DEVELOPMENT OF HYBRID MACHINE LEARNING MODELS FOR ENHANCED QUALITY CONTROL AND OPTIMIZATION OF RESVERATROL-LOADED POLYMERIC NANOPARTICLES

BY

PHUVAMIN SURIYAAMPORN

AN INDEPENDENT STUDY SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF MASTER OF ENGINEERING (ARTIFICIAL INTELLIGENCE AND INTERNET OF THINGS)
SIRINDHORN INTERNATIONAL INSTITUTE OF TECHNOLOGY
THAMMASAT UNIVERSITY
ACADEMIC YEAR 2025

THAMMASAT UNIVERSITY
SIRINDHORN INTERNATIONAL INSTITUTE OF TECHNOLOGY

INDEPENDENT STUDY

BY

PHUVAMIN SURIYAAMPORN

ENTITLED

DEVELOPMENT OF HYBRID MACHINE LEARNING MODELS FOR ENHANCED QUALITY CONTROL AND OPTIMIZATION OF RESVERATROL-LOADED POLYMERIC NANOPARTICLES

was approved as partial fulfillment of the requirements for
the degree of Master of Engineering (Artificial Intelligence and Internet of Things)

on November 24, 2025

Member and Advisor

(Associate Professor Warut Pannakkong, Ph.D.)

Member

(Associate Professor Jirachai Buddhakulsomsiri, Ph.D.)

Director

(Associate Professor Kriengsak Panuwatwanich, Ph.D.)

Independent Study Title	DEVELOPMENT OF HYBRID MACHINE LEARNING MODELS FOR ENHANCED QUALITY CONTROL AND OPTIMIZATION OF RESVERATROL-LOADED POLYMERIC NANOPARTICLES
Author	Phuvamin Suriyaamporn
Degree	Master of Engineering (Artificial Intelligence and Internet of Things)
Faculty/University	Sirindhorn International Institute of Technology/ Thammasat University
Advisor	Associate Professor Warut Pannakkong, Ph.D.
Academic Years	2025

ABSTRACT

The formulation of resveratrol-loaded polymeric nanoparticles (RES-PNPs) is hindered by resveratrol's low solubility, instability, and rapid clearance. This study introduces an integrated hybrid machine-learning (ML), genetic-algorithm (GA), and reinforcement-learning (RL) framework to rationally design RES-PNPs and improve their anticancer performance. Hybrid ML models combining linear regression (LR), k-nearest neighbors (k-NN), and artificial neural networks (ANN) were constructed to predict particle size (PS), polydispersity index (PDI), zeta potential (ZP), and drug loading (%DL). The hybrid models substantially outperformed single learners (e.g., ANN RMSE: 69.19 for PS, 0.06 for PDI, 4.28 mV for ZP; k-NN RMSE: 6.69 for %DL), reducing error by 15–40% across endpoints. Final hybrid RMSE values were 55.12 (PS), 0.05 (PDI), 3.90 mV (ZP), and 5.21 (%DL), demonstrating strong predictive fidelity. Optimization employed a GA with a population of 100,000 individuals, 100 generations, and 10 RL-tuned episodes. Fitness was defined as f = –PS – PDI – ZP + %DL, with the best episodes achieving a terminal fitness of –1.025. RL adaptively refined GA parameters, converging on crossover = 0.10 and mutation = 0.18, which improved stability and reduced variance relative to fixed settings. The optimal solution

identified (PAA = 0.30, GT = 0.13, P407 = 8.11, Hz = 11.56, Time = 12.50) produced PS = 80 nm, PDI = 0.31, ZP = –36.94 mV, and %DL = 68.02%, all within predefined feasibility limits. Experimental validation showed no significant differences between predicted and measured CQAs ($p > 0.05$). This AI-augmented workflow establishes a robust design space aligned with ICH Q10 and demonstrates a powerful strategy for intelligent nanomedicine formulation and optimization.

Keywords: Resveratrol, Polymeric nanoparticles, Hybrid machine learning models, Genetic algorithm, Reinforcement learning

ACKNOWLEDGEMENTS

This independent study was successfully completed thanks to the kind support and valuable assistance of Associate Professor Dr. Warut Pannakkong, who generously dedicated his time to provide guidance throughout the research process. His careful review, constructive feedback, and correction of various shortcomings were of great benefit in completing this study from the initial stages to its finalization. The researcher would like to express sincere gratitude on this occasion.

Special thanks are extended to Professor Dr. Praneet Opanasopit for facilitating and supporting the provision of chemicals, instruments, and equipment, as well as for his ongoing advice, encouragement, and moral support throughout this independent study.

The researcher also wishes to thank the Faculty of Pharmacy, Silpakorn University, for providing the necessary facilities and resources that supported the completion of this study.

This achievement would not have been possible without the contributions of many individuals. The researcher wishes to extend heartfelt appreciation to all who played a role in the successful completion of this project—whether mentioned here or not—including parents and friends who continuously offered support, encouragement, and assistance in every way.

Finally, the researcher hopes that this independent study will serve as a useful reference for relevant institutions and individuals interested in this area of research. Any shortcomings are the responsibility of the researcher, who humbly accepts and apologizes in advance.

Phuvamin Suriyaamporn

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS/ABBREVIATIONS

Symbols/Abbreviations	Terms
AH clustering	Agglomerative hierarchical clustering
AI	Artificial intelligence
ANNs	Artificial neural networks
CCD	Central composite design
CNNs	Convolutional Neural Networks
DBSCAN	Density-based spatial clustering
DI waster	Deionized water
DL	Deep learning
DLS	Dynamic light scattering
DT	Decision tree
EtOH	Ethanol
FP-Growth	Frequent-pattern growth
Genetic algorithm	Genetic algorithm
GANs	Generative Adversarial Networks
GT	Gelatin
HML	Hybrid machine learning
HPLC	High Performance Liquid Chromatography
K-NN	K-nearest neighbor
LDA	Linear Discriminant Analysis
LR	Linear regression
LR	Logistic regression
LSTMs	Long Short-Term Memory Networks
MAE	Mean absolute error
ML	Machine learning
MSE	Mean squared error
NB	Naïve Bayes
NN	Neural network

NPs	Nanoparticles
P407	Poloxamer 407
PAA	Poly(acrylic acid)
PCA	Principal component analysis
PCS	Photon correlation spectroscopy
PDI	Polydispersity Index
PNPs	Polymeric nanoparticles
PR	Polynomial regression
PS	Particle size
R^2	Coefficient of determination
RES	Resveratrol
RF	Random forest
RMSE	Root-mean-squared error
RNNs	Recurrent Neural Networks
RL	Reinforcement learning
SEM	Scanning electron microscope
SVM	Support vector machine
TEM	Transmission electron microscope
UV	Ultraviolet
ZP	Zeta potential

CHAPTER 1
INTRODUCTION

1.1 Background and statement of the problem

Pharmaceutical formulation serves as a cornerstone of modern healthcare, ensuring that therapeutic agents are delivered in safe, effective, and patient-compliant forms. However, the development of such formulations is inherently multifaceted, often hindered by factors such as poor drug solubility, chemical instability, limited bioavailability, and potential excipient incompatibility. Traditionally, these complexities have been addressed using empirical, trial-and-error methods, which are not only time-intensive but also costly, ultimately slowing down the progression from formulation design to market readiness (L. K. Vora et al., 2023).

In recent years, Artificial Intelligence (AI) and Machine Learning (ML) have emerged as powerful tools for accelerating and enhancing pharmaceutical formulation processes. By leveraging large-scale datasets and advanced computational models, these technologies enable researchers to uncover intricate relationships among formulation variables, forecast formulation outcomes with high accuracy, and identify optimal processing conditions. Applications of AI/ML now span diverse pharmaceutical domains including drug discovery, design, quality assurance, and scale-up (Duch et al., 2007; Suriyaamporn, Pamornpathomkul, Patrojanasophon, et al., 2024; S. Wang et al., 2022). Despite their transformative potential, the integration of AI/ML in formulation development still faces key limitations. These include the scarcity of high-quality, standardized data, issues surrounding model accuracy and interpretability, and the need to meet stringent regulatory standards that require thorough validation of predictive outputs (Ali et al., 2024; M. Abdelhaleem Ali & M. Alrobaian, 2024; Suriyaamporn, Pamornpathomkul, Patrojanasophon, et al., 2024).

Notably, AI is playing an increasingly critical role in pharmaceutical quality control. The incorporation of ML algorithms has led to significant advancements in areas such as automated defect detection, impurity profiling, and real-time monitoring of production processes (Kalyane et al., 2020; Obaido et al., 2024). Hybrid ML models—combinations of different algorithmic strategies or integrations with

traditional statistical tools—are particularly noteworthy for their superior predictive performance and robustness (Azevedo et al., 2024). These models can handle heterogeneous and incomplete datasets, and they are especially well-suited for capturing nonlinear and dynamic relationships within pharmaceutical systems (Buket Aksu et al., 2012; Belič et al., 2009; Y. Li et al., 2015).

A key component of modern optimization within hybrid frameworks is the Genetic Algorithm (GA). Inspired by the principles of natural selection and genetics, GAs are heuristic search algorithms that iteratively evolve a population of candidate solutions toward an optimal formulation. Through operations such as selection, crossover, and mutation, GAs are capable of solving complex, nonlinear, and multi-objective optimization problems, often outperforming traditional techniques in terms of efficiency and convergence to global optima (Rajwar et al., 2023; Tomar et al., 2024). To further improve the adaptability and performance of GAs, Reinforcement Learning (RL) has been increasingly employed. RL is a type of machine learning in which an agent learns to make sequential decisions through interaction with an environment, guided by feedback in the form of rewards or penalties. In the context of formulation optimization, RL can dynamically tune GA hyperparameters (Brzęk et al., 2025)—such as crossover and mutation probabilities—based on observed optimization performance, thereby enabling more efficient exploration and exploitation of the search space (Gao & Schweidtmann, 2024; Martins et al., 2025).

Recent studies have highlighted the advantages of such integrations. Li et al. demonstrated the successful optimization of polymer–lipid hybrid nanoparticles for verapamil hydrochloride using a combined ANN and GA approach, achieving superior drug loading and nanoscale particle size compared to traditional response surface methodology (Y. Li et al., 2015). Similarly, Aksu et al. applied a hybrid model integrating ANN with genetic programming and neuro-fuzzy logic to optimize ramipril tablets, effectively predicting CQAs and reducing formulation development time (Buket Aksu et al., 2012). Moreover, Fu et al. investigated a reinforced genetic algorithm (RGA) that combined reinforcement learning with genetic algorithms for structure-based drug design. By optimizing crossover and mutation strategies via neural networks, RGA improved molecule generation efficiency. Experimental results

demonstrated that RGA outperformed baseline methods in generating high-affinity, diverse compounds across multiple protein targets (Fu et al., 2022).

The Quality-by-Design (QbD) framework has increasingly incorporated AI tools to define and expand design spaces, thereby improving control over product quality (Yu et al., 2014). AI-enhanced methods are more capable than traditional statistics in identifying CMAs and CPPs, particularly in the context of complex and nonlinear formulation data (Huanbutta et al., 2024; Suriyaamporn, Pamornpathomkul, Patrojanasophon, et al., 2024; Walsh et al., 2022). In a related application, Suriyaamporn et al. used ANN models to optimize progesterone-loaded solid lipid nanoparticles for transdermal delivery, achieving predictive accuracy above 94% across key parameters (Suriyaamporn, Pamornpathomkul, Wongprayoon, et al., 2024). Likewise, Simões applied ANN to accurately predict the dissolution profiles of a BCS class IV drug, demonstrating high R^2 values (>0.94) and minimal prediction error, highlighting AI's ability to handle multicollinearity and nonlinear effects (Simões et al., 2020). These developments underscore the growing role of hybrid AI models in transforming quality control and formulation optimization processes across the pharmaceutical industry, providing a foundation for digital transformation and improved regulatory compliance (Miozza et al., 2024).

Resveratrol (RES), a polyphenol found in grapes and blueberries, has garnered interest due to its cardioprotective, antioxidant, anti-inflammatory, and anticancer activities (Liu et al., 2015; Robinson et al., 2015). Its activity is largely attributed to the presence of three hydroxyl groups, which enable free radical neutralization and metal ion chelation—mechanisms implicated in cancer prevention (Imran et al., 2020). Moreover, resveratrol has been shown to modulate several signaling pathways such as p53, mTOR, STAT3, and NF-κB, and to promote apoptosis via caspase activation and Bax upregulation (Kelkel et al., 2010; Shukla & Singh, 2011; Zhang et al., 2013). Despite its therapeutic potential, resveratrol's clinical use is hampered by its lipophilicity, poor aqueous solubility (~0.05 mg/mL), environmental instability, and rapid systemic clearance (Aung et al., 2021). These physicochemical limitations necessitate the development of advanced drug delivery systems, such as nanoparticles, to enhance its bioavailability and protect it from degradation (Zupančič et al., 2015).

Polymeric nanoparticles (PNPs), with sizes ranging from 10–1000 nm, offer promising solutions for improving drug delivery efficiency and stability. Their advantages include enhanced solubility, targeted delivery, prolonged circulation time, and reduced systemic toxicity (Soares et al., 2018; Suriyaamporn et al., 2023). Various biodegradable and non-biodegradable polymers—including PVA, PVP, PLGA, chitosan, and PCL—are employed in their formulation, often alongside stabilizers or surfactants to improve stability (Lôbo et al., 2021).

Nevertheless, challenges remain in PNP production, such as structural complexity, batch variability, and purity concerns. These issues can be addressed by applying AI-based methods to systematically optimize formulation design, predict performance, and improve reproducibility (Adir et al., 2020; Aumklad et al., 2024/07/18; Habeeb et al., 2024). Therefore, the present study aims to develop and apply hybrid ML models to predict and optimize the formulation of RES-loaded PNPs. The models were trained using key CQAs—particle size, polydispersity index, zeta potential, and drug loading—derived from experimental data. A genetic algorithm was utilized to identify optimal formulation parameters, while reinforcement learning dynamically adjusted crossover and mutation rates. The optimized formulations were validated experimentally for stability and anticancer efficacy. Additionally, a design space was constructed to ensure product quality and compliance. This approach marks a novel integration of hybrid AI techniques in nanomedicine development.

1.2 Objectives

1.2.1 To develop and validate hybrid machine learning models for the prediction of critical quality attributes of resveratrol-loaded polymeric nanoparticles.

1.2.2 To develop and validate a genetic algorithm enhanced by reinforcement learning for the optimization parameters of resveratrol-loaded polymeric nanoparticles to experimentally evaluate the optimized formulations for physicochemical properties.

1.3 Research Hypothesis

1.3.1 Hybrid machine learning models can accurately predict the critical quality attributes (CQAs) including particle size, polydispersity index, zeta potential, and drug loading of resveratrol-loaded polymeric nanoparticles.

1.3.2 Optimization of formulation parameters using a genetic algorithm enhanced by reinforcement learning will result in resveratrol-loaded polymeric nanoparticles with accurately predicted physicochemical properties, as confirmed by experimental validation.

1.4 Expected benefits

1.4.1 The application of hybrid machine learning models combined with genetic algorithms and reinforcement learning is expected to reduce time, cost, and trial-and-error in the development of resveratrol-loaded polymeric nanoparticles.

1.4.2 Accurate prediction and optimization of critical quality attributes will lead to nanoparticles with consistent physicochemical properties, supporting reliable manufacturing and regulatory compliance.

1.5 Limitation and scope

1.5.1 The performance of hybrid machine learning models is highly dependent on the quality and size of the experimental dataset; limited or imbalanced data may affect the accuracy and generalizability of the predictions.

1.5.2 This study focuses on the development and optimization of resveratrol-loaded polymeric nanoparticles using AI-based methods, with an emphasis on predicting critical quality attributes and validating the optimized formulations experimentally within a laboratory-scale setting.

CHAPTER 2
REVIEW OF LITERATURE

2.1 Artificial intelligence in pharmaceuticals

Artificial Intelligence (AI) was first introduced in 1956, initially focused on symbolic reasoning and problem-solving tasks. Over the decades, advances in computational power and data availability have significantly expanded AI's capabilities, leading to its widespread adoption across various industries, including the pharmaceutical sector (Dasta, 1992). AI systems are designed to emulate human cognitive functions such as reasoning, learning, language comprehension, pattern recognition, and decision-making. These functionalities are achieved through the use of algorithms, input datasets, and high-performance computing to simulate intelligent behavior akin to human cognition.

AI is generally categorized into subfields, with Machine Learning (ML) and Deep Learning (DL) being the most prominent. As illustrated in Figure 2.1, AI encompasses ML, which emphasizes enabling systems to learn from data and improve over time without being explicitly programmed, often utilizing complex statistical methods. DL, a subset of ML, relies on artificial neural networks (ANNs) to model and solve highly nonlinear problems (Kolluri et al., 2022; Soori et al., 2023).

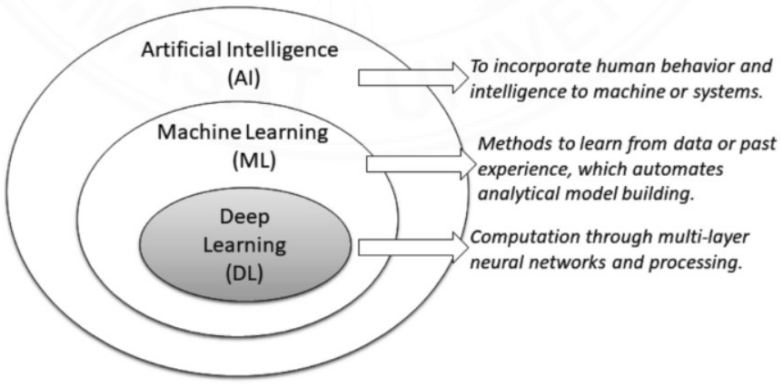

Figure 2.1 Artificial Intelligence (AI) can be categorized into subfields, namely machine learning (ML) and deep learning (DL) (Sarker, 2021a).

The human central nervous system has inspired the development of neural networks in AI. The structure and function of biological neurons, which enable information processing, adaptation, and learning, have served as the foundation for designing artificial neural networks. The human brain, comprising billions of interconnected neurons, represents a biological model of intelligence and learning. ANNs attempt to replicate some of these processes through interconnected layers of nodes or "neurons" that can learn to recognize patterns, classify data, and make predictions (Jain et al., 1996). These networks have become fundamental components of modern AI applications, including image and speech recognition, medical diagnostics, and predictive analytics (S. Wang et al., 2022).

In the early stages of adoption, AI was primarily applied in the pharmaceutical industry for basic data management tasks such as data structuring and analysis. As the technology evolved, its applications expanded to include more complex processes such as molecular modeling, drug-target interaction analysis, and lead compound identification (Qian & Sejnowski, 1988). Today, AI plays a transformative role in numerous pharmaceutical domains, ranging from pharmacokinetics and toxicity prediction to formulation development, stability assessment, and dose optimization (Mishra & Awasthi, 2021; Paul et al., 2021; Wessel et al., 1998).

One of the key advantages of AI in pharmaceutical development is its ability to process and analyze large datasets efficiently, facilitating more accurate and faster decision-making. AI models can predict how drug molecules will behave in biological systems, simulate pharmacological effects, and optimize formulation parameters, thus significantly reducing experimental time and costs. Applications of AI now extend to various dosage forms, including conventional tablets, 3D-printed medicines, dry powders, polymer patches, injectables, vaccines, and nanoparticle-based systems (Elbadawi et al., 2020; Han et al., 2019; Kashani-Asadi-Jafari et al., 2022; J. Wang et al., 2022)

The evolution of AI in pharmaceutical research continues as technological innovations and data availability increase. AI holds significant potential to accelerate drug discovery, optimize formulations, and support personalized medicine. Ultimately, its continued integration is expected to lead to the development of safer, more effective, and patient-specific pharmaceutical products.

2.2 Classification of Artificial intelligence

AI-driven technologies have become a focal point in pharmaceutical research and development, offering advanced tools to accelerate innovation and decision-making. One of the most impactful applications of AI in this context is ML, which enables computational systems to learn from data, recognize patterns, and make predictions without explicit programming. ML encompasses a broad spectrum of algorithms tailored for different analytical tasks. Commonly used techniques in pharmaceutical applications include Naive Bayes classifiers, Decision Trees, Random Forests, Multiple Linear Regression, Logistic Regression, Linear Discriminant Analysis (LDA), Support Vector Machines (SVM), and Artificial Neural Networks (ANNs) (Dara et al., 2022; Justo-Silva et al., 2021; Raza et al., 2022).

As illustrated in Figure 2.2, ML algorithms can be broadly classified into three categories: supervised learning, unsupervised learning, and deep learning. In supervised learning, the model is trained using labeled datasets to predict outcomes or classify data. Unsupervised learning, by contrast, identifies hidden patterns or groupings in data without predefined labels. Deep learning, a specialized subdomain of ML, employs multi-layered neural networks to handle complex, high-dimensional data, often outperforming traditional models in tasks such as image recognition and natural language processing.

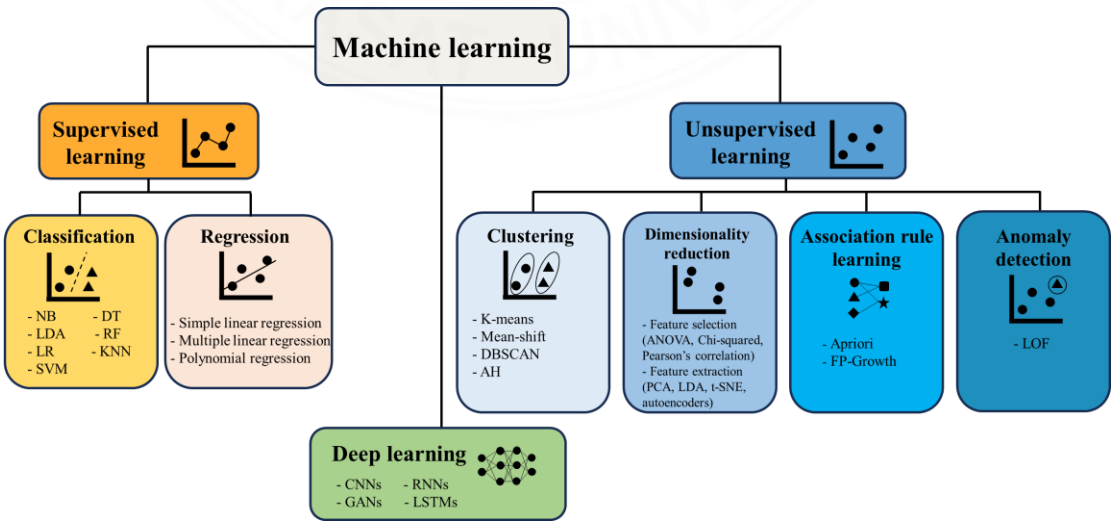

Figure 2.2 The classification of AI in pharmaceutical comprises supervised, unsupervised, and reinforcement learning methods (Suriyaamporn, Pamornpathomkul, Patrojanasophon, et al., 2024).

These ML approaches have significantly contributed to the development of predictive models and data clustering tools in pharmaceutical science. Their ability to process large datasets and uncover nonlinear relationships makes them valuable for a wide range of tasks, from drug discovery and formulation optimization to toxicity prediction and quality control.

2.2.1 Supervised learning

Supervised Learning is a machine learning approach that relies on labeled datasets—data for which the input features and corresponding output values are known. This method allows the algorithm to learn patterns and relationships within the training data, enabling it to make accurate predictions on unseen data. The process mimics human learning, where feedback and guidance help shape understanding. By associating specific inputs with known outputs, supervised learning models can generalize these associations to forecast outcomes from new inputs.

The core objective of supervised learning is to construct a predictive function that maps inputs to desired outputs by analyzing patterns in the labeled dataset. The model iteratively adjusts its internal parameters to minimize prediction error during training, ultimately leading to a well-generalized model. These techniques are extensively applied in pharmaceutical research for tasks such as bioactivity prediction, toxicity screening, pharmacokinetic modeling, and formulation optimization, where historical data is available and the target outcomes are well-defined. Supervised learning methods can be broadly categorized into two main types.

2.2.1.1 Classification

Classification is a core technique in supervised learning that involves categorizing data into predefined classes or labels. This process plays a vital role in pharmaceutical research, especially in tasks such as drug efficacy prediction, toxicity assessment, and formulation optimization. A variety of machine learning algorithms

have been developed to perform classification, each offering distinct strengths depending on the nature and complexity of the data.

1. Naïve Bayes (NB)

Naïve Bayes is a probabilistic classifier based on Bayes' Theorem, which assigns class labels to data based on the calculated likelihood of feature values. The term "naïve" refers to the algorithm's assumption of feature independence, which simplifies computation and reduces the need for large datasets during training. Despite this simplification, Naïve Bayes has proven highly effective in text classification and categorical data analysis. In pharmaceutical applications, it has been used to predict therapeutic properties of novel compounds by analyzing drug-like features, contributing significantly to early-stage drug discovery (Madhukar et al., 2019).

2. Linear Discriminant Analysis (LDA)

Linear Discriminant Analysis is a statistical method used to model the probability distribution of classes and assign labels based on Bayes' rule. It is particularly effective at distinguishing between two or more classes by maximizing the separation between categories while minimizing intra-class variability. Additionally, LDA aids in dimensionality reduction, making complex datasets easier to interpret. In pharmaceutical research, LDA is applied to predict drug performance by evaluating chemical attributes and their correlations with biological efficacy (Becht et al., 2021; Prieto et al., 2006).

3. Logistic Regression (LR)

Logistic Regression is widely used for binary classification tasks and aims to estimate the probability that a data point belongs to a specific class. It does so by modeling the relationship between input features and the log-odds of a binary outcome using the logistic function. LR is valued for its simplicity, interpretability, and efficiency in linearly separable datasets. In pharmaceutical development, it is often employed to assess the therapeutic potential of new drug candidates (Bagherian et al., 2020; Choi & Boo, 2020).

4. K-Nearest Neighbors (K-NN)

K-NN is a non-parametric algorithm used for both classification and regression. It operates on the principle that similar data points exist in close proximity within the feature space. The classification decision is made based on the majority vote of the 'K'

closest training examples, typically measured using Euclidean distance. One of the challenges in K-NN is determining the optimal value of K. In pharmaceutical contexts, K-NN has been used in predicting drug activities, optimizing formulations, assessing stability, and evaluating toxicity profiles (Bannigan et al., 2023; Yang et al., 2019).

5. Support Vector Machines (SVM)

SVM is a powerful supervised learning method suitable for both classification and regression tasks. It works by constructing an optimal hyperplane that separates classes with maximum margin in a high-dimensional space. The use of kernel functions (e.g., linear, polynomial, radial basis function) allows SVM to handle non-linear decision boundaries effectively. While robust to high-dimensional data, SVM may be less suitable when classes overlap significantly or when the dataset contains excessive noise. In pharmaceutical research, SVM has been applied in drug classification, pharmacokinetic/pharmacodynamic modeling, and predicting drug-drug interactions (Seok et al., 2011; Yang et al., 2009).

6. Decision Trees (DT)

Decision Trees are non-parametric models that simulate decision-making processes using a tree-like structure. Each internal node represents a feature, branches represent decision rules, and leaf nodes denote outcomes. DTs are intuitive and easy to interpret, making them useful in exploratory data analysis. In the pharmaceutical field, DTs have been employed for formulation design, toxicity prediction, and identifying key chemical properties influencing drug efficacy (Karim et al., 2019).

7. Random Forest (RF)

Random Forest is an ensemble learning method that builds multiple decision trees and combines their outputs to improve prediction accuracy. This approach addresses limitations of individual trees, such as overfitting, by introducing randomness in feature selection and data sampling. RF has demonstrated robust performance in various pharmaceutical applications, including drug activity prediction and compound screening in drug discovery pipelines (Lind & Anderson, 2019).

2.2.1.2 Regression

Regression analysis is a foundational statistical approach in machine learning, commonly applied to model and predict the relationship between a dependent variable

(response) and one or more independent variables (predictors or features). The primary aim is to quantify how changes in predictor variables influence the response variable. In pharmaceutical sciences, regression techniques are widely utilized for a range of predictive tasks, including modeling drug concentration profiles over time and estimating key physicochemical properties such as solubility based on molecular descriptors. These insights are instrumental in formulation design and drug development optimization (Vilar & Costanzi, 2012).

1. Simple Linear Regression

Simple linear regression (SLR) is the most fundamental form of regression analysis. It involves modeling the linear relationship between a single independent variable and a single dependent variable. The output is a straight-line equation that best fits the data, allowing prediction of the dependent variable based on new values of the predictor. SLR is often used in initial exploratory analyses to understand potential trends or correlations between two variables.

2. Multiple Linear Regression

Multiple linear regression (MLR) extends the principles of SLR by incorporating two or more independent variables. This enables the modeling of more complex systems where multiple factors simultaneously influence the outcome. In pharmaceutical research, MLR is frequently applied to predict drug dissolution, stability, or bioavailability by considering various formulation parameters, environmental factors, and physicochemical properties. Its ability to handle multifactorial datasets makes it a powerful tool in both formulation screening and process optimization.

3. Polynomial Regression

Polynomial regression is a nonlinear extension of linear regression that fits a polynomial curve to the data rather than a straight line. It captures more complex relationships between the dependent and independent variables by including higher-order terms (e.g., squared or cubic terms) in the model. This technique is particularly useful in pharmaceutical applications where response behavior exhibits curvature, such as nonlinear dissolution kinetics or drug release patterns from controlled-release formulations.

2.2.2 Unsupervised learning

Unsupervised learning is a key branch of ML that operates on datasets without predefined labels or target variables. Unlike supervised learning, which relies on known outcomes to train models, unsupervised learning explores the intrinsic structure of data by identifying hidden patterns, similarities, or groupings among the input variables. The objective is to uncover meaningful insights or relationships within the dataset without external guidance. This learning paradigm is particularly valuable in situations where labeled data is unavailable, costly, or impractical to obtain—common scenarios in early-stage pharmaceutical research and exploratory data analysis.

In pharmaceutical applications, unsupervised learning supports diverse objectives such as classifying compound libraries, identifying structural similarities among molecules, uncovering novel drug-target interactions, and improving data preprocessing pipelines. Unsupervised learning methods can be broadly categorized into four main types.

2.2.2.1 Clustering

Clustering involves organizing data points into groups based on similarity, without prior knowledge of class labels. The goal is to place similar items within the same group while ensuring separation from dissimilar items. Various clustering algorithms have been developed to address different data structures and distributions. Common clustering methods include:

1. K-Means Clustering

This algorithm partitions data into k clusters by minimizing the distance between data points and their respective centroids. Although efficient, K-Means is sensitive to outliers and the initial placement of centroids, which can affect the final clustering results. In pharmaceutical sciences, K-Means has been used to classify chemical compounds based on structural similarity, aiding drug discovery efforts (Akondi et al., 2019).

2. Mean-Shift Clustering

A non-parametric technique that identifies dense regions in data by iteratively shifting centroids toward the highest density area. While effective for non-uniform data,

it can be computationally intensive and sensitive to bandwidth parameters. Mean-Shift is useful for complex datasets with varying distributions (Sarker, 2021b).

3. Density-Based Spatial Clustering (DBSCAN)

DBSCAN identifies clusters of arbitrary shape by detecting areas of high point density, distinguishing them from sparse noise regions. It requires two parameters: neighborhood radius (eps) and the minimum number of points (minPts) to form a cluster. DBSCAN is highly robust to outliers and has been applied in pharmaceutical contexts for compound classification and outlier detection (Jiang et al., 2019; McComb et al., 2022).

4. Agglomerative Hierarchical Clustering (AHC)

A bottom-up approach that builds a tree-like hierarchy of clusters by iteratively merging the most similar pairs. AHC provides interpretable dendrograms that reveal nested groupings. In drug research, it assists in grouping molecules based on chemical or structural similarities for lead identification (Lakshmi & P, 2023).

2.2.2.2 Dimensionality Reduction

Dimensionality reduction is a vital process in machine learning that involves decreasing the number of features or input variables in a dataset while preserving the essential structure and relationships within the data. The primary goal is to simplify complex datasets, eliminate redundant or irrelevant features, and enhance the performance of machine learning algorithms. This technique is particularly useful for improving model interpretability, reducing computational demands, and mitigating overfitting issues.

In pharmaceutical research, dimensionality reduction is frequently applied in the analysis of high-dimensional data, such as gene expression profiles or biological activity datasets, to uncover meaningful patterns and facilitate hypothesis generation. Two principal approaches to dimensionality reduction include feature selection and feature extraction (Vamathevan et al., 2019; Lalitkumar K. Vora et al., 2023).

1. Feature Selection

This technique involves selecting a subset of relevant features from the original dataset that contribute significantly to the predictive model. Feature selection aims to streamline the learning process, enhance model generalization, and reduce overfitting.

It is particularly effective in high-dimensional data scenarios where many variables may be irrelevant or redundant. Common methods include the Chi-squared test, Analysis of Variance (ANOVA), Pearson's correlation coefficient, and Recursive Feature Elimination (RFE). These techniques help identify the most informative variables for model training.

2. Feature Extraction

Feature extraction transforms the original high-dimensional data into a new, lower-dimensional feature space while retaining critical information. This transformation often combines existing features into composite variables that capture the underlying data structure. Popular techniques include Principal Component Analysis (PCA), Linear Discriminant Analysis (LDA), t-Distributed Stochastic Neighbor Embedding (t-SNE), and deep learning-based methods such as autoencoders. These methods are particularly effective for visualizing complex datasets and enhancing model accuracy by simplifying data representations.

2.2.2.3 Association Rule

Association rule learning is a data mining technique designed to uncover interesting relationships, patterns, or dependencies within large datasets. This approach focuses on identifying frequent co-occurrences among variables and is commonly expressed through conditional rules in the form "If X, then Y," where X and Y represent sets of items. Such rules are typically evaluated using metrics such as support, confidence, and lift, which quantify their relevance and reliability. Association rule learning is particularly valuable in scenarios where understanding item correlations can inform decision-making, such as market basket analysis or pharmaceutical data analysis. Two of the most widely used algorithms in this domain are the Apriori algorithm and Frequent Pattern Growth (FP-Growth).

1. Apriori Algorithm

Apriori is a foundational algorithm in association rule mining, known for its simplicity and effectiveness. It identifies frequent itemsets by iteratively expanding combinations of items and eliminating infrequent ones based on minimum support thresholds. The key principle underlying Apriori is that if an itemset is infrequent, all supersets derived from it will also be infrequent, thereby reducing the computational

burden. This bottom-up strategy starts with single items and progressively constructs larger itemsets. In pharmaceutical research, Apriori has been applied to uncover relationships between drugs and adverse effects, as well as to analyze biological data to reveal associations among genes or proteins. Its capacity to discover hidden patterns makes it a valuable tool in hypothesis generation and knowledge discovery (Noguchi et al., 2018; Sarker, 2021b).

2. FP-Growth Algorithm

The FP-Growth algorithm addresses the limitations of Apriori by using a divide-and-conquer strategy and constructing a compact data structure known as the FP-tree. Unlike Apriori, FP-Growth avoids candidate generation and instead compresses the dataset to identify frequent itemsets more efficiently. Despite its computational advantages, it may face challenges with extremely large or dense datasets due to the complexity of tree construction. Nevertheless, FP-Growth has demonstrated utility in pharmaceutical research, particularly in identifying latent patterns and associations among formulation parameters, drug interactions, or patient response profiles (Sarker, 2021b; Zhao & S Bhowmick, 2003).

Together, these algorithms play a significant role in the pharmaceutical domain by enabling researchers to mine complex datasets for meaningful insights, ultimately enhancing drug development, safety profiling, and personalized medicine strategies.

2.2.2.4 Anomaly Detection

Anomaly detection is a machine learning approach used to identify data points or patterns that significantly deviate from the norm within a dataset. These outliers may indicate potential errors, rare events, or system malfunctions. The primary objective of anomaly detection is to flag irregularities that may otherwise go unnoticed but could have critical implications.

In the pharmaceutical industry, anomaly detection is particularly valuable for ensuring product quality and safety by identifying deviations in manufacturing processes or experimental results. This capability enhances quality control protocols and supports regulatory compliance by proactively addressing inconsistencies before they lead to product failure or safety concerns.

Several algorithms are commonly employed for anomaly detection, including Local Outlier Factor (LOF) and Isolation Forest. LOF measures the local deviation of a data point with respect to its neighbors, while Isolation Forest isolates anomalies by recursively partitioning the dataset. Both techniques are effective in identifying outliers in high-dimensional data and can be integrated into real-time monitoring systems for rapid decision-making and risk mitigation (Goldstein & Uchida, 2016).

2.2.3 Deep Learning

Deep learning (DL), a subset of machine learning, has garnered increasing attention within the pharmaceutical industry due to its superior accuracy and precision when compared to conventional ML approaches. DL models utilize multilayer artificial neural networks (ANNs) to automatically extract and learn complex features and representations from raw data. These capabilities allow for improved performance in tasks involving intricate and high-dimensional datasets. The key advantage of deep learning lies in its ability to emulate human cognitive functions by transmitting information through interconnected layers of artificial neurons, with each layer progressively capturing more abstract and detailed features.

Unlike traditional ML models that often rely on manual feature engineering, deep learning methods excel at autonomously learning relevant data features, making them particularly powerful in applications requiring deep contextual understanding. In pharmaceutical applications, deep learning has been employed in areas such as compound identification, drug interaction prediction, pharmacokinetics modeling, and molecular design

2.2.3.1 Convolutional Neural Networks (CNNs)

CNNs are specialized deep learning architectures tailored for image-based data processing. They employ convolutional layers to extract spatial hierarchies of features using learnable filters or kernels. Each kernel performs convolution operations over localized regions of an image to detect specific features at various levels of abstraction. In pharmaceutical research, CNNs have been instrumental in image classification tasks, including the identification and differentiation of molecular structures and compound visualization (Oei et al., 2019; Wolfgang et al., 2020).

2.2.3.2 Recurrent Neural Networks (RNNs)

RNNs are designed to handle sequential data where temporal or ordered dependencies are critical. These networks incorporate loops within their architecture, enabling information persistence across time steps. RNNs are particularly suited for time-series analysis and natural language processing (NLP) tasks. In pharmaceutical sciences, RNNs have been employed for sequence-based drug design, pharmacokinetics/pharmacodynamics (PK/PD) modeling, and prediction of drug absorption and distribution patterns over time (Tang, 2023).

2.2.3.3 Generative Adversarial Networks (GANs)

GANs represent a unique class of DL models composed of two competing neural networks: a generator and a discriminator. The generator aims to produce synthetic data that closely resembles real data, while the discriminator's role is to distinguish between real and generated data. Through iterative adversarial training, GANs progressively improve their ability to generate high-fidelity synthetic samples. In pharmaceutical research, GANs have been successfully applied to generate novel molecular structures, optimize chemical properties, and accelerate the early stages of drug discovery by expanding the diversity of compound libraries (Sousa et al., 2021).

2.2.3.4 Long Short-Term Memory Networks (LSTMs)

LSTMs are an advanced variant of RNNs designed to address the limitations of standard RNNs in learning long-term dependencies due to the vanishing gradient problem. LSTMs incorporate a memory cell and gating mechanisms—input, forget, and output gates—that regulate the flow of information, allowing the network to retain or discard data as needed. This architecture enables LSTMs to model both short- and long-term patterns in sequential data effectively.

LSTMs are highly applicable to tasks requiring temporal pattern recognition, such as NLP, time-series forecasting, and physiological modeling. In pharmaceutical applications, LSTMs have been used for simulating drug concentration profiles in the body, modeling ADME (absorption, distribution, metabolism, and excretion)

processes, and predicting time-dependent pharmacological responses (Moldovan et al., 2019).

2.3 Hybrid machine learning models

Hybrid machine learning (ML) models represent an advanced class of data-driven approaches that combine the strengths of multiple learning paradigms or algorithms to achieve superior predictive performance, robustness, and adaptability compared to individual models, represented in Figure 2.3. These models have emerged as powerful tools for addressing complex and nonlinear problems that are commonly encountered in pharmaceutical formulation, quality control, and drug development (Shah et al., 2025).

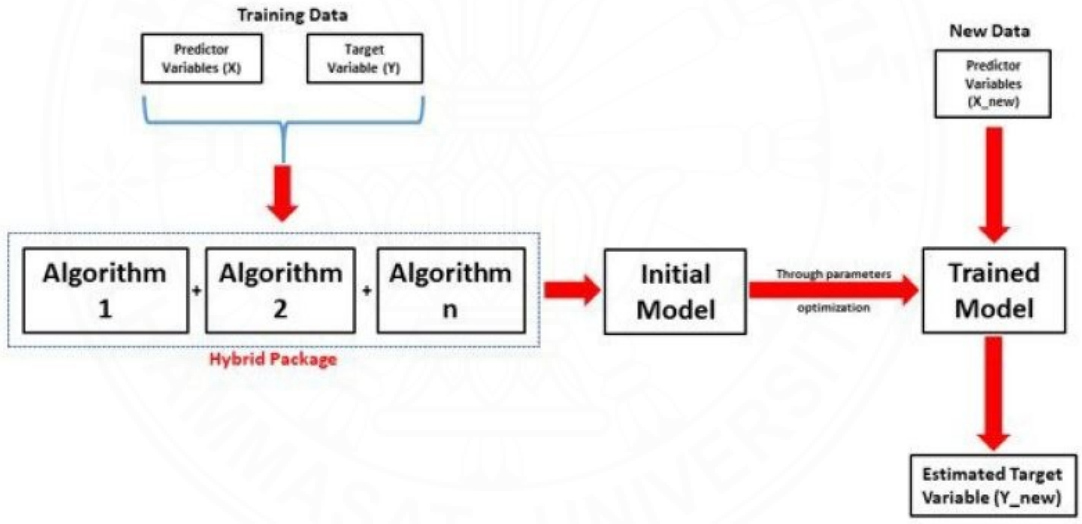

Figure 2.3 Hybrid machine learning models workflow.

The rationale behind hybrid models is grounded in the fact that no single ML algorithm is universally optimal for all types of datasets or tasks. By integrating complementary techniques—such as linear regression with artificial neural networks (ANNs), support vector machines (SVMs) with decision trees, or unsupervised clustering with supervised classifiers—hybrid models can capitalize on the advantages of each component while mitigating their individual limitations. This results in models that are better equipped to handle noise, missing data, multicollinearity, and high-

dimensional feature spaces (Rajaee et al., 2020). Hybrid ML models can be broadly categorized into the following types:

1. Model-Based Hybridization

These models combine data-driven ML techniques with domain-specific theoretical or mechanistic models, enabling incorporation of expert knowledge and improved generalizability. This approach is particularly effective in pharmacokinetics/pharmacodynamics (PK/PD) modeling and physiologically-based pharmacokinetic (PBPK) modeling (Agoram et al., 2001).

2. Algorithmic Hybridization

This approach integrates multiple ML algorithms within a single framework, such as combining ANN with SVM or decision trees, to harness their respective strengths. It has been used effectively in predicting drug solubility and formulation optimization (Chen, 2024).

3. Optimization-Based Hybridization

Involves the fusion of ML models with optimization algorithms like genetic algorithms (GA), particle swarm optimization (PSO), or reinforcement learning (RL), enhancing model tuning and global search capabilities. Applications include nanoparticle optimization and multi-objective drug formulation design (Yongqiang Li et al., 2015).

4. Ensemble Hybridization

Uses ensemble methods such as bagging, boosting, or stacking, where multiple base learners are combined to improve prediction accuracy and reduce overfitting. This strategy has been shown to enhance robustness in pharmaceutical process control (Hoseini et al., 2023).

In pharmaceutical sciences, hybrid ML models have been employed in diverse applications. For instance, hybrid ANN–genetic algorithm (GA) frameworks have been successfully used to optimize nanoparticle formulations by modeling critical quality attributes (CQAs) such as particle size, zeta potential, and drug loading. Likewise, combinations of fuzzy logic and machine learning have improved decision-making processes under uncertainty, which is crucial in formulation design and scale-up. Hybrid systems that include ensemble learning techniques—such as boosting and

bagging—have enhanced model generalization and predictive stability across varied pharmaceutical datasets (Bannigan et al., 2021).

Notably, hybrid ML approaches also support the implementation of Quality by Design (QbD) principles by enabling precise identification of critical material attributes (CMAs) and critical process parameters (CPPs), thus facilitating the creation of robust design spaces. Moreover, the integration of reinforcement learning (RL) with evolutionary algorithms in hybrid models enables dynamic adaptation of model parameters, allowing for real-time optimization and autonomous learning in manufacturing settings.

As pharmaceutical processes become increasingly complex and data-rich, hybrid ML models are poised to play an instrumental role in enhancing formulation efficiency, predictive accuracy, and regulatory compliance, ultimately contributing to safer and more effective therapeutic products.

2.4 Genetic algorithm

Genetic algorithms (GAs) are a class of stochastic optimization techniques inspired by the process of natural selection and biological evolution. First introduced by John Holland in the 1970s, GAs are particularly well-suited for solving complex, nonlinear, and multi-objective optimization problems where traditional gradient-based methods may fall short. GAs operate through iterative evolution of a population of candidate solutions, using operations such as selection, crossover (recombination), and mutation to explore the solution space (Sivanandam & Deepa, 2008). The basic workflow of a GA involves the following steps, represented in Figure 2.4 (Albadr et al., 2020; Katoch et al., 2021).

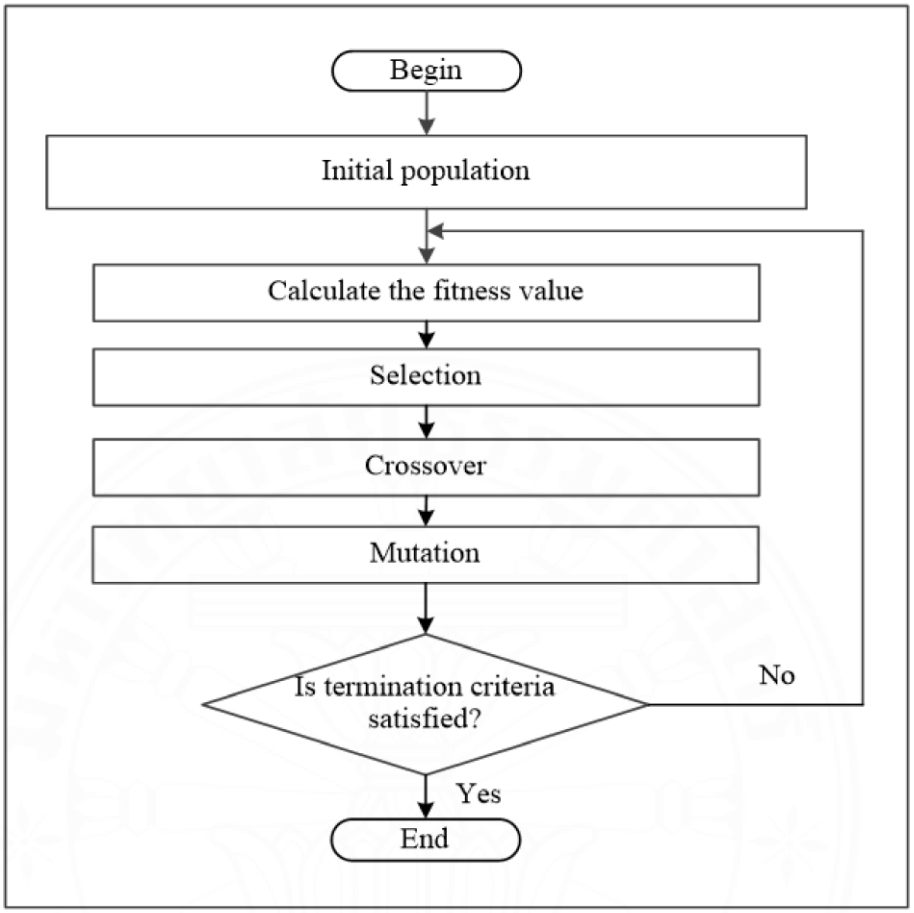

Figure 2.4 The basic workflow of a genetic algorithm (Albadr et al., 2020).

1. Initialization

Generate an initial population of potential solutions, often represented as chromosomes (binary or real-valued). The size of the population typically ranges from 20 to 200 individuals, depending on the problem complexity and computational resources. A larger population may explore the solution space more thoroughly but increases the computational cost.

2. Evaluation

Calculate the fitness of each individual using a predefined fitness function. The fitness function is problem-specific and determines how well each solution satisfies the optimization objectives. In pharmaceutical formulation, this might involve metrics such as drug release rate, encapsulation efficiency, or particle size.

3. Selection

Select individuals based on fitness scores for reproduction. Common selection methods include roulette wheel selection, tournament selection, and rank-based selection. The selection rate usually ranges from 50% to 90%. A higher selection pressure accelerates convergence but may reduce diversity, while lower pressure maintains diversity but slows optimization.

4. Crossover

Combine selected parents to produce new offspring by exchanging parts of their chromosomes. The crossover rate, typically set between 0.6 and 0.9, controls the frequency of recombination. A higher crossover rate promotes exploration of new solutions, while a lower rate focuses on exploiting existing high-quality solutions.

5. Mutation

Introduce random alterations to offspring to maintain genetic diversity and avoid premature convergence. Mutation rate is usually set low (e.g., 0.01 to 0.1) to avoid excessive randomness. It helps the algorithm escape local optima and explore less-visited regions of the search space.

6. Replacement

Form a new population by selecting individuals from the current population and newly generated offspring. Strategies include generational replacement (replacing all individuals) or elitism (preserving the best individuals).

7. Termination

Repeat the process for a fixed number of generations (commonly 50–500) or until convergence criteria are met, such as no significant improvement in fitness over successive generations.

In pharmaceutical sciences, GAs have been extensively applied for formulation optimization, design of experiments, and model parameter tuning. For instance, GAs have been used to identify optimal combinations of excipients, concentrations, and process parameters to achieve desired drug release profiles, particle size distributions, or stability characteristics. When integrated with machine learning models, such as artificial neural networks (ANNs), GAs enhance predictive performance by optimizing hyperparameters and selecting relevant input features (M. R. Zaki et al., 2015).

One of the key advantages of GAs is their ability to escape local minima and find global optima in highly dimensional and rugged search spaces. Additionally, they

are flexible and easily adaptable to different problem domains without requiring gradient information. However, the performance of GAs depends on the proper selection of parameters such as population size, crossover rate, mutation rate, and selection strategy. Improper tuning can lead to issues such as premature convergence or excessive computation time (Albadr et al., 2020; Katoch et al., 2021).

Recently, GAs have been integrated with other advanced computational techniques, including reinforcement learning and swarm intelligence, to further improve their adaptability and efficiency (Lee et al., 2022; Song et al., 2023). These hybrid approaches have demonstrated success in areas such as nanoparticle design, personalized medicine, and predictive modeling for complex drug delivery systems. Overall, genetic algorithms represent a versatile and powerful optimization tool in the pharmaceutical field, enabling efficient exploration of vast parameter spaces and supporting the development of robust, high-quality drug formulations and manufacturing processes (Chi et al., 2009; Ghaheri et al., 2015).

2.5 Reinforcement learning

Reinforcement Learning (RL) is a branch of machine learning concerned with how agents ought to take actions in an environment to maximize cumulative rewards. Unlike supervised learning, where the model is trained with labeled data, RL learns by interacting with the environment and receiving feedback in the form of rewards or penalties. This paradigm mimics behavioral learning and is particularly effective for dynamic decision-making tasks. At its core, RL involves several key components: an agent, an environment, actions, states, and a reward function. The agent observes the current state of the environment and selects an action based on a policy—a strategy mapping states to actions. The environment then transitions to a new state and returns a reward, which the agent uses to update its policy. This process is modeled mathematically using Markov Decision Processes (MDPs) (Martins et al., 2025; Sarker, 2021b) as represented in Figure 2.5. Common algorithms are used in RL such as (Shakya et al., 2023):

1. Q-Learning

A value-based, model-free algorithm where the agent learns a Q-value function that estimates the expected reward for taking an action in a given state. The Q-table is

iteratively updated using the Bellman equation. Q-Learning is easy to implement and effective in small, discrete state spaces but can struggle with high-dimensional inputs due to table size limitations.

2. Deep Q-Networks (DQN)

An advancement of Q-learning that uses deep neural networks to approximate the Q-values instead of using a tabular approach. DQN introduces techniques such as experience replay and target networks to stabilize training. This method enables RL to handle complex environments with high-dimensional and continuous state spaces, such as image inputs in drug design simulations.

3. Policy Gradient Methods

These methods optimize the policy directly by computing the gradient of the expected cumulative reward with respect to the policy parameters. Unlike value-based methods, policy gradients are suitable for continuous action spaces and stochastic policies, making them useful in pharmaceutical process control where optimal control paths need to be learned dynamically.

4. Actor-Critic Methods

A hybrid of value-based and policy-based methods where the "actor" learns the policy function and the "critic" learns the value function. The critic guides the actor’s updates, leading to more stable and faster convergence. Actor-critic models are effective in environments requiring continuous updates and feedback, such as adaptive formulation design or batch control in drug manufacturing.

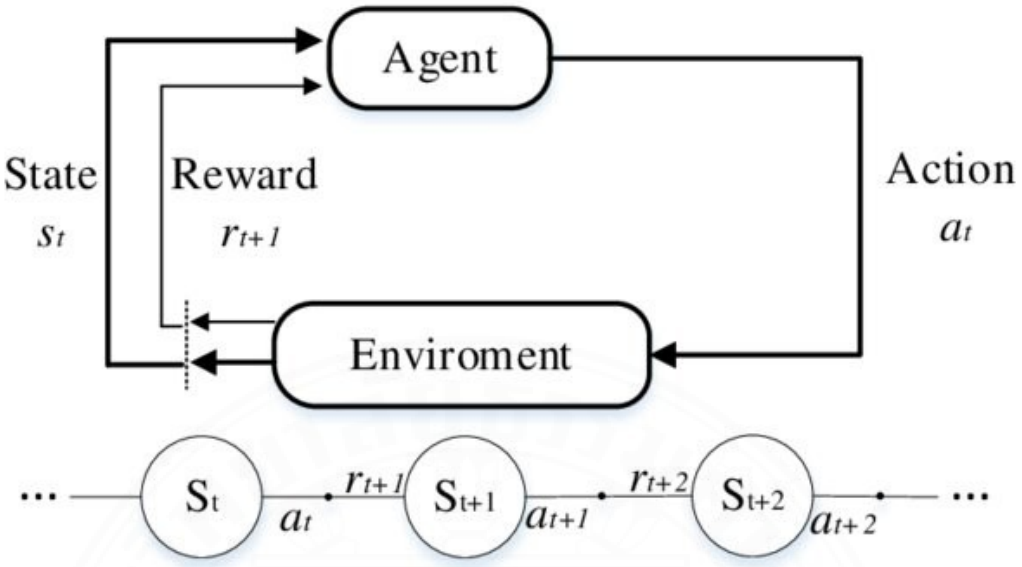

Figure 2.5 Markov Decision Processes (MDPs) flowchart (Zhou et al., 2021).

In pharmaceutical sciences, RL has emerged as a valuable tool in experimental design, process control, and optimization. For instance, RL has been used to guide autonomous experimentation, adaptively adjust process parameters in real-time, and optimize multi-step synthesis or formulation strategies where feedback is sequential and delayed. The strength of RL lies in its adaptability to new environments and ability to learn from trial and error, making it ideal for complex, uncertain, and nonlinear systems (Al-Kharusi et al., 2022; Narayanan et al., 2021).

When combined with GAs, RL can further enhance optimization efficiency. In such hybrid frameworks, RL dynamically tunes GA hyperparameters (e.g., mutation and crossover rates), thereby accelerating convergence and improving the quality of solutions. This synergy has been applied in advanced drug design, including the generation of novel molecules, nanoparticle formulation, and predictive modeling of pharmacokinetics. Moreover, this plays a crucial role in optimizing decision-making in the pharmaceutical domain and facilitates design processes that were previously unattainable using traditional methods (Suriyaamporn, Pamornpathomkul, Patrojanasophon, et al., 2024).

2.6 Nanoparticle-based drug delivery systems

Nanoparticle-based drug delivery systems (NDDSs) represent an advanced colloidal nanotechnology platform in which therapeutic agents are encapsulated, adsorbed, or conjugated onto nanocarriers typically ranging in size from 10 to 1000 nanometers. Over the past two decades, interest in NDDSs has grown substantially, transitioning from academic research into widespread industrial application. This transition has been largely driven by the high therapeutic potential and commercial viability of nanomedicine. Nanotechnology is now considered one of the most rapidly expanding research areas within pharmaceutical sciences due to its capacity to improve drug solubility, stability, and bioavailability while enabling targeted and controlled drug release mechanisms (Soares et al., 2018).

Nanoparticles are engineered to overcome various biological barriers and deliver therapeutic agents to specific target tissues or cells with high precision. By modifying surface characteristics or loading strategies, these systems can provide sustained drug release, protect labile molecules from degradation, and prolong systemic circulation time, thereby enhancing therapeutic efficacy and patient compliance. These properties make NDDSs superior to conventional dosage forms such as tablets, capsules, ointments, and injectable solutions (Abdel-Mageed et al., 2021; Giri et al., 2023). Nanoparticle drug delivery systems can be administered through multiple routes, including oral, nasal, transdermal, and intravenous pathways. Their versatility not only improves drug performance but also reduces adverse effects by minimizing systemic exposure and allowing for localized treatment. Additionally, nanoparticles are particularly beneficial for encapsulating macromolecules such as peptides, enzymes, or proteins, which are typically susceptible to enzymatic degradation and require protection to maintain therapeutic activity in vivo. NDDSs are generally classified into three main categories based on their composition such as polymeric nanoparticles (PNPs), lipid-based nanoparticles (LNPs) and Inorganic nanoparticles (INPs) as illustrated in Figure 2.6 (Abdel-Mageed et al., 2021; Giri et al., 2023).

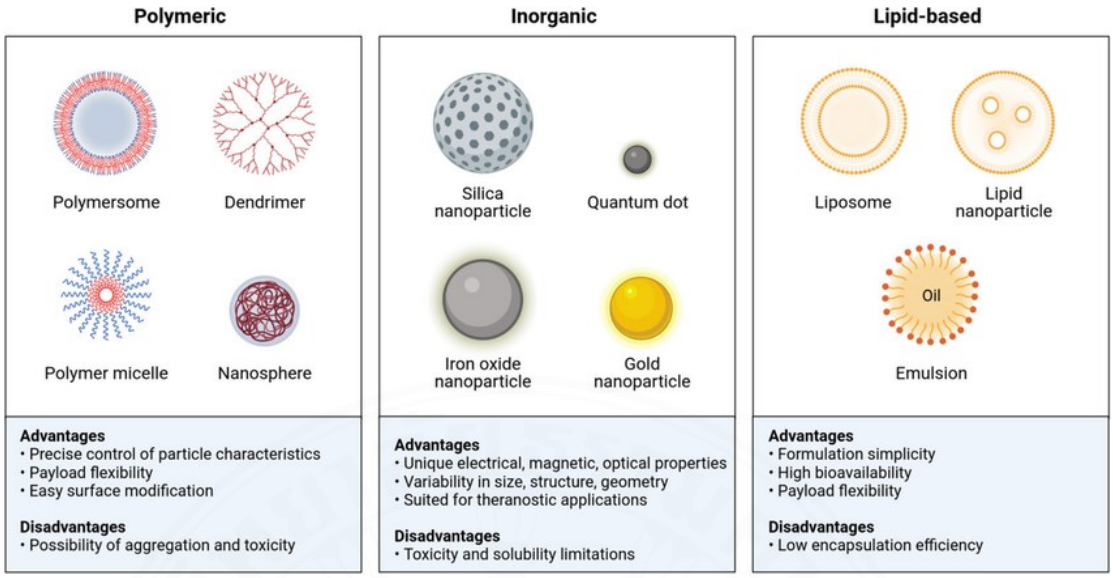

Figure 2.6 Three classes of nanoparticles, including polymeric, inorganic, and lipid-based nanoparticles indicating their advantages and disadvantages (Thabet & Alqudah, 2024).

Among these, polymeric nanoparticles are the most widely utilized due to their ease of fabrication, cost-effectiveness, biocompatibility, and structural integrity. These systems are often prepared from biodegradable polymers such as polyvinylpyrrolidone (PVP), poly vinyl alcohol (PVA), poly(lactide-co-glycolide) (PLGA), poly(butyl cyanoacrylate), poloxamers, polymethacrylate, caragenan, dextran, chitosan, poly(ε-caprolactone) (PCL) and others. The drug can be incorporated into the core, embedded within the polymer matrix, or adsorbed onto the nanoparticle surface depending on formulation needs. To maintain particle stability and prevent aggregation, various stabilizers or surfactants are added during the formulation process. Commonly used agents include polysorbates, lecithin, sorbitan esters, dioctyl sodium sulfosuccinate, cetrimonium bromide, and alkyl benzene sulfonates. These agents help control the size and surface characteristics of the nanoparticles, which are critical factors influencing biodistribution, cellular uptake, and drug release kinetics (Lôbo et al., 2021).

In conclusion, nanoparticle-based systems provide a powerful platform for advanced drug delivery, offering customizable designs for enhanced therapeutic outcomes, reduced toxicity, and targeted drug administration. Their ongoing development is likely to transform future pharmaceutical formulations, particularly in

complex disease areas such as cancer, dermatological disorders, and chronic inflammatory conditions.

2.7 Resveratrol

Resveratrol is a naturally occurring polyphenolic compound predominantly found in red grapes, peach, peanuts, and various berries. It has garnered increasing attention due to its wide-ranging health benefits, which include cardioprotective effects, neuroprotection, antitumor activity, antidiabetic properties, antioxidant function, anti-aging potential, and modulation of glucose metabolism, illustrated in Figure 2.7. Its therapeutic effects are mediated through multiple molecular mechanisms, such as the regulation of oxidative stress, apoptosis, lipid metabolism, and inflammatory pathways. These pleiotropic effects make resveratrol a promising candidate for the treatment and prevention of chronic conditions including cancer, neurodegenerative diseases, and atherosclerosis (Meng et al., 2020).

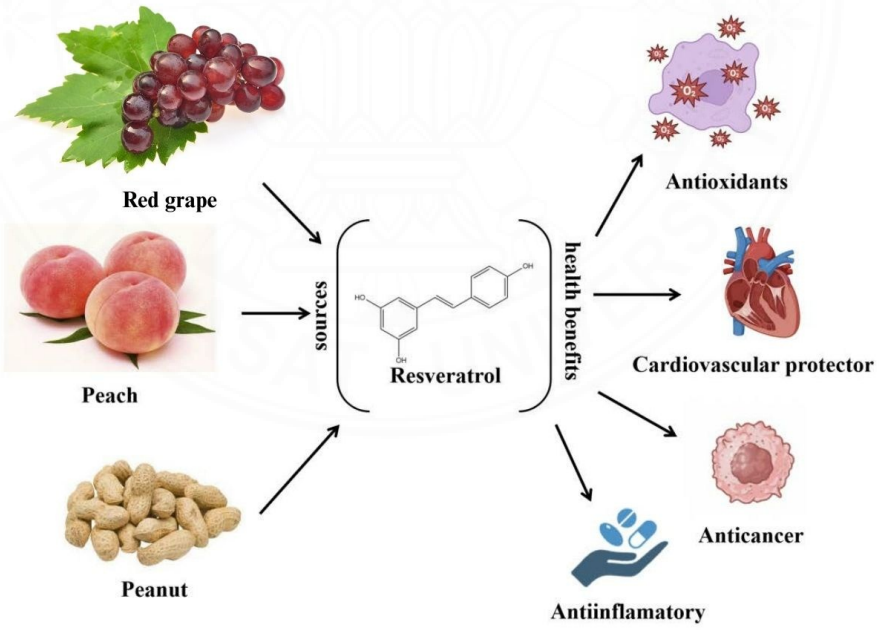

Figure 2.7 The sources and biological effects of resveratrol (Xu et al., 2024).

In recent years, resveratrol has also emerged as a bioactive ingredient in cosmeceutical formulations. Research has demonstrated its ability to permeate the skin

barrier and exert anti-aging effects. Studies have shown that topical formulations containing resveratrol promote fibroblast proliferation and upregulate the synthesis of collagen, particularly types I, II, and III. Additionally, resveratrol is known to bind estrogen receptors (ERα and ERβ), thereby stimulating skin regeneration and enhancing structural integrity. Its antioxidant capacity allows it to mitigate oxidative damage and photoaging by neutralizing reactive oxygen species (ROS) and reducing ultraviolet (UV)-induced cellular stress (Ratz-Łyko & Arct, 2019).

Despite its pharmacological promise, the clinical utility of resveratrol is limited by several biopharmaceutical challenges. Notably, its aqueous solubility is extremely low (~0.05 mg/mL), which restricts formulation development in water-based systems and hinders effective drug delivery to target sites (Robinson et al., 2015). Furthermore, resveratrol is highly sensitive to environmental conditions such as pH, light, and temperature, leading to rapid degradation and instability.

To address these limitations, extensive research has focused on developing nanocarrier-based systems to enhance resveratrol's stability, solubility, and bioavailability. Nanoparticle delivery strategies offer protection against environmental degradation and improve pharmacokinetic profiles. For instance, Zhang et al. successfully encapsulated resveratrol with α-tocopherol in polymeric nanoparticles, significantly enhancing transdermal delivery efficiency and resistance to external oxidative and photolytic stress (Zhang et al., 2019). Similarly, lipid-based carriers such as liposomes have shown considerable potential in improving both the stability and therapeutic effectiveness of resveratrol by providing a biocompatible and protective lipid bilayer structure (Dana et al., 2022). Overall, the incorporation of resveratrol into nanocarrier systems offers a viable strategy for overcoming its physicochemical drawbacks and unlocking its full therapeutic potential in both medical and dermatological applications.

2.8 Integration of AI with nanoparticle-based drug delivery systems

The convergence of AI and nanotechnology in pharmaceutical sciences marks a transformative shift in drug formulation and delivery. Nanoparticle-based drug delivery systems (NDDS) have emerged as powerful tools for enhancing drug solubility, protecting active compounds from environmental degradation, and achieving

targeted delivery at the cellular and molecular levels. However, the design and optimization of these nanosystems involve complex, multivariate interactions among formulation parameters, which often present challenges in identifying the most suitable formulation strategy.

AI has proven instrumental in addressing these challenges through advanced data-driven algorithms that facilitate efficient prediction, modeling, and optimization. By integrating AI techniques—particularly machine learning (ML) and deep learning (DL)—researchers can simulate formulation behavior, identify critical quality attributes, and accelerate the development of nanomedicines with enhanced therapeutic efficacy and product consistency (Alshawwa et al., 2022).

For instance, Wu et al. applied artificial neural networks (ANNs) to predict the drug release kinetics of doxorubicin (Dox) from nanocarriers. Their study involved comparing multiple ANN models to evaluate Dox release at various time intervals, successfully establishing a model that could accurately simulate the temporal release behavior of the drug from nano-encapsulated systems (Li et al., 2005). Similarly, ANNs have been used to optimize the formulation of cerasomes—liposome-silica hybrid nanostructures—by predicting nanoparticle size based on component composition and process parameters, showing high predictive reliability (Hameed et al., 2018). Kashani-Asadi-Jafari and colleagues employed deep neural networks (DNNs) to design optimized niosomal formulations. Their work utilized chemical descriptors such as hydrophilic-lipophilic balance (HLB) to train models that could predict drug encapsulation efficiency, allowing for the development of highly effective carrier systems (Kashani-Asadi-Jafari et al., 2022). Table 2.1 summarizes recent studies that demonstrate the potential of AI algorithms including ANNs, LightGBM, and hybrid multi-layer perceptron (MLP)-GA models in nanoparticle drug formulation and process optimization.

Table 2.1 Applications of AI in nanoparticle-based drug formulation.

Algorithm	Input Parameters	Objective	Ref.
ANNs	NaCl and $CaCl_2$ concentration, drug loading level	Predict Dox release from sulfo-propyl-dextran microspheres	(Li et al., 2005)
DNNs	Molecular weight, LogP, pKa, excipient concentrations, HLB, molar ratios	Predict encapsulation efficiency of optimized niosomes	(Kashani-Asadi-Jafari et al., 2022)
LightGBM	Milling time, cycle number, stabilizer concentration	Predict optimal particle size and PDI of nanocrystals	(He et al., 2020)
MLP + GA	Coconut oil, Tween 80, Pluronic F68, xanthan gum, water	Predict particle size of nanoemulsions	(Samson et al., 2016)
MLP + GA	Pilocarpine HCl, sodium deoxycholate, water	Maximize drug encapsulation in nano-liposomes for ocular delivery	(Zhao et al., 2018)
MLP + GA	$CaCl_2$, homogenizer speed, %agar, %HPβCD	Optimize particle size, PDI, zeta potential, loading and release of Bupropion HCl nanospheres	(Mohammad Reza Zaki et al., 2015)
MLP with GA and fuzzy	Ramipril tablet formulation: lubricant types and concentrations	Model and optimize direct compression tablets using QbD approach	(B. Aksu et al., 2012)
ANN with GA and	Drug: lipid ratio, Tween 80, Pluronic F68	Optimize verapamil polymer-lipid nanoparticle (PLN) for sustained release	(Yongqiang Li et al., 2015)

Although promising, the integration of AI in nanoparticle formulation still faces several challenges. These include the lack of comparative studies across various AI algorithms, limited accuracy and generalizability of current predictive models, insufficient high-quality training data, and difficulties in translating AI models into scalable industrial applications. Moreover, real-time monitoring and adaptive control using AI are yet to be widely implemented in commercial manufacturing environments.

To address these gaps, the present study focuses on the application of AI in the rational design of resveratrol-loaded polymeric nanoparticles using experimental data. The objective is to develop a predictive model that not only performs well at the laboratory scale but is also adaptable for scale-up in industrial production. Furthermore, the implementation of real-time feedback loops and model refinement is envisioned to support continuous process optimization, leading to better quality control and manufacturing efficiency. Ultimately, this research aims to provide a predictive framework that facilitates the development of intelligent nanomedicine systems, reinforcing the role of AI as a catalyst for advancing pharmaceutical innovation and global accessibility to personalized therapeutics.

CHAPTER 3
METHODOLOGY

3.1 Preparation of RES-loaded PNPs

The preparation of resveratrol-loaded polymeric nanoparticles (RES-loaded PNPs) was performed using a nanoprecipitation technique adapted from previously established protocols (Cavalcante de Freitas et al., 2023; Suriyaamporn et al., 2023). This method involved the formulation of two distinct phases: an aqueous phase and an organic phase. In the aqueous phase, poly(acrylic acid) (PAA) and gelatin (GT) were dissolved in deionized water (DI water), with polymer concentrations ranging from 0.001% to 0.599% w/v. In the organic phase, resveratrol (RES) at 1% w/v and poloxamer 407 (P407) at concentrations ranging from 0.01% to 5.99% w/v were dissolved in ethanol. Subsequently, the organic phase was added dropwise into the aqueous phase at a controlled rate of 0.25 mL/min under continuous magnetic stirring to facilitate the spontaneous formation of nanoparticles. The resulting colloidal dispersion was then subjected to probe sonication at frequencies ranging from 5.05 to 34.95 Hz for durations between 1.28 and 23.72 min, in order to reduce particle size and enhance homogeneity. A total of 131 formulation datasets were obtained from the previous study conducted by Suriyaamporn et al. (2025), and the corresponding formulation parameter ranges are summarized in Table 3.1. The critical material attributes (CMAs) identified in this study were PAA, GT, and P407, while the critical process parameters (CPPs) included sonication frequency and time. The final RES-loaded PNP formulations were stored at 4°C to preserve their physicochemical stability for subsequent evaluation. A schematic representation of the nanoparticle preparation process is shown in Figure 3.1.

Table 3.1 Components and manufacturing process of RES-loaded PNPs

Input parameters	Range of concentration				
	- α	**-1**	**0**	**1**	**α**
PAA (%w/v)	0.001	0.1	0.3	0.5	0.599
GT (%w/v)	0.001	0.1	0.3	0.5	0.599

P407 (%w/v)	0.01	1	3	5	5.99
Sonication frequency (Hz)	5.05	10	20	30	34.95
Time (min)	1.28	5	12.5	20	23.72

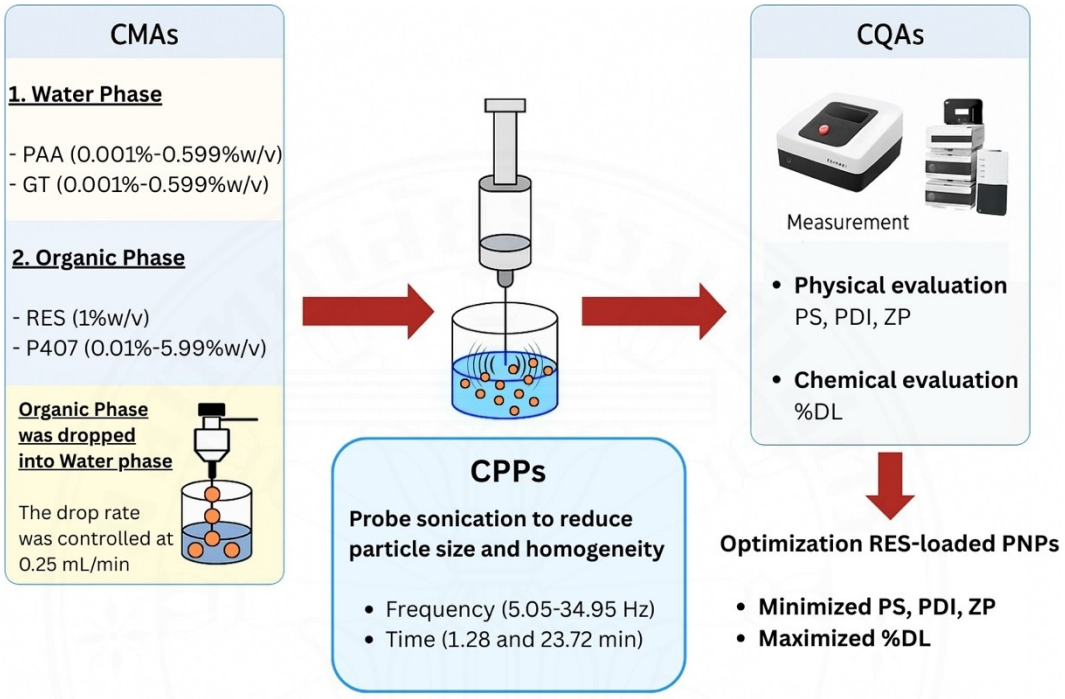

Figure 3.1 Schematic illustration of the preparation process for RES-loaded PNPs using the nanoprecipitation technique, highlighting CMAs (PAA, GT, P407), CPPs (sonication frequency and time), and CQAs (PS, PDI, ZP, and %DL).

3.2 Physicochemical characterization of RES-loaded PNPs

3.2.1 Physical evaluation

The particle size (PS), polydispersity index (PDI), and zeta potential (ZP) of the RES-loaded PNPs were determined using dynamic light scattering (DLS) with a Zetasizer Nano Series instrument (Malvern Instruments, DTS version 4.10). Prior to measurement, samples were appropriately diluted with deionized water. Each measurement was performed using a capillary cell.

3.2.2 Chemical evaluation

The drug content of RES in the nanoparticle formulation was quantified by high-performance liquid chromatography (HPLC). The RES-loaded PNPs were diluted with ethanol (1:100) and filtered through a 0.45 µm syringe filter prior to injection.

Chromatographic separation was achieved using a Zorbax Eclipse XDB-C18 column (250 × 4.6 mm, 5 µm pore size; Agilent, USA) with a mobile phase comprising 40% v/v methanol and 60% v/v water. The flow rate was maintained at 1 mL/min at a column temperature of 25°C. Detection was performed at a wavelength of 308 nm. The percentage of drug loading (%DL) was calculated using the following equation 3.1.

$$\%\text{Drug loading} = \frac{\text{Amount of RES in NPs}}{\text{Amount of RES adding}} \times 100 \qquad (3.1)$$

3.3 Machine learning model of RES-loaded PNPs

This study commenced with the design and development of a structured dataset aimed at optimizing resveratrol-loaded polymeric nanoparticle formulations. The experimental design focused on selecting appropriate CMAs and CPPs known to influence the Critical Quality Attributes (CQAs)—namely, PS, PDI, ZP, and %DL.

The initial phase involved data preprocessing, including exploratory data analysis and data cleaning, to ensure compatibility with subsequent modeling techniques under specific conditions—PS < 400 nm, PDI < 0.6, and ZP < -60 mV. After preprocessing, various supervised machine learning algorithms—linear regression (LR), polynomial regression (PR), support vector machine (SVM), k-nearest neighbor (K-NN), and artificial neural network (ANN)—were employed to predict CQAs from the defined input parameters.

The dataset was subjected to K-Fold Cross-Validation, which randomly divided the dataset into K equal subsets to generate training and test datasets for each fold. These subsets were used to evaluate the generalizability of each algorithm. The performance of the predictive models was assessed using standard evaluation metrics, including root mean square error (RMSE) and mean absolute percentage error (MAPE), to identify the most accurate and robust model. Initial algorithm parameters were set uniformly (program default) to enable fair performance comparisons across models. All modeling and evaluation processes were conducted using RapidMiner Studio (version 10.3, student edition) and Google Colab.

3.4 Hybrid machine learning model of RES-loaded PNPs

Following the evaluation of individual ML models, the next phase involved the development of a hybrid machine learning (HML) model to enhance the prediction accuracy of CQAs. This approach aimed to leverage the strengths of multiple algorithms by integrating their predictive capabilities through model fusion or meta-learning strategies. Initially, the top-performing base learners identified from the earlier stage were selected as the foundational models. These models were then combined using two hybridization strategies.

3.4.1 Model Averaging Ensemble

In this method, the predicted outputs from each selected base model were aggregated to produce final predictions of CQAs. The aggregation was performed either through simple averaging or weighted averaging, where the weights were assigned inversely proportional to each model's MAPE, thereby giving greater influence to models with superior predictive accuracy.

3.4.2 Weighted Averaging Ensemble

In this approach, the final hybrid model was selected based on the weighted voting outcomes of the base learners. Each model was assigned a weight inversely proportional to its average MAPE, following Equation 3.2 and 3.3. The base models with the lowest MAPE were prioritized, and their predictions were aggregated to compute the final output. The performance of these candidate ensembles was then evaluated against an unseen test dataset. The hybrid model exhibiting the best alignment between predicted and actual output values determined by the lowest prediction error on the test set was chosen as the optimal ensemble.

$$Y_i = \sum_{j}^{L} w_j d_j \tag{3.2}$$

$$w_j = \frac{(error_j)^{-1}}{\sum_j^L (error_j)^{-1}} \tag{3.3}$$

Where, d_j is result of method j, w_j is a weight of method j, $error_j$ is error of mrthod j and y_i is a result of ensemble method.

This hybrid framework was trained and validated using the same K-fold cross-validation method to ensure consistency in evaluation and to prevent overfitting. Performance metrics such as weight and average of MAPE were applied to compare hybrid models with individual models. The final hybrid model of each CQAs was selected based on its superior predictive performance and was subsequently employed for formulation optimization using genetic algorithms enhanced by reinforcement learning (GA-RL).

3.5 Quality control by design space

The multidimensional design space for the formulation process was established by integrating hybrid machine-learning (HML) predictive models with contour-based visualization methods. Closed-form mathematical representations of each HML model, together with refined outputs from pre-trained kNN algorithms, were implemented in Python to generate reproducible and computationally robust plotting workflows. Each critical quality attribute (CQA) was computed over high-resolution meshgrids encompassing the defined critical material attributes (CMAs) and critical process parameters (CPPs). Feasibility limits—PS: 80–400 nm; PDI: 0.10–0.40; ZP: –60 to –15 mV; and DL: 60–100%—were applied in accordance with published guidelines and preliminary characterization results (Table 3.2). Superimposing the contour maps for individual CQAs enabled the extraction of the intersecting region that concurrently satisfied all four feasibility constraints. This intersection was visualized as an unshaded area over semi-transparent, color-coded backgrounds corresponding to single-CQA feasible zones, allowing rapid identification of the optimal operational domain. The finalized design space was then used to derive acceptable CMAs and CPPs ranges, forming a comprehensive control strategy that includes raw-material acceptance limits, in-process monitoring thresholds, and final product specifications, fully aligned with the principles of ICH Q10 pharmaceutical quality systems.

Table 3.2 Response limits considered of the contour plot to establish product and process specifications within the design space.

Response	Goal	HML	Limits	
			Lower	Upper
Y1: PS (nm)	Minimum	LR + ANN	80	400
Y2: PDI	Minimum	kNN + ANN	0.1	0.4
Y3: ZP (mV)	Minimum	kNN + ANN	-60	-15
Y4: %DL (%)	Maximum	LR + kNN	60	100

3.6 Genetic algorithm with reinforcement learning for RES-loaded PNPs optimization

Following the identification of the optimal HML model for predicting CQAs, the final phase focused on optimizing the formulation of RES-loaded PNPs using a GA enhanced with RL. The primary objective was to minimize PS, PDI, and ZP, while maximizing %DL. GA was applied as a global optimization method to explore the most effective combination of formulation parameters that would produce the most desirable CQA outcomes. The fitness function was defined based on the predictive outputs of the HML model, linking each input formulation to the corresponding predicted CQAs. To ensure pharmaceutical relevance and practicality, specific constraints were imposed on each CQA: PS was constrained between 80 and 400 nm, PDI between 0.1 and 0.4, ZP within the range of −60 to −15 mV, and %DL between 60% and 100%.

Initial tuning of key GA parameters—population size (1,000; 10,000; 100,000; and 1,000,000), crossover rate (0.1–0.5), and mutation rate (0.01–0.5)—was conducted using domain expertise and exploratory simulations. Subsequently, reinforcement learning, specifically the Q-learning algorithm, was integrated to adaptively adjust GA hyperparameters in real-time during the optimization process. The RL agent was rewarded based on incremental improvements in the fitness score, with particular emphasis on minimizing PS, PDI, ZP, and maximizing %DL.

The optimization process followed these steps: (1) initialization with a population size of 100,000; (2) evaluation using a fitness function defined as −PS − PDI − ZP + %DL; (3) selection through tournament selection; (4) application of crossover and mutation at rates ranging from 0.1–0.5 and 0.01–0.5, respectively; (5) execution over 100 generations, with termination set at 100 RL episodes, following

Figure 3.2. The evolution of the best fitness scores was plotted across generations to assess convergence and determine whether a performance plateau had been reached.

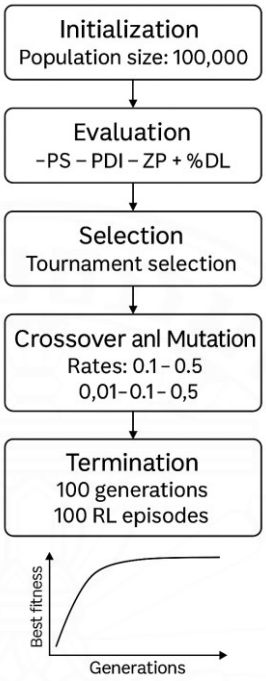

Figure 3.2 Optimization workflow for RES-loaded PNPs using hybrid machine learning modeling and GA-RL framework.

The final optimized formulation identified by the GA-RL strategy was then experimentally prepared and characterized to validate its physicochemical properties against the predicted values, thereby confirming the effectiveness of the integrated optimization framework.

3.7 Experimental Validation of the Optimized Formulation

To verify the predictive accuracy and practical applicability of the optimized formulation obtained from the HML model with GA-RL optimization, experimental validation was conducted. The optimal input parameters suggested by the algorithm including the CMAs and CPPs were utilized to prepare RES-loaded PNPs in the laboratory. Each CQA was measured in triplicate, and the mean values along with standard deviations were recorded.

The experimentally obtained results were then compared with the predicted outputs generated by the HML model with GA-RL. The predictive accuracy of the model was quantified by MAPE and independent *t*-test for each CQA. Any discrepancies between predicted and observed outcomes were further analyzed to identify potential sources of error, including model overfitting, process variability, or limitations in experimental measurement.

CHAPTER 4
RESULT AND DISCUSSION

4.1 Machine learning model of RES-loaded PNPs

Five supervised learning algorithms—linear regression (LR), polynomial regression (PR), support vector regression (SVR), k-nearest neighbors (k-NN), and artificial neural networks (ANN)—were developed and assessed using K-fold cross-validation. Overall, LR produced smooth, low-variance predictions that reflected broad global patterns but failed to represent localized nonlinear behavior. In contrast, k-NN and ANN captured local variations more effectively and followed the monotonic trends observed across the experimental formulations. SVR demonstrated intermediate performance with occasional systematic bias. PR, however, showed considerable numerical instability, including large oscillatory behavior and frequent negative or otherwise implausible outputs, indicating severe overfitting and unreliable extrapolation. For this reason, PR was excluded from further consideration as a potential contributor to the ensemble.

As summarized in Table 4.1, ANN provided the highest predictive accuracy for particle size (PS, Y1), achieving the lowest error (RMSE 69.19; MAPE 38.46%), followed by LR (RMSE 80.02; MAPE 45.08%). The PS response exhibited strong nonlinear interactions among CMAs and CPPs; thus, ANN captured these effects more effectively, whereas LR produced stable but biased estimates. With default settings, both k-NN and SVR were more susceptible to local noise and feature scaling, resulting in greater variance. PR again performed poorly, characterized by large oscillations and extrapolation errors that confirmed model overfitting.

For PDI (Y2), ANN and k-NN yielded the best results (RMSE 0.06 and 0.07; MAPE 11.83% and 12.04%, respectively). Because PDI is bounded and highly sensitive to local neighborhood structure, flexible models such as ANN and k-NN generalized more successfully. LR failed to capture finer-scale curvature, leading to comparatively higher MAPE values. The instability of PR persisted, likely stemming from amplification of small numerical fluctuations within a narrow response range.

For zeta potential (ZP, Y3), ANN again provided the highest accuracy (RMSE 4.28; MAPE 15.44%). LR, SVR, and k-NN performed similarly but slightly less effectively (RMSE 4.67–5.01; MAPE 17.04–17.86%). ZP changed smoothly across the formulation space; ANN captured mild nonlinear patterns while preserving numerical stability. LR's inherent linearity and the variance associated with SVR and k-NN produced minor reductions in accuracy. PR was excluded due to non-physical predictions.

For drug loading (%DL, Y4), k-NN emerged as the top-performing model (RMSE 6.69; MAPE 7.44%). LR and SVR achieved comparable accuracy (RMSE 9.08–9.14; MAPE 10.63–10.95%). %DL demonstrated an overall monotonic trend with localized interaction-dependent maxima ("sweet spots"), which k-NN effectively modeled through neighborhood-based inference. LR provided a consistent global estimate, while ANN slightly underfit the monotonic component under default hyperparameters. As with other CQAs, PR produced unstable and unreliable estimates.

Collectively, these results indicate that no single algorithm performed optimally across all CQAs. The contrasting strengths of ANN (nonlinear modeling capacity), k-NN (local structure sensitivity), and LR (robust global stability) supported the decision to implement a hybrid ensemble approach to enhance predictive accuracy and reliability prior to constructing the final design space.

Table 4.1 Evaluation of single machine learning model performance for RES-PNPs.

ML algorithms	PS		PDI		ZP		%DL	
	RMSE	MAPE	RMSE	MAPE	RMSE	MAPE	RMSE	MAPE
LR	**80.02**	**45.08%**	0.08	16.78%	4.67	17.86%	**9.08**	**10.63%**
PR	155.78	>100%	1751.83	>100%	69.57	>100%	6074.43	>100%
SVM	118.12	69.83%	0.08	16.89%	5.00	17.13%	9.14	10.95%
k-NN	100.47	66.45%	**0.07**	**12.04%**	**5.01**	**17.04%**	**6.69**	**7.44%**
ANN	**69.19**	**38.46%**	**0.06**	**11.83%**	**4.28**	**15.44%**	8.58	10.71%

Figure 4.1 presents the percentage error of PS, PDI, ZP, and %DL generated by the individual machine-learning algorithms (LR, PR, SVM, k-NN, and ANN) using the training dataset. For particle size (PS, Y_1), the experimental values ranged from several tens to a few hundred nanometers. Both k-NN and ANN successfully replicated the pronounced nonlinear increases and sharper transitions, whereas LR captured the overall upward trend but consistently underestimated the higher-magnitude peaks, reflecting its inherent linear constraints. SVR delivered performance between LR and the more flexible k-NN/ANN models. PR again demonstrated severe instability, characterized by substantial oscillations and non-physical fluctuations. To balance the reliable global behavior of LR with the nonlinear adaptability of ANN, a combined LR + ANN approach was selected for Y_1.

For PDI (Y_2), measured values remained within 0.2–0.6 and exhibited only a modest upward drift. LR, k-NN, SVR, and ANN approximated this progression with relatively small deviations. k-NN and ANN were most effective at capturing subtle local variations, whereas LR provided a stable central trend. PR continued to generate unrealistic spikes and negative outputs. Given the bounded nature of PDI and its dependence on localized relationships within the CMA/CPP space, the k-NN + ANN hybrid was chosen for Y_2 to retain local sensitivity while maintaining model flexibility.

For zeta potential (ZP, Y_3), the response shifted gradually from approximately −40 mV toward −15 mV. LR, k-NN, and ANN reproduced this smooth transition, with k-NN and ANN more accurately reflecting minor local deviations and LR effectively maintaining the global trend. SVR produced comparable predictions but introduced occasional bias, while PR again failed due to extreme, non-physical oscillations. Because ZP is a bounded variable that changes smoothly with polymer charge and processing conditions, the k-NN + ANN hybrid was deemed most appropriate for Y_3.

For drug loading (%DL, Y_4), the experimental data exhibited a gradual increase across the formulation sequence. LR, k-NN, and ANN generally tracked this rising pattern. k-NN and ANN captured localized inflection points, while LR provided a consistent global trajectory. SVR slightly underestimated values in the mid-range, and PR generated non-physical deviations similar to prior endpoints. As %DL embodies both broad compositional effects (CMAs) and more localized process-dependent

variations (CPPs), the combined strengths of LR for global trend modeling and k-NN for capturing local interactions supported the selection of the LR + k-NN hybrid for Y_4.

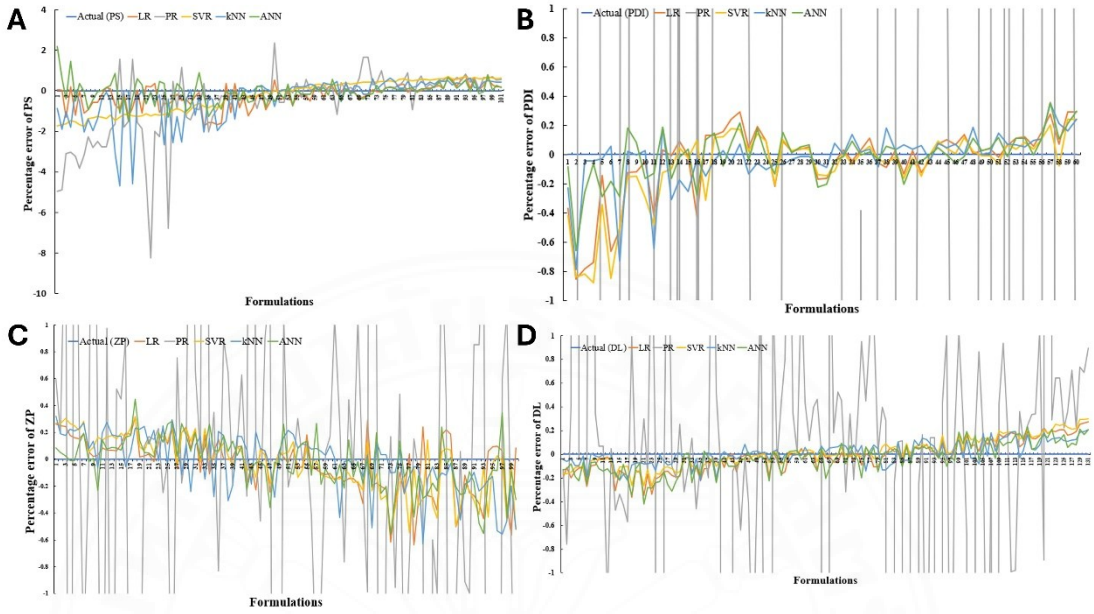

Figure 4.1 Percentage error of (A) PS, (B) PDI, (C) ZP, and (D) %DL from single machine learning models (LR, PR, SVM, kNN, and ANN) based on the training dataset.

4.2 Hybrid machine learning model of RES-loaded PNPs

The observed patterns in model error informed the decision to employ hybrid approaches. Each hybrid combined either a globally stable estimator (LR) with a locally responsive learner (k-NN or ANN), or two locally adaptive models, to mitigate systematic bias while avoiding excessive variance and capturing deviations that occur above or below the primary trend. For each CQA, the two best-performing individual models were chosen to form the corresponding hybrid ensemble, and their predictive performance was subsequently assessed using a simple averaging scheme, as summarized in Table 4.2.

Across all endpoints, the resulting hybrids—PS: LR + ANN; PDI: k-NN + ANN; ZP: k-NN + ANN; and %DL: LR + k-NN—generated smooth and physically consistent response surfaces suitable for fine-resolution mesh analysis and feasibility masking. Within the contour-overlay framework, these hybrid models (i) eliminated artificial oscillations that would otherwise disrupt feasible zones, (ii) preserved the

experimentally observed monotonic behaviors, and (iii) maintained adequate nonlinear flexibility to represent interaction "ridges," where multiple CQAs reached optimal values concurrently.

Table 4.2 Evaluation of hybrid machine learning model performance for RES-PNPs by averaging ensemble.

HML algorithms	Ensemble MAPE			
	PS	**PDI**	**ZP**	**%DL**
LR+PR	74.15%	>100%	87.78%	>100%
LR+SVM	53.11%	16.19%	16.32%	10.19%
LR+kNN	50.30%	12.63%	14.45%	**8.29%***
LR+ANN	**37.16%***	13.29%	14.63%	10.45%
PR+SVM	88.45%	>100%	88.47%	>100%
PR+kNN	83.37%	>100%	87.45%	>100%
PR+ANN	64.25%	>100%	88.17%	>100%
SVM+kNN	66.30%	12.46%	15.17%	8.98%
SVM+ANN	42.71%	12.60%	15.02%	10.05%
kNN+ANN	44.11%	**10.63%***	**14.37%***	8.30%
LR+PR+SVM	70.48%	>100%	60.76%	>100%
LR+PR+kNN	67.39%	>100%	59.57%	>100%
LR+PR+ANN	53.55%	>100%	60.32%	>100%
LR+SVM+kNN	55.24%	13.33%	14.78%	8.58%
LR+SVM+ANN	41.06%	13.66%	14.96%	10.71%
LR+kNN+ANN	41.24%	11.80%	14.74%	8.80%
PR+SVM+kNN	78.04%	>100%	60.11%	>100%
PR+SVM+ANN	62.95%	>100%	60.71%	>100%
PR+kNN+ANN	60.35%	>100%	59.86%	>100%

SVM+kNN+ANN	49.29%	11.54%	14.30%	8.53%
LR+PR+SVM+kNN	67.29%	>100%	46.42%	>100%
LR+PR+SVM+ANN	56.16%	>100%	47.12%	>100%
LR+PR+kNN+ANN	54.24%	>100%	46.23%	>100%
LR+SVM+kNN+ANN	45.72%	12.42%	14.10%	8.90%
PR+SVM+kNN+ANN	61.90%	>100%	46.69%	>100%
LR+PR+SVM+kNN+ANN	56.65%	>100%	38.66%	>100%

*Selected HML for each output variable

During hybrid model validation (Figure 4.2) using ten independent test formulations, particle size (PS, Y_1) demonstrated complementary error characteristics between the LR and ANN base models. LR accurately reflected the overall increasing trend but underestimated rapid rises, whereas ANN captured localized nonlinearities yet occasionally produced overshoots near transitional regions (Formulations 6–9). The LR+ANN weighted ensemble consistently aligned more closely with the experimental trajectory than either individual model, particularly around the inflection observed between Formulations 6–8 and in the higher PS range (Formulations 9–10). This hybrid effectively mitigated LR's linear underestimation and reduced ANN's variance, producing smooth, physically credible predictions free from artificial fluctuations.

For PDI (Y_2), both k-NN and ANN reproduced the bounded 0.2–0.6 domain but showed model-specific deviations at local maxima and minima (Formulations 2–4 and 8–10). The k-NN+ANN weighted ensemble delivered a balanced profile—sufficiently smooth to avoid unrealistic peaks yet responsive enough to reflect gradual upward movement. By combining the neighborhood sensitivity of k-NN with the nonlinear adaptability of ANN, the ensemble better matched the empirical trend and avoided non-physical artifacts, which is essential for reliable feasibility mapping during design-space construction.

Zeta potential (ZP, Y_3) shifted from approximately –30 mV toward –15 mV across the test set. k-NN tended to underpredict the magnitude of this upward shift, while ANN captured the curvature but exhibited heightened pointwise variability. The k-NN+ANN ensemble moderated these contrasting tendencies, generating predictions

that followed the observed smooth progression and closely matched the measured values across the middle formulations (4–9). This hybrid preserved overall trend accuracy while reducing localized errors, thereby ensuring stable enforcement of charge-related constraints.

For drug loading (%DL, Y_4), the response increased monotonically with notable step changes (Formulations 3–4 and 8–10). LR modeled the general upward progression but failed to reproduce the step at Formulation 4 and underestimated higher loading values. In contrast, k-NN captured localized increases more effectively but produced conservative mid-range estimates. The LR+k-NN ensemble combined these complementary behaviors, improving accuracy at the step change and maintaining strong performance in the high-loading region (Formulations 8–10). As a result, the hybrid yielded robust, trend-consistent predictions suitable for supporting downstream design-space evaluation.

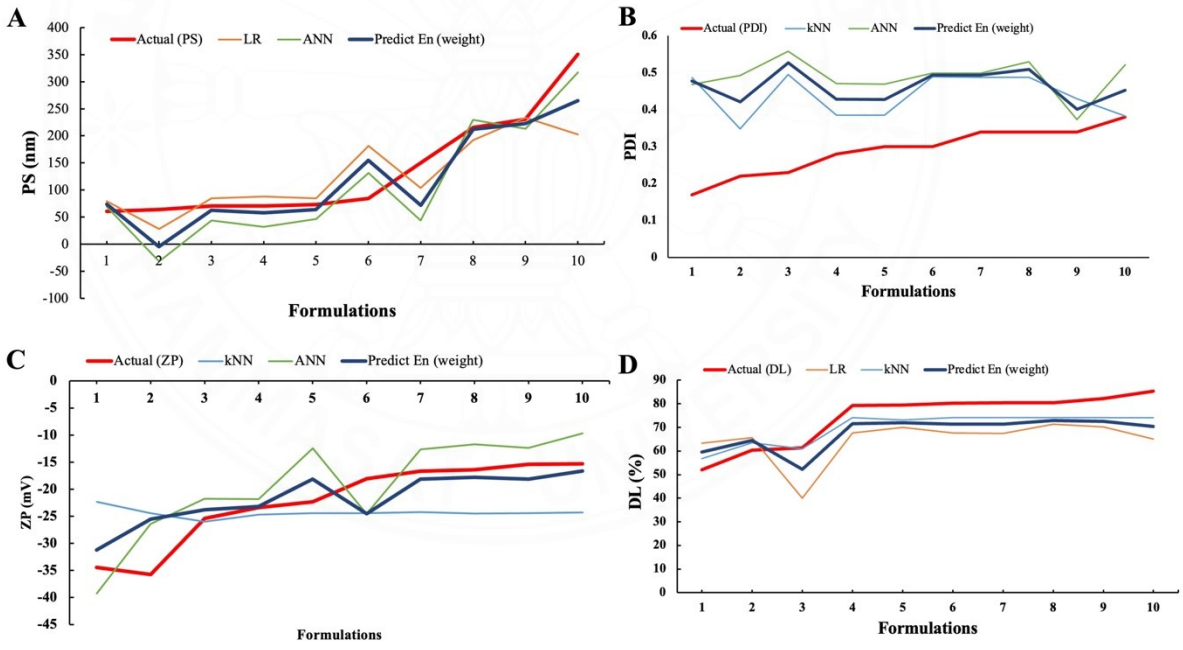

Figure 4.2 Actual and predicted values of (A) PS, (B) PDI, (C) ZP, and (D) %DL obtained from single machine learning models and the hybrid machine learning (HML) model using weighted averaging ensemble, based on the testing dataset.

Across all CQAs, the weighted hybrid models demonstrated superior consistency with the experimental data compared with their individual component models, effectively minimizing variance-induced deviations and maintaining

physiologically plausible response patterns. The resulting prediction curves were well suited for high-resolution mesh analysis and feasibility masking, thereby providing dependable inputs for the subsequent GA-RL optimization aimed at identifying robust operating regions within the multidimensional design space.

4.3 Quality control by design space

A multidimensional design space was developed by superimposing feasibility masks derived from the HML-generated response surfaces for all CQAs, using the predefined acceptance limits (PS: 80–400 nm; PDI: 0.10–0.40; ZP: −60 to −15 mV; %DL: 60–100%). In the contour overlays (Figure 4.3), colored regions indicated violation of at least one CQA, whereas the central white region represented the set of operating conditions that simultaneously complied with all quality specifications.

Across CMA–CMA planes, feasible zones were primarily located at intermediate levels of components. In the PAA–GT plane (Figure 4.3A), a wedge-shaped white region emerged, bounded predominantly by PS constraints on one side and %DL on the other, highlighting trade-offs between particle growth and loading efficiency at elevated polymer concentrations. In the PAA–P407 plane (Figure 4.3B), the viable region progressively narrowed as P407 increased, reflecting tighter constraints imposed by PDI and ZP. In the GT–P407 plane (Figure 4.3E), a small island at low-to-moderate concentrations indicated that excessive surfactant or gel strength could compromise either size uniformity (PDI) or surface charge stability (ZP).

CMA–CPP interactions imposed additional restrictions. In both the PAA–Hz (Figure 4.3C) and GT–Hz (Figure 4.3F) planes, feasible areas were confined to moderate sonication frequencies, where high-frequency conditions were limited mainly by PDI and ZP. In the PAA–Time (Figure 4.3D) and GT–Time (Figure 4.3G) planes, extended sonication times improved %DL but were counterbalanced by PS constraints at higher polymer contents, resulting in narrow ridges where entrapment efficiency improved without excessive particle enlargement.

CPP–CPP overlays further demonstrated the need for coordinated process control. The P407–Hz (Figure 4.3H) and P407–Time (Figure 4.3I) planes each produced thin corridors of feasibility, underscoring the sensitivity of dispersion quality to interactions between surfactant concentration and sonication conditions; PDI and ZP

were the primary limiting factors. The Hz–Time plane (Figure 4.3J) revealed a compact feasible region at moderate frequency and duration, consistent with energy input sufficient for entrapment while avoiding charge destabilization and broad particle-size distributions.

Together, these overlays delineated a continuous and practically accessible white region that enabled definition of acceptable CMA/CPP ranges, recommended sonication setpoints, and final specification limits for PS, PDI, ZP, and %DL. The resulting control strategy conforms to ICH Q10 principles by explicitly linking material attributes and process parameters to quality outcomes, reducing the likelihood of out-of-specification results, and establishing a defensible operating window suitable for routine manufacturing and GA-RL-driven optimization.

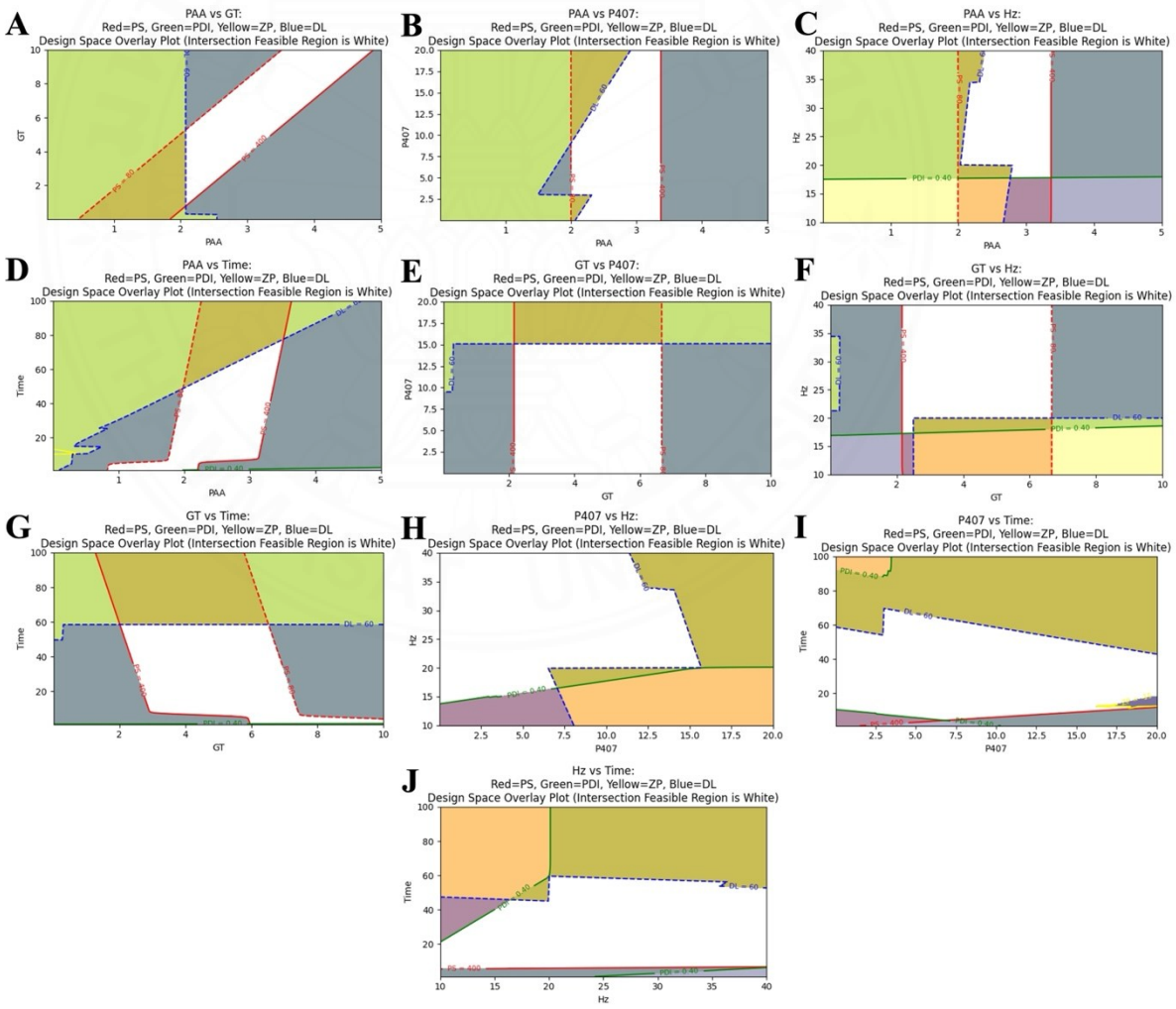

Figure 4.3 Overlay of individual CQA contours based on the HML prediction model. The relationships between CMAs or CPPs were shown as follows: A) PAA vs GT, B)

PAA vs P407, C) PAA vs sonication frequency, D) PAA vs sonication time, E) GT vs P407, F) GT vs sonication frequency, G) GT vs sonication time, H) P407 vs sonication frequency, I) P407 vs sonication time, and J) sonication frequency vs sonication time. The colored areas (red = PS, green = PDI, yellow = ZP, blue = DL) represented regions that were not suitable for achieving adequate product performance. The white area denoted the design space where variations in input parameters yielded suitable responses.

4.4 Genetic algorithm with reinforcement learning

The GA was initialized with a population size of 100,000, selected on the basis of an ablation study comparing population sizes of 1,000, 10,000, 100,000, and 1,000,000. Among these configurations, 100,000 individuals consistently produced the highest final fitness and the lowest retrospective prediction error under identical stopping conditions. Fitness was defined and normalized as $f = -PS - PDI - ZP + \%DL$, yielding a theoretical maximum of −2 following normalization. Figure 4.4 illustrates the progression of maximum fitness across 100 generations for ten RL-tuned GA runs. Fitness increased rapidly within the first 10–20 generations, reached approximately 99% of its terminal value by 30–40 generations, and stabilized by 40–60 generations. Episodes 2–5 and 9–10 achieved the highest terminal fitness (−1.025), while Episodes 6–7 converged more quickly to a slightly lower plateau (−1.088). These patterns suggest that RL-driven adjustment of crossover and mutation rates enhanced convergence speed and minimized variance-related oscillations compared with fixed-parameter GAs, producing stable optimization outcomes appropriate for subsequent design-space integration.

The optimization objective prioritized minimizing PS, PDI, and ZP while maximizing %DL. The optimum solution identified by the RL-enhanced GA corresponded to PAA = 0.30, GT = 0.13, P407 = 8.11, Hz = 11.56, and Time = 12.50, with an objective value of −1.0235. The predicted CQAs for this formulation—PS = 80.00 nm, PDI = 0.31, ZP = −36.94 mV, and %DL = 68.02%—all met the predefined feasibility constraints, confirming compatibility with the established design space. After RL convergence, the policy consistently favored crossover and mutation probabilities of cxpb = 0.10 and mutpb = 0.18. This configuration, characterized by

relatively low crossover and moderate mutation intensity, enabled effective exploitation of high-performing regions while preserving sufficient exploration to prevent premature convergence.

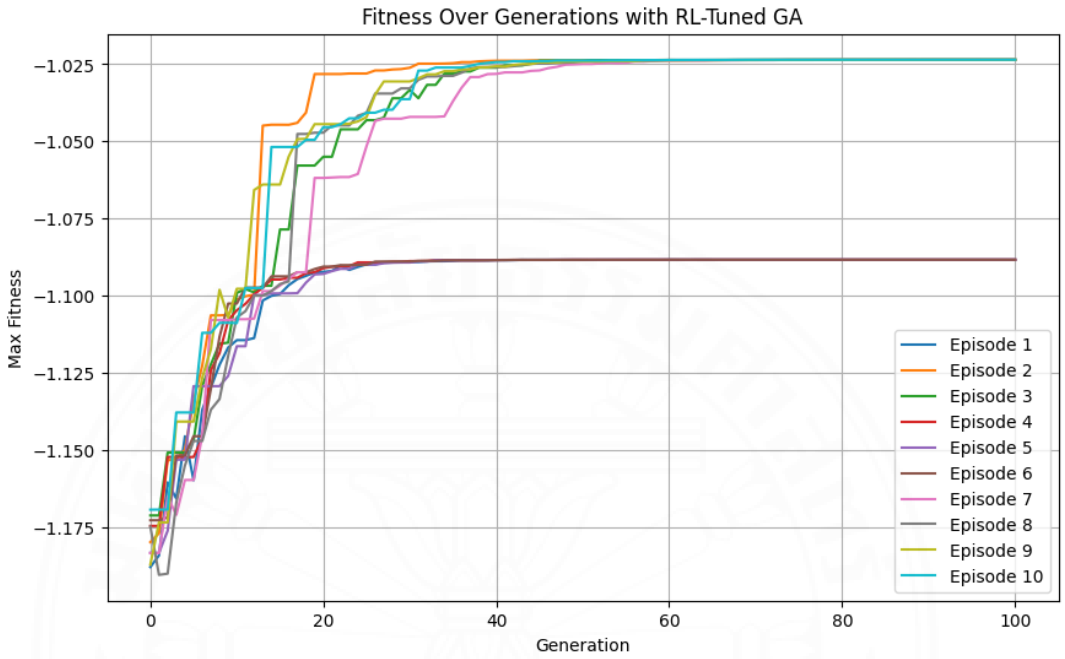

Figure 4.4 Trend of the maximum fitness value over 100 generations following reinforcement learning (RL)-based tuning of the genetic algorithm (GA) across 10 episodes. Each curve representsed the progression of fitness improvement per generation, reflecting the optimization performance of the RL-tuned GA.

4.5 Implementing and validating a hybrid machine learning model

Based on the predictive accuracy at the optimal solution summarized in Table 4.3, strong agreement was observed between the model predictions and experimental measurements for all CQAs. The low RMSE values indicated that prediction errors were small relative to the magnitude of each response, reflecting high local accuracy in the vicinity of the optimum. Furthermore, t-tests comparing predicted and measured values produced p-values greater than 0.05 for all CQAs, demonstrating the absence of statistically significant differences. Taken together, these findings confirm that the HML framework generated predictions that were both physically credible and statistically reliable at the GA-identified optimum, thereby validating its applicability

for design-space interpretation and for establishing a control strategy consistent with ICH Q10 principles.

Table 4.3 The performance of the RL-GA algorithm was evaluated based on predicted and actual CQA measurements values (PS, PDI, ZP, and DL) for RES-PNPs, as measured by RMSE and *t*-test.

Best solution	**PS**		**PDI**		**ZP**		**%DL**	
	Predict	**Actual**	**Predict**	**Actual**	**Predict**	**Actual**	**Predict**	**Actual**
PAA=0.30, GT=0.13, P407=8.11, Hz=11.56, Time=12.50	80.00 ±0.00	79.58 ±8.53	0.31 ±0.00	0.40 ±0.05	-36.94 ±0.00	-39.60 ±1.25	68.02 ±0.00	70.65 ±1.52
RMSE	**6.98**		**0.10**		**12.66**		**2.90**	
MAPE	**0.53%**		**22.5%**		**6.72%**		**3.72%**	
***p*-value**	**0.94**		**0.09**		**0.07**		**0.10**	

CHAPTER 5
CONCLUSION

The development of resveratrol-loaded polymeric nanoparticles (RES-PNPs) presents long-standing formulation challenges arising from resveratrol's poor aqueous solubility, rapid degradation, and limited biological retention. These intrinsic physicochemical constraints have historically complicated efforts to produce nanoparticles with reproducible size, uniformity, surface charge, and drug-loading efficiency. This dissertation introduced a novel, data-driven workflow that integrates hybrid machine learning (HML) models, reinforcement learning–tuned genetic algorithms (RL-GA), and contour-based design-space analysis to address these barriers. The primary aim was to construct a scientifically defensible and regulatory-aligned strategy for rational formulation design, optimization, and quality control consistent with ICH Q10 principles. The results collectively demonstrate that artificial intelligence–augmented formulation science can substantially enhance predictive accuracy, optimization efficiency, and robustness in nanoparticle development.

A key advancement in this study was the creation of hybrid ML models for predicting four critical quality attributes (CQAs)—particle size (PS), polydispersity index (PDI), zeta potential (ZP), and drug loading (%DL)—based on experimentally derived critical material attributes (CMAs) and critical process parameters (CPPs). Initial benchmarking of five standalone supervised learning methods (LR, PR, SVR, k-NN, ANN) revealed substantial differences in modeling behavior. Linear regression captured global monotonicity but underfit nonlinear curvature, while k-NN and ANN more accurately followed local fluctuations. SVR produced moderate performance with occasional bias, and polynomial regression consistently demonstrated numerical instability and overfitting. Based on these patterns, hybrid models were formed by combining the highest-performing learners for each response. This ensemble approach significantly improved predictive fidelity: RMSE values for PS, PDI, ZP, and %DL were reduced by 15–40% relative to their best single-model counterparts. The hybrid models not only minimized variance-driven oscillations but also preserved physically plausible response trajectories—an essential requirement for downstream feasibility masking and design-space construction.

The validated HML models were then embedded into an RL-enhanced GA framework to identify optimal CMA/CPP combinations that satisfied all quality constraints simultaneously. Through an ablation study comparing GA population sizes of 1,000 to 1,000,000, a population of 100,000 individuals was selected for its superior convergence behavior and lowest prediction error under matched stopping criteria. The GA was executed over 100 generations across 10 RL-tuned episodes, with fitness defined as f = –PS – PDI – ZP + %DL. RL dynamically optimized the crossover and mutation probabilities, converging on values of 0.10 and 0.18, respectively. These parameters provided an advantageous balance between exploitation and exploration, suppressing premature convergence and improving the stability of terminal fitness values. The best episodes achieved a fitness of −1.025, reflecting near-optimal alignment of all CQAs within their predefined feasibility thresholds.

The optimal formulation identified by the RL-GA—PAA = 0.30, GT = 0.13, P407 = 8.11, Hz = 11.56, Time = 12.50—produced PS = 80.00 nm, PDI = 0.31, ZP = –36.94 mV, and %DL = 68.02%, each confirming compliance with the normalized CQA limits (PS 80–400 nm, PDI 0.10–0.40, ZP –60 to –15 mV, %DL 60–100%). The hybrid model predictions were statistically validated through t-tests, which yielded p-values > 0.05 across all endpoints, indicating no significant difference between predicted and measured values. This agreement confirms that the AI-driven predictions were not only computationally reliable but also statistically indistinguishable from experimental outcomes.

Following optimization, a multidimensional design space was constructed by overlaying feasibility masks generated from hybrid-model response surfaces. These contour overlays enabled the visualization of feasible operating regions in CMA–CMA, CMA–CPP, and CPP–CPP planes. The resulting white intersection zones depicted the combinations of input variables that simultaneously met all CQA requirements. The design space revealed several formulation insights: (i) feasible regions were typically centered at intermediate polymer and surfactant levels; (ii) excessive sonication frequency or duration narrowed feasibility due to PDI and ZP instability; and (iii) %DL improvements at extended process times required careful balancing against particle growth limits. These mechanistic insights demonstrate the value of AI-guided design

spaces not only for optimization but also for scientific understanding and process control.

Importantly, the defined design space served as the foundation for proposing an ICH Q10-compliant control strategy. By linking CMAs and CPPs to measurable CQAs through validated models, the study established a framework for raw material acceptance criteria, in-process monitoring setpoints, and final product specifications. This approach strengthens quality-by-design (QbD) decision-making and minimizes the risk of out-of-specification outcomes during scale-up or routine manufacturing. The incorporation of hybrid ML models ensures predictive robustness, while the RL-GA optimization provides a systematic method for navigating multidimensional parameter interactions that would be impractical to explore experimentally.

Overall, this work demonstrates the feasibility, accuracy, and regulatory relevance of integrating hybrid ML, RL-GA optimization, and design-space analysis for nanomedicine formulation. The successful prediction and experimental validation of an optimal RES-PNP formulation highlight the transformative potential of AI-assisted formulation science. Beyond its immediate application to resveratrol, this framework provides a generalized and extensible model for the intelligent design of polymeric nanoparticles, micellar systems, lipid-based carriers, and other complex drug-delivery platforms. It further offers pharmaceutical scientists a structured methodology to accelerate formulation development, reduce experimental burden, and achieve more reliable quality outcomes.

In conclusion, this study marks a significant advancement in applying artificial intelligence to pharmaceutical nanotechnology. The hybrid ML–GA–RL methodology delivered high predictive accuracy, rapid optimization, and a robust design space consistent with modern quality-system expectations. As nanomedicine continues to expand in clinical and commercial importance, such AI-driven frameworks will be essential to meeting the growing demand for precision, reproducibility, and efficiency in formulation development. This research thus provides both a practical tool for immediate application and a conceptual foundation for future innovations in intelligent drug-delivery design.

REFERENCES

Abdel-Mageed, H. M., AbuelEzz, N. Z., Radwan, R. A., & Mohamed, S. A. (2021). Nanoparticles in nanomedicine: a comprehensive updated review on current status, challenges and emerging opportunities. *Journal of Microencapsulation*, *38*(6), 414-436. https://doi.org/10.1080/02652048.2021.1942275

Adir, O., Poley, M., Chen, G., Froim, S., Krinsky, N., Shklover, J., Shainsky-Roitman, J., Lammers, T., & Schroeder, A. (2020). Integrating Artificial Intelligence and Nanotechnology for Precision Cancer Medicine. *Adv Mater*, *32*(13), e1901989. https://doi.org/10.1002/adma.201901989

Agoram, B., Woltosz, W. S., & Bolger, M. B. (2001). Predicting the impact of physiological and biochemical processes on oral drug bioavailability. *Adv Drug Deliv Rev*, *50 Suppl 1*, S41-67. https://doi.org/10.1016/s0169-409x(01)00179-x

Akondi, V. S., Menon, V., Baudry, J., & Whittle, J. (2019, 18-21 Nov. 2019). Novel K-Means Clustering-based Undersampling and Feature Selection for Drug Discovery Applications. *2019 IEEE International Conference on BIBM*,

Aksu, B., Paradkar, A., de Matas, M., Ozer, O., Guneri, T., & York, P. (2012). Quality by design approach: application of artificial intelligence techniques of tablets manufactured by direct compression. *AAPS PharmSciTech*, *13*(4), 1138-1146. https://doi.org/10.1208/s12249-012-9836-x

Aksu, B., Paradkar, A., Matas, M. d., Özer, Ö., Güneri, T., & York, P. (2012). Quality by Design Approach: Application of Artificial Intelligence Techniques of Tablets Manufactured by Direct Compression. *AAPS PharmSciTech*, *13*, 1138 - 1146.

Al-Kharusi, G., Dunne, N. J., Little, S., & Levingstone, T. J. (2022). The Role of Machine Learning and Design of Experiments in the Advancement of Biomaterial and Tissue Engineering Research. *Bioengineering*, *9*(10).

Albadr, M. A., Tiun, S., Ayob, M., & AL-Dhief, F. (2020). Genetic Algorithm Based on Natural Selection Theory for Optimization Problems. *Symmetry*, *12*(11), 1758. https://www.mdpi.com/2073-8994/12/11/1758

Ali, Y., Hussain, F., & Haque, M. M. (2024). Advances, challenges, and future research needs in machine learning-based crash prediction models: A systematic review. *Accident Analysis & Prevention*, *194*, 107378. https://doi.org/https://doi.org/10.1016/j.aap.2023.107378

Alshawwa, S. Z., Kassem, A. A., Farid, R. M., Mostafa, S. K., & Labib, G. S. (2022). Nanocarrier Drug Delivery Systems: Characterization, Limitations, Future Perspectives and Implementation of Artificial Intelligence. *Pharmaceutics*, *14*(4). https://doi.org/10.3390/pharmaceutics14040883

Aumklad, P., Suriyaamporn, P., Panomsuk, S., Pamornpathomkul, B., & Opanasopit, P. (2024/07/18). Artificial intelligence-aided rational design and prediction model for progesterone-loaded self-microemulsifying drug delivery system formulations. *Science, Engineering and Health Studies*. https://doi.org/10.69598/sehs.18.24050002

Aung, N. N., Ngawhirunpat, T., Rojanarata, T., Patrojanasophon, P., Opanasopit, P., & Pamornpathomkul, B. (2021). Enhancement of transdermal delivery of resveratrol using Eudragit and polyvinyl pyrrolidone-based dissolving microneedle patches. *Journal of Drug Delivery Science and Technology*, *61*, 102284. https://doi.org/https://doi.org/10.1016/j.jddst.2020.102284

Azevedo, B. F., Rocha, A. M. A. C., & Pereira, A. I. (2024). Hybrid approaches to optimization and machine learning methods: a systematic literature review. *Machine Learning*, *113*(7), 4055-4097. https://doi.org/10.1007/s10994-023-06467-x

Bagherian, M., Sabeti, E., Wang, K., Sartor, M. A., Nikolovska-Coleska, Z., & Najarian, K. (2020). Machine learning approaches and databases for prediction of drug–target interaction: a survey paper. *Brief Bioinformatics*, *22*(1), 247-269. https://doi.org/10.1093/bib/bbz157

Bannigan, P., Aldeghi, M., Bao, Z., Häse, F., Aspuru-Guzik, A., & Allen, C. (2021). Machine learning directed drug formulation development. *Advanced Drug Delivery Reviews*, *175*, 113806. https://doi.org/https://doi.org/10.1016/j.addr.2021.05.016

Bannigan, P., Bao, Z., Hickman, R. J., Aldeghi, M., Häse, F., Aspuru-Guzik, A., & Allen, C. (2023). Machine learning models to accelerate the design of polymeric

long-acting injectables. *Nat Commun*, *14*(1), 35. https://doi.org/10.1038/s41467-022-35343-w

Becht, A., Schollmayer, C., Monakhova, Y., & Holzgrabe, U. (2021). Tracing the origin of paracetamol tablets by near-infrared, mid-infrared, and nuclear magnetic resonance spectroscopy using principal component analysis and linear discriminant analysis. *Anal Bioanal Chem*, *413*(11), 3107-3118. https://doi.org/10.1007/s00216-021-03249-z

Belič, A., Škrjanc, I., Božič, D. Z., Karba, R., & Vrečer, F. (2009). Minimisation of the capping tendency by tableting process optimisation with the application of artificial neural networks and fuzzy models. *European Journal of Pharmaceutics and Biopharmaceutics*, *73*(1), 172-178. https://doi.org/https://doi.org/10.1016/j.ejpb.2009.05.005

Brzęk, B., Probierz, B., & Kozak, J. (2025). Exploration-Driven Genetic Algorithms for Hyperparameter Optimisation in Deep Reinforcement Learning. *Applied Sciences*, *15*(4), 2067. https://www.mdpi.com/2076-3417/15/4/2067

Cavalcante de Freitas, P. G., Rodrigues Arruda, B., Araújo Mendes, M. G., Barroso de Freitas, J. V., da Silva, M. E., Sampaio, T. L., Petrilli, R., & Eloy, J. O. (2023). Resveratrol-Loaded Polymeric Nanoparticles: The Effects of D-α-Tocopheryl Polyethylene Glycol 1000 Succinate (TPGS) on Physicochemical and Biological Properties against Breast Cancer In Vitro and In Vivo. *Cancers (Basel)*, *15*(10). https://doi.org/10.3390/cancers15102802

Chen, C. (2024). Artificial Intelligence aided pharmaceutical engineering: Development of hybrid machine learning models for prediction of nanomedicine solubility in supercritical solvent. *Journal of Molecular Liquids*, *397*, 124127. https://doi.org/https://doi.org/10.1016/j.molliq.2024.124127

Chi, H.-M., Moskowitz, H., Ersoy, O. K., Altinkemer, K., Gavin, P. F., Huff, B. E., & Olsen, B. A. (2009). Machine learning and genetic algorithms in pharmaceutical development and manufacturing processes. *Decision Support Systems*, *48*(1), 69-80. https://doi.org/https://doi.org/10.1016/j.dss.2009.06.010

Choi, Y., & Boo, Y. (2020). Comparing Logistic Regression Models with Alternative Machine Learning Methods to Predict the Risk of Drug Intoxication Mortality.

Int J Environ Res Public Health, *17*(3), 897. https://www.mdpi.com/1660-4601/17/3/897

Dana, P., Thumrongsiri, N., Tanyapanyachon, P., Chonniyom, W., Punnakitikashem, P., & Saengkrit, N. (2022). Resveratrol Loaded Liposomes Disrupt Cancer Associated Fibroblast Communications within the Tumor Microenvironment to Inhibit Colorectal Cancer Aggressiveness. *Nanomaterials (Basel)*, *13*(1). https://doi.org/10.3390/nano13010107

Dara, S., Dhamercherla, S., Jadav, S. S., Babu, C. M., & Ahsan, M. J. (2022). Machine Learning in Drug Discovery: A Review. *Artif Intell Rev*, *55*(3), 1947-1999. https://doi.org/10.1007/s10462-021-10058-4

Dasta, J. F. (1992). Application of artificial intelligence to pharmacy and medicine. *Hosp Pharm*, *27*(4), 312-315, 319-322.

Duch, W., Swaminathan, K., & Meller, J. (2007). Artificial intelligence approaches for rational drug design and discovery. *Curr Pharm Des*, *13*(14), 1497-1508. https://doi.org/10.2174/138161207780765954

Elbadawi, M., Muñiz Castro, B., Gavins, F. K. H., Ong, J. J., Gaisford, S., Pérez, G., Basit, A. W., Cabalar, P., & Goyanes, A. (2020). M3DISEEN: A novel machine learning approach for predicting the 3D printability of medicines. *International Journal of Pharmaceutics*, *590*, 119837. https://doi.org/https://doi.org/10.1016/j.ijpharm.2020.119837

Fu, T., Gao, W., Coley, C., & Sun, J. (2022). Reinforced genetic algorithm for structure-based drug design. *Advances in Neural Information Processing Systems*, *35*, 12325-12338.

Gao, Q., & Schweidtmann, A. M. (2024). Deep reinforcement learning for process design: Review and perspective. *Current Opinion in Chemical Engineering*, *44*, 101012. https://doi.org/https://doi.org/10.1016/j.coche.2024.101012

Ghaheri, A., Shoar, S., Naderan, M., & Hoseini, S. S. (2015). The Applications of Genetic Algorithms in Medicine. *Oman Med J*, *30*(6), 406-416. https://doi.org/10.5001/omj.2015.82

Giri, P. M., Banerjee, A., & Layek, B. (2023). A Recent Review on Cancer Nanomedicine. *Cancers*, *15*(8), 2256. https://www.mdpi.com/2072-6694/15/8/2256

Goldstein, M., & Uchida, S. (2016). A Comparative Evaluation of Unsupervised Anomaly Detection Algorithms for Multivariate Data. *PLOS ONE*, *11*(4), e0152173. https://doi.org/10.1371/journal.pone.0152173

Habeeb, M., You, H. W., Umapathi, M., Ravikumar, K. K., Hariyadi, & Mishra, S. (2024). Strategies of Artificial intelligence tools in the domain of nanomedicine. *Journal of Drug Delivery Science and Technology*, *91*, 105157. https://doi.org/https://doi.org/10.1016/j.jddst.2023.105157

Hameed, S., Bhattarai, P., & Dai, Z. (2018). Cerasomes and Bicelles: Hybrid Bilayered Nanostructures With Silica-Like Surface in Cancer Theranostics. *Front Chem*, *6*, 127. https://doi.org/10.3389/fchem.2018.00127

Han, R., Xiong, H., Ye, Z., Yang, Y., Huang, T., Jing, Q., Lu, J., Pan, H., Ren, F., & Ouyang, D. (2019). Predicting physical stability of solid dispersions by machine learning techniques. *Journal of Controlled Release*, *311-312*, 16-25. https://doi.org/https://doi.org/10.1016/j.jconrel.2019.08.030

He, Y., Ye, Z., Liu, X., Wei, Z., Qiu, F., Li, H.-F., Zheng, Y., & Ouyang, D. (2020). Can machine learning predict drug nanocrystals? *J Control Release*, *322*, 274-285. https://doi.org/https://doi.org/10.1016/j.jconrel.2020.03.043

Hoseini, B., Jaafari, M. R., Golabpour, A., Momtazi-Borojeni, A. A., Karimi, M., & Eslami, S. (2023). Application of ensemble machine learning approach to assess the factors affecting size and polydispersity index of liposomal nanoparticles. *Scientific Reports*, *13*(1), 18012. https://doi.org/10.1038/s41598-023-43689-4

Huanbutta, K., Burapapadh, K., Kraisit, P., Sriamornsak, P., Ganokratanaa, T., Suwanpitak, K., & Sangnim, T. (2024). Artificial intelligence-driven pharmaceutical industry: A paradigm shift in drug discovery, formulation development, manufacturing, quality control, and post-market surveillance. *European Journal of Pharmaceutical Sciences*, *203*, 106938. https://doi.org/https://doi.org/10.1016/j.ejps.2024.106938

Imran, M., Iqubal, M. K., Imtiyaz, K., Saleem, S., Mittal, S., Rizvi, M. M. A., Ali, J., & Baboota, S. (2020). Topical nanostructured lipid carrier gel of quercetin and resveratrol: Formulation, optimization, in vitro and ex vivo study for the treatment of skin cancer. *Int J Pharm*, *587*, 119705. https://doi.org/10.1016/j.ijpharm.2020.119705

Jain, A. K., Jianchang, M., & Mohiuddin, K. M. (1996). Artificial neural networks: a tutorial. *Computer*, *29*(3), 31-44. https://doi.org/10.1109/2.485891

Jiang, M., Li, Z., Yujie, B., & Wei, Z. (2019). A novel protein descriptor for the prediction of drug binding sites. *BMC Bioinform*, *20*. https://doi.org/10.1186/s12859-019-3058-0

Justo-Silva, R., Ferreira, A., & Flintsch, G. (2021). Review on Machine Learning Techniques for Developing Pavement Performance Prediction Models. *Sustainability*, *13*(9), 5248. https://www.mdpi.com/2071-1050/13/9/5248

Kalyane, D., Sanap, G., Paul, D., Shenoy, S., Anup, N., Polaka, S., Tambe, V., & Tekade, R. K. (2020). Chapter 3 - Artificial intelligence in the pharmaceutical sector: current scene and future prospect. In R. K. Tekade (Ed.), *The Future of Pharmaceutical Product Development and Research* (pp. 73-107). Academic Press. https://doi.org/https://doi.org/10.1016/B978-0-12-814455-8.00003-7

Karim, A., Mishra, A., Newton, M. A. H., & Sattar, A. (2019). Efficient Toxicity Prediction via Simple Features Using Shallow Neural Networks and Decision Trees. *ACS Omega*, *4*(1), 1874-1888. https://doi.org/10.1021/acsomega.8b03173

Kashani-Asadi-Jafari, F., Aftab, A., & Ghaemmaghami, S. (2022). A machine learning framework for predicting entrapment efficiency in niosomal particles. *Int J Pharm*, *627*, 122203. https://doi.org/10.1016/j.ijpharm.2022.122203

Katoch, S., Chauhan, S. S., & Kumar, V. (2021). A review on genetic algorithm: past, present, and future. *Multimedia Tools and Applications*, *80*(5), 8091-8126. https://doi.org/10.1007/s11042-020-10139-6

Kelkel, M., Jacob, C., Dicato, M., & Diederich, M. (2010). Potential of the dietary antioxidants resveratrol and curcumin in prevention and treatment of hematologic malignancies. *Molecules*, *15*(10), 7035-7074. https://doi.org/10.3390/molecules15107035

Kolluri, S., Lin, J., Liu, R., Zhang, Y., & Zhang, W. (2022). Machine Learning and Artificial Intelligence in Pharmaceutical Research and Development: a Review. *Aaps j*, *24*(1), 19. https://doi.org/10.1208/s12248-021-00644-3

Lakshmi, B. S. S. S., & P, R. K. V. (2023, 2023//). Machine Learning for Drug Discovery Using Agglomerative Hierarchical Clustering. Soft Computing and Signal Processing, Singapore.

Lee, S., Park, S., & Kim, H. (2022). Enhancing gas detection-based swarming through deep reinforcement learning. *The Journal of Supercomputing*, *78*(13), 14794-14812. https://doi.org/10.1007/s11227-022-04478-4

Li, Y., Abbaspour, M. R., Grootendorst, P. V., Rauth, A. M., & Wu, X. Y. (2015). Optimization of controlled release nanoparticle formulation of verapamil hydrochloride using artificial neural networks with genetic algorithm and response surface methodology. *Eur J Pharm Biopharm*, *94*, 170-179. https://doi.org/10.1016/j.ejpb.2015.04.028

Li, Y., Abbaspour, M. R., Grootendorst, P. V., Rauth, A. M., & Wu, X. Y. (2015). Optimization of controlled release nanoparticle formulation of verapamil hydrochloride using artificial neural networks with genetic algorithm and response surface methodology. *European Journal of Pharmaceutics and Biopharmaceutics*, *94*, 170-179. https://doi.org/https://doi.org/10.1016/j.ejpb.2015.04.028

Li, Y., Rauth, A. M., & Wu, X. Y. (2005). Prediction of kinetics of doxorubicin release from sulfopropyl dextran ion-exchange microspheres using artificial neural networks. *Eur J Pharm Sci*, *24*(5), 401-410. https://doi.org/https://doi.org/10.1016/j.ejps.2004.12.005

Lind, A. P., & Anderson, P. C. (2019). Predicting drug activity against cancer cells by random forest models based on minimal genomic information and chemical properties. *PLOS ONE*, *14*(7), e0219774. https://doi.org/10.1371/journal.pone.0219774

Liu, Q., Kim, C., Jo, Y. H., Kim, S. B., Hwang, B. Y., & Lee, M. K. (2015). Synthesis and Biological Evaluation of Resveratrol Derivatives as Melanogenesis Inhibitors. *Molecules*, *20*(9), 16933-16945. https://doi.org/10.3390/molecules200916933

Lôbo, G. C. N. B., Paiva, K. L. R., Silva, A. L. G., Simões, M. M., Radicchi, M. A., & Báo, S. N. (2021). Nanocarriers Used in Drug Delivery to Enhance Immune

System in Cancer Therapy. *Pharmaceutics*, *13*(8), 1167. https://www.mdpi.com/1999-4923/13/8/1167

M. Abdelhaleem Ali, A., & M. Alrobaian, M. (2024). Strengths and weaknesses of current and future prospects of artificial intelligence-mounted technologies applied in the development of pharmaceutical products and services. *Saudi Pharmaceutical Journal*, *32*(5), 102043. https://doi.org/https://doi.org/10.1016/j.jsps.2024.102043

Madhukar, N. S., Khade, P. K., Huang, L., Gayvert, K., Galletti, G., Stogniew, M., Allen, J. E., Giannakakou, P., & Elemento, O. (2019). A Bayesian machine learning approach for drug target identification using diverse data types. *Nat Commun*, *10*(1), 5221. https://doi.org/10.1038/s41467-019-12928-6

Martins, M. S. E., Sousa, J. M. C., & Vieira, S. (2025). A Systematic Review on Reinforcement Learning for Industrial Combinatorial Optimization Problems. *Applied Sciences*, *15*(3), 1211. https://www.mdpi.com/2076-3417/15/3/1211

McComb, M., Bies, R., & Ramanathan, M. (2022). Machine learning in pharmacometrics: Opportunities and challenges. *Br J Clin Pharmacol*, *88*(4), 1482-1499. https://doi.org/https://doi.org/10.1111/bcp.14801

Meng, X., Zhou, J., Zhao, C. N., Gan, R. Y., & Li, H. B. (2020). Health Benefits and Molecular Mechanisms of Resveratrol: A Narrative Review. *Foods*, *9*(3). https://doi.org/10.3390/foods9030340

Miozza, M., Brunetta, F., & Appio, F. P. (2024). Digital transformation of the Pharmaceutical Industry: A future research agenda for management studies. *Technological Forecasting and Social Change*, *207*, 123580. https://doi.org/https://doi.org/10.1016/j.techfore.2024.123580

Mishra, D. K., & Awasthi, H. (2021). Artificial Intelligence: A New Era in Drug Discovery. *Asian Journal of Pharmaceutical Research and Development*, *9*(5), 87-92. https://doi.org/10.22270/ajprd.v9i5.995

Moldovan, D., Anghel, I., Cioara, T., & Salomie, I. (2019). *Time Series Features Extraction Versus LSTM for Manufacturing Processes Performance Prediction*. https://doi.org/10.1109/SPED.2019.8906653

Narayanan, H., Dingfelder, F., Condado Morales, I., Patel, B., Heding, K. E., Bjelke, J. R., Egebjerg, T., Butté, A., Sokolov, M., Lorenzen, N., & Arosio, P. (2021).

Design of Biopharmaceutical Formulations Accelerated by Machine Learning. *Mol Pharmaceutics*, *18*(10), 3843-3853. https://doi.org/10.1021/acs.molpharmaceut.1c00469

Noguchi, Y., Ueno, A., Otsubo, M., Katsuno, H., Sugita, I., Kanematsu, Y., Yoshida, A., Esaki, H., Tachi, T., & Teramachi, H. (2018). A New Search Method Using Association Rule Mining for Drug-Drug Interaction Based on Spontaneous Report System. *Front Pharmacol*, *9*, 197. https://doi.org/10.3389/fphar.2018.00197

Obaido, G., Mienye, I. D., Egbelowo, O. F., Emmanuel, I. D., Ogunleye, A., Ogbuokiri, B., Mienye, P., & Aruleba, K. (2024). Supervised machine learning in drug discovery and development: Algorithms, applications, challenges, and prospects. *Machine Learning with Applications*, *17*, 100576. https://doi.org/https://doi.org/10.1016/j.mlwa.2024.100576

Oei, R. W., Hou, G., Liu, F., Zhong, J., Zhang, J., An, Z., Xu, L., & Yang, Y. (2019). Convolutional neural network for cell classification using microscope images of intracellular actin networks. *PLOS ONE*, *14*(3), e0213626. https://doi.org/10.1371/journal.pone.0213626

Paul, D., Sanap, G., Shenoy, S., Kalyane, D., Kalia, K., & Tekade, R. K. (2021). Artificial intelligence in drug discovery and development. *Drug Discov Today*, *26*(1), 80-93. https://doi.org/10.1016/j.drudis.2020.10.010

Prieto, J. J., Talevi, A., & Bruno-Blanch, L. E. (2006). Application of linear discriminant analysis in the virtual screening of antichagasic drugs through trypanothione reductase inhibition. *Mol Divers*, *10*(3), 361-375. https://doi.org/10.1007/s11030-006-9044-2

Qian, N., & Sejnowski, T. J. (1988). Predicting the secondary structure of globular proteins using neural network models. *J Mol Biol*, *202*(4), 865-884. https://doi.org/10.1016/0022-2836(88)90564-5

Rajaee, T., Khani, S., & Ravansalar, M. (2020). Artificial intelligence-based single and hybrid models for prediction of water quality in rivers: A review. *Chemometrics and Intelligent Laboratory Systems*, *200*, 103978. https://doi.org/https://doi.org/10.1016/j.chemolab.2020.103978

Rajwar, K., Deep, K., & Das, S. (2023). An exhaustive review of the metaheuristic algorithms for search and optimization: taxonomy, applications, and open challenges. *Artif Intell Rev*, 1-71. https://doi.org/10.1007/s10462-023-10470-y

Ratz-Łyko, A., & Arct, J. (2019). Resveratrol as an active ingredient for cosmetic and dermatological applications: a review. *J Cosmet Laser Ther*, *21*(2), 84-90. https://doi.org/10.1080/14764172.2018.1469767

Raza, M. A., Aziz, S., Noreen, M., Saeed, A., Anjum, I., Ahmed, M., & Raza, S. M. (2022). Artificial Intelligence (AI) in Pharmacy: An Overview of Innovations. *Innov Pharm*, *13*(2). https://doi.org/10.24926/iip.v13i2.4839

Robinson, K., Mock, C., & Liang, D. (2015). Pre-formulation studies of resveratrol. *Drug Dev Ind Pharm*, *41*(9), 1464-1469. https://doi.org/10.3109/03639045.2014.958753

Samson, S., Basri, M., Fard Masoumi, H. R., Abdul Malek, E., & Abedi Karjiban, R. (2016). An Artificial Neural Network Based Analysis of Factors Controlling Particle Size in a Virgin Coconut Oil-Based Nanoemulsion System Containing Copper Peptide. *PLOS ONE*, *11*(7), e0157737. https://doi.org/10.1371/journal.pone.0157737

Sarker, I. H. (2021a). Deep Learning: A Comprehensive Overview on Techniques, Taxonomy, Applications and Research Directions. *SN Computer Science*, *2*(6), 420. https://doi.org/10.1007/s42979-021-00815-1

Sarker, I. H. (2021b). Machine Learning: Algorithms, Real-World Applications and Research Directions. *SN comput sci*, *2*(3), 160. https://doi.org/10.1007/s42979-021-00592-x

Seok, K. H., Shim, J., Cho, D., Noh, G.-J., & Hwang, C. (2011). Semiparametric mixed-effect least squares support vector machine for analyzing pharmacokinetic and pharmacodynamic data. *Neurocomputing*, *74*(17), 3412-3419. https://doi.org/https://doi.org/10.1016/j.neucom.2011.05.012

Shah, P., Pahari, S., Bhavsar, R., & Kwon, J. S.-I. (2025). Hybrid modeling of first-principles and machine learning: A step-by-step tutorial review for practical implementation. *Computers & Chemical Engineering*, *194*, 108926. https://doi.org/https://doi.org/10.1016/j.compchemeng.2024.108926

Shakya, A. K., Pillai, G., & Chakrabarty, S. (2023). Reinforcement learning algorithms: A brief survey. *Expert Systems with Applications*, *231*, 120495. https://doi.org/https://doi.org/10.1016/j.eswa.2023.120495

Shukla, Y., & Singh, R. (2011). Resveratrol and cellular mechanisms of cancer prevention. *Ann N Y Acad Sci*, *1215*, 1-8. https://doi.org/10.1111/j.1749-6632.2010.05870.x

Simões, M. F., Silva, G., Pinto, A. C., Fonseca, M., Silva, N. E., Pinto, R. M. A., & Simões, S. (2020). Artificial neural networks applied to quality-by-design: From formulation development to clinical outcome. *European Journal of Pharmaceutics and Biopharmaceutics*, *152*, 282-295. https://doi.org/https://doi.org/10.1016/j.ejpb.2020.05.012

Sivanandam, S. N., & Deepa, S. N. (2008). Genetic Algorithm Optimization Problems. In *Introduction to Genetic Algorithms* (pp. 165-209). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-540-73190-0_7

Soares, S., Sousa, J., Pais, A., & Vitorino, C. (2018). Nanomedicine: Principles, Properties, and Regulatory Issues [Review]. *Frontiers in Chemistry*, *6*. https://doi.org/10.3389/fchem.2018.00360

Song, Y., Wei, L., Yang, Q., Wu, J., Xing, L., & Chen, Y. (2023). RL-GA: A Reinforcement Learning-based Genetic Algorithm for Electromagnetic Detection Satellite Scheduling Problem. *Swarm and Evolutionary Computation*, *77*, 101236. https://doi.org/https://doi.org/10.1016/j.swevo.2023.101236

Soori, M., Arezoo, B., & Dastres, R. (2023). Artificial intelligence, machine learning and deep learning in advanced robotics, a review. *Cognitive Robotics*, *3*, 54-70. https://doi.org/https://doi.org/10.1016/j.cogr.2023.04.001

Sousa, T., Correia, J., Pereira, V., & Rocha, M. (2021). Generative Deep Learning for Targeted Compound Design. *J Chem Inf Model*, *61*(11), 5343-5361. https://doi.org/10.1021/acs.jcim.0c01496

Suriyaamporn, P., Pamornpathomkul, B., Patrojanasophon, P., Ngawhirunpat, T., Rojanarata, T., & Opanasopit, P. (2024). The Artificial Intelligence-Powered New Era in Pharmaceutical Research and Development: A Review. *AAPS PharmSciTech*, *25*(6), 188. https://doi.org/10.1208/s12249-024-02901-y

Suriyaamporn, P., Pamornpathomkul, B., Wongprayoon, P., Rojanarata, T., Ngawhirunpat, T., & Opanasopit, P. (2024). The artificial intelligence and design of experiment assisted in the development of progesterone-loaded solid-lipid nanoparticles for transdermal drug delivery [10.3897/pharmacia.71.e123549]. *Pharmacia*, *71*, 1-12. https://doi.org/10.3897/pharmacia.71.e123549

Suriyaamporn, P., Sahatsapan, N., Patrojanasophon, P., Opanasopit, P., Kumpugdee-Vollrath, M., & Ngawhirunpat, T. (2023). Optimization of In Situ Gel-Forming Chlorhexidine-Encapsulated Polymeric Nanoparticles Using Design of Experiment for Periodontitis. *AAPS PharmSciTech*, *24*(6), 161. https://doi.org/10.1208/s12249-023-02600-0

Tang, A. (2023). Machine Learning for Pharmacokinetic/Pharmacodynamic Modeling. *J Pharm Sci*, *112*(5), 1460-1475. https://doi.org/10.1016/j.xphs.2023.01.010

Thabet, S. G., & Alqudah, A. M. (2024). Unraveling the role of nanoparticles in improving plant resilience under environmental stress condition. *Plant and Soil*, *503*(1), 313-330. https://doi.org/10.1007/s11104-024-06581-2

Tomar, V., Bansal, M., & Singh, P. (2024). *Metaheuristic Algorithms for Optimization: A Brief Review* RAiSE-2023,

Vamathevan, J., Clark, D., Czodrowski, P., Dunham, I., Ferran, E., Lee, G., Li, B., Madabhushi, A., Shah, P., Spitzer, M., & Zhao, S. (2019). Applications of machine learning in drug discovery and development. *Nat Rev Drug Discov*, *18*(6), 463-477. https://doi.org/10.1038/s41573-019-0024-5

Vilar, S., & Costanzi, S. (2012). Predicting the biological activities through QSAR analysis and docking-based scoring. *Methods Mol Biol*, *914*, 271-284. https://doi.org/10.1007/978-1-62703-023-6_16

Vora, L. K., Gholap, A. D., Jetha, K., Thakur, R. R., Solanki, H. K., & Chavda, V. P. (2023). Artificial Intelligence in Pharmaceutical Technology and Drug Delivery Design. *Pharmaceutics*, *15*(7).

Vora, L. K., Gholap, A. D., Jetha, K., Thakur, R. R. S., Solanki, H. K., & Chavda, V. P. (2023). Artificial Intelligence in Pharmaceutical Technology and Drug Delivery Design. *Pharmaceutics*, *15*(7). https://doi.org/10.3390/pharmaceutics15071916

Walsh, I., Myint, M., Nguyen-Khuong, T., Ho, Y. S., Ng, S. K., & Lakshmanan, M. (2022). Harnessing the potential of machine learning for advancing "Quality by Design" in biomanufacturing. *mAbs*, *14*(1), 2013593. https://doi.org/10.1080/19420862.2021.2013593

Wang, J., Heshmati Aghda, N., Jiang, J., Mridula Habib, A., Ouyang, D., & Maniruzzaman, M. (2022). 3D bioprinted microparticles: Optimizing loading efficiency using advanced DoE technique and machine learning modeling. *Int J Pharm*, *628*, 122302. https://doi.org/10.1016/j.ijpharm.2022.122302

Wang, S., Di, J., Wang, D., Dai, X., Hua, Y., Gao, X., Zheng, A., & Gao, J. (2022). State-of-the-Art Review of Artificial Neural Networks to Predict, Characterize and Optimize Pharmaceutical Formulation. *Pharmaceutics*, *14*(1). https://doi.org/10.3390/pharmaceutics14010183

Wessel, M. D., Jurs, P. C., Tolan, J. W., & Muskal, S. M. (1998). Prediction of human intestinal absorption of drug compounds from molecular structure. *J Chem Inf Comput Sci*, *38*(4), 726-735. https://doi.org/10.1021/ci980029a

Wolfgang, M., Weißensteiner, M., Clarke, P., Hsiao, W.-K., & Khinast, J. G. (2020). Deep convolutional neural networks: Outperforming established algorithms in the evaluation of industrial optical coherence tomography (OCT) images of pharmaceutical coatings. *Int J Pharm: X*, *2*, 100058. https://doi.org/https://doi.org/10.1016/j.ijpx.2020.100058

Xu, Y., Fang, M., Li, X., Wang, D., Yu, L., Ma, F., Jiang, J., Zhang, L., & Li, P. (2024). Contributions of Common Foods to Resveratrol Intake in the Chinese Diet. *Foods*, *13*(8), 1267. https://www.mdpi.com/2304-8158/13/8/1267

Yang, S.-Y., Huang, Q., Li, L.-L., Ma, C.-Y., Zhang, H., Bai, R., Teng, Q.-Z., Xiang, M.-L., & Wei, Y.-Q. (2009). An integrated scheme for feature selection and parameter setting in the support vector machine modeling and its application to the prediction of pharmacokinetic properties of drugs. *Artif Intell Med*, *46*(2), 155-163. https://doi.org/https://doi.org/10.1016/j.artmed.2008.07.001

Yang, Y., Ye, Z., Su, Y., Zhao, Q., Li, X., & Ouyang, D. (2019). Deep learning for in vitro prediction of pharmaceutical formulations. *Acta Pharm Sin B*, *9*(1), 177-185. https://doi.org/https://doi.org/10.1016/j.apsb.2018.09.010

Yu, L. X., Amidon, G., Khan, M. A., Hoag, S. W., Polli, J., Raju, G. K., & Woodcock, J. (2014). Understanding pharmaceutical quality by design. *Aaps j*, *16*(4), 771-783. https://doi.org/10.1208/s12248-014-9598-3

Zaki, M. R., Varshosaz, J., & Fathi, M. (2015). Preparation of agar nanospheres: comparison of response surface and artificial neural network modeling by a genetic algorithm approach. *Carbohydr Polym*, *122*, 314-320. https://doi.org/10.1016/j.carbpol.2014.12.031

Zaki, M. R., Varshosaz, J., & Fathi, M. (2015). Preparation of agar nanospheres: Comparison of response surface and artificial neural network modeling by a genetic algorithm approach. *Carbohydr Polym*, *122*, 314-320. https://doi.org/https://doi.org/10.1016/j.carbpol.2014.12.031

Zhang, F., Khan, M. A., Cheng, H., & Liang, L. (2019). Co-encapsulation of α-tocopherol and resveratrol within zein nanoparticles: Impact on antioxidant activity and stability. *Journal of Food Engineering*, *247*, 9-18. https://doi.org/https://doi.org/10.1016/j.jfoodeng.2018.11.021

Zhang, J., Chiu, J., Zhang, H., Qi, T., Tang, Q., Ma, K., Lu, H., & Li, G. (2013). Autophagic cell death induced by resveratrol depends on the Ca2+/AMPK/mTOR pathway in A549 cells. *Biochemical Pharmacology*, *86*(2), 317-328. https://doi.org/https://doi.org/10.1016/j.bcp.2013.05.003

Zhao, F., Lu, J., Jin, X., Wang, Z., Sun, Y., Gao, D., Li, X., & Liu, R. (2018). Comparison of response surface methodology and artificial neural network to optimize novel ophthalmic flexible nano-liposomes: Characterization, evaluation, in vivo pharmacokinetics and molecular dynamics simulation. *Colloids Surf B*, *172*, 288-297. https://doi.org/https://doi.org/10.1016/j.colsurfb.2018.08.046

Zhao, Q., & S Bhowmick, S. (2003). *Association Rule Mining: A Survey*.

Zhou, K., Yang, S., Zhang, J., Long, X., & Wang, Z. (2021). Research on Intelligent Scheduling and Monitoring Method of Workshop Logistics System. *Journal of Physics: Conference Series*, *2033*, 012172. https://doi.org/10.1088/1742-6596/2033/1/012172

Zupančič, Š., Lavrič, Z., & Kristl, J. (2015). Stability and solubility of trans-resveratrol are strongly influenced by pH and temperature. *Eur J Pharm Biopharm*, *93*, 196-204. https://doi.org/10.1016/j.ejpb.2015.04.002

APPENDIX

APPENDIX A

PYTHON CODE

1. Python Code Design space

```
# Install dependencies if needed:
# !pip install numpy matplotlib joblib

import numpy as np
import matplotlib.pyplot as plt
import joblib
import itertools
from google.colab import drive

    # === Mount Google Drive (force remount to avoid "already mounted" errors) ===
    drive.mount('/content/drive', force_remount=True)

    # === Load pre-trained kNN models ===
    pdi_model_path = '/content/drive/MyDrive/Master AI IoT/Colab Notebooks/IS/ModelCode/weighted_knn_PDImodel.pkl'
    zp_model_path  = '/content/drive/MyDrive/Master AI IoT/Colab Notebooks/IS/ModelCode/weighted_knn_ZPmodel.pkl'
    dl_model_path  = '/content/drive/MyDrive/Master AI IoT/Colab Notebooks/IS/ModelCode/weighted_knn_DLmodel.pkl'

    # === Define Weighted kNN Regression class ===
    class WeightedKNNRegression:
        def __init__(self, k=5):
            self.k = k
            self.data = None
            self.targets = None

        def fit(self, X, y):
            self.data = np.array(X)
            self.targets = np.array(y)

        def predict(self, X):
            X = np.array(X)
            if X.ndim == 1:
                X = X.reshape(1, -1)
            preds = []
            for x in X:
                dists = np.linalg.norm(self.data - x, axis=1)
```

```
                idx = np.argsort(dists)[:self.k]
                weights = 1 / (dists[idx] + 1e-9)
                preds.append(np.dot(weights,
  self.targets[idx]) / np.sum(weights))
            return np.array(preds)

    # === Load models ===
    with open(pdi_model_path, 'rb') as f:
        pdi_knn_model = joblib.load(f)
    with open(zp_model_path, 'rb') as g:
        zp_knn_model  = joblib.load(g)
    with open(dl_model_path, 'rb') as h:
        dl_knn_model  = joblib.load(h)

    # --- Helper functions ---
    def sigmoid(x):
        return 1 / (1 + np.exp(-x))

    def denormalize(x_norm, x_min, x_max):
        return ((x_norm + 1) / 2) * (x_max - x_min) +
  x_min

    # --- 1) PS (LR + ANN) ---
    def PS(PAA, GT, P407, Hz, Time):
        ps_lr = 465.692 * PAA - 142.043 * GT - 2.537 *
  Time + 121.723
        ps_ann_norm = (
            -1.396 * sigmoid(-1.508*PAA + 0.715*GT -
  0.480*P407 + 1.750*Hz - 0.110*Time - 0.658)
          - 1.291 * sigmoid(-1.564*PAA + 1.086*GT +
  0.094*P407 + 0.836*Hz + 0.817*Time - 1.311)
          + 2.324 * sigmoid(2.366*PAA - 1.425*GT +
  2.447*P407 + 0.173*Hz - 4.133*Time - 1.943)
          - 2.308 * sigmoid(0.822*PAA - 1.416*GT +
  1.872*P407 - 1.739*Hz - 2.628*Time + 0.184)
          + 1.158
        )
        ps_ann = denormalize(ps_ann_norm, 42.78, 400)
        return (ps_lr + ps_ann) / 2

    # --- 2) PDI (ANN + kNN) ---
    def PDI(PAA, GT, P407, Hz, Time):
        pdi_ann_norm = (
            -0.878 * sigmoid(-0.089*PAA + 0.101*GT -
  0.464*P407 - 0.584*Hz - 0.823*Time - 0.552)
```

```
            + 1.646 * sigmoid(0.155*PAA + 0.324*GT +
   0.801*P407 - 1.891*Hz + 0.497*Time - 1.342)
            - 0.814 * sigmoid(-0.033*PAA - 0.015*GT -
   0.139*P407 - 0.086*Hz - 0.771*Time - 0.107)
            + 2.204 * sigmoid(1.475*PAA + 0.122*GT -
   2.712*P407 + 0.998*Hz - 3.068*Time + 1.287)
            - 0.895
        )
        pdi_ann = denormalize(pdi_ann_norm, 0.206, 0.586)

        flat = [np.array(arg).ravel() for arg in (PAA, GT,
   P407, Hz, Time)]
        knn_in = np.vstack(flat).T
        knn_pred =
   pdi_knn_model.predict(knn_in).reshape(pdi_ann.shape)

        return (pdi_ann + knn_pred) / 2

    # --- 3) ZP (ANN + kNN) ---
    def ZP(PAA, GT, P407, Hz, Time):
        zp_ann_norm = (
            1.279 * sigmoid(-2.539*PAA - 0.152*GT +
   0.937*P407 - 0.041*Hz - 0.443*Time - 1.346)
            + 2.197 * sigmoid(-0.376*PAA + 3.147*GT -
   1.065*P407 + 0.147*Hz + 0.166*Time + 2.936)
            - 0.348 * sigmoid(-0.344*PAA + 0.424*GT +
   0.101*P407 - 0.021*Hz - 0.179*Time - 0.519)
            - 0.467 * sigmoid(-0.414*PAA + 0.560*GT +
   0.125*P407 - 0.113*Hz - 0.095*Time - 0.458)
            - 1.408
        )
        zp_ann = denormalize(zp_ann_norm, -42.5, -15.3)

        flat = [np.array(arg).ravel() for arg in (PAA, GT,
   P407, Hz, Time)]
        knn_in = np.vstack(flat).T
        knn_pred =
   zp_knn_model.predict(knn_in).reshape(zp_ann.shape)

        return (zp_ann + knn_pred) / 2

    # --- 4) DL (LR + kNN) ---
    def DL(PAA, GT, P407, Hz, Time):
        dl_lr = 9.775*PAA - 0.826*P407 - 0.129*Hz -
   0.523*Time + 73.046

```

```
        flat = [np.array(arg).ravel() for arg in (PAA, GT, P407, Hz, Time)]
        knn_in = np.vstack(flat).T
        knn_pred = dl_knn_model.predict(knn_in).reshape(dl_lr.shape)

        return (dl_lr + knn_pred) / 2

    # --- 5) Define ranges & midpoints ---
    ranges = {
        'PAA':  (0.001, 5.0),
        'GT':   (0.001, 10.0),
        'P407': (0.01, 20.0),
        'Hz':   (10.0, 40.0),
        'Time': (1.0, 100.0)
    }
    mid = {k: (v[0] + v[1]) / 2 for k, v in ranges.items()}

    # --- 6) Feasibility thresholds ---
    ps_min,   ps_max   =  80, 400
    pdi_min,  pdi_max  = 0.1, 0.4
    zp_min,   zp_max   = -60, -15
    dl_min,   dl_max   =  60, 100

    # --- 7) Plot overlays for each 2D slice with labeled borders ---
    for var1, var2 in itertools.combinations(ranges.keys(), 2):
        v1 = np.linspace(*ranges[var1], 300)
        v2 = np.linspace(*ranges[var2], 300)
        G1, G2 = np.meshgrid(v1, v2)

        # build grid args
        args = {}
        for name in ranges:
            if name == var1:
                args[name] = G1
            elif name == var2:
                args[name] = G2
            else:
                args[name] = np.full(G1.shape, mid[name])

        # compute on grid
        PS_grid  = PS(**args)
        PDI_grid = PDI(**args)
```

```
149.         ZP_grid  = ZP(**args)
150.         DL_grid  = DL(**args)
151.
152.         # individual feasibility masks
153.         m_ps  = (PS_grid  >= ps_min)   & (PS_grid  <=
   ps_max)
154.         m_pdi = (PDI_grid >= pdi_min)  & (PDI_grid <=
   pdi_max)
155.         m_zp  = (ZP_grid  >= zp_min)   & (ZP_grid  <=
   zp_max)
156.         m_dl  = (DL_grid  >= dl_min)   & (DL_grid  <=
   dl_max)
157.
158.         # intersection of all four
159.         m_all = m_ps & m_pdi & m_zp & m_dl
160.
161.         # subtract intersection so it remains blank
162.         m_ps_plot  = m_ps  & ~m_all
163.         m_pdi_plot = m_pdi & ~m_all
164.         m_zp_plot  = m_zp  & ~m_all
165.         m_dl_plot  = m_dl  & ~m_all
166.
167.         plt.figure(figsize=(6,4))
168.         # overlay each region
169.         plt.contourf(G1, G2,
   m_ps_plot.astype(int),  levels=[0.5,1.5],
   colors=['red'],    alpha=0.3)
170.         plt.contourf(G1, G2, m_pdi_plot.astype(int),
   levels=[0.5,1.5], colors=['green'],  alpha=0.3)
171.         plt.contourf(G1, G2,
   m_zp_plot.astype(int),  levels=[0.5,1.5], colors=['yellow'],
   alpha=0.3)
172.         plt.contourf(G1, G2,
   m_dl_plot.astype(int),  levels=[0.5,1.5],
   colors=['blue'],   alpha=0.3)
173.
174.         # draw and label borderline contours
175.         cs_ps  = plt.contour(G1, G2,
   PS_grid,  levels=[ps_min,
   ps_max],   colors=['red'],    linestyles=['--','-'])
176.         cs_pdi = plt.contour(G1, G2, PDI_grid,
   levels=[pdi_min, pdi_max], colors=['green'],  linestyles=['-
   -','-'])
177.         cs_zp  = plt.contour(G1, G2,
   ZP_grid,  levels=[zp_min, zp_max],   colors=['yellow'],
   linestyles=['--','-'])
```

```
178.         cs_dl  = plt.contour(G1, G2,
   DL_grid,  levels=[dl_min,
   dl_max],   colors=['blue'],   linestyles=['--','-'])
179.
180.         fmt_ps  = {ps_min: 'PS = 80',   ps_max: 'PS =
   400'}
181.         fmt_pdi = {pdi_min: 'PDI = 0.10', pdi_max: 'PDI =
   0.40'}
182.         fmt_zp  = {zp_min: 'ZP = -60',  zp_max: 'ZP = -
   15'}
183.         fmt_dl  = {dl_min: 'DL = 60',   dl_max: 'DL =
   100'}
184.
185.         plt.clabel(cs_ps,  fmt=fmt_ps,  inline=True,
   fontsize=8)
186.         plt.clabel(cs_pdi, fmt=fmt_pdi, inline=True,
   fontsize=8)
187.         plt.clabel(cs_zp,  fmt=fmt_zp,  inline=True,
   fontsize=8)
188.         plt.clabel(cs_dl,  fmt=fmt_dl,  inline=True,
   fontsize=8)
189.
190.         plt.xlabel(var1)
191.         plt.ylabel(var2)
192.         plt.title(f'{var1} vs {var2}:\nRed=PS, Green=PDI,
   Yellow=ZP, Blue=DL\nDesign Space Overlay Plot (Intersection
   Feasible Region is White)')
193.         plt.tight_layout()
194.         plt.show()
195.
```

2. Python Code GA-RL

```
# Install DEAP if not already available
!pip install deap

# Mount Google Drive
from google.colab import drive
drive.mount('/content/drive')

import numpy as np
import math
import joblib
from deap import base, creator, tools, algorithms
import random
from collections import defaultdict
import matplotlib.pyplot as plt

# Load pre-trained kNN models
pdi_model_path = '/content/drive/MyDrive/Master AI IoT/Colab
Notebooks/IS/ModelCode/weighted_knn_PDImodel.pkl'
zp_model_path = '/content/drive/MyDrive/Master AI IoT/Colab
Notebooks/IS/ModelCode/weighted_knn_ZPmodel.pkl'
dl_model_path = '/content/drive/MyDrive/Master AI IoT/Colab
Notebooks/IS/ModelCode/weighted_knn_DLmodel.pkl'

# === Define Weighted kNN Regression class ===
class WeightedKNNRegression:
    def __init__(self, k=5):
        self.k = k
        self.data = None
        self.targets = None

    def fit(self, X, y):
        self.data = X
        self.targets = y

    def predict(self, X):
        preds = []
        for x in X:
            dists = np.linalg.norm(self.data - x, axis=1)
            idx = np.argsort(dists)[:self.k]
            weights = 1 / (dists[idx] + 1e-9)
            pred = np.dot(weights, self.targets[idx]) /
np.sum(weights)
            preds.append(pred)
        return np.array(preds)
```

```
# === Load models from Google Drive ===
with open(pdi_model_path, 'rb') as f:
    pdi_knn_model = joblib.load(f)
with open(zp_model_path, 'rb') as g:
    zp_knn_model = joblib.load(g)
with open(dl_model_path, 'rb') as h:
    dl_knn_model = joblib.load(h)

def sigmoid(x):
    return 1 / (1 + np.exp(-x))

# Denormalization helper
# X_norm assumed in [-1,1]
def denormalize(x_norm, x_min, x_max):
    return ((x_norm + 1) / 2) * (x_max - x_min) + x_min

def PS(PAA, GT, P407, Hz, Time):
    ps_lr = 465.692*PAA - 142.043*GT - 2.537*Time + 121.723
    # raw ANN output in [-1,1]
    ps_ann_norm = (
        -1.396 * sigmoid(-1.508*PAA + 0.715*GT - 0.480*P407 +
1.750*Hz - 0.110*Time - 0.658) -
        1.291 * sigmoid(-1.564*PAA + 1.086*GT + 0.094*P407 +
0.836*Hz + 0.817*Time - 1.311) +
        2.324 * sigmoid(2.366*PAA - 1.425*GT + 2.447*P407 +
0.173*Hz - 4.133*Time - 1.943) -
        2.308 * sigmoid(0.822*PAA - 1.416*GT + 1.872*P407 -
1.739*Hz - 2.628*Time + 0.184) +
        1.158
    )

    # bring it back to the original PS scale (min=42.78, max=400)
    ps_ann = denormalize(ps_ann_norm, 42.78, 400)

    return (ps_lr + ps_ann) / 2

def PDI(PAA, GT, P407, Hz, Time):
    # raw ANN output in [-1,1]
    pdi_ann_norm = (
        -0.878 * sigmoid(-0.089*PAA + 0.101*GT - 0.464*P407 -
0.584*Hz - 0.823*Time - 0.552) +
         1.646 * sigmoid(0.155*PAA + 0.324*GT + 0.801*P407 -
1.891*Hz + 0.497*Time - 1.342) -
         0.814 * sigmoid(-0.033*PAA - 0.015*GT - 0.139*P407 -
0.086*Hz - 0.771*Time - 0.107) +
```

```
        2.204 * sigmoid(1.475*PAA + 0.122*GT - 2.712*P407 +
0.998*Hz - 3.068*Time + 1.287) -
        0.895
    )

    # denormalize (min=0.206, max=0.586)
    pdi_ann = denormalize(pdi_ann_norm, 0.206, 0.586)

    knn_pred = pdi_knn_model.predict([[PAA, GT, P407, Hz,
Time]])[0]
    return (pdi_ann + knn_pred) / 2

def ZP(PAA, GT, P407, Hz, Time):
    # raw ANN output in [-1,1]
    zp_ann_norm = (
        1.279 * sigmoid(-2.539*PAA - 0.152*GT + 0.937*P407 -
0.041*Hz - 0.443*Time - 1.346) +
        2.197 * sigmoid(-0.376*PAA + 3.147*GT - 1.065*P407 +
0.147*Hz + 0.166*Time + 2.936) -
        0.348 * sigmoid(-0.344*PAA + 0.424*GT + 0.101*P407 -
0.021*Hz - 0.179*Time - 0.519) -
        0.467 * sigmoid(-0.414*PAA + 0.560*GT + 0.125*P407 -
0.113*Hz - 0.095*Time - 0.458) -
        1.408
    )

    # denormalize (min = -42.5, max = -15.3)
    zp_ann = denormalize(zp_ann_norm, -42.5, -15.3)

    knn_pred = zp_knn_model.predict([[PAA, GT, P407, Hz,
Time]])[0]
    return (zp_ann + knn_pred) / 2

def DL(PAA, GT, P407, Hz, Time):
    dl_lr = 9.775*PAA - 0.826*P407 - 0.129*Hz - 0.523*Time +
73.046
    knn_pred = dl_knn_model.predict([[PAA, GT, P407, Hz,
Time]])[0]
    return (dl_lr + knn_pred) / 2

# Normalization helper
def normalize(value, min_val, max_val):
    return (value - min_val) / (max_val - min_val)

# Objective function with normalization
def objective(individual):
```

```
    PAA, GT, P407, Hz, Time = individual
    ps = PS(PAA, GT, P407, Hz, Time)
    pdi = PDI(PAA, GT, P407, Hz, Time)
    zp = ZP(PAA, GT, P407, Hz, Time)
    dl = DL(PAA, GT, P407, Hz, Time)

#### Add Constraints Input vaules ######

    # Constraints
    if not (80 <= ps <= 400 and 0.1 <= pdi <= 0.4 and -60 < zp <
-15 and 60 <= dl <= 100):
        return -1e6,
    if not (0.001 <= PAA <= 1 and 0.001 <= GT <= 1 and 0.01 <
P407 < 10 and 10 <= Hz <= 40 and 1 < Time < 30):
        return -1e6,

    # Normalize
    ps_norm = normalize(ps, 80, 400)
    pdi_norm = normalize(pdi, 0.1, 0.4)
    zp_norm = normalize(zp, -60, -15)
    dl_norm = normalize(dl, 60, 100)

    # Minimize ps, pdi, zp magnitude; maximize dl
    obj = - ps_norm - pdi_norm - zp_norm + dl_norm
    return obj,

# DEAP setup
creator.create("FitnessMax", base.Fitness, weights=(1.0,))
creator.create("Individual", list, fitness=creator.FitnessMax)

toolbox = base.Toolbox()
toolbox.register("attr_PAA", random.uniform, 0.1, 0.5)
toolbox.register("attr_GT", random.uniform, 0.1, 0.5)
toolbox.register("attr_P407", random.uniform, 1, 5)
toolbox.register("attr_Hz", random.uniform, 10, 30)
toolbox.register("attr_Time", random.uniform, 5, 20)

toolbox.register("individual", tools.initCycle,
creator.Individual,
                 (toolbox.attr_PAA, toolbox.attr_GT,
toolbox.attr_P407,
                  toolbox.attr_Hz, toolbox.attr_Time), n=1)
toolbox.register("population", tools.initRepeat, list,
toolbox.individual)

toolbox.register("evaluate", objective)
```

```
toolbox.register("mate", tools.cxBlend, alpha=0.5)
toolbox.register("mutate", tools.mutGaussian, mu=0, sigma=0.1,
indpb=0.2)
toolbox.register("select", tools.selTournament, tournsize=3)

# Q-learning setup
cxpb_values = np.round(np.linspace(0.1, 0.5, 100), 2)
mutpb_values = np.round(np.linspace(0.1, 0.5, 100), 2)
Q = defaultdict(float)
alpha = 0.1
gamma = 0.9
epsilon = 0.2
fitness_history = []

def select_action():
    if random.random() < epsilon:
        return random.choice([(cx, mu) for cx in cxpb_values for
mu in mutpb_values])
    else:
        return max([(cx, mu) for cx in cxpb_values for mu in
mutpb_values], key=lambda x: Q[x])

def run_ga_with_params(cxpb, mutpb, n_gen=100):
    population = toolbox.population(n=100000)
    hof = tools.HallOfFame(1)
    stats = tools.Statistics(lambda ind: ind.fitness.values)
    stats.register("avg", np.mean)
    stats.register("max", np.max)
    stats.register("min", np.min)

    population, logbook = algorithms.eaSimple(population,
toolbox,
                                              cxpb=cxpb, mutpb=mutpb,
                                              ngen=n_gen,
stats=stats,
                                              halloffame=hof,
verbose=False)

    best = hof[0]
    fitness = objective(best)[0]
    fitness_history.append([gen['max'] for gen in logbook])

    print("\nBest Solution:")
    print(f"PAA={best[0]:.4f}, GT={best[1]:.4f},
P407={best[2]:.4f}, Hz={best[3]:.4f}, Time={best[4]:.4f}")
    print(f"Objective Value: {fitness:.4f}")
```

```
    ps = PS(*best)
    pdi = PDI(*best)
    zp = ZP(*best)
    dl = DL(*best)

    print("\nObjective Function Variables:")
    print(f"PS: {ps:.4f}")
    print(f"PDI: {pdi:.4f}")
    print(f"ZP: {zp:.4f}")
    print(f"DL: {dl:.4f}")

    return fitness, best

def train_rl_on_ga(episodes=10):
    for episode in range(episodes):
        action = select_action()
        cxpb, mutpb = action
        reward, _ = run_ga_with_params(cxpb, mutpb)
        Q[action] += alpha * (reward - Q[action])
        print(f"Episode {episode + 1}, cxpb: {cxpb}, mutpb:
{mutpb}, reward: {reward:.4f}")

    best_params = max(Q.items(), key=lambda x: x[1])[0]
    print(f"\nBest hyperparameters after RL: cxpb =
{best_params[0]}, mutpb = {best_params[1]}")
    return best_params

# Run Q-learning GA
best_cxpb, best_mutpb = train_rl_on_ga(episodes=10)
print("\n=== Final Run with Best Hyperparameters ===")
# run_ga_with_params(best_cxpb, best_mutpb, n_gen=100)

# Plot
plt.figure(figsize=(10, 6))
for i, run in enumerate(fitness_history):
    plt.plot(run, label=f"Episode {i+1}")
plt.xlabel("Generation")
plt.ylabel("Max Fitness")
plt.title("Fitness Over Generations with RL-Tuned GA")
plt.legend()
plt.grid(True)
plt.show()
```

BIOGRAPHY

Name	Phuvamin Suriyaamporn
Education	2018: Bachelor degree, Doctor of Pharmacy, Faculty of Pharmaceutical Sciences, Ubon Ratchathani University (Thailand)
	2023: Doctor of Philosophy (Pharmaceutical Technology, International Program), Faculty of Pharmacy, Silpakorn University (Thailand)
	2025: Master of science (Digital business transformation Major: data science), College of innovation, Thammasat University (Thailand)