Sci. 2018;107(1):286–96.

[22] Stanzione F, Giangreco I, Cole JC. Use of molecular docking computational tools in drug discovery. Prog Med Chem. 2016;55:273–343.

[23] Vemula SK, Daravath B, Repka M. Quality by design (QbD) approach to develop fast-dissolving tablets using melt-dispersion paired with surface-adsorption method: formulation and pharmacokinetics of flurbiprofen melt-dispersion granules. Drug Deliv Transl Res. 2023;13(12):3204–22.

[24] Shafique U, Qaiser H. A comparative study of data mining process models (KDD,

CRISP-DM and SEMMA). Int J Innov Sci Res.

[2014;12(1):217–22.

[25] Wang H, Cao H, Yang L. Machine learning-driven multidomain nanomaterial design: from bibliometric analysis to applications. ACS Appl Nano Mater. 2024;7(23):26579–600.

[26] Hayashi T, Latag GV. Self-assembled monolayers as platforms for nanobiotechnology and biointerface research: fabrication, analysis, mechanisms, and design. ACS Appl Nano Mater. 2025;8(17):8570–87.

[27] An J, Kim IS, Kim KJ, et al. Efficacy of automated machine learning models and feature engineering for diagnosis of equivocal appendicitis using clinical and computed tomography findings. Sci Rep. 2024;14:22658.

[28] Dietterich TG. An experimental comparison of three methods for constructing ensembles of decision trees: bagging, boosting, and randomization. Mach Learn. 2000;40(2):139–57.

[29] Jara MO, Landin M, Morales JO. Screening of critical variables in fabricating polycaprolactone nanoparticles using neuro fuzzy logic. Int J Pharm. 2021;601:120558.

[30] Silveira RF, Lima AL, Gross IP, Gelfuso GM, Gratieri T, Cunha-Filho M. The role of artificial intelligence and data science in nanoparticles development: a review. Nanomedicine. 2024;19:1271.

[31] Rasmy HE, Abouelmagd SA, Ibrahim EA. New ionic liquid forms of antituberculosis drug combinations for optimized stability and dissolution. AAPS PharmSciTech. 2025;26(1):27.

[32] Ferreira-Nunes R, Cunha-Filho M, Gratieri T, Gelfuso GM. Follicular-targeted delivery of spironolactone provided by polymeric nanoparticles. Colloids Surf B Biointerfaces. 2021;208:112101.

[33] Zaibudeen AW. Recent advances in different nanoprecipitation methods for efficient drug loading and controlled release. J Nanotechnol Nanomater. 2025;6(1):12–25.

[34] Khan I, Saeed K, Khan I. Nanoparticles: properties, applications and toxicities. Arab J Chem. 2019;12(7):908–31.35. Yadav N, Khatak S. Eudragit-based nanocarriers: a review of recent applications in drug delivery. J Drug Deliv Sci Technol. 2020;55:101478.

[36] Song Y, et al. Recent advances in polymeric nanoparticle based drug delivery systems. Colloids Surf B Biointerfaces. 2021;204:111812.

[37] Pinho LA, Lima AL, Sa-Barreto LL, Gratieri T, Gelfuso GM, Marreto RN, et al. Preformulation studies to guide the production of medicines by fused deposition modeling 3D printing. AAPS PharmSciTech. 2021;22(8):263.

[38] Shahiwala AF, Qawoogha SS, Faruqui N. Designing optimum drug delivery systems using machine learning approaches: a prototype study of niosomes. AAPS PharmSciTech. 2023;24(4):94.

[39] Khakpour A, Florescu L, Tilley R, Jiang H, Iyer KS, Carneiro G. (2025). AI-powered prediction of nanoparticle pharmacokinetics: A multi-view learning approach Materials Today Communications, 49, 113742. https:// doi. org/ 10. 1016/j. mtcomm. 2025. [113742.]

[40] Zhang W, Wu C, Zhong H, Li Y, Wang L. Prediction of undrained shear strength using extreme gradient boosting and random forest based on Bayesian optimization. Geosci Front. 2021;12(1):469–77.

[41] Wood SN. Fast stable restricted maximum likelihood and marginal likelihood estimation of semiparametric generalized linear models. J R Stat Soc Series B Stat Methodol. 2011;73(1):3–36.

[42] Wu X, Kumar V, Quinlan JR, et al. Top 10 algorithms in data mining. Knowl Inf Syst. 2008;14(1):1–37.

[43] Pannakkong W, Thiwa-Anont K, Singthong K, Parthanadee P, Buddhakulsomsiri J. Hyperparameter tuning of machine learning algorithms using response surface methodology: a case study of ANN, SVM, and DBN. Math Probl Eng. 2022;2022:7958647.

[44] Zhou Z, Qiu C, Zhang Y. A comparative analysis of linear regression, neural networks and random forest regression for predicting air ozone employing soft sensor models. Sci Rep. 2023;13:22420.

[45] Vabalas A, Gowen E, Poliakoff E, Casson AJ. Machine learning algorithm validation with a limited sample size. PLoS ONE. 2019;14(11):e0224365.

[46] Pradhan D, Biswasroy P, Goyal A, et al. Recent advancement in nanotechnology-based drug delivery system against viralinfections. AAPS PharmSciTech. 2021;22:47. https:// doi. org/ 10. 1208/ s12249- 020- 01908-5.

[47] Gneiting T, Raftery AE. Strictly proper scoring rules, prediction, and estimation. J Am Stat Assoc. 2007;102(477):359–78.

[48] Zhang D. A coefficient of determination for generalized linear models. Stat Med. 2017;36(30):4821–37.

[49] Lo WW, Yang X, Wang Y. An Xception convolutional neural network for malware classification with transfer learning. Proc 10th IFIP Int Conf New Technol Mobility Secur (NTMS); IEEE. 2019. p. 1–5

[50] Pan SJ, Yang Q. A survey on transfer learning. IEEE Trans Knowl Data Eng. 2010;22(10):1345

[51] Mujahid M, Kına E, Rustam F, et al. Data oversampling and imbalanced datasets: an investigation of performance for machine learning and feature engineering. J Big Data.

2024;11: 87. https://doi.org/10.1186/s40537-024-00943-4.

[52] Furxhi I, Murphy F, Mullins M, et al. Machine learning prediction of nanoparticle in vitro toxicity: a comparative study of classifiers

[and ensemble-classifiers using the Copeland index. Toxicol Lett. 2019;312:157–66.

[53] Alom MZ, Taha TM, Yakopcic C, et al. A state-of-the-art survey on deep learning theory and architectures. Electronics. 2019;8(3):292.

4 A Machine Learning Framework for Polymeric Nanoparticle Design with Zero-Shot Generalization Across Active Pharmaceutical Ingredients*

*manuscript currently being prepared for submission to a peer-reviewed journal

Abstract

The development of polymeric nanoparticles is inherently complex due to the interplay between formulation composition, process conditions, and drug-specific physicochemical properties. Machine learning (ML) has emerged as a promising tool to support formulation design; however, most existing models are limited to single-drug systems and lack validation under realistic experimental variability. In this study, we present the NANO-AI framework, a data-driven approach that integrates formulation variables, process parameters, and molecular descriptors within a unified predictive workflow. The framework is structured as a two-step model: a classification stage to predict nanoparticle formation feasibility, followed by regression models to estimate key physicochemical properties, including particle size, polydispersity index (PDI), zeta potential, and encapsulation efficiency. Using a dataset comprising 925 formulations across nine chemically diverse active pharmaceutical ingredients (APIs), the classification model achieved an AUC of 0.823, enabling reliable identification of viable formulation conditions. For successful systems, regression models demonstrated high predictive performance, with R^2 values exceeding 0.95 for particle size, encapsulation efficiency, and zeta potential, while maintaining functionally relevant predictions for PDI. Solid results were confirmed through interlaboratory validation involving different operators and experimental environments, with prediction errors remaining within expected variability ranges. Furthermore, zero-shot validation using unseen APIs demonstrated the model's ability to generalize across chemical spaces, with typical prediction errors below 20–30% for key properties. Rather than focusing solely on exact point prediction, the NANO-AI framework provides reliable directional guidance for formulation design, enabling efficient identification of viable conditions and expected property ranges.

Keywords: Polymeric nanoparticle; Nanoprecipitation; Machine learning; Formulation design; Interlaboratory validation; Zero-shot generalization.

4.1 Introduction

The integration of machine learning (ML) into pharmaceutical sciences has emerged as a promising strategy for the rational design and accelerated development of drug delivery systems [1], [2], [3]. In particular, the production of nanomedicines represents a highly multivariate process, in which subtle variations in formulation composition or operational parameters can significantly influence the key physicochemical properties of the final product [4], [5]. Traditional statistical tools applied within the Quality by Design (QbD) framework, such as response surface methodologies, often struggle to capture the nonlinear, high-dimensional, and dynamic nature of nanoparticulate systems [6], [7]. In contrast, ML approaches, particularly ensemble methods and deep learning architectures, offer a powerful alternative by identifying complex patterns and relationships within experimental data, thereby supporting QbD principles while reducing experimental workload and reliance on empirical trial-and-error strategies [8] .

Despite its potential, the practical application of ML in nanotechnology remains limited by recurring methodological challenges. A critical analysis of the current literature reveals that most predictive models are developed using small, unbalanced datasets, frequently lacking robust validation strategies [1]. Moreover, the studies are restricted to a single active pharmaceutical ingredient (API) and are evaluated under conditions closely resembling the training environment, with minimal consideration of external or interlaboratory variability. These limitations often result in overfitted models with reduced generalization capacity when exposed to novel molecules or real-world experimental noise [9], [10]. Therefore, there is a clear need for ML frameworks capable of integrating diverse experimental conditions while maintaining predictive reliability under realistic variability conditions.

To address these limitations, the present study builds upon a previously proposed multistep ML framework developed by our research group [4]. The original architecture incorporated an intermediate multiclassification stage to address dataset limitations and imbalance associated with single-drug systems. However, the expansion of the experimental database to over 900 formulations, encompassing nine physicochemically diverse APIs, enabled a simplification and generalization of the predictive pipeline. The resulting framework adopts a two-step architecture consisting of (i) a binary classification model to predict nanoparticle formation and (ii) regression models to estimate key physicochemical properties of the system. This modification allows the model to operate across a broader chemical space while maintaining predictive performance. Unlike previous studies restricted to single-API systems, the present framework integrates multi-API data and incorporates external validation strategies, including interlaboratory testing and zero-shot generalization.

The objective of this study is to present an optimized and generalized two-step ML platform, termed NANO-AI, as a scalable framework for the design of polymeric nanoparticles. The proposed system aims to extend predictive capability beyond the original training domain by leveraging molecular descriptors and formulation variables to guide the development of nanocarriers for previously unseen APIs. To assess its robustness and practical applicability, the models were subjected to external validation protocols, including interlaboratory experiments designed to evaluate sensitivity to operator and environmental variability. In addition, the generalization capacity of the platform was investigated through blind simulations involving novel APIs not included in the training dataset. By demonstrating consistent predictive behavior across diverse chemical and experimental conditions, this work supports the advancement of ML as a practical tool for reducing experimental burden and improving decision-making in nanomedicine development.

4.2 Materials and Methods

4.2.1 Materials

The experimental database was constructed using nine physicochemical diverse active pharmaceutical ingredients (APIs): isoniazid, paracetamol, propranolol, dutasteride, spironolactone, minoxidil, fluconazole, naringenin, and finasteride. For validation, the APIs caffeine, nifurtimox, and itraconazole were also used. The polymers Eudragit® E100 and Eudragit® L100 were provided by Evonik Industries (Essen, Germany). Polysorbate 80 (Tween® 80) and poly(vinyl alcohol) (PVA 40-88) were utilized as stabilizers [11]. All other reagents and solvents were of analytical grade and used as received without further purification [12], [13].

4.2.2 Nanoparticle preparation

Nanoparticles were prepared using the nanoprecipitation technique [14], [15]. Briefly, the selected polymer and API were dissolved in an organic phase, which was subsequently added to an aqueous phase containing a stabilizer under controlled conditions. The formation of nanoparticles occurred spontaneously upon solvent displacement and polymer precipitation. Methanol was subsequently removed under reduced pressure using a rotary evaporator to ensure complete solvent elimination and stabilization of the colloidal system. The range of formulation and process variables explored is detailed in Section 4.2.5.

To ensure reproducibility while preserving realistic experimental variability, all formulations were prepared by up to six previously trained operators following standardized written protocols. This approach allowed the incorporation of controlled inter-operator variability into the dataset, improving the robustness and generalization capacity of the subsequent ML models.

4.2.3 Physicochemical characterization

The physicochemical properties of the nanoparticles were evaluated immediately after preparation. The mean hydrodynamic diameter and polydispersity index (PDI) were determined

by dynamic light scattering (DLS), while the zeta potential was measured by electrophoretic light scattering, using a Zetasizer Nano ZS (Malvern Panalytical, UK) at 25 °C.

Prior to analysis, samples were appropriately diluted in ultrapure water to avoid multiple scattering effects and to ensure optimal signal quality. Measurements were performed in triplicate, and results are reported as mean ± standard deviation.

Encapsulation efficiency (EE) was determined using an indirect method by quantifying the non-encapsulated drug present in the filtrate. Briefly, nanoparticle dispersions were subjected to a separation step using centrifugation to isolate the free drug fraction from the nanoparticulate system. The filtrate, containing the non-encapsulated drug, was then quantitatively analyzed using previously validated analytical methods specific to each active pharmaceutical ingredient. The encapsulation efficiency was calculated based on the difference between the total amount of drug initially added to the formulation and the amount of free drug detected in the filtrate [16], according to Equation 1.

EE (%) = (Total drug amount – Free drug amount / Total drug amount) × 100 (Eq. 1)

4.2.4 Dataset construction and Experimental design

To support the development of a robust and generalizable ML framework, a comprehensive experimental dataset was systematically constructed based on nanoprecipitation experiments performed under a wide range of formulation and process conditions. The dataset comprised 925 formulations encompassing nine physicochemically diverse active pharmaceutical ingredients (APIs), selected to ensure broad coverage of chemical space and variability in molecular properties (Figure 1).

The experimental design followed a hybrid strategy combining exploratory and

data-driven approaches. In the initial phase, formulations were designed to broadly explore the formulation and process parameter space, capturing both successful nanoparticle formation and failure events, such as macroscopic precipitation during phase mixing or solvent removal. In subsequent stages, additional experiments were iteratively conducted to improve dataset representativeness, including balancing the proportion of successful and failed formulations and populating underrepresented regions of the response space identified during preliminary data analysis. This approach enabled the construction of a dataset suitable for both classification and regression modeling tasks.

The final dataset inherently reflects the multivariate complexity and stochastic nature of nanoprecipitation processes, incorporating variability associated with formulation composition, process conditions, and operator-dependent factors. Importantly, both successful nanoparticle systems and failed formulations were intentionally retained, allowing the model to learn the physicochemical boundaries that govern nanoparticle formation.

For ML purposes, the dataset was partitioned into two subsets according to the requirements of the predictive framework. The first subset included the complete dataset (925 formulations), comprising both successful nanoparticle systems (n = 422) and failed formulations (n = 503), and was used to train the binary classification model responsible for predicting nanoparticle formation. The second subset consisted exclusively of the successful nanoparticle formulations (n = 422) and was used to train regression models for the prediction of nanoparticle properties.

This dataset construction strategy enabled the development of a predictive framework capable of both identifying viable formulation conditions and estimating physicochemical properties, while preserving the inherent variability observed in experimental nanomedicine development.

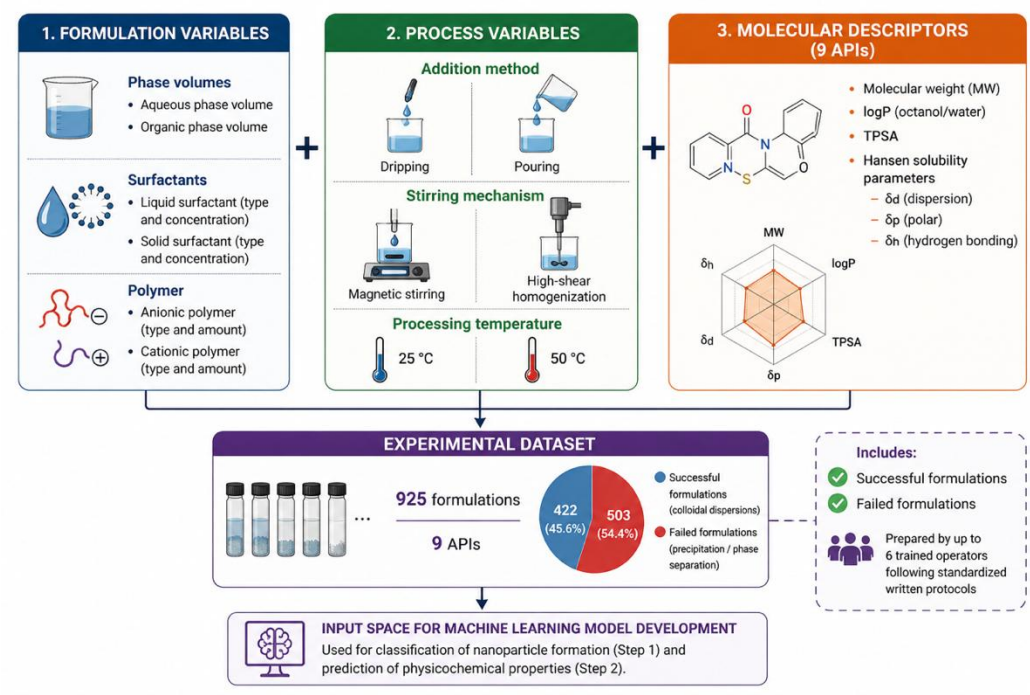


Figure 1. Schematic representation of the experimental design space, including formulation variables, process parameters, and molecular descriptors used to construct the dataset.

4.2.5 Feature Engineering and Input Variables

The experimental variables described in Section 4.2.2 were structured into a unified feature space to enable their use as inputs for ML model development. The feature set integrates formulation, process, and molecular information, allowing the models to capture both controllable experimental conditions and intrinsic physicochemical properties of the APIs.

Continuous variables included phase volumes, surfactant concentrations, and polymer amounts. In addition, selected molecular descriptors were incorporated to enhance the model's generalizability to previously untested drug compounds. These descriptors comprised molecular weight, logP, topological polar surface area (TPSA), and Hansen solubility parameters (HSP). The total HSP (δt), along with its dispersive, polar, and hydrogen-bonding components (δD, δP, and δH), were determined using HSPiP software (version 5.4.08, Ipswich, UK). For compounds not available in the software database, molecular structures were manually introduced via the Simplified Molecular Input Line Entry System (SMILES) [17]. Prior to model training, all variables were systematically assessed for consistency and appropriately scaled to ensure

numerical stability across the different algorithms.

Categorical variables, such as the method of organic phase addition and stirring mechanism, were encoded into numerical representations suitable for ML models. This encoding enabled the incorporation of discrete process conditions while preserving their functional distinctions in the experimental workflow.

No artificial reduction of variability or aggressive feature selection was performed, as preserving the natural dispersion of the experimental dataset was considered essential to support model generalization across diverse formulation conditions and chemical spaces.

The resulting feature space provides a comprehensive representation of the nanoprecipitation system, combining formulation composition, process dynamics, and molecular-level descriptors, thereby enabling the learning of complex nonlinear relationships governing nanoparticle formation and properties.

4.2.6 ML model development

ML models were developed to capture the complex relationships between formulation composition, process conditions, and the resulting nanoparticle behavior. Based on the structured feature set described in Section 4.2.5, multiple algorithms were systematically trained and evaluated to identify the most suitable approaches for both classification and regression tasks.

For the binary classification problem associated with nanoparticle formation (Step 1), the following algorithms were considered: Generalized Linear Models (GLM), Naive Bayes, Random Forest, Extreme Gradient Boosting (XGBoost), and Deep Learning (Artificial Neural Networks) [16]. For the regression tasks (Step 2), which involve the prediction of continuous nanoparticle properties, the analysis focused on Random Forest, XGBoost, and Deep Learning models, as these approaches are more suitable for capturing nonlinear relationships inherent to nanoprecipitation systems[17], [18].

Hyperparameter optimization was performed using a randomized search strategy to efficiently explore the parameter space while minimizing computational cost [19]. The tuning process included adjustments to key parameters such as the number of trees, tree depth, learning

rate, and regularization terms for ensemble models, as well as network architecture, activation functions, and regularization strategies for neural networks [17], [20]. Detailed hyperparameter configurations for the final models are provided in the Supplementary Material.

To ensure model robustness and reduce the risk of overfitting, all models were validated using k-fold cross-validation (with k ranging from 3 to 7), depending on the dataset size and task complexity [21]. In addition, early stopping criteria were applied during training based on performance convergence to prevent unnecessary model complexity and to improve generalization [22].

Model development was conducted with an emphasis on balancing predictive performance and generalization capacity. Given the heterogeneity of the dataset—comprising multiple APIs, diverse formulation conditions, and inherent experimental variability—model selection prioritized approaches capable of handling nonlinear interactions and high-dimensional feature spaces [23]. This strategy supports the development of predictive models that are not only accurate within the training domain but also transferable to new experimental conditions and previously unseen molecular systems.

All models were implemented in the R programming language (Free Software Foundation, version 4.3.3) and executed within the RStudio development environment. The H2O package (version 3.44.0.3) for R was employed, as it natively provides implementations of algorithms such as Random Forest, XGBoost, Artificial Neural Networks (ANNs), Naive Bayes, and Generalized Linear Models (GLMs), in addition to a comprehensive suite of model validation metrics and ensuring reproducibility of the modeling workflow.

4.2.7 Two-Step predictive framework

The computational framework developed in this study (NANO-AI) was structured as a sequential two-step predictive pipeline designed to reflect the rational development of nanoparticulate systems, as schematically illustrated in Figure 2. This architecture aligns with QbD principles, in which the establishment of system feasibility precedes the optimization of product properties [24] .

In the first stage (Step 1), a binary classification model was implemented to predict the likelihood of nanoparticle formation under a given set of formulation and process conditions. This step functions as a computational screening filter, distinguishing between successful colloidal dispersions and failed systems characterized by macroscopic precipitation. The model integrates formulation variables, process parameters, and molecular descriptors to estimate the probability of obtaining a stable nanoparticulate system. Failure events considered during model training included precipitation occurring during organic phase addition, solvent removal, or immediately after nanoparticle formation. By identifying non-viable conditions in advance, this step reduces unnecessary experimental effort and defines the boundaries of the feasible design space.

In the second stage (Step 2), regression models were applied exclusively to the subset of successful formulations identified in Step 1 to predict the key physicochemical properties of the nanoparticles, including mean hydrodynamic diameter, PDI, zeta potential, and EE [25], [26]. By restricting the regression analysis to physically viable systems, the models focus on capturing the nonlinear relationships governing nanoparticle refinement, rather than attempting to learn from unstable or non-representative data.

This two-step structure represents an evolution of previously proposed multistep frameworks [4], in which an intermediate multiclassification stage was required to compensate for limited and unbalanced datasets. In the present study, the expansion and diversification of the dataset enabled the simplification of the pipeline, allowing direct regression modeling following the classification filter without loss of predictive performance.

Overall, the proposed framework integrates feasibility screening and property prediction into a unified workflow, enabling both the identification of viable formulation

conditions and the estimation of nanoparticle characteristics. This approach supports a more efficient exploration of the experimental design space and facilitates data-driven decision-making in the development of polymeric nanocarriers.

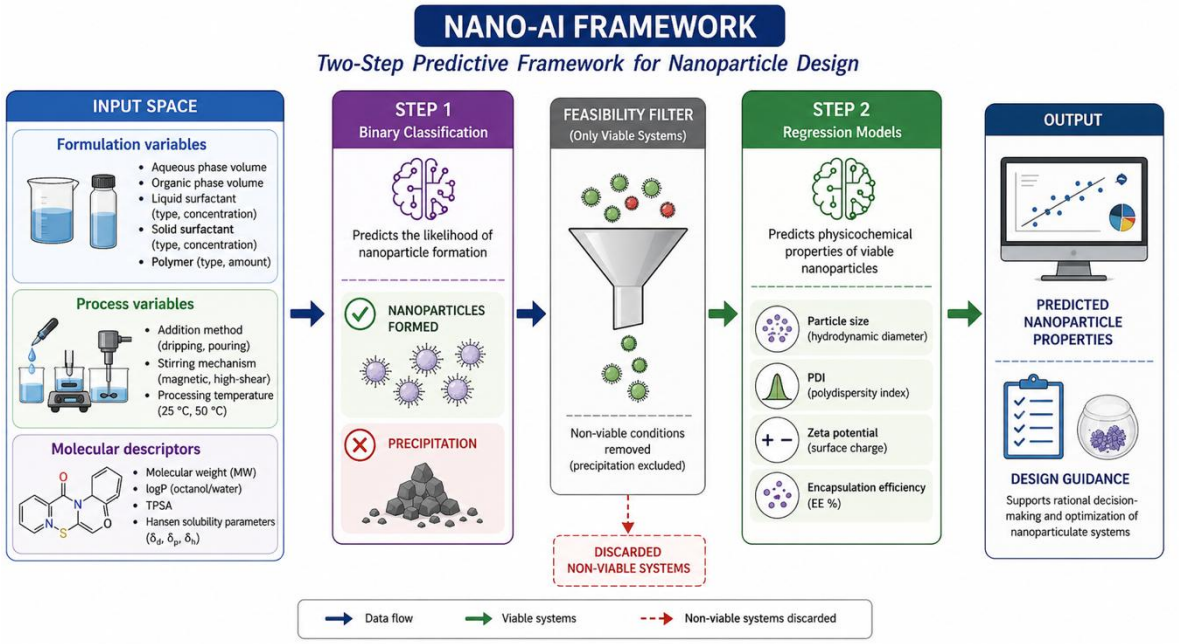


Figure 2. Schematic representation of the two-step machine learning framework (NANO-AI) for nanoparticle design, integrating feasibility screening and property prediction.

4.2.8 Model evaluation and interpretation

The predictive performance of the ML models was evaluated using statistical metrics selected according to the nature of each task [27]. For the binary classification models in Step 1, performance was assessed using the Area Under the Receiver Operating Characteristic Curve (AUC), F1-score, Precision, and Recall. Among these, particular emphasis was placed on Precision and AUC, as minimizing false-positive predictions is critical in the context of virtual formulation screening. Incorrectly classifying a non-viable formulation as successful may lead to unnecessary experimental effort, increased material consumption, and inefficient use of resources [28].

For the regression models in Step 2, performance was evaluated using the coefficient of determination (R^2), root mean square error (RMSE), and mean absolute error (MAE). The R^2 metric was used to quantify the proportion of variance in the nanoparticle properties explained by

the model, while RMSE and MAE provided complementary measures of prediction error in the original units of each variable. RMSE was particularly relevant for assessing model robustness, as it penalizes larger deviations more strongly than MAE [29].

This combined evaluation strategy allows not only the assessment of predictive accuracy but also the extraction of physically meaningful relationships from the models, supporting their application as tools for rational formulation design rather than purely statistical predictors.

To align the predictive logic of the computational models with Quality by Design (QbD) principles, a feature importance analysis was conducted for the best-performing algorithms. The relative contribution of each independent variable was evaluated to determine its physicochemical impact on the formulation outcomes:

- **Nanoparticle Formation (Step 1):** Feature extraction from the XGBoost model (Step 1) revealed that the polar solubility parameter (δp) and the concentration of anionic polymers were the most critical factors influencing the physical stability and successful formation of the colloidal system. These variables were followed in importance by the dispersion solubility parameter (δd), liquid surfactant concentration, and aqueous phase volume (Vaq). These findings indicate that the electrostatic interactions provided by the anionic polymers, combined with the specific polar affinity of the API, serve as the primary drivers of the nanoprecipitation process.

- **Particle Size and PDI (Step 2):** For the estimation of the hydrodynamic diameter, the Deep Learning algorithm prioritized process and macro-formulation variables. Aqueous and organic phase volumes, stirring mechanism (e.g., high-shear homogenization versus magnetic stirring), and processing temperature were the most significant inputs. Similarly, the prediction of the polydispersity index (PDI) via the XGBoost algorithm was influenced by phase volumes and the concentration of liquid and solid surfactants.

- **Encapsulation Efficiency and Zeta Potential (Step 2):** The XGBoost algorithm demonstrated that Encapsulation Efficiency (EE%) was fundamentally governed by the API's molecular descriptors, particularly molecular weight, polar solubility (deltap), Topological Polar Surface Area (TPSA), and logP. Conversely, the Zeta Potential was almost entirely dictated by the formulation's electrostatic components, being extremely sensitive to the exact quantities of cationic and anionic polymers added to the system.

4.2.9 External validation strategy

To assess the robustness and generalization capacity of the proposed predictive framework, external validation was conducted through two complementary strategies: (i) interlaboratory validation and (ii) zero-shot generalization using previously unseen APIs. These approaches were designed to evaluate model performance under conditions that more closely resemble real-world variability, addressing a common limitation in ML studies applied to nanomedicine [30].

4.2.10 Interlaboratory validation

To investigate the resilience of the predictive framework to experimental variability, an interlaboratory validation study was conducted using isoniazid as a model drug. Experimental formulations guided by the predictive framework were reproduced in an independent laboratory facility, distinct from the environment in which the training dataset was generated.

The validation protocol was designed to separately assess the impact of instrumental/environmental variability and operator-dependent variability. In the first scenario, the original operator responsible for dataset generation conducted experiments in the external laboratory using different equipment and environmental conditions. In the second scenario, an independent operator, with no prior involvement in the study, reproduced the same formulations following standardized protocols.

This dual approach enabled the evaluation of model performance under both controlled and realistic variability conditions, reflecting typical sources of deviation in nanoparticle preparation. By comparing predicted and experimentally obtained nanoparticle properties across different operators and facilities, the study assessed the extent to which the model captures underlying physicochemical behavior rather than laboratory-specific patterns.

4.2.11 Unseen drug generalization (Zero-shot validation)

To evaluate the ability of the model to generalize across chemical spaces not represented in the training dataset, a zero-shot validation strategy was implemented using three APIs not included during model development: caffeine, nifurtimox, and itraconazole [31], [32], [33]. These compounds were selected to represent a broad range of physicochemical properties, including differences in molecular weight, polarity, and lipophilicity [34].

The predictive framework was used to estimate both the likelihood of nanoparticle formation (Step 1) and the corresponding nanoparticle properties (Step 2) based solely on formulation variables and molecular descriptors. The predicted conditions were subsequently validated through experimental nanoprecipitation, and the resulting properties were compared to model outputs.

This validation strategy was designed to assess the capacity of the model to extrapolate learned relationships to novel APIs without prior exposure to their experimental data. Rather than focusing exclusively on point prediction accuracy, this approach evaluates the ability of the framework to provide reliable directional guidance for formulation design across diverse chemical contexts

4.3 Results and Discussion

4.3.1 Step 1 (Binary classification): Model selection and Feasibility prediction performance

The first stage of the NANO-AI framework focuses on predicting the feasibility of nanoparticle formation, acting as a computational screening layer to distinguish stable

colloidal systems from formulations prone to precipitation. This step is critical in practical formulation workflows, as false-positive predictions directly translate into unnecessary experimental trials, material consumption, and time expenditure.

To identify the most suitable classification model, five ML algorithms (GLM, Naive Bayes, Random Forest, XGBoost, and Deep Learning) were systematically evaluated using the complete dataset of 925 formulations. As shown in Figure 3, XGBoost achieved the highest overall performance, with an AUC of 0.823, outperforming Deep Learning (0.765), GLM (0.728), Random Forest (0.718), and Naive Bayes (0.682).

Beyond global discrimination metrics, the evaluation of Precision is particularly relevant in this context. From a formulation design perspective, incorrectly predicting nanoparticle formation (false positives) leads to failed experimental attempts. XGBoost exhibited the highest Precision (0.752), indicating a more reliable identification of truly viable systems. In contrast, Random Forest, despite achieving the highest Recall (0.908), showed significantly lower Precision (0.560), reflecting a tendency to overestimate formulation feasibility.

Deep Learning presented a balanced performance (Precision = 0.722; Recall = 0.731), but XGBoost achieved the best compromise between sensitivity and reliability, as indicated by its superior F1-score (0.772). These results demonstrate that gradient boosting methods are particularly effective in capturing the nonlinear boundaries governing nanoparticle formation across diverse formulation conditions and molecular descriptors.

The observed shift in algorithmic superiority, relative to the previous study in which Deep Learning outperformed XGBoost for the STEP1 model of isoniazid [4], can be primarily attributed to the transition from a specialized, single-drug chemical space to a highly heterogeneous, multi-component formulation landscape. In the earlier work, the dataset

comprised only 90 formulations of a single active pharmaceutical ingredient (API), resulting in a constrained and chemically consistent domain. Under these conditions, the Deep Learning (DL) model achieved superior predictive performance, indicating its capacity to capture subtle, drug-specific patterns governing isoniazid behavior. Such performance is consistent with the strengths of neural networks in low-diversity settings, where the underlying data-generating process remains relatively stable [18]. The chemical uniformity of a single API reduces variability associated with solute-specific interactions, effectively limiting the "noise" to process-related factors. This enables DL architectures to converge toward a well-defined and highly optimized global solution, enhancing their ability to model nuanced, non-linear relationships within a narrowly defined formulation space [18].

Conversely, the present study introduces a substantial increase in system complexity, encompassing 925 formulations across nine physicochemically diverse drugs, including compounds with pronounced precipitation tendencies such as spironolactone, finasteride, and dutasteride. Within this expanded and heterogeneous context, XGBoost emerged as the superior predictive model, achieving a precision of 0.75 compared to 0.72 for Deep Learning. This inversion in performance underscores fundamental differences in how these algorithms handle structurally heterogeneous datasets.

Deep Learning models typically aim to learn a continuous and smooth mapping across the feature space, which can become challenging when confronted with conflicting patterns arising from multiple APIs. The coexistence of drugs with markedly distinct physicochemical behaviors introduces competing gradients during training, potentially leading to suboptimal convergence. Each API exhibits specific interactions with the polymeric matrix and solvent system; for example, formulation conditions that stabilize hydrophilic compounds may induce rapid precipitation in highly hydrophobic drugs such as dutasteride.

In contrast, XGBoost—an ensemble of gradient-boosted decision trees—demonstrates greater robustness in this type of tabular and heterogeneous setting. By recursively partitioning the feature space into discrete regions, the model can effectively allocate different subspaces to distinct physicochemical regimes. This enables the implicit specialization of tree branches to

capture drug-specific precipitation thresholds and discontinuities. As a result, XGBoost is better suited to modeling the abrupt, non-linear transitions in stability that characterize multi-drug datasets, where no single, unified response surface adequately describes the system behavior.

Importantly, the observed performance aligns with the expected complexity of nanoprecipitation processes, in which multiple interacting variables define a relatively narrow feasibility window. In this context, the classification step effectively defines the accessible design space for subsequent optimization.

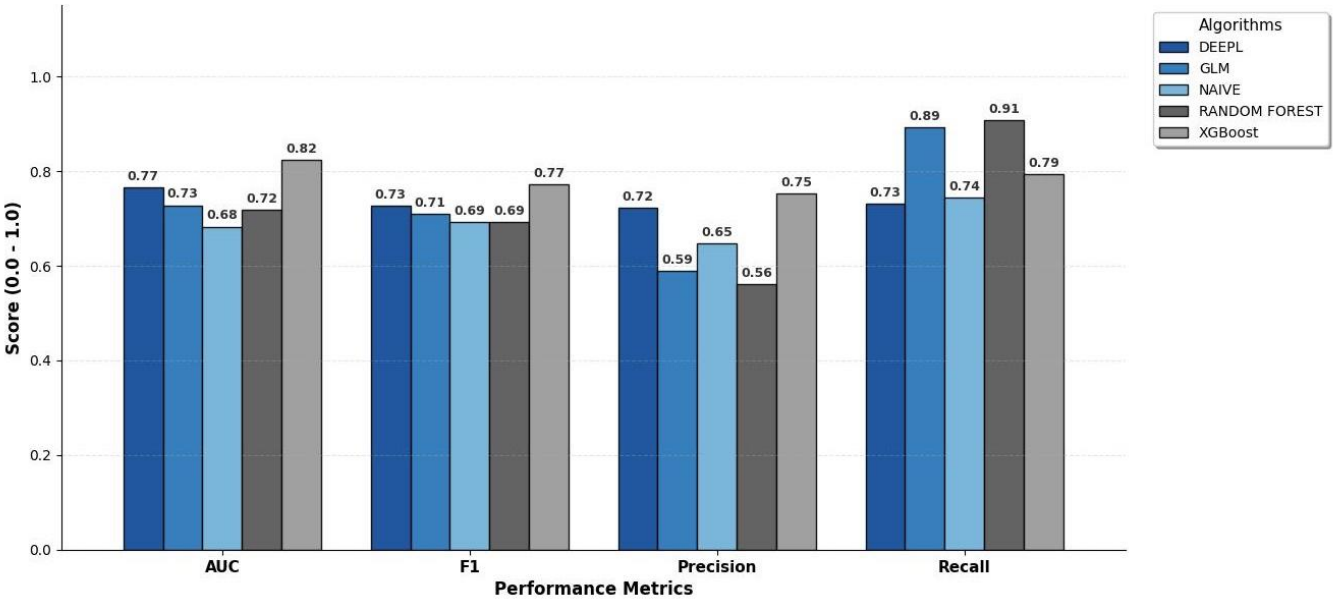


Figure 3. Comparative performance of classification models for nanoparticle formation prediction (Step 1). Model performance is evaluated using (A) Area Under the ROC Curve (AUC), (B) F1-score, (C) Precision, and (D) Recall.

4.3.2 Step 2 (Regression): Prediction of nanoparticle properties

Following the identification of viable formulations, regression models were applied to predict nanoparticle physicochemical properties using the subset of successful formulations (n = 422). The predictive performance of Deep Learning, XGBoost, and Random Forest models is summarized in Figure 4.

For mean hydrodynamic diameter, the Deep Learning model demonstrated

superior performance, achieving an impressive R^2 of 0.964, with RMSE and MAE values of 32.1 nm and 22.5 nm, respectively. It is important to note that lower RMSE and MAE values indicate better predictive accuracy, as these metrics quantify the magnitude of prediction errors in absolute units. The strong performance of the neural network suggests its ability to capture complex nonlinear relationships associated with nucleation, growth, and stabilization phenomena.

In contrast, Random Forest exhibited significantly lower predictive capacity (R^2 = 0.552), indicating limitations in modeling high-dimensional interactions across multiple APIs. This reinforces the importance of selecting models capable of capturing both formulation and molecular-level effects.

For encapsulation efficiency and zeta potential, XGBoost provided the most accurate predictions, achieving R^2 values of 0.959 and 0.970, respectively. These results reflect the strong dependence of these properties on compositional variables and electrostatic interactions, which are effectively captured by tree-based ensemble methods.

The prediction of PDI resulted in comparatively lower performance (R^2 = 0.818), which is consistent with the inherently stochastic nature of particle size distribution. However, from a practical standpoint, precise prediction of PDI is not strictly required. In pharmaceutical applications, ensuring a monomodal distribution with PDI values below 0.5 is generally sufficient for system stability and reproducibility. Notably, all predicted systems met this criterion, indicating that the model provides functionally relevant predictions despite variability in exact values.

The differential performance between Deep Learning (DL) and Extreme Gradient Boosting (XGBoost) in predicting nanoparticle critical quality attributes (CQAs) follows the same rationale described in Section 4.3.1. Hydrodynamic size is a continuous physical property governed by complex, multi-layered interactions between solvent viscosity, polymer chain entanglement, and shear forces during nanoprecipitation. DL architectures,

particularly multilayer perceptrons, excel at capturing these high-order, non-linear dependencies through their hierarchical structure [18]. By utilizing weighted interconnections and non-linear activation functions (such as ReLU), DL models act as universal function approximators capable of modeling the smooth, manifold-like transitions in particle growth [28]. The backpropagation algorithm allows the model to refine its internal representation of the "size" variable across a global loss surface, providing a more structured signal for neural networks to internalize [17].

Conversely, PDI, Zeta Potential, and Encapsulation Efficiency (EE%) exhibit a more discrete nature in the feature space, where abrupt changes in formulation—such as a specific polymer concentration threshold or a change in API polarity—trigger significant shifts in the outcome. XGBoost, as an ensemble of gradient-boosted decision trees, is inherently superior at handling these tabular datasets with sharp decision boundaries [35]. While DL attempts to find a global continuous mapping, XGBoost partitions the feature space into hyper-rectangles through recursive binary splitting [23]. For Zeta Potential, which is dominated by the stoichiometric ratio of ionic groups, and EE%, which depends on solubility limits and molecular affinities, the categorical-like behavior of these shifts is more effectively captured by the tree-based "if-then" logic. Furthermore, XGBoost's regularization parameters, L1 and L2, and its handling of sparsity make it highly resilient to the inherent noise and higher experimental variance associated with EE measurements, which are often more susceptible to operator-dependent artifacts than the relatively stable laser diffraction signals used for sizing.

Overall, the high predictive accuracy observed for size, encapsulation efficiency, and zeta potential, parameters typically difficult to model simultaneously across multiple APIs, demonstrates the robustness of the proposed framework. Compared to previous studies, that exhibit performance metrics approaching 100% when evaluated on very small

datasets—often not exceeding samples of more than 50 observations—thereby indicating a substantial risk of single-drug model overfitting to the data [10], [14], [36], [37], [38], [39], the results obtained here highlight the advantage of integrating diverse and larger datasets into a unified modeling approach .

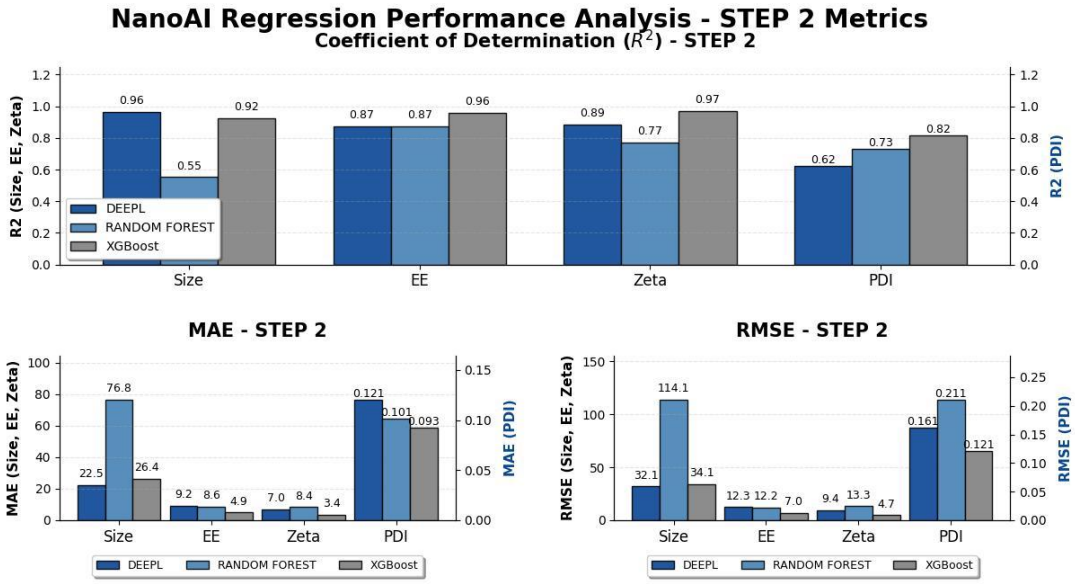


Figure 4. Regression performance of machine learning models for nanoparticle property prediction (Step 2). R^2 values and error metrics (RMSE and MAE) are shown for size, PDI, encapsulation efficiency, and zeta potential.

4.3.3 Physicochemical Insights and Quality by Design (QbD)

A primary advantage of implementing machine learning in pharmaceutical development is the capacity to perform feature importance analysis, which aligns computational predictions with Quality by Design (QbD) principles. By evaluating the relative contribution of each independent variable, the mechanisms governing the multivariate nanoprecipitation process can be scientifically elucidated.

In Step 1, feature importance analysis from the XGBoost algorithm indicated that the polar solubility parameter (δp) and the concentration of anionic polymers were the primary factors influencing the physical stability and successful formation of the nanoparticles. These were followed by the dispersion solubility parameter (δd), liquid surfactant concentration, and the

aqueous phase volume (Vaq). These findings suggest that the initial stability of the colloidal system is predominantly governed by the polar affinity between the API and the solvent-antisolvent phases, as well as the electrostatic stabilization provided by the anionic polymeric matrix. This alignment with Quality by Design (QbD) principles confirms that the interplay between specific thermodynamic descriptors and formulation components dictates the boundaries of the experimental design space for successful nanoprecipitation.

Once thermodynamic viability is established in Step 1, the determination of nanoparticle dimensions in Step 2 is governed by process and macro-formulation variables. For the prediction of the mean hydrodynamic diameter, the Deep Learning algorithm prioritized the volumes of the aqueous and organic phases, the mechanical stirring mechanism (e.g., high-shear homogenization versus conventional magnetic stirring), and the addition method. The ratio between the aqueous and organic phases influences the solvent diffusion rate into the continuous phase, while the stirring mechanism provides the mechanical energy required to overcome interfacial tension and promote droplet fragmentation at the nanometric scale. Similarly, the prediction of the polydispersity index (PDI) was influenced by phase volumes and the concentration of liquid and solid surfactants. Stabilizers are necessary to secure the newly formed interfacial area and prevent coalescence, ensuring a uniform size distribution

Finally, the prediction of encapsulation efficiency (EE%) and zeta potential reflects chemical interactions at the molecular and surface levels. The XGBoost model indicated that EE% is influenced by intrinsic structural properties of the API, specifically molecular weight, the polar solubility parameter (δp), topological polar surface area (TPSA), and logP. These results suggest that lipophilicity and molecular bulkiness determine the affinity of the drug for the polymeric core, limiting diffusion into the external aqueous phase. Conversely, the zeta potential was almost entirely determined by the concentration of cationic and anionic polymers. This aligns with QbD principles, as the electrostatic surface charge of the nanocarrier is a physical manifestation of the ionized functional groups present in the structural polymers utilized in the formulation.

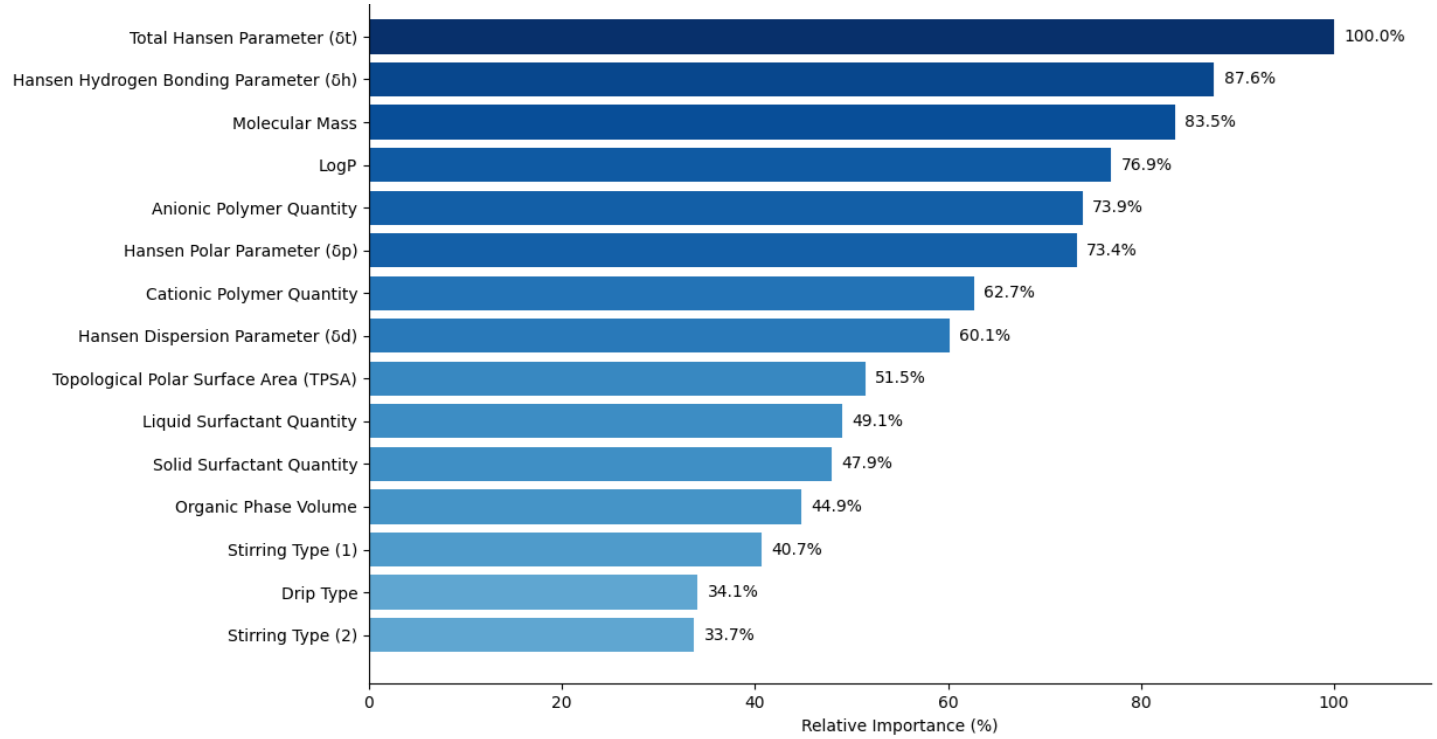


Figure 5. Feature extraction from the XGBoost model (Step 1) identified the polar solubility parameter (δp) and anionic polymer concentration as the main factors governing colloidal stability and nanoparticle formation. Additional influential variables included the dispersion solubility parameter (δd), liquid surfactant concentration, and aqueous phase volume (Vaq). Overall, the results suggest that electrostatic interactions from anionic polymers, together with the API's polar affinity, are key determinants of the nanoprecipitation process.

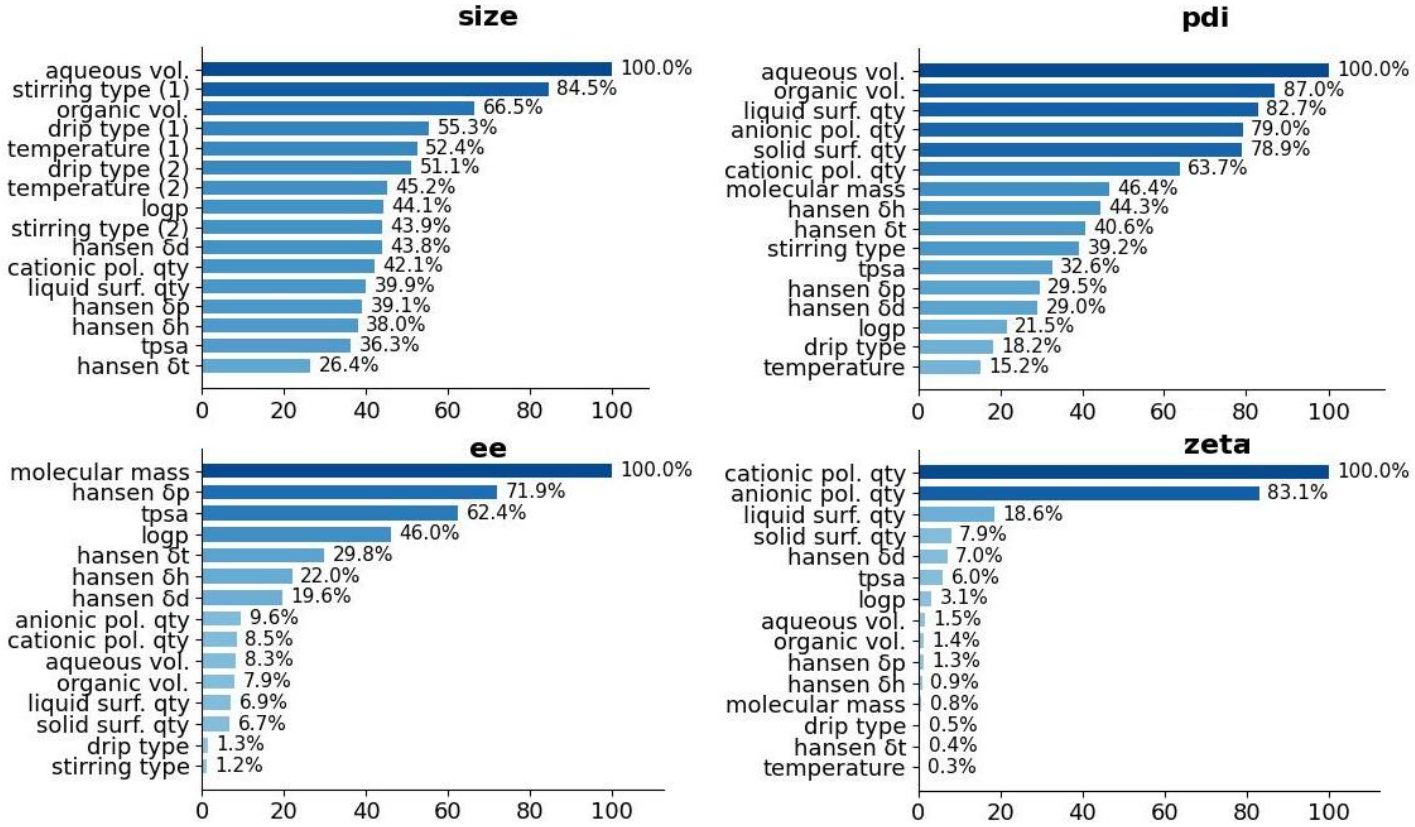


Figure 6. For particle size and PDI prediction (Step 2), the models emphasized the importance of process and formulation parameters. Deep Learning identified aqueous and organic phase volumes, stirring mechanism, and processing temperature as the main determinants of hydrodynamic diameter, while XGBoost indicated that PDI was primarily influenced by phase volumes and the concentrations of liquid and

solid surfactants. Regarding encapsulation efficiency and zeta potential, XGBoost demonstrated that EE% was mainly governed by the API's molecular descriptors, particularly molecular weight, polar solubility (δp), TPSA, and logP. In contrast, zeta potential was predominantly controlled by the formulation's electrostatic composition, especially the concentrations of cationic and anionic polymers.

4.3.4 Robustness under interlaboratory and Operator variability

A major limitation in ML-based nanomedicine studies is the lack of validation under realistic experimental variability [1]. To address this, the robustness of the NANO-AI framework was evaluated through interlaboratory experiments involving different operators and equipment. The aggregated prediction error and variability across laboratories are summarized in Figure 5, while detailed individual results for each formulation, operator, and location are provided in Table S2 (Supplementary Material).

In Step 1, nanoparticle formation rates remained within the expected predictive range (70–90%). While the success rate decreased from 90% (City 1, Operator 1) to 70% (City 2, Operator 1), the independent operator in City 2 achieved 80% success. These values are consistent with the classification performance metrics reported in Section 3.1, indicating that the observed variability falls within expected prediction uncertainty rather than representing model failure.

For Step 2, variability in predictive accuracy was observed across operators and locations. For particle size, prediction errors for Operator 1 remained relatively consistent across laboratories, typically ranging from approximately 10% to 25%, whereas higher deviations were observed for Operator 2, occasionally exceeding 50% in specific cases. As summarized in Figure 5, this increased dispersion likely reflects differences in manual handling, mixing efficiency, and local environmental conditions, which are known to significantly influence nanoprecipitation processes.

For PDI, larger deviations were observed, including cases of overestimation. However, as previously discussed, exact prediction of PDI is less critical than ensuring acceptable distribution profiles. All experimental systems remained within acceptable PDI ranges, supporting the practical utility of the model.

Zeta potential predictions also showed variability across operators and experimental

conditions, with errors ranging from below 10% under favorable conditions to values exceeding 90% in more variable scenarios. These deviations are likely associated with sensitivity to experimental factors such as ionic strength, temperature, and measurement conditions, rather than intrinsic limitations of the predictive model.

The interlaboratory validation of Encapsulation Efficiency (EE%) reveals the high sensitivity of relative error calculations in formulations with low drug entrapment. As experimental values approach zero, small absolute deviations—arising from analytical and operational differences—lead to disproportionately large percentage errors (up to 92.31% or 100%). These deviations reflect the inherent noise of nanomedicine experiments near detection limits rather than a flaw in the model. Nevertheless, the NANO-AI platform showed consistent performance, achieving low errors (0.00% to 9.09%) for stable formulations across different locations. Overall, despite high point-specific errors due to experimental variability, the model remains statistically reliable for estimating entrapment scale and identifying unviable regions of the design space, as evidenced by the aggregated results in Figure 5 and the detailed data presented in guidance within experimentally acceptable ranges, Table S2.

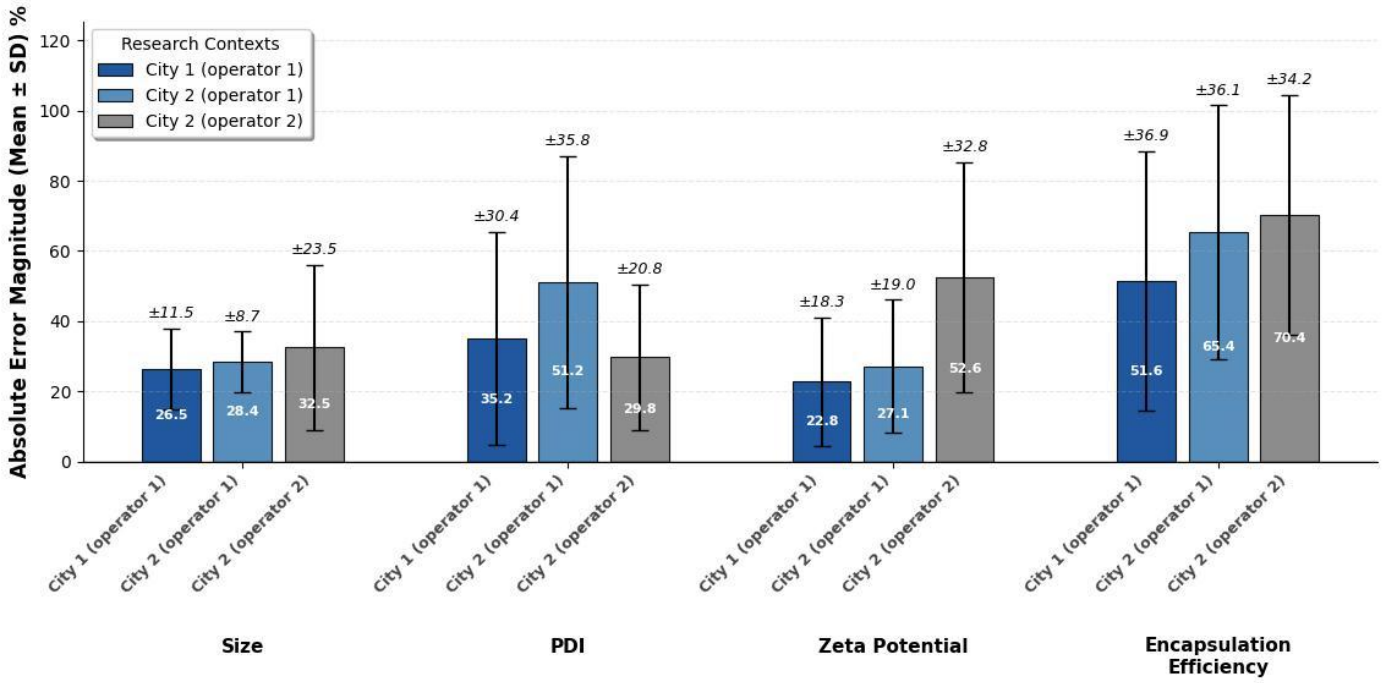


Figure 5. Mean prediction error (±SD) of the NANO-AI framework across operators and laboratories for nanoparticle size, PDI, zeta potential and encapsulation efficiency.

4.3.5 Generalization to Unseen APIs

The ability to generalize across unseen APIs represents a critical challenge in ML-driven formulation design. To evaluate this capability, the NANO-AI framework was tested using caffeine, nifurtimox and itraconazole—three compounds not included in the training dataset and selected to represent a broad range of physicochemical properties (Figure 6).

To the best of our knowledge, this level of cross-API generalization has not been systematically demonstrated for a ML model in pharmaceutical nanoparticle systems, particularly within a unified modeling framework integrating formulation, process, and molecular descriptors.

In Step 1, the model achieved high success rates in predicting nanoparticle formation, ranging from 80% to 100% across the three APIs. These results indicate that the classification model is capable of identifying viable formulation conditions even for previously unseen chemical entities, supporting its role as an effective screening tool for defining feasible design spaces.

In Step 2, predictive performance varied depending on both the property and the physicochemical profile of each API. For mean particle size, the model maintained good predictive accuracy within the most populated range of the training dataset (<200 nm), with typical errors below 20–30%. However, deviations increased for larger particle sizes (>250 nm), occasionally exceeding 40–50%, likely reflecting reduced representation of these conditions in the training data and increased sensitivity to process variability.

For PDI, larger relative deviations were observed, including values exceeding 100% in some cases. However, consistent with previous observations, all experimentally obtained systems exhibited PDI values below 0.5, indicating acceptable monomodal distributions. From a practical standpoint, this confirms that the model provides

functionally relevant predictions, even when exact numerical agreement is limited.

Zeta potential predictions showed highest accuracy within moderate negative ranges (approximately −10 to −30 mV), where errors typically remained below 20%. Increased deviations were observed at more extreme values, occasionally exceeding 50%, reflecting the nonlinear dependence of surface charge on formulation composition, ionic strength, and experimental conditions.

EE exhibited the most consistent predictive performance across all APIs, with typical deviations below 15–20%. Notably, higher discrepancies observed for caffeine were not associated with visible instability or phase separation, suggesting that experimental variability in quantification, rather than model limitations, may have contributed to these differences.

Overall, the zero-shot validation demonstrates that the NANO-AI framework is capable of extrapolating learned relationships to previously unseen chemical spaces with considerable robustness. Rather than focusing exclusively on exact point prediction, the model provides reliable directional guidance for formulation design, enabling the identification of viable processing conditions and expected physicochemical property ranges across structurally diverse APIs. The results further suggest that the framework successfully captures underlying formulation–property relationships that can be generalized beyond the compounds included in the training dataset. This capability is particularly valuable during early-stage pharmaceutical development, where rapid screening, reduction of experimental workload, and informed decision-making are critical for accelerating formulation optimization and minimizing resource consumption.

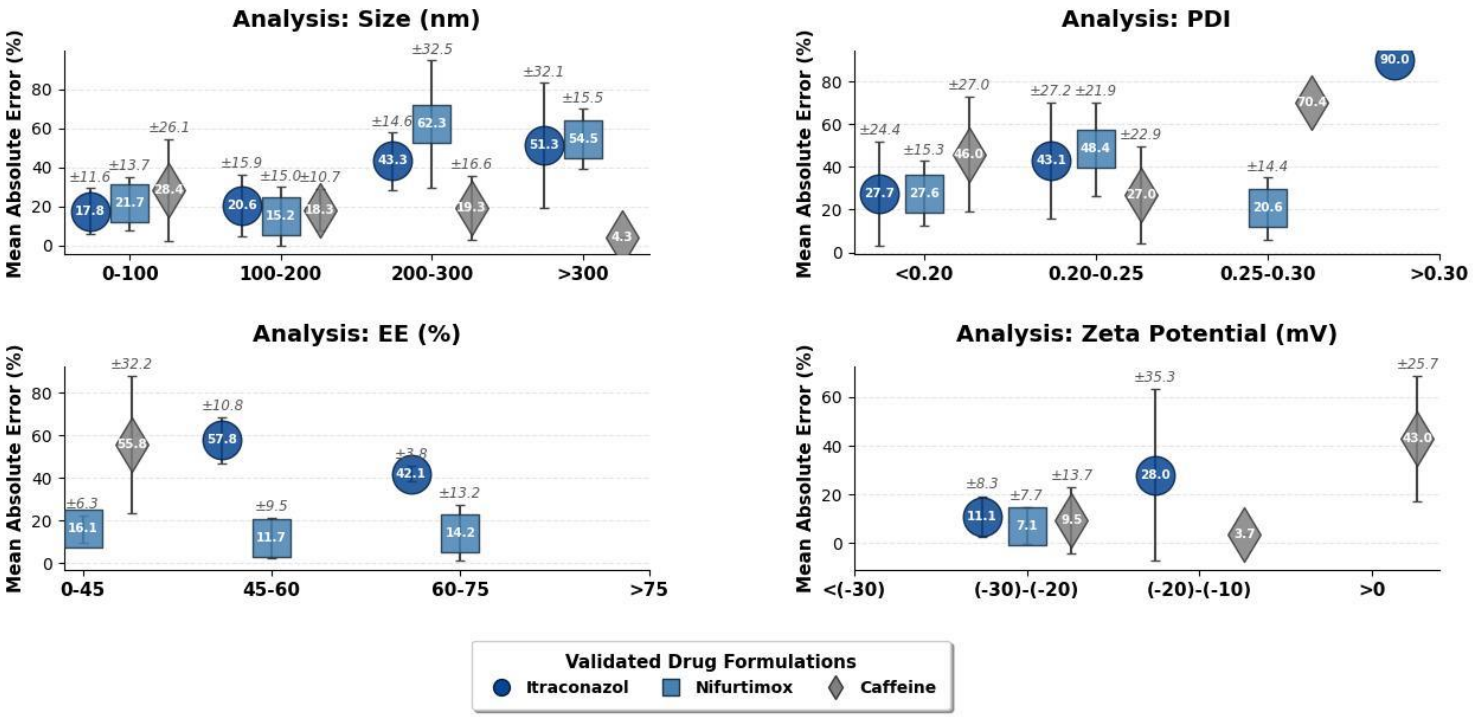


Figure 6. Zero-shot validation of the NANO-AI framework for nanoparticle property prediction using unseen APIs (caffeine, nifurtimox, and itraconazole). Prediction errors (%) are shown as a function of experimentally measured values for (A) mean hydrodynamic diameter, (B) polydispersity index (PDI), (C) zeta potential, and (D) encapsulation efficiency (EE). Each point represents an individual formulation.

4.4 Conclusion

This study presents the NANO-AI framework, a data-driven approach for the design of polymeric nanoparticles that integrates formulation variables, process parameters, and molecular descriptors within a unified ML workflow. By structuring the predictive process into a two-step framework, the model enables efficient screening of viable formulations and estimation of key physicochemical properties.

The classification model demonstrated reliable identification of nanoparticle formation conditions, effectively defining the feasible design space and reducing unnecessary experimental trials. For successful formulations, regression models achieved high predictive performance for particle size, encapsulation efficiency, and zeta potential across multiple APIs, while maintaining functionally relevant predictions for PDI.

Interlaboratory validation confirmed the robustness of the framework under realistic experimental variability, demonstrating consistent predictive behavior across different operators

and laboratory environments. Furthermore, zero-shot validation using unseen APIs showed that the model can generalize to new chemical spaces, providing reliable directional guidance for formulation design even in the absence of prior experimental data.

Rather than focusing solely on exact point prediction, the NANO-AI framework supports informed decision-making by identifying viable formulation conditions and expected property ranges. This capability represents a step toward more efficient, data-driven development of nanomedicines, with characteristics consistent with an advanced level of technological maturity and potential for future translation into applied and industrial contexts.

4.5 References

[1] R. F. Silveira, A. L. Lima, I. P. Gross, G. M. Gelfuso, T. Gratieri, and M. Cunha-Filho, "The role of artificial intelligence and data science in nanoparticles development: a review," *Nanomedicine*, vol. 19, p. 1271, 2024.

[2] H. Wang, H. Cao, and L. Yang, "Machine learning-driven multidomain nanomaterial design: from bibliometric analysis to applications," *ACS Appl. Nano Mater.*, vol. 7, no. 23, pp. 26579–26600, 2024.

[3] D. B. Olawade, A. O. Ige, A. G. Olaremu, J. O. Ijiwade, and A. O. Adeola, "The synergy of artificial intelligence and nanotechnology towards advancing innovation and sustainability: a mini-review," *Nano Trends*, vol. 8, p. 100052, 2024.

[4] R. F. Silveira *et al.*, "Multistep Machine Learning Pipeline For Polymeric Nanoparticle Design," *AAPS PharmSciTech*, vol. 27, no. 1, p. 8, Oct. 2025, doi: 10.1208/s12249-025-03248-8.

[5] A. L. Lima, T. Gratieri, M. Cunha-Filho, and G. M. Gelfuso, "Polymeric nanocapsules: a review on design and production methods for pharmaceutical purpose," *Methods*, vol. 199, pp. 54–66, 2022.

[6] S. R. Ansari *et al.*, “Pharmaceutical quality by design approach to develop high-performance nanoparticles for magnetic hyperthermia,” *ACS Nano*, vol. 18, no. 23, p. 15284, 2024.

[7] F. Stanzione, I. Giangreco, and J. C. Cole, “Use of molecular docking computational tools in drug discovery,” *Prog. Med. Chem.*, vol. 55, pp. 273–343, 2016.

[8] T. B. Taha, A. A. Barzinjy, F. H. S. Hussain, and T. Nurtayeva, “Nanotechnology and computer science: trends and advances,” *Memories*, vol. 2, p. 100011, 2022.

[9] M. O. Jara, J. Catalan-Figueroa, M. Landin, and J. O. Morales, “Finding key nanoprecipitation variables for achieving uniform polymeric nanoparticles using neurofuzzy logic technology,” *Drug Deliv. Transl. Res.*, vol. 8, pp. 1797–1806, 2018.

[10] J. Youshia, M. E. Ali, and A. Lamprecht, “Artificial neural network based particle size prediction of polymeric nanoparticles,” *European Journal of Pharmaceutics and Biopharmaceutics*, vol. 119, pp. 333–342, 2017.

[11] I. Khan, K. Saeed, and I. Khan, “Nanoparticles: properties, applications and toxicities,” *Arabian Journal of Chemistry*, vol. 12, no. 7, pp. 908–931, 2019.

[12] N. Yadav and S. Khatak, “Eudragit-based nanocarriers: a review of recent applications in drug delivery,” *J. Drug Deliv. Sci. Technol.*, vol. 55, p. 101478, 2020.

[13] Y. Song and others, “Recent advances in polymeric nanoparticle-based drug delivery systems,” *Colloids Surf. B Biointerfaces*, vol. 204, p. 111812, 2021.

[14] R. Ferreira-Nunes, M. Cunha-Filho, T. Gratieri, and G. M. Gelfuso, “Follicular-targeted delivery of spironolactone provided by polymeric nanoparticles,” *Colloids Surf. B Biointerfaces*, vol. 208, p. 112101, 2021.

[15] A. W. Zaibudeen, “Recent advances in different nanoprecipitation methods for efficient drug loading and controlled release,” *Journal of Nanotechnology and Nanomaterials*, vol. 6, no. 1, pp. 12–25, 2025.

[16] F. Q. Pires *et al.*, “Use of mixture design in drug-excipient compatibility determinations: thymol nanoparticles case study,” *J. Pharm. Biomed. Anal.*, vol. 137, pp. 196–203, 2017.

[17] P. A. Granados *et al.*, “Application of Theoretical Solubility Calculations and Thermal and Spectroscopic Measurements to Guide the Processing of Triamcinolone Acetonide by Hot-Melt Extrusion,” *Pharmaceutics*, vol. 17, no. 5, p. 586, Apr. 2025, doi: 10.3390/pharmaceutics17050586.

[18] Z. Zhou, C. Qiu, and Y. Zhang, “A comparative analysis of linear regression, neural networks and random forest regression for predicting air ozone employing soft sensor models,” *Sci. Rep.*, vol. 13, p. 22420, 2023.

[19] M. Z. Alom, T. M. Taha, C. Yakopcic, and others, “A state-of-the-art survey on deep learning theory and architectures,” *Electronics (Basel).*, vol. 8, no. 3, p. 292, 2019.

[20] X. Wu, V. Kumar, J. R. Quinlan, and others, “Top 10 algorithms in data mining,” *Knowl. Inf. Syst.*, vol. 14, no. 1, pp. 1–37, 2008.

[21] W. Pannakkong, K. Thiwa-Anont, K. Singthong, P. Parthanadee, and J. Buddhakulsomsiri, “Hyperparameter tuning of machine learning algorithms using response surface methodology: a case study of ANN, SVM, and DBN,” *Math. Probl. Eng.*, vol. 2022, p. 7958647, 2022.

[22] L. A. Yates, Z. Aandahl, S. A. Richards, and B. W. Brook, “Cross validation for model selection: A review with examples from ecology,” *Ecol. Monogr.*, vol. 93, no. 1, Feb. 2023, doi: 10.1002/ecm.1557.

[23] F. Mohr and J. N. van Rijn, “Learning curves for decision making in supervised machine learning: a survey,” *Mach. Learn.*, vol. 113, no. 11–12, pp. 8371–8425, Dec. 2024, doi: 10.1007/s10994-024-06619-7.

[24] T. G. Dietterich, “An experimental comparison of three methods for constructing

ensembles of decision trees: bagging, boosting, and randomization," *Mach. Learn.*, vol. 40, no. 2, pp. 139–157, 2000.

[25] T. Bastogne, "Quality-by-design of nanopharmaceuticals—a state of the art," *Nanomedicine*, vol. 13, no. 7, pp. 2151–2157, 2017.

[26] S. A. Abd-Algaleel, H. M. Abdel-Bar, A. A. Metwally, and R. M. Hathout, "Evolution of the computational pharmaceutics approaches in the modeling and prediction of drug payload in lipid and polymeric nanocarriers," *Pharmaceuticals*, vol. 14, no. 7, p. 645, 2021.

[27] A. Yusuf, A. R. Z. Almotairy, H. Henidi, O. Y. Alshehri, and M. S. Aldughaim, "Nanoparticles as Drug Delivery Systems: A Review of the Implication of Nanoparticles' Physicochemical Properties on Responses in Biological Systems," *Polymers (Basel).*, vol. 15, no. 7, p. 1596, Mar. 2023, doi: 10.3390/polym15071596.

[28] G. Naidu, T. Zuva, and E. M. Sibanda, "A Review of Evaluation Metrics in Machine Learning Algorithms," 2023, pp. 15–25. doi: 10.1007/978-3-031-35314-7_2.

[29] M. Imani and H. R. Arabnia, "Hyperparameter Optimization and Combined Data Sampling Techniques in Machine Learning for Customer Churn Prediction: A Comparative Analysis," *Technologies (Basel).*, vol. 11, no. 6, p. 167, Nov. 2023, doi: 10.3390/technologies11060167.

[30] T. O. Hodson, "Root-mean-square error (RMSE) or mean absolute error (MAE): when to use them or not," *Geosci. Model Dev.*, vol. 15, no. 14, pp. 5481–5487, Jul. 2022, doi: 10.5194/gmd-15-5481-2022.

[31] S. Đorđević *et al.*, "Current hurdles to the translation of nanomedicines from bench to the clinic," *Drug Deliv. Transl. Res.*, vol. 12, no. 3, pp. 500–525, Mar. 2022, doi: 10.1007/s13346-021-01024-2.

[32] V. K. Sharma, A. Sharma, K. K. Verma, P. K. Gaur, R. Kaushik, and B. Abdali,

"A COMPREHENSIVE REVIEW ON PHARMACOLOGICAL POTENTIALS OF CAFFEINE," *Journal of Applied Pharmaceutical Sciences and Research*, vol. 6, no. 3, pp. 16–26, Dec. 2023, doi: 10.31069/japsr.v6i3.04.

[33] E. C. Arrúa, K. P. Seremeta, G. R. Bedogni, N. B. Okulik, and C. J. Salomon, "Nanocarriers for effective delivery of benznidazole and nifurtimox in the treatment of chagas disease: A review," *Acta Trop.*, vol. 198, p. 105080, Oct. 2019, doi: 10.1016/j.actatropica.2019.105080.

[34] J. Heykants *et al.*, "The Clinical Pharmacokinetics of Itraconazole: An Overview," *Mycoses*, vol. 32, no. s1, pp. 67–87, Nov. 1989, doi: 10.1111/j.1439-0507.1989.tb02296.x.

[35] F. Rommasi and N. Esfandiari, "Liposomal Nanomedicine: Applications for Drug Delivery in Cancer Therapy," *Nanoscale Res. Lett.*, vol. 16, no. 1, p. 95, May 2021, doi: 10.1186/s11671-021-03553-8.

[36] W. Zhang, C. Wu, H. Zhong, Y. Li, and L. Wang, "Prediction of undrained shear strength using extreme gradient boosting and random forest based on Bayesian optimization," *Geoscience Frontiers*, vol. 12, no. 1, pp. 469–477, 2021.

[37] P. V. Devarajan, G. S. Sonavane, and M. Doble, "Computer-Aided Molecular Modeling: A Predictive Approach in the Design of Nanoparticulate Drug Delivery System," *J. Biomed. Nanotechnol.*, vol. 1, no. 4, pp. 375–383, Dec. 2005, doi: 10.1166/jbn.2005.051.

[38] H. Asadi, K. Rostamizadeh, D. Salari, and M. Hamidi, "Preparation of biodegradable nanoparticles of tri-block PLA–PEG–PLA copolymer and determination of factors controlling the particle size using artificial neural network," *J. Microencapsul.*, vol. 28, no. 5, pp. 406–416, Aug. 2011, doi: 10.3109/02652048.2011.576784.

[39] R. Furtuna, S. Curteanu, and C. Racles, "NSGA-II-RJG applied to multi-objective

optimization of polymeric nanoparticles synthesis with silicone surfactants," *Open Chem.*, vol. 9, no. 6, pp. 1080–1095, Dec. 2011, doi: 10.2478/s11532-011-0096-5.

[40] F. Zabihi, N. Xin, J. Jia, T. Chen, and Y. Zhao, "High Yield and High Loading Preparation of Curcumin–PLGA Nanoparticles Using a Modified Supercritical Antisolvent Technique," *Ind. Eng. Chem. Res.*, vol. 53, no. 15, pp. 6569–6574, Apr. 2014, doi: 10.1021/ie404215h.

5 *Software Architecture and Technological Implementation of the NANO-AI Platform**

**software currently undergoing registration with the Brazilian National Institute of Industrial Property (INPI)*

This chapter elucidates the technological infrastructure, data engineering frameworks, and computational workflows that constitute the NANO-AI platform. To ensure methodological rigor within the architectural exposition, the operational pipeline encompasses twelve distinct active pharmaceutical ingredients (APIs) central to this study: paracetamol, propranolol, dutasteride, spironolactone, minoxidil, fluconazole, naringenin, isoniazid, caffeine, itraconazole, finasteride, and nifurtimox. The software ecosystem was engineered to transmute raw bench-scale experimental data into high-fidelity interactive predictions, achieved through the integration of high-level programming languages, relational database management systems (RDBMS), and Business Intelligence (BI) analytics, as illustrated in Figure 1.

5.1 *Data Acquisition, Ingestion, and Structuring (ETL Pipeline)*

The predictive evolution of the NANO-AI platform is underpinned by a continuous active learning framework. Experimental nanoparticle formulations, documented by operators via cloud-synchronized spreadsheets, are integrated into a local Linux Ubuntu computational environment for processing. Data ingestion is initiated by a custom Python-based automated pipeline that retrieves raw data and systematically partitions it into three structured CSV (Comma-Separated Values) datasets: experimental formulation parameters, quantitative physicochemical characterization, and qualitative outcome classifications.

Following extraction, these datasets are migrated to a PostgreSQL relational

database. The architectural design segregates data into three primary tables:

- **Independent Variables:** Captures process and compositional parameters, such as phase volumes, surfactant concentrations, polymer ratios, and specific thermodynamic conditions (e.g., stirring type and temperature).

- **Physicochemical Metrics:** Records quantitative outputs including encapsulation efficiency, mean hydrodynamic diameter, polydispersity index (PDI), and zeta potential.

- **Qualitative Classifications:** Documents experimental success or failure (e.g., after production precipitation).

To augment this operational data with chemical intelligence, a supplementary relational table provides molecular descriptors of the active pharmaceutical ingredients (APIs), including molecular weight, LogP, topological polar surface area (TPSA), and Hansen solubility parameters.Once the data have been successfully ingested into the source tables, a structured SQL routine is executed to perform comprehensive data cleansing and standardization procedures. This transformation pipeline ensures numerical consistency by systematically replacing commas with decimal points across all physicochemical measurements, while categorical process variables are standardized through whitespace removal and translation into English. In addition, stringent quality-control filters are applied to automatically eliminate inconsistent experimental records, specifically excluding entries in which the sum of phase volumes differs from 20 mL or the total surfactant mass deviates from 50 mg. During this stage, the script also generates

the target variables required for formulation success prediction, assigning a value of “1” to systems exhibiting a formation probability greater than 0.5 (50%) and “0” otherwise, besides the formation probability itself as a variable. Subsequently, the refined tables are integrated through relational joins and consolidated into a definitive destination table. This unified matrix aggregates formulation parameters, physicochemical outcomes, qualitative classifications, and molecular descriptors into a single standardized dataset. Finally, a third Python script establishes a direct connection to the PostgreSQL database and exports the consolidated data into two large CSV files: one containing the complete dataset and another comprising only successful formulations. These files are then automatically stored within a synchronized local directory, serving as the standardized input source for subsequent predictive model training, including both binary classification and regression tasks.

5.2 *Machine Learning Model Training*

The initial dataset described in Section 5.1 serves as the basis for the STEP 1 binary classification model. Developed via the H2O framework in R, this computational stage quantifies the likelihood of formulation viability. Addressing the challenge of class imbalance, the methodology integrates a strategic preprocessing routine: k-medoids clustering via the CLARA algorithm is utilized to perform informed undersampling. This approach prioritizes the retention of representative data points, maintaining structural and chemical diversity while neutralizing the statistical skew of the majority class prior to model training.

The computational pipeline performs a Random Discrete Grid Search, systematically evaluating hundreds of hyperparameter combinations across multiple

machine learning algorithms, including Generalized Linear Models (GLM), Naive Bayes, Random Forest, Extreme Gradient Boosting (XGBoost), and Deep Learning-based Artificial Neural Networks. The optimization grids extensively explore critical algorithmic parameters, such as the number of trees, maximum tree depth, sampling rates, and learning rates for ensemble-based methods. For Deep Learning models, the search space includes diverse hidden-layer architectures, activation functions incorporating dropout regularization mechanisms (e.g., *RectifierWithDropout* and *TanhWithDropout*), as well as L1 and L2 regularization penalties to enhance generalization performance. Model convergence is dynamically controlled through early stopping criteria based on the Area Under the Precision–Recall Curve (AUCPR) and a stringent stopping tolerance threshold, thereby interrupting training once performance stabilization is detected and minimizing the risk of structural overfitting. At the conclusion of the project, the resulting dataset comprised 1006 experimental records, with the distribution of drugs represented in Figure 2. As discussed in Chapter 4, the XGBoost algorithm demonstrated superior predictive performance among all evaluated models and was therefore selected as the final model for the binary classification stage.

Subsequently, the second CSV file, a dataset comprising 485 experimental records enriched with the nanoparticle formation probability generated by the first predictive model, selectively filters viable colloidal systems and serves as the empirical input for the regression stage, which was also developed in the R programming language. This stage quantitatively predicts the critical quality attributes (CQAs) of the resulting nanoparticles, including hydrodynamic diameter, polydispersity index (PDI), encapsulation efficiency, and zeta potential.

To effectively capture the highly nonlinear physicochemical relationships governing these continuous variables, the hyperparameter optimization process in this

phase focuses primarily on Random Forest, XGBoost, and Deep Learning algorithms. The tuning boundaries permit increased model complexity to accommodate substantial functional variability, including deeper decision-tree structures (with maximum depths of up to 30) and expanded estimator limits (reaching up to 1,000 trees) for XGBoost models, while simultaneously incorporating stringent L1 and L2 regularization constraints to reduce the risk of overfitting.

For the Deep Learning architectures, conservative learning rates ranging from 0.001 to 0.01 and carefully condensed network topologies are employed to stabilize gradient descent during nanoparticle size and surface charge prediction tasks. Model selection in this regression stage is strictly governed by the minimization of the Root Mean Square Error (RMSE). Furthermore, to ensure an unbiased estimation of model generalization performance, k-fold cross-validation strategies are systematically applied, with the number of folds varying between 3 and 7 according to the target physicochemical attribute. This stratified approach to cross-validation serves to ensure that the predictive performance remains consistent across different subsets of the data, thereby reducing the likelihood of overfitting and enhancing the reliability of the findings within the context of nanobiotechnological characterization.

As discussed in Chapter 4, the XGBoost algorithm demonstrated superior predictive performance for the variables polydispersity index (PDI), encapsulation efficiency, and zeta potential, whereas Deep Learning models outperformed all other evaluated approaches in predicting nanoparticle size.

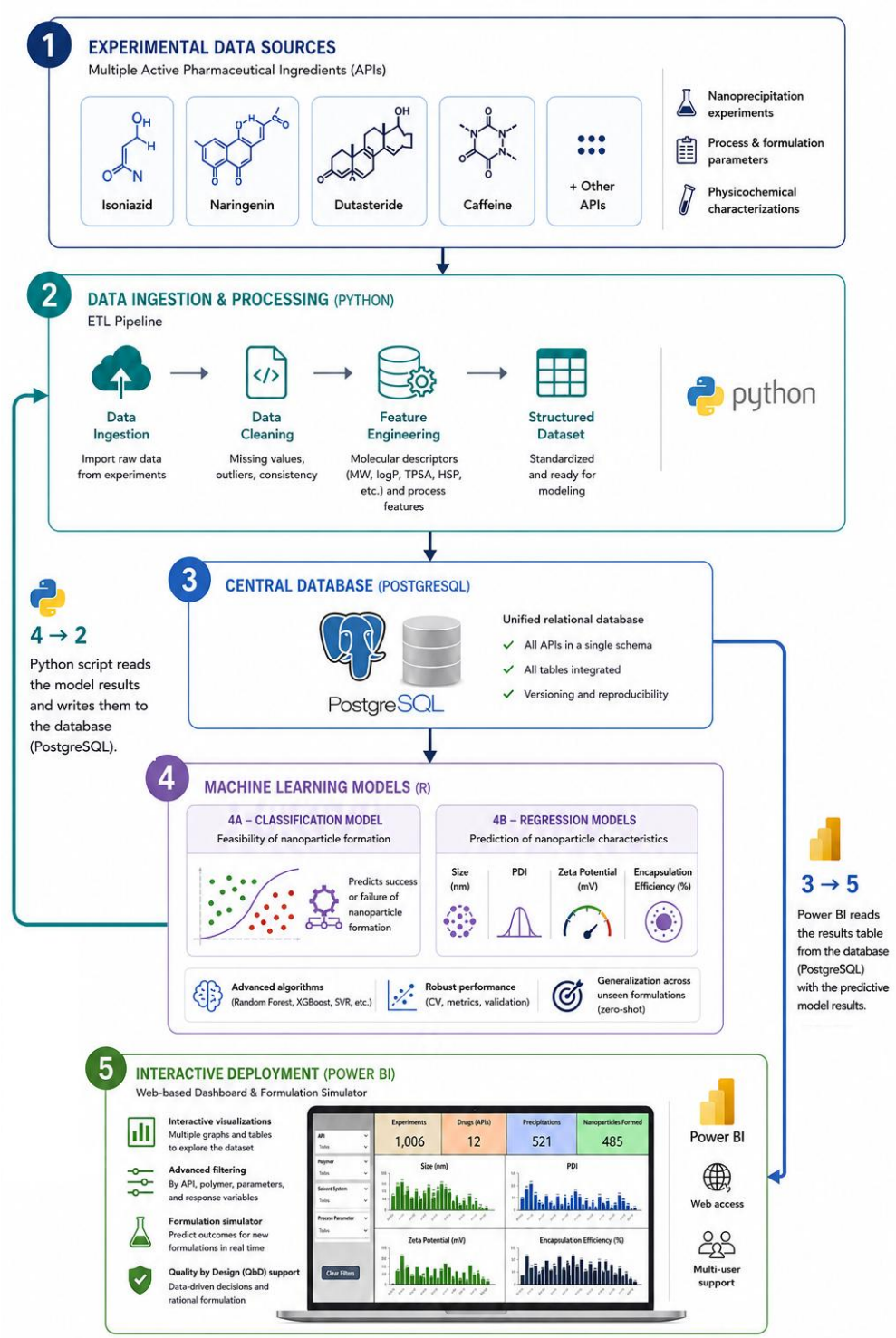


Figure 1. The computational architecture of the NANO-AI platform is organized into five distinct functional layers. The first layer comprises the primary experimental data source, consisting of researcher-generated spreadsheets documenting nanoparticle formulations. The second layer constitutes the Python-based ETL (Extract, Transform, Load) pipeline, encompassing four principal scripts responsible for data transformation, database ingestion, record retrieval, and the generation of synthetic datasets. The third layer utilizes a PostgreSQL relational database system, which structures and stores the processed data within interconnected tables. The fourth layer serves as the predictive engine, housing the trained classification and regression models. Finally, the fifth layer is the visualization and deployment interface, utilizing Power BI to host the NANO-AI formulation simulator dashboard.

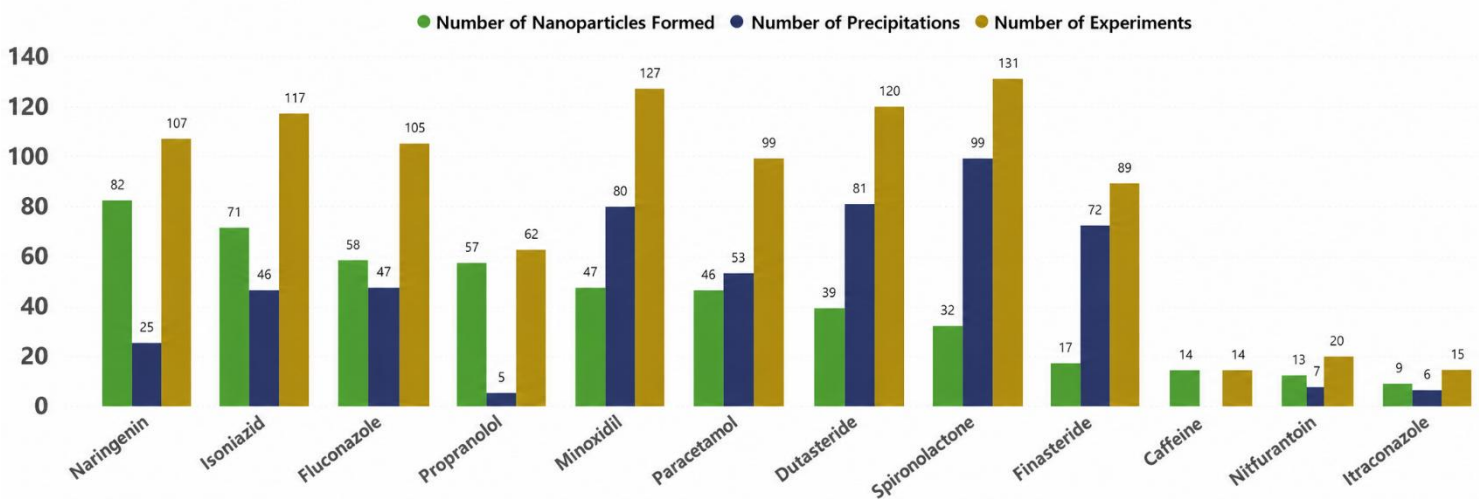


Figure 2. Illustration of the distribution of the 1,006 experimental records across the investigated drug set. The core developmental datasets were generated following the protocol established for isoniazid (Chapter 3), with each drug undergoing 3 experimental rounds to yield 90 or more unique formulations. Although caffeine, nifurtimox, and itraconazole were utilized for external validation, the NANO-AI platform employs an active learning framework; consequently, these validation data were reincorporated into the relational database to further optimize model accuracy as part of a continuous refinement cycle.

5.3 Deployment and the Interactive Power BI Simulator

After the classification and regression models had been fully trained, an extensive dataset was generated containing all possible formulation combinations for the 12 investigated drugs. Regarding the input variables, formulations were systematically created using combinations of aqueous and organic phase volumes ranging from 5 to 15 mL, liquid and solid surfactant quantities varying between 0 and 50 mg, and anionic and cationic polymer concentrations ranging from 10 to 100 mg. In addition, random and equally distributed assignments were applied to categorical process variables, including the phase mixing method (pouring or burette addition), agitation type (magnetic stirring or Ultra-Turrax homogenization), and processing temperature (25 or 50 °C).

Following dataset generation, all simulated formulations violating predefined physicochemical constraints were automatically excluded, specifically those in which the combined aqueous and organic phase volumes differed from 20 mL or the total surfactant mass

deviated from 50 mg. At the conclusion of this process, approximately 60,000 virtual formulations were generated for each drug, resulting in a final dataset comprising nearly 720,000 simulated formulations.

All generated formulations were subsequently processed by the STEP 1 classification model to estimate the probability of nanoparticle formation. The resulting dataset was then forwarded to the regression model, which generated approximately 720,000 predictions for nanoparticle size, polydispersity index (PDI), encapsulation efficiency, and zeta potential across the 12 investigated drugs. These predictive outputs were consolidated and stored as an additional large-scale CSV file within the synchronized cloud-mapped local directory.

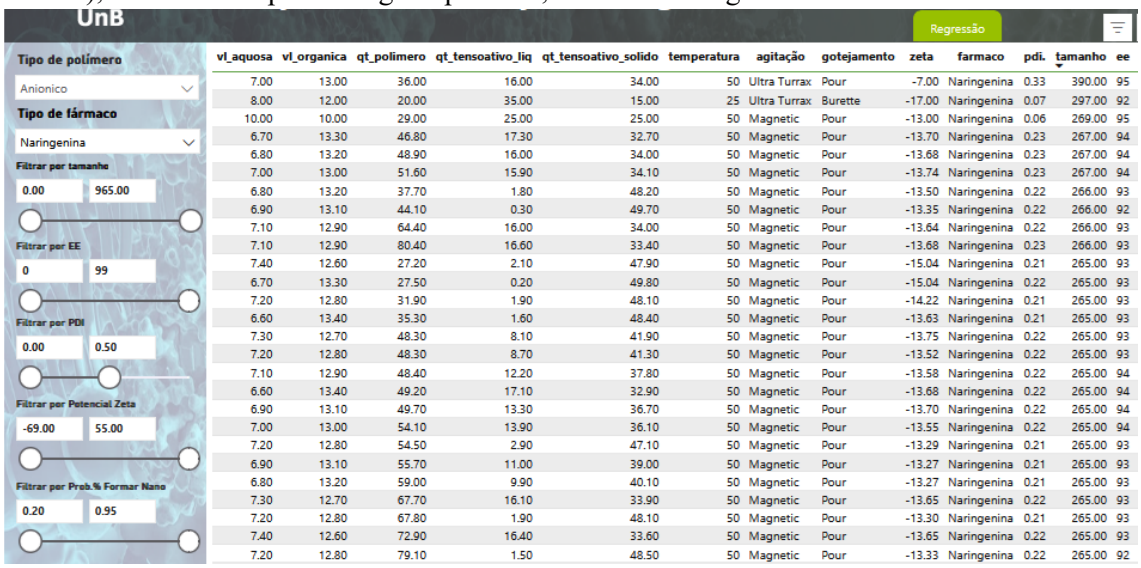
Finally, a fourth Python script reads this CSV file and imports its contents into a dedicated PostgreSQL database table, which is subsequently accessed and processed by the Power BI analytical platform, which materializes the NANO-AI platform interface. A dashboard of this BI tool functions as an interactive formulation simulator, presenting statistical charts of the simulated data and allowing the application of dynamic filters tailored to the researcher's preferences (e.g., specific API, polymer type, target size range, etc.). By enabling the prior virtual exploration of formulations based on algorithmic predictions, the innovation of this platform potentially minimizes the reliance on empirical trial-and-error strategies, thereby conserving research time and preventing the waste of laboratory materials.

5.4 *Interactive Formulation Simulator and Predictive Dashboard*

The NANO-AI platform reaches technical maturity with the implementation of its interactive formulation simulator. Within this Business Intelligence (BI) interface, formulators are presented with a digital environment engineered to facilitate the reverse-engineering of polymeric nanoparticles tailored to specific developmental exigencies. The dashboard encompasses a array of interactive parameters, enabling users to delineate target criteria by selecting the polymer matrix, the active pharmaceutical ingredient (API), and the requisite numerical ranges for Critical Quality Attributes (CQAs), specifically hydrodynamic diameter, encapsulation efficiency, polydispersity index (PDI), and zeta potential. Moreover, researchers

can impose probability thresholds (confidence intervals) regarding successful nanoparticle synthesis, ensuring the system prioritizes the recommendation of highly viable colloidal dispersions (Figure 3). By leveraging a pre-calculated multi-dimensional database, the system enables nearly instantaneous computational interrogation of the entire formulation space.

Upon applying these target constraints, the simulator dynamically queries the underlying predictive dataset and outputs the exact formulation and process parameters necessary to synthesize nanoparticles matching the requested profile. The system provides a detailed, ready-to-execute experimental protocol that dictates the optimal volumes for the aqueous and organic phases, the precise required mass of liquid and solid surfactants, the appropriate stirring mechanism (differentiating between standard magnetic stirring and high-shear homogenization via Ultra Turrax), the phase mixing methodology (direct pouring versus controlled dripping via a burette), and the ideal processing temperature, as shown in Figure 3.



vl_aquosa	vl_organica	qt_polimero	qt_tensoativo_liq	qt_tensoativo_solido	temperatura	agitação	gotejamento	zeta	farmaco	pdi.	tamanho	ee
7.00	13.00	36.00	16.00	34.00	50	Ultra Turrax	Pour	-7.00	Naringenina	0.33	390.00	95
8.00	12.00	20.00	35.00	15.00	25	Ultra Turrax	Burette	-17.00	Naringenina	0.07	297.00	92
10.00	10.00	29.00	25.00	25.00	50	Magnetic	Pour	-13.00	Naringenina	0.06	269.00	95
6.70	13.30	46.80	17.30	32.70	50	Magnetic	Pour	-13.70	Naringenina	0.23	267.00	94
6.80	13.20	48.90	16.00	34.00	50	Magnetic	Pour	-13.68	Naringenina	0.23	267.00	94
7.00	13.00	51.60	15.90	34.10	50	Magnetic	Pour	-13.74	Naringenina	0.23	267.00	94
6.80	13.20	37.70	1.80	48.20	50	Magnetic	Pour	-13.50	Naringenina	0.22	266.00	93
6.90	13.10	44.10	0.30	49.70	50	Magnetic	Pour	-13.35	Naringenina	0.22	266.00	92
7.10	12.90	64.40	16.00	34.00	50	Magnetic	Pour	-13.64	Naringenina	0.22	266.00	93
7.10	12.90	80.40	16.60	33.40	50	Magnetic	Pour	-13.68	Naringenina	0.23	266.00	93
7.40	12.60	27.20	2.10	47.90	50	Magnetic	Pour	-15.04	Naringenina	0.21	265.00	93
6.70	13.30	27.50	0.20	49.80	50	Magnetic	Pour	-15.04	Naringenina	0.22	265.00	93
7.20	12.80	31.90	1.90	48.10	50	Magnetic	Pour	-14.22	Naringenina	0.21	265.00	93
6.60	13.40	35.30	1.60	48.40	50	Magnetic	Pour	-13.63	Naringenina	0.21	265.00	93
7.30	12.70	48.30	8.10	41.90	50	Magnetic	Pour	-13.75	Naringenina	0.22	265.00	93
7.20	12.80	48.30	8.70	41.30	50	Magnetic	Pour	-13.52	Naringenina	0.22	265.00	93
7.10	12.90	48.40	12.20	37.80	50	Magnetic	Pour	-13.58	Naringenina	0.22	265.00	94
6.60	13.40	49.20	17.10	32.90	50	Magnetic	Pour	-13.68	Naringenina	0.22	265.00	94
6.90	13.10	49.70	13.30	36.70	50	Magnetic	Pour	-13.70	Naringenina	0.22	265.00	94
7.00	13.00	54.10	13.90	36.10	50	Magnetic	Pour	-13.55	Naringenina	0.22	265.00	94
7.20	12.80	54.50	2.90	47.10	50	Magnetic	Pour	-13.29	Naringenina	0.21	265.00	93
6.90	13.10	55.70	11.00	39.00	50	Magnetic	Pour	-13.27	Naringenina	0.21	265.00	93
6.80	13.20	59.00	9.90	40.10	50	Magnetic	Pour	-13.27	Naringenina	0.21	265.00	93
7.30	12.70	67.70	16.10	33.90	50	Magnetic	Pour	-13.65	Naringenina	0.22	265.00	93
7.20	12.80	67.80	1.90	48.10	50	Magnetic	Pour	-13.30	Naringenina	0.21	265.00	93
7.40	12.60	72.90	16.40	33.60	50	Magnetic	Pour	-13.65	Naringenina	0.22	265.00	93
7.20	12.80	79.10	1.50	48.50	50	Magnetic	Pour	-13.33	Naringenina	0.22	265.00	92

Figure 3. The NANO-AI simulator interface provides a dual-panel configuration for real-time formulation design. The lateral control panel features a suite of interactive filters that allow for the dynamic modulation of nanoparticle attributes, including polymer matrix selection, the target active pharmaceutical ingredient (API), and specific numerical intervals for hydrodynamic size, encapsulation efficiency, polydispersity index (PDI), and zeta potential. Furthermore, users can calibrate a statistical probability threshold for nanoparticle formation. Adjustments to these parameters trigger instantaneous updates in the primary results table, which enumerates the corresponding formulation variables. These variables encompass phase volumes (aqueous and organic), surfactant concentrations, and specific process parameters, such as the mode of agitation and the thermodynamic conditions maintained during the mixing procedure.

Appendix A — Supplementary Material for Chapter IV: Hyperparameter Configurations and Detailed Results

Table S1. Molecular descriptors for each active pharmaceutical ingredient (API) used to build the NANO-AI platform, i.e., the Hansen solubility parameters including dispersion (δ_D), polar (δ_P), and hydrogen bonding (δ_H) components, as well as the total solubility parameter (δ_t), the molecular weight (MW) and topological polar surface area (tpsa).

API	δ_D	δ_P	δ_H	δ_t	MW	logP	tpsa
Isoniazid	20.9	16.3	13.4	29.7	137.14	-0.70	68.01
Paracetamol	20.2	13.3	14.4	28.1	151.16	0.46	49.30
Propranolol	19.1	5.4	8.9	21.7	259.34	3.48	41.49
Dutasteride	17.8	8.0	3.6	19.9	528.53	5.79	58.20
Spironolactone	19.6	3.5	3.0	20.1	416.57	2.20	79.90
Minoxidil	18.7	8.1	21	29.3	289.31	-0.79	138.48
Fluconazole	19.7	11.6	8.7	24.5	306.27	0.61	55.10
Naringenin	20.4	9.5	12.9	25.9	272.25	2.52	98.30
Finasteride	18.2	8.2	3.6	20.3	372.55	3.03	58.20

Table S2. Comparison between predicted and experimentally measured nanoparticle properties obtained during interlaboratory validation. Results are presented for different operators and laboratory environments (City 1 and City 2), including mean hydrodynamic diameter (nm), polydispersity index (PDI), zeta potential (mV) and efficiency encapsulation (%). Percentage errors are calculated relative to experimental values for each condition. "NanoAI" represents model predictions, while C1-OP1, C2-OP1, and C2-OP2 correspond to experimental results obtained by Operator 1 in City 1, Operator 1 in City 2, and Operator 2 in City 2, respectively. Not determined (nd) indicate conditions not experimentally evaluated.

SIZE				PDI				ZETA POTENTIAL				EE			
Predicte	Error(%)	Error(%)	Error(%)	Predicted	Error(%)	Error(%)	Error(%)	Predicted	Error(%)	Error(%)	Error(%)	Predicted	Error(%)	Error(%)	Error(%)
NanoAI	C1-OP1	C2-OP1	C2-OP2	NanoAI	C1-OP1	C2-OP1	C2-OP2	NanoAI	C1-OP1	C2-OP1	C2-OP2	NanoAI	C1-OP1	C2-OP1	C2-OP2
76.37	23.9%	24.1%	46.3%	**0.23**	-65.22%	-82.61%	-65.22%	**-13.99**	-2.72%	-35.03%	49.96%	**13.0**	46.15%	30.77%	92.31%
75.97	11.69%	31.37%	53.93%	**0.29**	-93.10%	-31.03%	10.34%	**-15.44**	11.27%	-2.72%	22.28%	**16.0**	12.5%	68.75%	96.25%
295.32	nd	nd	nd	**0.09**	nd	nd	nd	**55.87**	nd	nd	nd	**13.0**	nd	nd	nd
70.69	11.77%	22.80%	36.34%	**0.24**	12.50%	4.17%	25.00%	**-16.48**	-54.37%	-54.43%	2.91%	**11.0**	9.09%	0%	63.64%
74.35	-41.63%	nd	-61.40%	**0.21**	33.33%	nd	38.10%	**53.61**	40.37%	nd	17.93%	**9.0**	100%	nd	77.78%
276.30	21.50%	-18.36%	-0.62%	**0.26**	57.69%	80.77%	57.69%	**-21.84**	-12.36%	-15.43%	-73.99%	**7.0**	-100%	-100%	0%
133.14	-16.43%	33.50%	-0.65%	**0.13**	-7.69%	-107.69%	7.69%	**-21.35**	-43.37%	-1.97%	-92.04%	**12.0**	-41.67%	-58.33%	33.33%

141.92	-43.48%	nd	nd	**0.13**	0.00%	nd	nd	**-12.24**	4.41%	nd	nd	**5.0**	100%	nd	nd
146.09	-33.84%	-22.84%	-9.52%	**0.24**	-4.17%	20.83%	8.33%	**27.21**	-29.47%	-37.71%	-94.78%	**5.0**	-100%	100%	100%
84.47	-34.09%	45.95%	51.46%	**0.35**	42.86%	-31.43%	-25.71%	**-13.76**	6.83%	-42.66%	-67.15%	**5.0**	100%	100%	-100%

Table S3. Summary of the validation of the three drugs—itraconazole, caffeine, and nifurtimox—regarding particle size, encapsulation efficiency, zeta potential, and polydispersity index (PDI), in terms of the mean absolute error (MAE) and the standard deviation of percentage values (SDV)

	SIZE		PDI		EE		ZETA	
Drug	**MAE**	**SDV**	**MAE**	**SDV**	**MAE**	**SDV**	**MAE**	**SDV**
Itraconazole	28,55%	± 20,425%	43,12%	± 29,05%	46,02%	± 9,07%	13,89%	± 14,60%
Nifurtimox	34,52%	± 24,46%	39,31%	± 21,97%	12,96%	± 9,38%	7,14%	± 7,65%
Caffeine	21,15%	± 18,74%	38,75%	± 26,55%	55,84%	± 32,15%	15,84%	± 20,69%

Table S4. The models were optimized through hyperparameter tuning and cross-validation. The XGBoost model was configured with 100 trees and a low learning rate (0.01), enabling gradual and stable learning. Model complexity was controlled with a max depth of 10, while both L1 (alpha = 0.1) and L2 (lambda = 5) regularization reduced overfitting by penalizing excessive weight growth. Additional robustness was introduced through stochastic subsampling of rows and features (sample_rate and col_sample_rate = 0.8), which improves generalization in heterogeneous datasets. A low min_rows value (2) allowed deeper tree structures, enhancing the model's ability to capture complex nonlinear relationships. Five-fold cross-validation ensured reliable performance estimation. The Deep Learning model was designed with a compact architecture to avoid overfitting, using a single hidden layer and Rectifier activation with dropout. Regularization was a central component: L1 and L2 penalties controlled weight magnitude, while dropout (10%) randomly deactivated neurons during training, improving robustness. Training stability was enhanced through a moderate learning rate (0.02) with annealing, and momentum (0.6 → 0.95) to accelerate convergence and reduce oscillations. Early stopping based on rmse (patience of 20 rounds) prevented overtraining, while 500 epochs allowed sufficient learning. Cross-validation (5-fold) further ensured model reliability. In comparison, Random Forest used a shallow and efficient configuration (10 trees, max_depth = 15), focusing on stability and reduced variance. GLM applied elastic net regularization (alpha = 0.5, lambda = 0.1) to balance feature selection and coefficient shrinkage, while Naive Bayes relied on smoothing techniques to handle numerical instability.

	Hyperparameters	
Algorithm	***Name***	***Value***
Random Forest	ntrees	10
	maxdepth	15
	samplerate	1
	mtries	0
	minrows	15
	histtype	QuantilesGlobal
	kfold	5

XGBoost	ntrees	100
	learnrate	0.01
	maxdepth	10
	reglambda	5
	regalpha	0.1
	samplerate	0.8
	colsamplerate	0.8
	minrows	2
	kfold	5
GLM	alfa	0.5
	lamba	1.00E-01
	kfold	3
Naive Bayes	laplace	0.1
	minsdev	0.05
	epssdev	1.00E-02
	kfold	3
Deep Learning	hiddenlayers	10
	activfunction	ReLU
	l1	1.00E-04
	l2	1.00E-04
	inputDrp	0.3
	hideDrp	0.1
	learnrate	1.00E-02
	rateanne	1.00E-07
	momstart	0.6
	momstable	0.95
	stoppround	20
	stopmetr	rmse
	stoptol	1.00E-03
	epochs	500
	kfold	5