Advanced Drug Delivery Reviews 175 (2021) 113806

Contents lists available at ScienceDirect

Advanced Drug Delivery Reviews

journal homepage: www.elsevier.com/locate/adr

Machine learning directed drug formulation development ☆

Pauric Bannigan [a], Matteo Aldeghi [b,c,d], Zeqing Bao [a], Florian Häse [b,c,d], Alán Aspuru-Guzik [b,c,d,e,*], Christine Allen [a,*]

[a] Leslie Dan Faculty of Pharmacy, University of Toronto, Toronto, ON M5S 3M2, Canada
[b] Chemical Physics Theory Group, Department of Chemistry, University of Toronto, Toronto, ON M5S 3H6, Canada
[c] Department of Computer Science, University of Toronto, Toronto, ON M5S 3H6, Canada
[d] Vector Institute for Artificial Intelligence, Toronto, ON M5S 1M1, Canada
[e] Lebovic Fellow, Canadian Institute for Advanced Research, Toronto, ON M5S 1M1, Canada

ARTICLE INFO

Article history:
Received 13 January 2021
Revised 31 March 2021
Accepted 14 May 2021
Available online 19 May 2021

Keywords:
Machine learning
Deep learning
Drug delivery
Drug development

ABSTRACT

Machine learning (ML) has enabled ground-breaking advances in the healthcare and pharmaceutical sectors, from improvements in cancer diagnosis, to the identification of novel drugs and drug targets as well as protein structure prediction. Drug formulation is an essential stage in the discovery and development of new medicines. Through the design of drug formulations, pharmaceutical scientists can engineer important properties of new medicines, such as improved bioavailability and targeted delivery. The traditional approach to drug formulation development relies on iterative trial-and-error, requiring a large number of resource-intensive and time-consuming *in vitro* and *in vivo* experiments. This review introduces the basic concepts of ML-directed workflows and discusses how these tools can be used to aid in the development of various types of drug formulations. ML-directed drug formulation development offers unparalleled opportunities to fast-track development efforts, uncover new materials, innovative formulations, and generate new knowledge in drug formulation science. The review also highlights the latest artificial intelligence (AI) technologies, such as generative models, Bayesian deep learning, reinforcement learning, and self-driving laboratories, which have been gaining momentum in drug discovery and chemistry and have potential in drug formulation development.

1. Introduction

Drug formulation typically involves combining inert materials and excipients with active pharmaceutical ingredients (APIs) to produce viable drug products with desired properties. The improvements associated with the development of an optimized drug formulation can include enhanced efficacy, longer acting therapeutic effects, reduced side effects, extended API stability and shelf-life, as well as better patient compliance [1]. Depending on the desired route of administration and specific requirements for the indication, APIs can be formulated using a diverse set of materials (including, inert excipients, such as polymers, lipids and surfactants, as well as other APIs) and in a wide range of delivery systems including various types of microparticles (MPs), nanoparticles (NPs), and multicomponent systems [2–5]. These delivery systems are usually further manufactured into final drug products, e.g., solid, liquid, or parenteral dosage forms [1]. Bringing effective medicines to the market in a timely manner requires innovative drug delivery systems and an economically efficient development process. Although the traditional approach to formulation development has delivered successful drug products to patients, it relies on several inherently time-consuming and often inefficient steps. In general, the current formulation development pathway involves preparing and characterizing several candidate formulations through an extensive API-material matching process. The

Abbreviations: APIs, Active pharmaceutical ingredients; T_{agg}, Aggregation onset temperature; ASP, Antisolvent precipitation; AI, Artificial intelligence; DL, Deep learning; k_D, Diffusion interaction parameter; GI, Gastrointestinal; GPs, Gaussian processes; HPH, High-pressure homogenization; IR, Immediate release; IND, Indomethacin; LightGBM, Light gradient boosting machine; ML, Machine learning; MAE, Mean absolute error; T_m, Melting temperature; MPs, Microparticles; mAbs, Monoclonal antibodies; NPs, Nanoparticles; NN, Neural network; PDI, Polydispersity index; PLGA, poly(lactic-co-glycolic acid); RF, Random forests; RMSE, Root-mean-square error; siRNA, Small interfering ribonucleic acid; FDA, United States Food and Drug Administration; WBM, Wet ball milling.

☆ This review is part of the Advanced Drug Delivery Reviews theme issue on "Editor's collection 2021".

* Corresponding authors at: Chemical Physics Theory Group, Department of Chemistry, University of Toronto, Toronto, ON M5S 3H6, Canada (A. Aspuru-Guzik); Leslie Dan Faculty of Pharmacy, University of Toronto, Toronto, ON M5S 3M2, Canada (C. Allen).

E-mail addresses: alan@aspuru.com (A. Aspuru-Guzik), cj.allen@utoronto.ca (C. Allen).

https://doi.org/10.1016/j.addr.2021.05.016
0169-409X/

performance of these potential formulations is then evaluated and compared to the API alone, and competitor formulations (should they exist) in a variety of *in vitro* and/or *ex vivo* assays. Several predetermined product requirements, such as API loading, API release rate, formulation stability, particle size and/or particle charge are then used to select lead-candidate formulation(s) to be carried forward into preclinical animal studies. Failure of a potential formulation to meet the desired criteria (e.g., release rate, particle size, etc.) at any stage of development can require its refinement and the need to repeat several steps in the development process or in some cases the abandonment of a formulation and the need to begin the process anew.

The setbacks encountered during formulation development are largely related to an inability to predict how the composition, or the combination, of APIs and materials influences the performance-related parameters of the formulation. In an attempt to bridge this knowledge gap, pharmaceutical scientists have adopted computational modelling approaches such as molecular dynamics simulations [6], molecular docking studies [7], and cheminformatics tools [8]. Molecular modeling tools can provide new insights into complex drug delivery systems at the molecular level that are not always accessible by experimental techniques. A prominent example is the progress made in predicting properties such as small molecule solubility and affinity using molecular dynamics simulations [9–11]. While an in-depth discussion of these techniques, and their applications, is beyond the scope of this review, their application in drug formulation development has seen increasing success in recent years and these advances are well summarized in a recent review by Casalini [12]. Yet, these physics-based simulations have limitations that hinder their application in formulation development. In fact, the prediction of properties such as API release involves the simulation of large, multicomponent drug delivery systems over long timescales, such that the use of approaches like atomistic molecular dynamics simulations would be computationally intractable. The timescales involved, sometimes days long, are out of reach for coarse-grained approaches as well [13].

Machine learning (ML) is a branch of artificial intelligence (AI) that aims to model processes by training computational models based on a body of data. For instance, ML might allow one to predict the stability of a specific drug formulation by considering data from many previous experiments that examined formulation stability. Recent advances in ML algorithms, the wide availability of faster computing hardware, as well as the release of user-friendly ML toolkits, have significantly improved accessibility to powerful ML models. These trends have led to an explosion in the real-world applications of ML and AI, including within the healthcare and pharmaceutical sectors. The application of ML in these sectors has led to improved cancer diagnostics [14–16], the discovery of new antifibrotic [17] and antibiotic [18] molecules, and the development of so-called self-driving laboratories [19,20]. Other notable applications include using supervised learning algorithms to predict the products of chemical reactions [21], the use of deep reinforcement learning to optimize chemical reactions [22], as well as the use of deep learning (DL) to determine the three-dimensional structure of a protein from its amino acid sequence [23].

The invention of new drug products and the steps involved in optimizing such formulations poses similar challenges to those which have already been addressed using ML in other sectors. For example, a major barrier in the current drug formulation development process is the number of expensive, laborious and time-consuming experiments that must be conducted to select appropriate materials to achieve a desirable formulation property (such as increased API solubility). By harnessing the predictive power of AI and ML, pharmaceutical scientists may be able to streamline the development of such formulations using existing data or through optimal experimental planning. To date, ML models have been developed to address several of the inherent challenges faced by formulation scientists, including prediction of the effect of excipients on API solubility, determination of the chemical and colloidal stability of proteins, prediction of the physical stability of API formulations, determination of API loading capacity as well as release rates of APIs from advanced delivery platforms such as MPs and NPs. The purpose of this review is to provide drug delivery and formulation scientists with a brief introduction to ML and to highlight recent formulation development projects that have overcome substantial obstacles using ML tools. Efforts are also made to highlight promising directions to achieve further success in this area. Overall, this review aims to make the case for a new data-driven formulation development process.

2. Machine learning tools and techniques

In this review, we focus primarily on the use of *supervised* ML to predict properties of drug formulations. Supervised ML tasks aim to predict a numerical value or class for a specific data sample. The prediction of numerical values, for example the prediction of API solubility in various surfactant solutions (Box 1), is referred to as a *regression* task. A *classification* task, in contrast, determines a category to which a sample belongs, for example predicting whether a molecule primarily acts as an antibiotic, a chemotherapeutic agent, or an antidepressant. ML algorithms differ from the aforementioned computational models (i.e., molecular dynamic simulations, molecular docking studies, and cheminformatics) in several ways. Most notably, ML models require training data from which they can infer task-relevant information to generate predictions. For supervised prediction tasks, this training data includes input features as well as examples of the desired predictions associated with those specific inputs. For example, a hypothetical model that predicts API solubility in surfactant solutions may rely on inputs such as the physico-chemical properties of the API and the solvent (e.g., logP, melting point, and boiling point). This model might then return a value for API solubility in units of concentration (e.g., mg/mL). During training, parameters of a ML model are modified to mathematically approximate the physical relation between inputs and outputs through optimization algorithms [23]. The fact that approximative ML models can be constructed exclusively from collected data is particularly useful for scientific problems where physical models are unavailable, are computationally intractable, or where the relationships between experimental variables and outcomes are unknown. Thus, training ML models on datasets of successful and unsuccessful formulations might reveal materials which are best suited to achieve a desirable formulation property for an API, such as improved water solubility, sustained API release, or improved long-term formulation stability.

Box 1 Example of a data-driven workflow in drug formulation development.

Predicting the solubility of an API in various surfactant solutions.

Data preparation

We assemble a dataset with different APIs and their associated solubilities in surfactant solutions. We decide how to describe each API and surfactant numerically, (e.g., via physico-chemical properties). For instance, we may consider the water solubility or logP of the API. We also decide which data points will be used to train the model, and which to test it (e.g., 20% of the data points are randomly set aside for testing).

Model selection and evaluation

We are trying to predict a numerical value (solubility); hence we select a supervised regression ML algorithm. We collected 2,000 previous experiments from the literature and want to explore the use of *random forest* (RF) and *neural network* (NN) models. We train both models on 1,600 experiments, and then assess whether they can accurately predict the solubility of the remaining 400. We obtain the root mean square error (RMSE) between the predicted and actual solubility values as a measure of model performance. The RF and NN models give us RMSE values of 0.2 mg/mL and 0.05 mg/mL, respectively. Thus, we conclude that the NN model is more accurate and proceed to use this model to predict solubility. If the RMSE achieved was not satisfactory, we would consider collecting more data, using other ML algorithms, or using different input features.

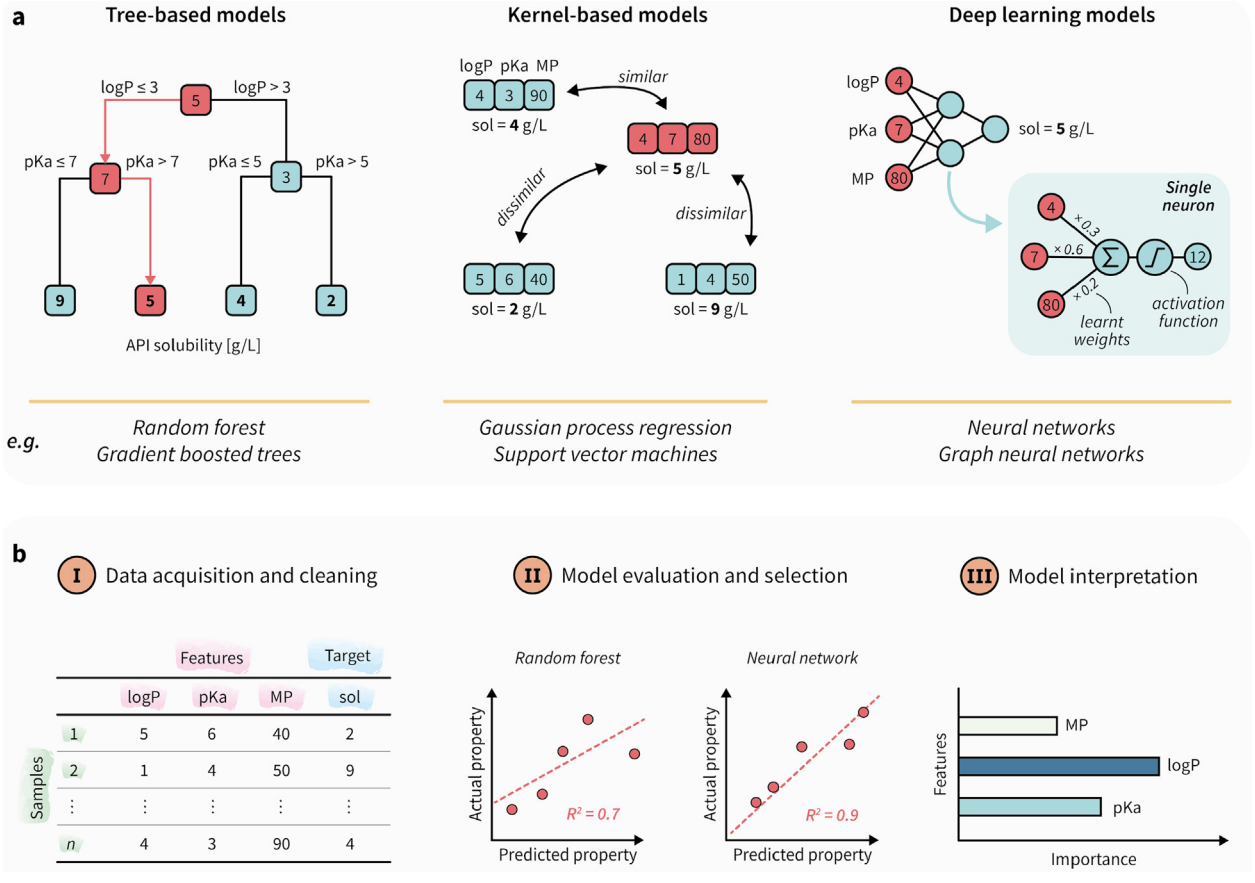
Model interpretation

We now have a reliable NN model, which we can interrogate to understand which features result in higher API solubility. For instance, after we predict solubility in many different surfactant solutions, we may find that Tween 80 is generally the surfactant that most improves solubility. In addition, we can study the importance of each feature used to describe the system, and we may find that logP of the API and surfactant concentration are the two features that are major determinants of solubility.

In the following section, we first introduce the most prevalent supervised learning algorithms, and then provide an overview of the steps typically involved in a supervised ML workflow: (i) data acquisition and cleaning, (ii) model evaluation and selection, and (iii) interpretation, as shown in Scheme 1. We will briefly discuss each of these steps.

2.1. Overview of ML algorithms

ML models differ in their assumptions about the data, the process generating it, and the complexity of the relationships between features and targets. Generally, constructing a ML model requires a balance of predictive capacities, interpretability and robustness.



Scheme 1. Overview of supervised ML algorithm classes and outline of a typical data-driven model-building pipeline. (a) Schematic diagrams show the core elements of each algorithm class. We consider a simple numerical example in which we try predicting API solubility (sol) based on the compound's octanol-water partition coefficient (logP), acid dissociation constant (pKa), and melting point (MP). For tree-based algorithms, a small decision tree is shown. For kernel-based algorithms, a qualitative comparison of data points to be predicted and those in the training set is shown. In reality, these models would learn a quantitative measure of similarity. For deep-learning algorithms, we show the structure of a single neuron in more detail. (b) Typical data science pipeline for building predictive ML models. First, data needs to be collected from the literature or public and/or private databases. Second, multiple supervised ML models are trained and evaluated to identify the best performing option. If the desired accuracy is reached, any chosen model can then be deployed for prospective predictions. Finally, the ML model can be interpreted to extract scientific insights about the problem being addressed.

Some models can reproduce more complex relationships but tend to be harder to interpret, while others are easier to interpret but can be too simplistic to resolve all variations in the underlying physical relation. A large number of ML algorithms are now readily available via well-established libraries and software packages. Some notable examples include *scikit-learn* [24], *Keras* [25], *PyTorch* [26], and *TensorFlow* [27]. While a detailed description of supervised ML models would be beyond the scope of this review, we provide a high-level description of some of the most commonly used models in the studies summarized herein.

The simplest ML model is perhaps linear regression, which assumes that the properties to be predicted, such as API solubility, linearly depend on input features. Given its simplicity, this model is straightforward to interpret, yet it also implies severe limitations in its modeling capacities [28]. Non-linear models generally achieve better performance. The most commonly used non-linear ML models may be classified into tree-based, kernel-based, and DL methods, while both hybrid and more exotic formulations of ML models have been introduced.

Decision trees are the foundation of all tree-based models and rely on splitting the feature space into separate partitions. Feature-target relationships are derived from linear approximations (or "branches") which connect the various partitions. The flexibility of the model arises from a training process that determines optimal partitions from a provided dataset. One of the advantages of decision trees is their interpretability, as all decisions that lead to a specific prediction can be visualized (Scheme 1**a**). Well-known examples of tree-based models include random forests (RFs) [29] and boosted trees [30]. The former makes predictions using multiple independent trees trained on subsamples of the training dataset, while the latter trains multiple trees in a sequence to correct for the variance not explained by the preceding trees.

Kernel methods rely on the comparison of new samples to those instances present in the training dataset to make predictions (Scheme 1**a**). Among the most popular kernel methods are Gaussian processes (GPs), which can provide well-calibrated estimates of uncertainty on their predictions [31]. A downside of GPs is that the computational cost for training grows cubically with the size of the dataset. With this limitation, GPs are primarily applied to small datasets (e.g., <1000 samples). Sparse kernel methods mitigate this adverse computational scaling, with the most well-known example being support vector machines [32]. Many more kernel methods have been developed, including kernel ridge regression, relevance vector machines, and radial basis function networks [28].

In the last decade, DL models have gained increasing popularity [33]. These models rely on the composition of simple non-linear functions, referred to as *artificial neurons*. A neuron implements a linear regressor whose output is rescaled by a selection of non-linear functions (*activation functions*). DL models combine and transform input features through multiple stacked *layers* of neurons (Scheme 1**a**). The most well-known example of such models is the feedforward *neural network* (NN). One of the advantages of DL models is that their computational cost increases slowly as larger datasets are used. For this reason, and thanks to their established predictive power, they are often employed for supervised tasks where large amounts of data (e.g., millions of samples) are available. Due to the flexibility with which layers and nodes of the networks can be assembled, many different DL architectures are now available (e.g., convolutional NN, recurrent NN) [33], each with specific advantages for certain prediction tasks. For example, in chemistry, graph NNs have been increasingly employed for molecular property prediction [34–36].

2.2. Data acquisition and cleaning

The collection, organization and cleaning of a dataset is usually the first step when constructing predictive ML models (Scheme 1**b**). For supervised tasks, datasets should contain features describing the system to be studied (e.g., a specific drug formulation) and the properties of interest that emerge from these features (e.g., API solubility). Common molecular input features can be derived from computer simulations or experimental measurements and could include physico-chemical properties, such as molecular weight, logP, polar surface area, melting point or aggregation temperature. The process of identifying and constructing relevant features (i.e., the most relevant characteristics of the drug formulation that determines API solubility) is referred to as *feature engineering* and can greatly enhance the predictive capabilities of a ML model. Yet, while the inclusion of additional features has the potential to provide more relevant information, extending the set of features also increases the chance of obstructing a model with spurious correlations.

Generally, more data is better when utilizing ML algorithms. However, data collection in drug formulation development is slow and expensive compared to other scientific disciplines. Certain *in vitro* drug release experiments can take upwards of three months to complete, while extensive *in vivo* testing is expensive and restricted by ethical concerns. A recent trend across chemistry and biology has been data mining from the published literature [37,38]. Published scientific studies are a readily available source of data, and there are now many publications that have extracted formulation data from previous publications and used it to train ML models. Nonetheless, drug formulation datasets extracted from the literature rarely exceed one thousand samples and are usually only a few hundred in size. Hence, limited data availability is a major challenge in ML for drug formulation development. In addition, it is widely accepted that there is a reproducibility crisis in the life sciences, and many scientific studies are often difficult or impossible to replicate or reproduce. Automation and high-throughput experimentation might allow us to increase the rate at which new data is generated. In materials science and chemistry, automation has not only allowed for an increase in experimental throughput, but it has also been shown to improve reproducibility and data quality [19,20,39].

Similarly, advances in digitalization [40] will ensure better data integrity, avoiding the loss of information or consistency that can happen when information on experimental conditions and results are manually recorded. In addition, better data-sharing tools within the drug formulation community would enable the frictionless distribution of drug formulation results. Other disciplines are also moving in this direction to facilitate the application of ML. The organic synthesis community is creating a chemical reaction database (i.e., OpenReactionDatabase.org) to enable data-driven reaction prediction and organic synthesis planning, and the drug design community has for years been collecting ligand-protein structures and affinities into online databases [41,42] to build affinity prediction models.

Feature selection and engineering are also challenging in drug formulation development. The more informative the features chosen are to the predicted property, the more accurate and reliable the ML model will be. Nevertheless, in drug formulation, it is challenging to know a *priori* which features are most useful. Properties of single molecules (e.g., partition coefficients) depend on these molecules alone, and it is relatively straightforward to select features (e.g., polar surface area) that are relevant to the predictive task. However, the *in vitro* and *in vivo* performance of formulations depends on complex interactions between the API, excipients, and the host organism. Thus, it is not immediately clear which features

are most informative to predict the performance of complex drug delivery systems. In recent years, the ML community has favored a representation learning [43] approach in which ML models are provided with enough raw data to learn the best features independently. For example, image recognition is achieved by providing raw pixel values to the ML model for a large dataset of images [44]. Similarly, in chemistry, this can be done by providing a large dataset of molecules represented as graph objects [34]. However, the lack of accessible and large formulation datasets hinders the successful application of such approaches. In the future, larger bodies of experimental data, combined with more experience in applying data-driven approaches to drug formulation challenges, are likely to inform better selection of relevant features for property prediction in this field.

2.3. Model evaluation and selection

Typical data science pipelines split the dataset into a *training* and a *test* set. The training set is used to train the ML model, and the test set for its evaluation. In fact, once a ML model has been trained, it is important to estimate its future accuracy (Scheme 1**b**). Here, the test set plays a pivotal role. Since the model is not trained using this data, the test set provides an objective estimate of the prospective performance of the model. It is important to split the dataset such that the test set is representative of how the model will be deployed in the future. For instance, if the model is to be used to predict solubility for new APIs, the test set used should not contain APIs also available in the training set. The predictive accuracy of ML models is assessed on the test set using quantitative metrics of performance. For regression tasks, metrics that capture the prediction errors (e.g., mean absolute error, MAE; root-mean-square error, RMSE) or the correlation between predictions and targets (e.g., coefficient of determination, R^2; Pearson's linear correlation; Spearman's rank correlation) are typically used. For classification tasks, metrics that capture the discrepancy between predicted and known class labels are commonly employed, with various options depending on whether only two (e.g., binary accuracy, precision, recall) or more classes (e.g., cross entropy) are considered. Finally, it is important to note that a model trained to reproduce experimental data cannot achieve accuracies beyond the uncertainties inherent to the original dataset without overfitting.

It is difficult to discern which ML algorithm will provide the best predictive performance. As a result, it has become common practice to evaluate different models to identify the model that provides the best performance. However, the performance of many ML algorithms also depends on the choice of their hyperparameters. These are user-defined parameters (i.e., fixed before model training), and as such, are not inferred (i.e., optimized) by the model during training. Hyperparameters vary with the type of model being trained, and examples include the number of trees in RF regression; the choice of the kernel function in GP regression; and the number of layers, neurons per layer, and type of activation function in DL models. The chosen set of input features provided to the model may also be seen as hyperparameters. As these choices affect model performance, a natural question is how to choose the most suitable hyperparameters for a given ML algorithm. One possibility is to base the choice on one's ML expertise. However, similar to how a model's predictive performance is assessed empirically against a test set, hyperparameters are often tuned based on model performance on a validation set. This third dataset is often similar in size to the test set (e.g., a common approach uses a 60/20/20% dataset split for the training, validation, and test sets) and is used to tune hyperparameters before the model performance is assessed on the test set. It is essential not to use the test set for model optimization and selection to avoid overfitting the model to the details of a specific test set. As the number of possible hyperparameter settings may be significant and an intuition for the best settings may be lacking, there are now algorithms that enable automated hyperparameter optimization [45]. These methods may help accelerate hyperparameter tuning for less experienced users.

When the collected dataset is relatively small (e.g., only a few hundred samples), the composition of the training, validation, and test sets can have a significant impact on model performance. One model may perform better than another just by chance. *Cross-validation* is often used to mitigate this effect. This approach involves creating multiple splits of the dataset (e.g., typically 3, 5, or 10), and the model is trained and tested for each of these splits. In this way, multiple evaluations of the model's performance are available, and one can determine the average performance. Models trained in this way can be more robust against variations in test set composition. The only downside of cross-validation is the increased computational cost of training and testing a ML model multiple times. However, this is generally not an issue for algorithms that can be trained quickly, such as tree-based algorithms or linear regression, but can be time-consuming for DL models.

2.4. Model interpretation

Once a model has been trained it can be used to predict the properties of interest (e.g., API solubility) for new experimental samples (e.g., variations of a drug formulation). If a trained ML model has achieved the desired prediction accuracy on the test set, one can assume it has identified relevant statistical correlations. This model can then also be analyzed (Scheme 1**b**) to gain insight into which features the model considers most important for prediction (e.g., logP, polarity, and molecular weight). This type of analysis, called *feature importance*, can shed light on the main determinants of the property being studied [46].

The details of feature importance analysis depend on the ML algorithm used and aim to rank the features to indicate which the model deems most informative for training or making accurate predictions. For instance, tree-based models are built by iteratively selecting features that provide the highest reduction in training set error [29]. Hence, one can readily analyze during training which features provide the most significant improvements to the model. Similarly, in DL algorithms, one may analyze the weight that the model has learned to associate with each feature; the larger the weight, the more important it is to the task. Kernel-based methods can learn parameters that inform the model on how much variation in each feature is needed to affect the outcome of a prediction [47]. Features for which even a slight variation can greatly affect the model's prediction may be considered more consequential to the predicted property of interest. Model-agnostic feature importance techniques are also available. For instance, one may assign importance based on a decrease in predictive accuracy when the values of a specific feature are scrambled.

Feature importance analysis may be considered as part of the set of techniques referred to as explainable ML. These methods try to explain the decisions of a ML model via *post-hoc* analysis [48]. The field of explainable ML is currently expanding as complex "black-box" ML models such as DL have become increasingly adopted across disciplines. Consequently, more sophisticated analysis techniques that aim to gain insight from complex ML models are currently the subject of active research in the ML community. For instance, recent advances in ML for molecular property prediction have resulted in algorithms that can highlight the atoms or substructures in a molecule that are most relevant for a model's predictions [46,49].

While a ML model may provide accurate predictions, and while one might analyze a model to explain its decisions, it may still not be clear exactly how the algorithm makes specific predictions.

Complex models, like those created by DL algorithms, which may contain thousands or millions of interacting variables, are not interpretable by humans. This limited interpretability can hinder our ability to extract scientific understanding. For some applications, it may limit the trust in the algorithms. This issue is particularly relevant for healthcare and criminal justice applications [50,51], where high-stakes decisions require accountability and justification. As such, interpretability is a critical aspect of ML model choice that may need to be balanced against predictive accuracy. More straightforward, logic-based, or sparse ML models, like decision trees and linear regression, are intrinsically more interpretable than DL approaches. Once trained, a person can follow the algorithm's procedure and understand how a particular prediction is obtained.

Explainable and interpretable ML models hold the potential to provide accurate property predictions and scientific insight. For these reasons, we expect the interpretable and explainable ML fields to continue to expand in the future, leading to the development of ML algorithms that can go beyond property prediction and contribute to the creation of new knowledge in a number of scientific fields, including drug formulation development.

3. Applications of machine learning in formulation development

3.1. Conventional oral dosage forms

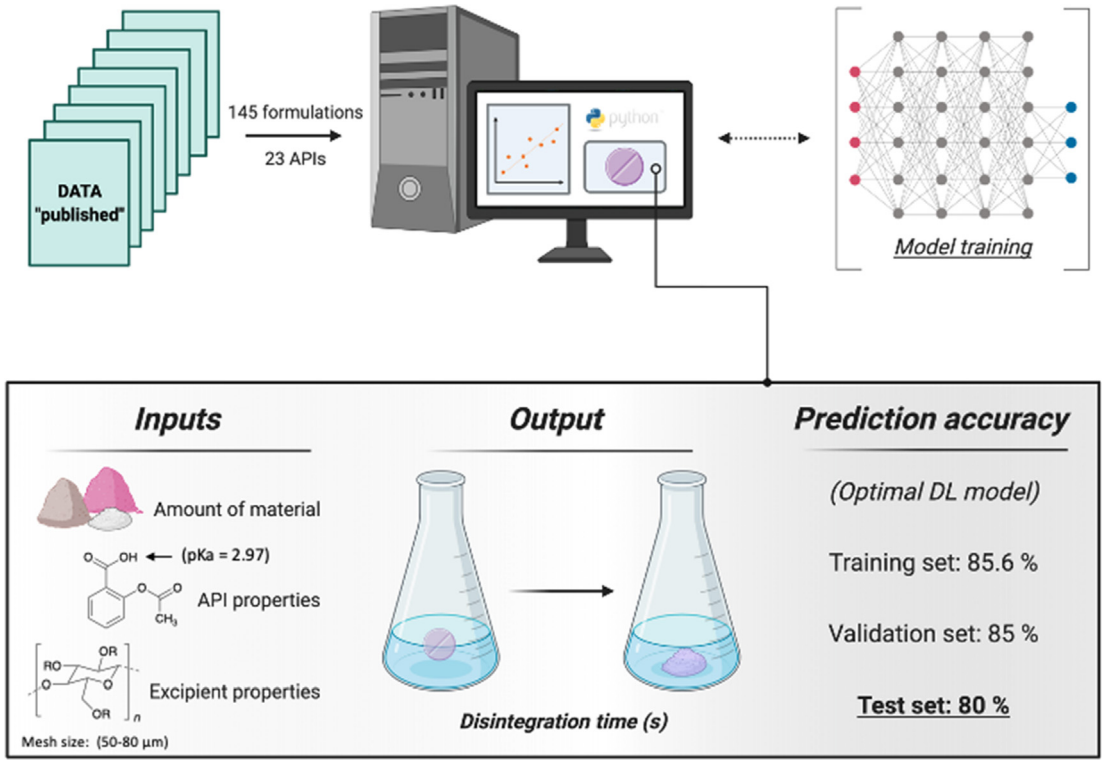
The first applications of ML in drug formulation development date back to the 1990s when NNs were used to predict properties of immediate release (IR) oral tablets. These studies involved the preparation and evaluation of a range of tablet formulations. The resulting data was then used to train NNs and/or decision trees to predict various outputs (e.g., disintegration time, dissolution rate, and friability). These studies were some of the first to demonstrate how ML could be used to predict drug formulation performance, however the overall results were quite varied [52–55]. Bourquin et al. used NNs to predict an array of tablet properties (i.e., tensile strength, friability, capping, disintegration time, and dissolution at various time points) for a direct compression tablet of a single drug and used relative w/w % of excipients as input features. The authors used random holdout to split the dataset (205 samples) into training (24 samples), testing (177 samples) and validation sets (4 samples). Given the small size of the dataset, they implemented *early stopping* to avoid overfitting. In early stopping the training of a DL model is interrupted when the validation error increases. The overall performance (R^2) of the model ranged from 0.41−0.75 [52]. In some of these early studies there are concerns regarding model overfitting, for example, Kesavan et al. [54] and Turkoglu et al. [55] trained NNs on 32 and 42 samples, respectively. Both authors attempted similar predictions to those described by Bourquin et al. (i.e., tablet properties and dissolution based on excipient composition for a single drug tablet), both employed manual holdout validation, reported very high model accuracies (i.e., in most cases $R^2 > 0.9$) and neither included any methodical effort to avoid overfitting.

Since the publication of these studies, many others have investigated how ML can be used to address a range of challenges associated with the development of different types of drug formulations. In recent years, as interest in the application of ML has become more mainstream, some researchers have revisited the idea of using ML to predict tablet performance properties using specific inputs. For example, Takagaki et al. trained a NN on five APIs in 500 different tablet formulations [56]. In contrast to the aforementioned studies [52–54], Takagaki et al. incorporated fundamental API properties (including solubility, mean particle size, and specific surface area) to predict tensile strength, disintegration, as well as the effects of accelerated stability conditions on the performance properties of the formulations. The authors also used various training strategies to minimize the overfitting of their models. These included *leave-some-out cross-validation* (four-fold) as well as early stopping. Overall, the performance of these models was assessed using the Pearson correlation coefficient (R), and there was a strong positive correlation between the predicted and experimental results (R = 0.95–0.99). In another example, Han et al., compiled data on 23 APIs in 145 different tablet formulations from published studies to predict the disintegration time of oral tablets using a range of NN structures. These models used molecular descriptors of the APIs and excipients as well as tablet process parameters to generate their predictions [57], Scheme 2. The authors split the dataset to ensure maximum dissimilarity between their training (70%), validation (15%), and test (15%) sets. In this study, model performance was assessed based on the number of accurate predictions, where accuracy was based on the predicted disintegration time being within 10 s of the experimental time.

3.2. Advanced oral delivery systems

In contrast to conventional oral dosage forms, advanced oral delivery systems consist of drug products formulated to overcome clinical limitations of an API (such as poor solubility or intestinal permeability), or to control the release rate of the API in the gastrointestinal (GI) tract (e.g., via sustained release matrix tablets). Oral drug products, including conventional and advanced delivery systems, remain the most popular in the pharmaceutical industry. For example, in 2019, almost 70% of new drug approvals by the United States Food and Drug Administration (US FDA) were formulations intended for oral administration [58]. Despite this demand, designing advanced oral formulations is not without its challenges. The greatest of these is often the need to overcome the poor solubility and/or permeability of APIs, properties that are strongly correlated with oral bioavailability [59,60]. In recent years, ML techniques have been developed to predict a diverse range of parameters for these systems, including estimations of the effects of excipients on API solubility [61]; prediction of API release rates from sustained release matrix tablets [62–67]; and determination of the long-term physical stability of amorphous solid dispersions [68]. In addition, ML models have been developed to aid in the design of push-pull osmotic pump tablets for hydrophobic drugs [69,70], microemulsion formulations [71], and drug-phospholipid/cyclodextrin complexes [72,73]. The outputs of these models (i.e., API solubility, API release rates, long-term physical stability, etc.) represent inherent challenges to the successful development of advanced oral delivery systems, and these studies demonstrate how ML can aid in the design of such systems.

To date, most of the ML models highlighted here have been trained on relatively small datasets (i.e., <200 samples) [61,62,64–67], and while there are some exceptions (i.e., Gao et al., 341 samples [72], Han et al., 646 samples [57], and Zhao et al., 3000 samples [73]), using such small datasets can increase the risk of overfitting during training, which, as mentioned in Section 2, can result in ML models that do not generalize well when presented with new data. However, appropriate data splitting strategies can highlight overfitting issues and help mitigate them, even in incidences where datasets are relatively small, resulting in more robust ML models. In some of these studies where small data sets were used for training [61,62] the authors took precautions to avoid model overfitting (i.e., cross-validation and diversity-based data splitting, as well as early stopping protocols). However, in other studies, an evaluation strategy based on random splits was employed [64–67]. In the latter case, very high model



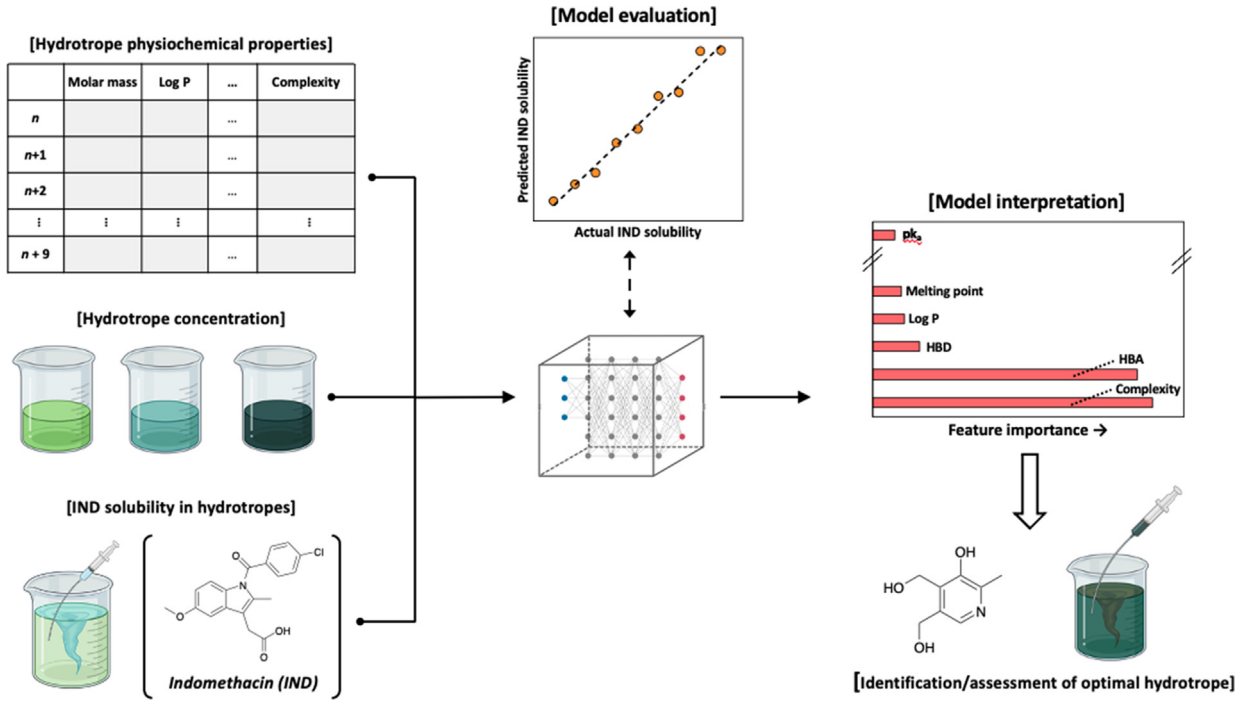
Scheme 2. Illustration of the study conducted by Han et al. In this study, the authors developed DL models to predict the disintegration time (in seconds) of IR oral tablets. To train these DL models, the authors used numerical values to describe the physico-chemical properties of APIs and excipients and the relative amount of each ingredient in the formulation. This study's best performing DL model predicted the disintegration time of formulations in the test set with 80% accuracy [57].

accuracy was reported (i.e., $R^2 > 0.9$), which could indicate sample carryover between the training and test set and ultimately model overfitting. Models that overfit data typically display low accuracy in prospective studies if not used in the same way they were tested. The test set used to evaluate a model needs to challenge it in a way that is representative of its future use. For example, if a dataset is constructed with 10 APIs and future use will not extend beyond these 10 APIs, a random data splitting strategy should be sufficient, as all 10 APIs will be included in both the training and test sets. However, if the goal is to train the model to predict outcomes for new APIs, a random split would not pick up on overfitting during training, and the model would not perform well for new APIs when used prospectively. In this case, a better approach would be to make sure that the data is appropriately split for model training (e.g., trained on 8 APIs and tested on 2 APIs it has never seen before). This would provide a better estimate of the real performance of the model and highlight any overfitting issues during testing of the two "unseen" APIs in the test set.

For example, in some of the studies mentioned above, ML models trained through cross-validation, have been successfully used to prospectively predict performance of new API-material combinations, resulting in the identification of novel drug formulations. For example, Damiati et al. used a NN (trained on a small dataset through cross-validation) to determine the effects of hydrotropes on the solubility of indomethacin (IND), a non-steroidal anti-inflammatory agent [61]. Poor solubility is a challenge that must often be overcome for successful oral delivery of hydrophobic APIs, such as IND [58–60], and in this study the authors aimed to use hydrotropes (slightly amphiphilic organic molecules) to increase the solubility of IND [74]. By determining the solubility of IND in aqueous solutions of different hydrotropes and various concentrations of hydrotropes, the authors were able to train an NN to predict IND solubility in hypothetical hydrotrope solutions. Through a combination of model interpretation (i.e., NN connection weight interrogation), and *in silico* screening of 16 additional hydrotropes, the authors were able to identify key features that were important for hydrotrope-mediated solubilization of IND. This analysis allowed the authors to deduce that an "ideal" hydrotrope for IND would be a low complexity compound, that included a pyridine ring as a hydrophobe, a low hydrogen bond acceptor count and an alkyl-substituted amide moiety. Based on these features the authors identified pyridoxine (vitamin B6) as a good candidate. This "ideal" hydrotrope hypothesis was then tested and confirmed experimentally: in a 0.5 M solution of pyridoxine the water solubility of IND was increased by 727-fold, almost double what was observed by the next best hydrotrope (sodium nicotinate), which was part of the initial training dataset, Scheme 3.

The use of ML models for advanced oral delivery systems has extended beyond formulation development and the prediction of *in vitro* performance. ML models have also been used to guide selection when deciding on a particular oral formulation strategy [75]. Branchu et al. used data from commercially available oral formulations to train classification models to determine which oral formulation strategy (including conventional formulations, crystalline nanoparticles, solid dispersions, lipidic/surfactant systems) would be most appropriate given the physico-chemical properties of an API [75]. While the resulting models operated under several assumptions (i.e., passive diffusion is the primary mode of API absorption from the GI tract), this idea of using data from regulatory approved formulations to guide future development could increase the probability of success of new oral formulations.

Additionally, attempts have been made to use ML to improve *in vitro - in vivo* correlations (IVIVC) of oral dosage forms [76–78]. The development of reliable ML models that can predict *in vivo* performance, such as the pharmacokinetic profile of a drug, has obvious ethical and economic advantages. Initial attempts to



Scheme 3. Illustration of the study conducted by Damiati et al. In this study the authors developed and evaluated a NN to predict the effect of hydrotrope type and concentration on indomethacin (IND) solubility. Using a combination of in-silico screening and the model interpretation (i.e., NN connection weight interrogation) the authors were able to identify the key features of hydrotrope compounds that affect IND solubility and thereby identify an "idea" hydrotrope for the drug [61].

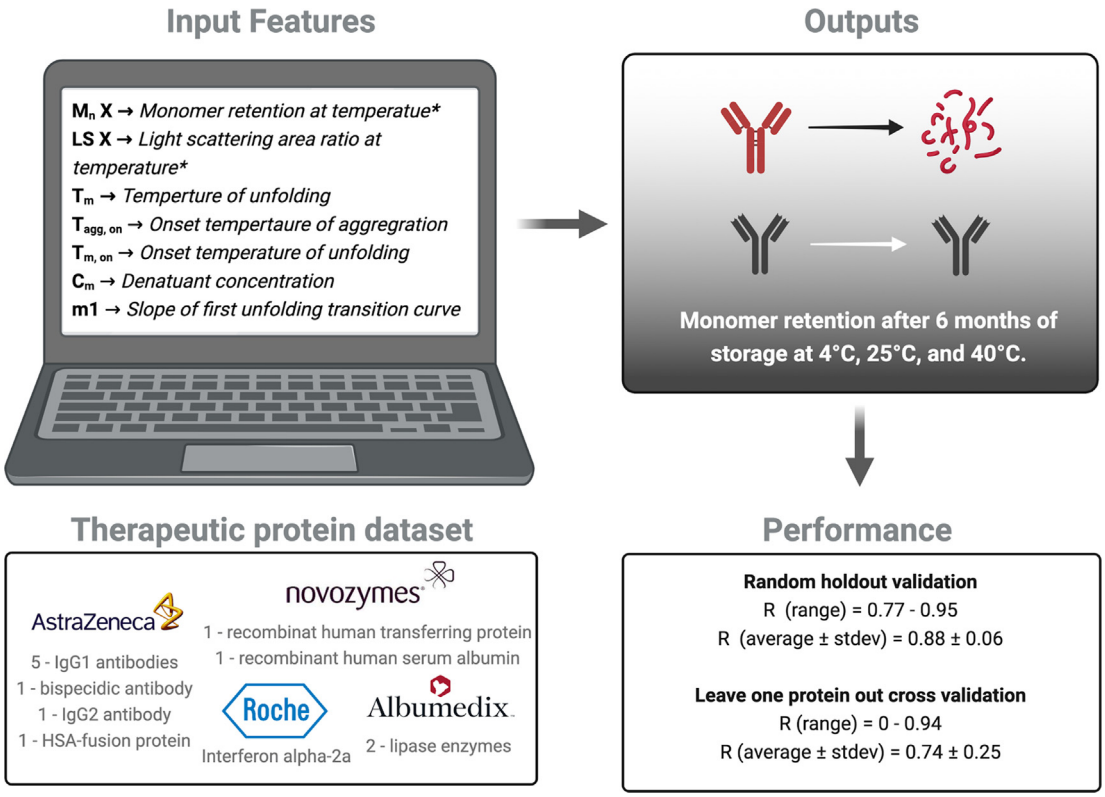
do this can be dated back to the late 1990s [76,77]. One early study by Dowell et al. used *in vitro* release data for two drug formulations (i.e., USP dissolution apparatus, n = 6) as inputs and pharmacokinetic profiles from nine patients enrolled in a clinical trial as outputs. Four different model structures were evaluated; these varied in numbers of input and output features. Generally, the aim was to use *in vitro* release data to predict the pharmacokinetic profile in different patients. The authors also examined random (i.e., ~10% of data) and manual (i.e., an individual subject with a single formulation used for test set) data splitting strategies. Overall, it was found that, models trained on manually split data (i.e., where data from an individual patient was withheld for testing) performed worse than their randomly split counterparts [76].

Similarly, Hussain et al. developed NN models to correlate the *in vitro* release (i.e., USP dissolution apparatus, n = 6) of metoprolol tartrate from extended-release tablets and diltiazem from extended-release capsules (i.e., three different formulations for each drug: fast, medium, and slow-releasing) to their respective *in vivo* performance (i.e., bioavailability study in nine healthy volunteers). In both cases, models were evaluated using a *leave-one-formulation-out cross-validation* approach. The NN models provided excellent predictions for the medium release rate formulations and were unable to extrapolate to the slow and fast release drug formulations [77]. While these early studies demonstrate the potential feasibility of ML for the development of IVIVC models, they also focused on very small datasets that contained only a single API and took only *in vitro* drug release profiles as input features, rather than the physico-chemical properties of the drug in question. In addition, these models were not trained on information related to the type of formulations, the dose of drug administered, details of the *in vivo* study design, or information related to the subject's demographics. However, this is still an active area of research, and more recent publications in this area have tried to develop larger datasets to train more generalizable ML models. For example, Mendyk et al. constructed a dataset containing 93 formulations of 13 drugs and used it to train NN models. The original dataset consisted of 307 initial input features (i.e., *in vitro* and *in vivo* study conditions, quantitative and qualitative formulation composition features, complete *in vitro* release profiles, and the time points corresponding to the *in vivo* pharmacokinetics samples). This was reduced to 28 input features through model sensitivity analysis. The best model could predict the *in vivo* release profiles with 37.6% accuracy (i.e., where successful predictions were declared if the normalized-root-mean-squared error did not exceed 20%) [78]. Any extension to existing IVIVC models can only be beneficial during the drug development process. Increased research in this space may provide us with increased foresight and enable better decision-making early in the formulation development process.

3.3. Protein therapeutics

While a considerable amount of research has examined how ML might be used to improve the development of conventional and advanced oral delivery systems, many APIs are difficult to administer orally. This is particularly true for biopharmaceuticals, which often exhibit physico-chemical instability in the GI tract, poor absorption across the GI wall and/or short half-life in the bloodstream [79]. Despite these difficulties, biopharmaceuticals can, in some cases, offer several advantages over small molecule drugs, including highly specific mechanisms of action as well as good tolerability [80]. More than 239 therapeutic proteins have been approved by the US FDA since the 1980s [80,81], and more than 600 biopharmaceuticals have been approved worldwide [82]. Given their potential as medicines, there is significant interest in the identification of methods to expedite their development. A ML-mediated, data-driven approach to formulate such biopharmaceuticals may facilitate such an expedited development process.

For example, Gentiluomo et al., used a NN model to predict various biophysical properties (including, melting temperature (T_m), aggregation onset temperature (T_{agg}) and the diffusion interaction parameter (k_D)) of therapeutic proteins, as a function of pH and



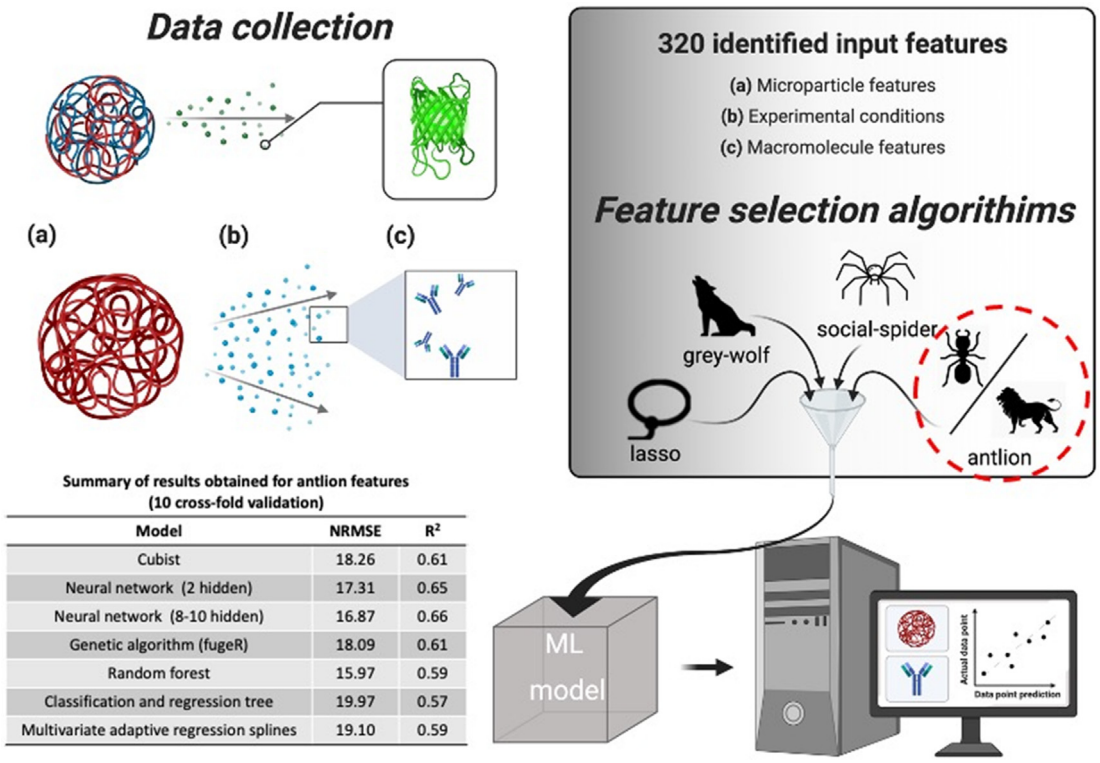
Scheme 4. Illustration of the study conducted by Gentiluomo et al. The authors investigated various DL model architectures and data splitting strategies to predict protein stability (i.e., monomer retention) after six months storage at various temperatures (i.e., 4 °C, 25 °C and 40 °C). Four pharmaceutical companies donated the dataset of therapeutic proteins used in this study, and the authors used experimentally determined metrics as well as preliminary stability data (i.e., up to two weeks) to conduct predictions of long-term stability. The leave-one-protein-out cross-validation approach provided a better means of testing the robustness of the DL model and would likely perform better at predicting the stability of new proteins never seen by the model [83].

ionic strength, based on the amino acid composition of the proteins [82]. To predict these properties, the authors collected experimental data on six immunoglobulin G antibodies (24 conditions per protein and a total dataset of 144 samples). This dataset was then used to train three distinct NNs to predict the T_m, T_{agg} and k_D, with varying degrees of accuracy (R^2 values of 0.98, 0.94 and 0.6 were obtained for T_m, T_{agg} and k_D, respectively). Moreover, an additional study conducted by the same group used NNs to predict the long-term stability (at six months) of 24 therapeutic protein formulations under various storage temperatures: 4℃, 25℃ and 40℃ (i.e., 72 total entries in the dataset). In this case, the NNs predicted the long-term stability of the therapeutic proteins was based on initial stability data collected over a two-week period under accelerated storage conditions as well as biophysical properties of the individual proteins [83], Scheme 4. In both cases described here [82,83], the authors used various cross-validation strategies (i.e., 5- and 10-fold cross-validation, as well as "leave-one-protein-out cross-validation") to avoid overfitting in these relatively small datasets (i.e., <200 samples). These studies demonstrate how ML can be beneficial in predicting the stability of biopharmaceuticals. This could be highly advantageous given that conformational, chemical and colloidal stability of such APIs are a constant concern during biopharmaceutical development. ML models that can determine API stability in this way will undoubtedly accelerate the future development of therapeutic proteins.

3.4. Microparticle and nanoparticle mediated drug delivery

MPs and NPs can offer several advantages over conventional formulations. Encapsulation of both small molecules and/or biopharmaceuticals into MPs and NPs can protect the API from external environmental conditions, provide continuous and/or controlled API release , an ability to maintain API concentrations within a safe therapeutic range as well as the ability to deliver combinations of APIs in a synergistic ratio to elicit an improved therapeutic effect [84–87]. However, relative to the thousands of sustained release oral delivery systems that have received regulatory approval over the past 30 years, the number of NP and MP formulations is negligibly low, approximately 50 and 20 formulations, respectively [84,86]. The difficulty in bringing such advanced drug delivery systems to the market is largely due to their complexity, with a significant number of parameters that must be considered in order to engineer an optimal formulation. The challenges associated with designing these systems presents a further opportunity for application of ML.

To this end, a growing number of studies have begun to use ML to guide the design and optimization of NP and MP delivery platforms. For example, ML has been used to predict the release rates of APIs from polymeric MPs [88–90], which is one of the most tedious and time-consuming aspects of MP development. Mendyk et al. trained various ML models to predict the release profiles of macromolecules from PLGA MPs [88]. The authors collected release data from previous publications that described 68 different PLGA MP formulations of 14 macromolecules. The final dataset consisted of API release data as well as 319 input parameters which described various aspects of the systems, including characteristics of the formulations, experimental release conditions, as well as molecular descriptors for both macromolecules and excipients used in the formulations. The trained NN models were able to predict macromolecule release with a high degree of accuracy (15.4% and 14.3% normalized RMSE, respectively), and with a final list of less than 20 of the aforementioned input features. In a follow up



Summary of results obtained for antlion features (10 cross-fold validation)

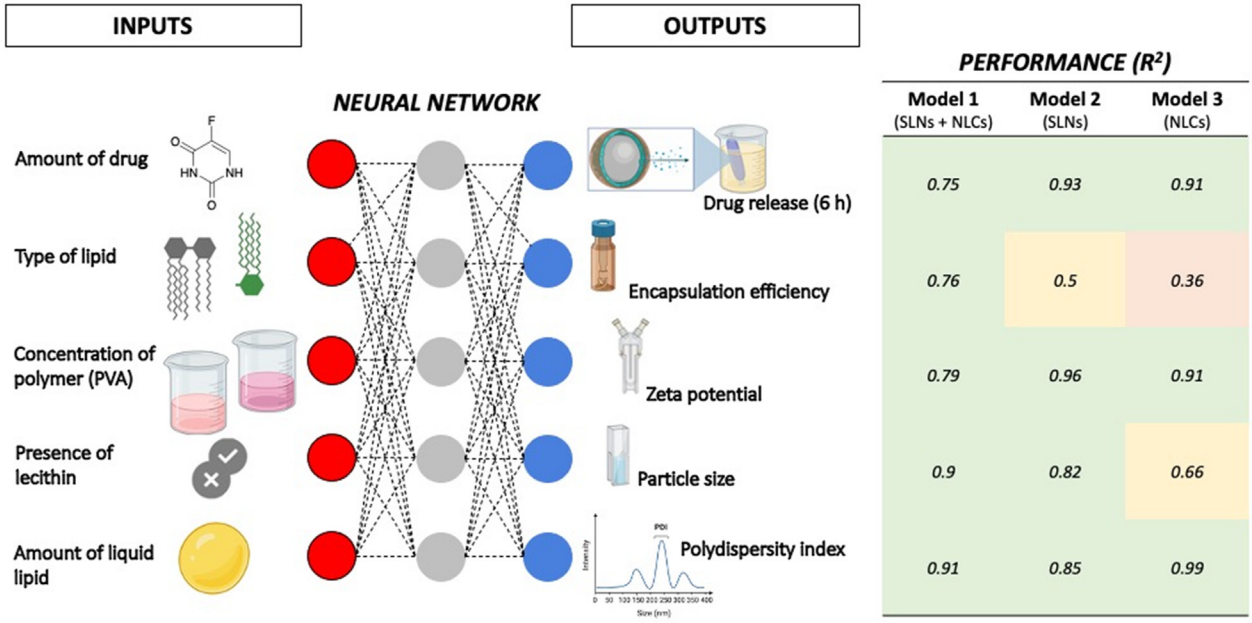
Model	NRMSE	R^2
Cubist	18.26	0.61
Neural network (2 hidden)	17.31	0.65
Neural network (8-10 hidden)	16.87	0.66
Genetic algorithm (fugeR)	18.09	0.61
Random forest	15.97	0.59
Classification and regression tree	19.97	0.57
Multivariate adaptive regression splines	19.10	0.59

Scheme 5. Illustration of the study conducted by Zawbaa et al. In this study, the authors, collected experimental data for macromolecule release from PLGA microparticles and then aimed to develop a ML model to predict macromolecule release from these delivery systems. The authors identified 320 potential input features that could be used to describe (i) the MPs, (ii) the experimental conditions for release, and (iii) the physico-chemical properties of the macromolecule. The authors proceeded to evaluate a variety of feature selection algorithms (i.e., grey-wolf, lasso, social-spider, and antlion) and various ML models to construct a ML model with high accuracy and the optimal number of input features. The authors identified the so-called binary antlion optimization (BALO) algorithm as the best at narrowing the pool of potential features (i.e., nine features selected). The performance of the various ML models trained on the BALO features varied, with the neural network (8–10 layers) marginally outperforming the others in terms of R2 and normalized-root-mean-square error (NRMSE) [89].

study by the same group, the authors further investigated the effects of distinct input parameters on the resulting prediction accuracy of different ML models using feature selection algorithms to narrow the pool of 319 potential input features, see Scheme 5. Here the authors found that a RF model with even fewer input features (i.e., nine parameters), selected via a feature selection algorithm (i.e., binary antlion [91]), predicted macromolecule release from the PLGA MPs as well as if not better than the previously reported NN models [89]. In both of these studies [82,83], the authors used 10-fold cross-validation to split their dataset, each time excluding all of the data belonging to a particular formulation, in order to simulate the real application of the model, which was to predict the performance of a new macromolecule-PLGA formulation. Along with the prediction of macromolecule release from PLGA MPs, other researchers have utilized ML models to predict the release of small molecules. For example, Wu et al., used NNs to predict the *in vitro* release kinetics of doxorubicin from sulfopropyl dextran ion-exchange MPs [90]. However, the small size of this dataset (i.e., 18 samples), the splitting strategy employed (i.e., random holdout), and the high accuracy reported (i.e., R^2 = 0.98–0.99) could be an indicator of model overfitting. Other applications in this space include the design of customized injection system (i.e., that increased the injectability of large MPs six-fold in comparison to commercial syringes) using a combination of computational fluid dynamics, experimental design and ML [92].

ML has also been integrated into the development of nanomedicines. Nano-sized advanced delivery platforms, such as polymeric or lipid-based NPs, have the potential to overcome critical barriers associated with conventional medicines, including the ability to deliver combinations of APIs in a synergistic ratio to elicit an improved therapeutic effect (e.g., Vyxeos®) [86]. The applications of nanomedicines extend beyond the delivery of small molecule chemotherapeutics. NP-based formulations have been approved for the targeted delivery of small interfering ribonucleic acid (siRNA) (e.g., Onpattro®), in radiotherapy (e.g., Hensify®) and as imaging agents (e.g., Technicoll®). There are currently more than 50 drug products (approved for human use by multiple regulatory agencies) that rely on NP formulation strategies, with many other nanomedicines in clinical development [86]. However, as with MPs, the design and development of effective nanomedicines is non-trivial.

To date, most of the studies that have employed ML models in the design of NPs have focused on predicting the properties of drug loaded NPs using NNs [93–96]. Most of these studies have used small datasets (i.e., ~50 data samples or less) and report very high prediction accuracies (R^2 > 0.9) raising concerns with model overfitting. Nonetheless, some of the resulting models demonstrated themselves to be fit for purpose when applied to formulation optimization. For example, Amasya et al., conducted an extensive literature review to identify the ideal range for a number of key NP properties which are commonly observed for transdermal delivery (i.e., particle size, polydispersity index (PDI), charge and encapsulation efficiency values) [96]. These key properties were then used as target formulation parameters in a quality by design approach to NP development. The authors prepared 32 variations of 5-fluorouracil lipid NPs (including 16 solid lipid NPs and 16 nanostructured lipid carriers). This data was then used to train NNs to predict the aforementioned key NP properties, see Scheme 6. The resulting trained NNs were evaluated and used to conduct virtual experiments to determine which experimental conditions would



Scheme 6. Illustration of the study conducted by Amasya et al. The authors collected experimental data for 16 solid lipid nanoparticle (SLN) and 16 nanostructured lipid carrier (NLC) formulations of 5-fluorouracil. The authors used this data to train a variety of NNs to predict physico-chemical properties and performance of the drug-loaded nanoparticles (i.e., amount of drug release after 6 h *in vitro*, drug encapsulation efficiency, as well as particle size, zeta potential, and polydispersity index), using the input features listed above. Three NNs in total were trained; Model 1 (i.e., trained on both SLN and NLC data), Model 2 (i.e., trained on SLN data only), and Model 3 (i.e., trained on NLC data only). The authors evaluated the models based on the following metrics: R2 $\geq$ 0.7 = "reliable" (i.e., shown in green), 0.5 $\geq$ R2 $\geq$ 0.7 = "model to need for caution" (i.e., shown in yellow), and R2 $\leq$ 0.5 = "unreliable model" (i.e., shown in red). Once trained and evaluated in this way, the authors used the NNs, in conjunction with quality by design, to conduct virtual experiments and derive an optimal 5-fluorouracil loaded nanoparticle formulation transdermal delivery. *While this study demonstrates an interesting application for ML models in formulation design, the optimized 5-fluorouracil loaded nanoparticle formulation was not evaluated for transdermal delivery* [96].

yield a lipid NP of 5-fluorouracil with the optimal properties for transdermal delivery.

Furthermore, ML models have been developed to predict the experimental conditions necessary to prepare drug nanocrystals. He et al. developed several light gradient boosting machine (LightGBM) models to predict the size and PDI of drug nanocrystals prepared by three distinct techniques (i.e., wet ball milling (WBM), high-pressure homogenization (HPH) or antisolvent precipitation (ASP)) [97]. The authors trained these models based on data collected from previous publications (i.e., 523, 197 and 190 data samples for WBM, HPH, and ASP, respectively). The authors used both 10-fold cross-validation and various types of model regularization to avoid model overfitting during training. For each optimized size prediction model (i.e., WBM, HPH, and ASP), feature importance analysis was also conducted to rank the importance of the 20 most relevant features. In each case the trained models appeared to appropriately rank features based on existing pharmaceutical knowledge, for example, WBM (i.e., #1 = milling time), HPH (i.e., #1 = cycle index) and ASP (i.e., #1 = concentration of stabilizer). The logical ranking of input features in this way provides further confidence in the ability of trained models to generate useful predictions. Prospective predictions conducted as part of this study on "unseen" drugs (i.e., for celastrol, glipizide, and docetaxel) further tested the generalizability of the models. However, only the WBM and HPH models were able to successfully predict outcomes for these drug nanocrystals prospectively.

4. Outlook

ML models enable the users to analyze experimental results to uncover subtle patterns that are not immediately visible. While the majority of the studies summarized herein reported ML models with high predictive accuracy, many of these models have only been evaluated retrospectively. Only a limited number of studies have included prospective experimental validation and model interpretation steps. It is through such analytical steps that ML models can be used to generate new knowledge and afford innovative formulations with improved properties and performance. In addition, many recent advances in ML have yet to be harnessed by pharmaceutical scientists in the development of drug formulations. More sophisticated DL architectures are generally becoming more easily available to computational scientists thanks to continued improvements in the user-friendliness of ML libraries. For instance, quantifying prediction uncertainty provides invaluable information on whether a prediction is trustworthy and might affect the scientist's decision-making process. Bayesian versions of DL models, which are able to provide such uncertainty estimates, are available to researchers via established frameworks such as *TensorFlow* [27] and *PyTorch* [26]. Furthermore, new model architectures that are particularly suitable to handle chemistry problems have been developed. Graph neural networks are a special version of NNs that are able to take molecules directly as input, circumventing the need for hand-crafted feature engineering. When enough data is available, this approach has been shown to outperform more traditional ML models in the prediction of molecular properties [21,35] and these techniques have already been employed in the discovery of new antibiotics [18].

Generative ML models have also been rapidly gaining popularity in chemistry and drug discovery. Rather than providing property predictions, these models are able to directly suggest molecules that display properties of interest [98]. Generative models have been used, for instance, to propose new kinase inhibitors [3], and to suggest molecules likely to induce desired gene expression profiles [99]. We identify as a great area of opportunity the use of generative models to actively suggest new APIs and formulations based on gathered data. While these models typically require large amounts of data, the use of transfer learning and other ML advances may reduce the cost of setting up these generative mod-

els. Genetic algorithms have also proven to be a competitive alternative to generative models for molecular discovery tasks [100]. With specialized molecular representations for generative tasks [101], the recently introduced STONED algorithm [102] provides a simple and efficient means to perform interpolation and exploration in chemical space, with comparable performance to deep generative models. How these novel representations and algorithms will be translated to drug formulation development is a promising potential research direction to be explored.

While other ML techniques, such as interpretable and explainable ML, are becoming increasingly popular, most work in formulation development has focused only on prediction accuracy. In the future, we expect additional research efforts to focus on the interpretation of ML models to gain scientific insight in addition to accurate predictions. Other approaches, such as active and reinforcement learning, should also be further investigated as they enable the optimal planning of experiments. These algorithms have already been used in chemistry and materials science research, and it is only a matter of time before they are adopted in drug formulation development. Examples of these applications have been the optimization of chemical reaction conditions [22,103], of thin-film manufacturing protocols [20], and of organic photovoltaic blends [19]. These ML approaches may be interpreted as a modern incarnation of the field of design of experiments. In drug formulation development, these experiment-planning algorithms will likely soon be employed to inform researchers of which experiments to perform to achieve desired formulation properties with the least amount of experimental effort.

In summary, we have highlighted several examples of how ML tools might be harnessed to address inherent challenges encountered during drug formulation development, and to aid in the development and characterization of both conventional and non-conventional drug formulations. ML technologies have undoubtedly revolutionized pharmaceutical drug discovery pipelines in recent decades and are currently becoming more widely integrated into the healthcare system. The integration of ML techniques can enable pharmaceutical scientists to generate low-cost predictions and could serve to significantly expedite drug formulation development. We anticipate that further inclusion of ML models into the pharmaceutical sciences will provide us with the tools necessary to move away from trial-and-error based drug product development, to prioritize experimental work on the most promising material candidates and to move towards a more efficient data-driven formulation development process. The expanded use of ML in the pharmaceutical sciences in the coming years will require continued close interdisciplinary collaboration between pharmaceutical, data and computer scientists. Importantly, the opportunity offered by the integration of ML in drug formulation development, and more broadly pharmaceutical sciences, should not only be considered a means to fast-track efforts, but rather a door to uncovering new materials, innovative formulations, and new knowledge. For these reasons, we believe that ML is uniquely positioned to transform the way in which medicines are developed.

Author contributions

The manuscript was written through contributions of all authors. All authors have approved the final version of the manuscript.

Funding Sources

NSERC Discovery grant (RGPIN-2016-04293) to C. A. F.H., M.A., and A.A.G. acknowledge support by the Defense Advanced Research Projects Agency under the Accelerated Molecular Discovery Program under Cooperative Agreement No. HR00111920027 dated August 1, 2019. A.A.G. would like to thank Dr. Anders Frøseth for his support. M.A. is supported by a Postdoctoral Fellowship of the Vector Institute.

Acknowledgment

Images were created using BioRender.com, ChemDraw Professional, TheNounProject.com and Smart.Servier.com.

References

[1] Aulton's pharmaceutics: the design and manufacture of medicines, Elsevier, 2018.
[2] R.F. Pagels, R.K. Prud'homme, Polymeric nanoparticles and microparticles for the delivery of peptides, biologics, and soluble therapeutics, J. Controlled Release 219 (2015) 519–535.
[3] B.J. Boyd et al., Successful oral delivery of poorly water-soluble drugs both depends on the intraluminal behavior of drugs and of appropriate advanced drug delivery systems, Eur. J. Pharm. Sci. 137 (2019) 104967.
[4] P. Cerreia Vioglio, M.R. Chierotti, R. Gobetto, Pharmaceutical aspects of salt and cocrystal forms of APIs and characterization challenges, Adv. Drug Deliv. Rev. 117 (2017) 86–110.
[5] D.J. Berry, J.W. Steed, Pharmaceutical cocrystals, salts and multicomponent systems; intermolecular interactions and property based design, Adv. Drug Deliv. Rev. 117 (2017) 3–24.
[6] M. De Vivo, M. Masetti, G. Bottegoni, A. Cavalli, Role of molecular dynamics and related methods in drug discovery, J. Med. Chem. 59 (2016) 4035–4061.
[7] D.B. Kitchen, H. Decornez, J.R. Furr, J. Bajorath, Docking and scoring in virtual screening for drug discovery: methods and applications, Nat. Rev. Drug Discov. 3 (2004) 935–949.
[8] H. Chen, T. Kogej, O. Engkvist, Cheminformatics in drug discovery, an industrial perspective, Mol. Inform. 37 (2018) 1800041.
[9] V. Gapsys et al., Large scale relative protein ligand binding affinities using non-equilibrium alchemy, Chem. Sci. 11 (2020) 1140–1152.
[10] M. Aldeghi, A. Heifetz, M.J. Bodkin, S. Knapp, P.C. Biggin, Accurate calculation of the absolute free energy of binding for drug molecules, Chem. Sci. 7 (2016) 207–218.
[11] S. Hossain, A. Kabedev, A. Parrow, C.A.S. Bergström, P. Larsson, Molecular simulation as a computational pharmaceutics tool to predict drug solubility, solubilization processes and partitioning, Eur. J. Pharm. Biopharm. 137 (2019) 46–55.
[12] T. Casalini, Not only in silico drug discovery: molecular modeling towards in silico drug delivery formulations, J. Controlled Release 332 (2021) 390–417.
[13] M.G. Saunders, G.A. Voth, Coarse-graining methods for computational biology, Annu. Rev. Biophys. 42 (2013) 73–93.
[14] A. Hosny, C. Parmar, J. Quackenbush, L.H. Schwartz, H.J.W.L. Aerts, Artificial intelligence in radiology, Nat. Rev. Cancer 18 (2018) 500–510.
[15] N. Wu et al., Deep neural networks improve radiologists' performance in breast cancer screening, IEEE Trans. Med. Imaging 39 (2020) 1184–1194.
[16] S.M. McKinney et al., International evaluation of an AI system for breast cancer screening, Nature 577 (2020) 89–94.
[17] A. Zhavoronkov et al., Deep learning enables rapid identification of potent DDR1 kinase inhibitors, Nat. Biotechnol. 37 (2019) 1038–1040.
[18] J.M. Stokes et al., A deep learning approach to antibiotic discovery, Cell 180 (2020) 688–702.e13.
[19] S. Langner et al., Beyond ternary OPV: high-throughput experimentation and self-driving laboratories optimize multicomponent systems, Adv. Mater. 32 (2020) 1907801.
[20] B.P. MacLeod et al., Self-driving laboratory for accelerated discovery of thin-film materials, Sci. Adv. 6 (2020) eaaz8867.
[21] C.W. Coley et al., A graph-convolutional neural network model for the prediction of chemical reactivity, Chem. Sci. 10 (2019) 370–377.
[22] Z. Zhou, X. Li, R.N. Zare, Optimizing chemical reactions with deep reinforcement learning, ACS Cent. Sci. 3 (2017) 1337–1344.
[23] A.W. Senior et al., Improved protein structure prediction using potentials from deep learning, Nature 577 (2020) 706–710.
[24] F. Pedregosa, G. Varoquaux, A. Gramfort, et al., Scikit-learn: machine learning in Python, J. Mach. Learn. Res. 12 (2011) 2825–2830.
[25] F. Chollet, et al., Keras, https://keras.io (2015).
[26] A. Paszke et al., PyTorch: an imperative style, high-performance deep learning library, 33rd Conference on Neural Information Processing Systems (NeurIPS 2019), Vancouver, Canada (2019).
[27] M. Abadi et al., TensorFlow: large-scale machine learning on heterogeneous distributed systems, Computer Science > Distributed, Parallel, and Cluster Computing (2016) arXiv:1603.04467.
[28] C.M. Bishop, Pattern recognition and machine learning, Springer, 2006.
[29] L. Breiman, Bagging predictions, Mach. Learn. 45 (1996) 5–32.
[30] J.H. Friedman, Greedy function approximation: a gradient boosting machine, Ann. Stat. 29 (2001) 1189–1232.

[31] C.E. Rasmussen, C.K.I. Williams, Gaussian processes for machine learning, MIT Press, 2006.
[32] C. Cortes, V. Vapnik, Support-vector networks, Mach. Learn. 20 (1995) 273–297.
[33] Y. LeCun, Y. Bengio, G. Hinton, Deep learning, Nature 521 (2015) 436–444.
[34] D. Duvenaud et al., Convolutional Networks on Graphs for Learning Molecular Fingerprints. ArXiv150909292 Cs Stat (2015).
[35] K. Yang et al., Analyzing learned molecular representations for property prediction, J. Chem. Inf. Model. 59 (2019) 3370–3388.
[36] O. Wieder et al., A compact review of molecular property prediction with graph neural networks, Drug Discov. Today Technol. S1740674920300305 (2020), https://doi.org/10.1016/j.ddtec.2020.11.009.
[37] E. Kim et al., Materials synthesis insights from scientific literature via text extraction and machine learning, Chem. Mater. 29 (2017) 9436–9444.
[38] O.A. Tarasova, N.Yu. Biziukova, D.A. Filimonov, V.V. Poroikov, M.C. Nicklaus, Data mining approach for extraction of useful information about biologically active compounds from publications, J. Chem. Inf. Model. 59 (2019) 3635–3644.
[39] F. Häse, L.M. Roch, A. Aspuru-Guzik, Next-generation experimentation with self-driving laboratories, Trends Chem. 1 (2019) 282–291.
[40] L. Wilbraham, S.H.M. Mehr, L. Cronin, Digitizing chemistry using the chemical processing unit: from synthesis to discovery, Acc. Chem. Res. 54 (2021) 253–262.
[41] Z. Liu et al., PDB-wide collection of binding data: current status of the PDBbind database, Bioinformatics 31 (2015) 405–412.
[42] M.K. Gilson et al., BindingDB in 2015: a public database for medicinal chemistry, computational chemistry and systems pharmacology, Nucleic Acids Res. 44 (2016) D1045–D1053.
[43] Y. Bengio, A. Courville, P. Vincent, Representation Learning: A Review and New Perspectives. ArXiv12065538 Cs (2014).
[44] A. Krizhevsky, I. Sutskever, G.E. Hintion, ImageNet classification with deep convolutional neural networks, in: Proceedings of the 25th International Conference on Neural Information Processing System, 2012.
[45] J. Bergstra, D. Yamins, D. Cox, Making a Science of Model Search: Hyperparameter Optimization in Hundreds of Dimensions for Vision Architectures, in: Proceedings of the 30th International Conference on Machine Learning, 2013.
[46] P. Friederich, M. Krenn, I. Tamblyn, A. Aspuru-Guzik, Scientific intuition inspired by machine learning generated hypotheses, ArXiv201014236 Phys. Physicsquant-Ph (2020).
[47] F. Häse, I. Fdez. Galván, A. Aspuru-Guzik, R. Lindh, M. Vacher, How machine learning can assist the interpretation of ab initio molecular dynamics simulations and conceptual understanding of chemistry, Chem. Sci. 10 (2019) 2298–2307.
[48] C. Rudin, Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead, Nat. Mach. Intell. 1 (2019) 206–215.
[49] K. McCloskey, A. Taly, F. Monti, M.P. Brenner, L.J. Colwell, Using attribution to decode binding mechanism in neural network models for chemistry, Proc. Natl. Acad. Sci. 201820657 (2019), https://doi.org/10.1073/pnas.1820657116.
[50] G. Stiglic et al., Interpretability of machine learning-based prediction models in healthcare, WIREs Data Min. Knowl. Discov. 10 (2020).
[51] C. Wang, B. Han, B. Patel, F. Mohideen, C. Rudin, In Pursuit of Interpretable, Fair and Accurate Machine Learning for Criminal Recidivism Prediction, ArXiv200504176 Cs Stat (2020).
[52] J. Bourquin, H. Schmidli, P. van Hoogevest, H. Leuenberger, Comparison of artificial neural networks (ANN) with classical modelling techniques using different experimental designs and data from a galenical study on a solid dosage form, Eur. J. Pharm. Sci. 6 (1998) 287–300.
[53] J. Bourquin, H. Schmidli, P. van Hoogevest, H. Leuenberger, Application of Artificial Neural Networks (ANN) in the Development of Solid Dosage Forms, Pharm. Dev. Technol. 2 (1997) 111–121.
[54] J.G. Kesavan, G.E. Peck, Pharmaceutical granulation and tablet formulation using neural networks, Pharm. Dev. Technol. 1 (1996) 391–404.
[55] M. Turkoglu, I. Aydin, M. Murray, A. Sakr, Modeling of a roller-compaction process using neural networks and genetic algorithms, Eur. J. Pharm. Biopharm. 48 (1999) 239–245.
[56] K. Takagaki, H. Arai, K. Takayama, Creation of a tablet database containing several active ingredients and prediction of their pharmaceutical characteristics based on ensemble artificial neural networks, J. Pharm. Sci. 99 (2010) 4201–4214.
[57] R. Han, Y. Yang, X. Li, D. Ouyang, Predicting oral disintegrating tablet formulations by neural network techniques, Asian J. Pharm. Sci. 13 (2018) 336–342.
[58] P. Bannigan, J. Flynn, S.P. Hudson, The impact of endogenous gastrointestinal molecules on the dissolution and precipitation of orally delivered hydrophobic APIs, Expert Opin. Drug Deliv. 17 (2020) 677–688.
[59] P. van Hoogevest, X. Liu, A. Fahr, Drug delivery strategies for poorly water-soluble drugs: the industrial perspective, Expert Opin. Drug Deliv. 8 (2011) 1481–1500.
[60] G. Amidon, A theoretical basis for a biopharmaceutical classification system, Pharm. Res. 12 (1995) 413–420.
[61] S.A. Damiati, L.G. Martini, N.W. Smith, M.J. Lawrence, D.J. Barlow, Application of machine learning in prediction of hydrotrope-enhanced solubilisation of indomethacin, Int. J. Pharm. 530 (2017) 99–106.
[62] Y. Yang et al., Deep learning for in vitro prediction of pharmaceutical formulations, Acta Pharm. Sin. B 9 (2019) 177–185.
[63] J. Petrović, S. Ibrić, G. Betz, J. Parojčić, Z. Đurić, Application of dynamic neural networks in the modeling of drug release from polyethylene oxide matrix tablets, Eur. J. Pharm. Sci. 38 (2009) 172–180.
[64] S. Ibrić, M. Jovanović, Z. Djurić, J. Parojčić, L. Solomun, The application of generalized regression neural network in the modeling and optimization of aspirin extended release tablets with Eudragit® RS PO as matrix substance, J. Controlled Release 82 (2002) 213–222.
[65] J. Petrović, S. Ibrić, G. Betz, Z. Đurić, Optimization of matrix tablets controlled drug release using Elman dynamic neural networks and decision trees, Int. J. Pharm. 428 (2012) 57–67.
[66] A. Ghaffari et al., Performance comparison of neural network training algorithms in modeling of bimodal drug delivery, Int. J. Pharm. 327 (2006) 126–138.
[67] P. Barmpalexis, F.I. Kanaze, K. Kachrimanis, E. Georgarakis, Artificial neural networks in the optimization of a nimodipine controlled release tablet formulation, Eur. J. Pharm. Biopharm. 74 (2010) 316–323.
[68] R. Han et al., Predicting physical stability of solid dispersions by machine learning techniques, J. Controlled Release 311–312 (2019) 16–25.
[69] Z. Zhang, W. Pan, Expert system for the development and formulation of push–pull osmotic pump tablets containing poorly water-soluble drugs, in: Formulation Tools for Pharmaceutical Development 73–108, Elsevier, 2013. doi:10.1533/9781908818508.73.
[70] Z. Zhang et al., Design of an expert system for the development and formulation of push–pull osmotic pump tablets containing poorly water-soluble drugs, Int. J. Pharm. 410 (2011) 41–47.
[71] A. Mendyk, J. Szlęk, R. Jachowicz, ME_expert 2.0: a heuristic decision support system for microemulsions formulation development, in: Formulation Tools for Pharmaceutical Development 39–71, Elsevier, 2013. doi:10.1533/9781908818508.39.
[72] H. Gao et al., Predicting drug/phospholipid complexation by the lightGBM method, Chem. Phys. Lett. 747 (2020) 137354.
[73] Q. Zhao, Z. Ye, Y. Su, D. Ouyang, Predicting complexation performance between cyclodextrins and guest molecules by integrated machine learning and molecular modeling techniques, Acta Pharm. Sin. B 9 (2019) 1241–1252.
[74] M. Hopkins Hatzopoulos et al., Are hydrotropes distinct from surfactants?, Langmuir 27 (2011) 12346–12353.
[75] S. Branchu, P.G. Rogueda, A.P. Plumb, W.G. Cook, A decision-support tool for the formulation of orally active, poorly soluble compounds, Eur. J. Pharm. Sci. 32 (2007) 128–139.
[76] J.A. Dowell, A. Hussain, J. Devane, D. Young, Artificial neural networks applied to the in vitro-in vivo correlation of an extended-release formulation: initial trials and experience, J. Pharm. Sci. 88 (1999) 154–160.
[77] A.S. Hussain, Artificial Neural Network Based in Vitro-in Vivo Correlations, in: D. Young, J.G. Devane, J. Butler (Eds.), In Vitro-in Vivo Correlations (vol. 423 149–158 (Springer US, 1997).
[78] A. Mendyk, P. Tuszyński, Polak, Jachowicz, Generalized in vitro-in vivo relationship (IVIVR) model based on artificial neural networks, Drug Des. Devel. Ther. 223 (2013) doi:10.2147/DDDT.S41401.
[79] T.D. Brown, K.A. Whitehead, S. Mitragotri, Materials for oral delivery of proteins and peptides, Nat. Rev. Mater. 5 (2020) 127–148.
[80] B. Leader, Q.J. Baca, D.E. Golan, Protein therapeutics: a summary and pharmacological classification, Nat. Rev. Drug Discov. 7 (2008) 21–39.
[81] S.S. Usmani et al., THPdb: database of FDA-approved peptide and protein therapeutics, PLoS ONE 12 (2017) e0181748.
[82] L. Gentiluomo et al., Application of interpretable artificial neural networks to early monoclonal antibodies development, Eur. J. Pharm. Biopharm. 141 (2019) 81–89.
[83] L. Gentiluomo, D. Roessner, W. Frieß, Application of machine learning to predict monomer retention of therapeutic proteins after long term storage, Int. J. Pharm. 577 (2020) 119039.
[84] K. Park et al., Injectable, long-acting PLGA formulations: Analyzing PLGA and understanding microparticle formation, J. Controlled Release 304 (2019) 125–134.
[85] M.J. Mitchell et al., Engineering precision nanoparticles for drug delivery, Nat. Rev. Drug Discov. (2020), https://doi.org/10.1038/s41573-020-0090-8.
[86] M. Germain et al., Delivering the power of nanomedicine to patients today, J. Controlled Release 326 (2020) 164–171.
[87] C.I. Nkanga et al., Clinically established biodegradable long acting injectables: An industry perspective, Adv. Drug Deliv. Rev. 167 (2020) 19–46.
[88] A. Mendyk, J. Szlęk, R. Jachowicz, R. Lau, A. Paclawski, Heuristic modeling of macromolecule release from PLGA microspheres, Int. J. Nanomed. 4601 (2013), https://doi.org/10.2147/IJN.S53364.
[89] H.M. Zawbaa, J. Szlęk, C. Grosan, R. Jachowicz, A. Mendyk, Computational Intelligence Modeling of the Macromolecules Release from PLGA Microspheres—Focus on Feature Selection, PLoS ONE 11 (2016) e0157610.
[90] Y. Li, A.M. Rauth, X.Y. Wu, Prediction of kinetics of doxorubicin release from sulfopropyl dextran ion-exchange microspheres using artificial neural networks, Eur. J. Pharm. Sci. 24 (2005) 401–410.
[91] E. Emary et al., Binary ant lion approaches for feature selection, Neurocomputing 213 (C) (2016), https://doi.org/10.1016/j.neucom.2016.03.101.
[92] M. Sarmadi et al., Modeling, design, and machine learning-based framework for optimal injectability of microparticle-based drug formulations, Sci. Adv. 6 (2020) eabb6594.

[93] H. Asadi, K. Rostamizadeh, D. Salari, M. Hamidi, Preparation of biodegradable nanoparticles of tri-block PLA–PEG–PLA copolymer and determination of factors controlling the particle size using artificial neural network, J. Microencapsul. 28 (2011) 406–416.
[94] M. Soliman et al., Determination of factors controlling the particle size and entrapment efficiency of noscapine in PEG/PLA nanoparticles using artificial neural networks, Int. J. Nanomed. 4953 (2014), https://doi.org/10.2147/IJN.S68737.
[95] Y. Li, M.R. Abbaspour, P.V. Grootendorst, A.M. Rauth, X.Y. Wu, 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 (2015) 170–179.
[96] G. Amasya, U. Badilli, B. Aksu, N. Tarimci, Quality by design case study 1: design of 5-fluorouracil loaded lipid nanoparticles by the W/O/W double emulsion – solvent evaporation method, Eur. J. Pharm. Sci. 84 (2016) 92–102.
[97] Y. He et al., Can machine learning predict drug nanocrystals?, J. Controlled Release 322 (2020) 274–285.
[98] B. Sanchez-Lengeling, A. Aspuru-Guzik, Inverse molecular design using machine learning: generative models for matter engineering, Science 361 (2018) 360–365.
[99] O. Méndez-Lucio, B. Baillif, D.-A. Clevert, D. Rouquié, J. Wichard, De novo generation of hit-like molecules from gene expression signatures using artificial intelligence, Nat. Commun. 11 (2020) 10.
[100] A. Nigam, P. Friederich, M. Krenn, A. Aspuru-Guzik, Augmenting Genetic Algorithms with Deep Neural Networks for Exploring the Chemical Space. in (OpenReview.net, 2019).
[101] M. Krenn, F. Häse, A. Nigam, P. Friederich, A. Aspuru-Guzik, Self-referencing embedded strings (SELFIES): a 100% robust molecular string representation, Mach. Learn. Sci. Technol. 1 (2020) 045024.
[102] A. Nigam, R. Pollice, M. Krenn, G. dos Passos Gomes, A. Aspuru-Guzik, Beyond Generative Models: Superfast Traversal, Optimization, Novelty, Exploration and Discovery (STONED) Algorithm for Molecules using SELFIES. https://chemrxiv.org/articles/preprint/Beyond_Generative_Models_Superfast_Traversal_Optimization_Novelty_Exploration_and_Discovery_STONED_Algorithm_for_Molecules_using_SELFIES/13383266/1 (2020) doi:10.26434/chemrxiv.13383266.v1.
[103] M. Christensen, et al., Data-science driven autonomous process optimization, https://chemrxiv.org/articles/preprint/Data-science_driven_autonomous_process_optimization/13146404/2 (2020) doi:10.26434/chemrxiv.13146404.v2.