Technische Universität Berlin

Nghia Duong-Trung, Stefan Born, Jong Woo Kim, Marie-Therese Schermeyer, Katharina Paulick, Maxim Borisyak, Mariano Nicolas Cruz-Bournazou, Thorben Werner, Randolf Scholz, Lars Schmidt-Thieme, Peter Neubauer, Ernesto Martinez

When bioprocess engineering meets machine learning: A survey from the perspective of automated bioprocess development

Open Access via institutional repository of Technische Universität Berlin

Document type
Journal article | Accepted version
(i. e. final author-created version that incorporates referee comments and is the version accepted for publication; also known as: Author's Accepted Manuscript (AAM), Final Draft, Postprint)

This version is available at
https://doi.org/10.14279/depositonce-17066

Citation details
Duong-Trung, N., Born, S., Kim, J. W., Schermeyer, M.-T., Paulick, K., Borisyak, M., Cruz-Bournazou, M. N., Werner, T., Scholz, R., Schmidt-Thieme, L., Neubauer, P., & Martinez, E. (2023). When bioprocess engineering meets machine learning: A survey from the perspective of automated bioprocess development. In Biochemical Engineering Journal (Vol. 190, p. 108764). Elsevier BV. https://doi.org/10.1016/j.bej.2022.108764.

Terms of use
This work is protected by copyright and/or related rights. You are free to use this work in any way permitted by the copyright and related rights legislation that applies to your usage. For other uses, you must obtain permission from the rights-holder(s).

When Bioprocess Engineering Meets Machine Learning: A Survey from the Perspective of Automated Bioprocess Development

Nghia Duong-Trung[a,*], Stefan Born[a], Jong Woo Kim[a,d], Marie-Therese Schermeyer[a], Katharina Paulick[a], Maxim Borisyak[a], Mariano Nicolas Cruz-Bournazou[a], Thorben Werner[b], Randolf Scholz[b], Lars Schmidt-Thieme[b], Peter Neubauer[a], Ernesto Martinez[a,c,*]

[a] *Technische Universität Berlin, Faculty III Process Sciences, Institute of Biotechnology, Chair of Bioprocess Engineering. Straße des 17. Juni 135, 10623 Berlin, Germany.*
[b] *University of Hildesheim, Information Systems and Machine Learning Lab (ISMLL). Universitätspl. 1, 31141 Hildesheim, Germany.*
[c] *INGAR (CONICET-UTN), Avellaneda 3657, S3002GJC, Santa Fe, Argentina.*
[d] *Department of Energy and Chemical Engineering, Incheon National University, Incheon 22012, Republic of Korea.*

Abstract

Machine learning (ML) is becoming increasingly crucial in many fields of engineering but has not yet played out its full potential in bioprocess engineering. While experimentation has been accelerated by increasing levels of lab automation, experimental planning and data modeling are still largerly depend on human intervention. ML can be seen as a set of tools that contribute to the automation of the whole experimental cycle, including model building and practical planning, thus allowing human experts to focus on the more demanding and overarching cognitive tasks. First, probabilistic programming is used for the autonomous building of predictive models. Second, machine learning automatically assesses alternative decisions by planning experiments to test hypotheses and conducting investigations to gather informative data that focus on model selection based on the uncertainty of model predictions. This review provides a comprehensive overview of ML-based automation in bioprocess development. On the one hand, the biotech and bioengineering community should be aware of the potential and, most importantly, the limitation of existing ML solutions for their application in biotechnology and biopharma. On the other hand, it is essential to identify the missing links to enable the easy implementation of ML and Artificial Intelligence (AI) tools in valuable solutions for the bio-community. There is no one-fits-all procedure; however, this review should help identify the potential for automating model building by combining first-principles biotechnology knowledge and ML methods to address the

*Corresponding authors: Nghia Duong-Trung and Ernesto Martinez

reproducibility crisis in bioprocess development.

Keywords: Active Learning, Automation, Bioprocess Development, Reinforcement Learning, Reproducibility crisis.

1. Introduction

In the wake of climate change, many industries are turning to biotechnology to find sustainable solutions. The importance of biotechnological processes in pharmaceuticals is reflected in the growth figures for biopharmaceuticals (up 14 % to 30.8 % market share from 2020 to 2021)[1]. This trend is currently strongly inhibited by the long development times of biotechnological processes. To advance fast in bioprocess development, decisions must be taken under considerable high uncertainty, which does not enable a fast transition from laboratory to industrial production at scale with acceptable risks. Usually, different microorganisms or cells are tested to produce an industrial-relevant product, i.e., a pharmaceutical substance. The transfer of results from small to large scale represents a central challenge and is very time-consuming and error-prone.

Modern biolabs have automatized and parallelized many tasks aiming to run such a large number of experiments in short periods. These Robotic experimental facilities are equipped with Liquid Handling Stations (LHS) [2, 3], parallel cultivation systems, and High Throughput (HT) [4, 5] analytical devices which make them capable of timely generating informative data over a wide range of operating conditions. The past decade's focus was on hardware development and device integration with relatively simple data management systems lacking automatic association of the relevant metadata for the resulting experimental data. We have not been able to trigger the fruitful symbiosis expected between (i) robots that can perform thousands of complex tasks but are currently waiting for humans to design and operate the experiments; (ii) Active Learning (AL) algorithms that still rely on humans to perform the planned experiments, and (iii) Machine Learning (ML) tools are at the present waiting for humans to treat and deliver the data in a digital, machine-actionable format. Hence, end-to-end digitalization of experiments is a prerequisite to applying ML methods in bioprocessing.

Without complete annotations, the knowledge about how data were generated remains hidden, thus limiting the possible degree of automation for control and experimental design but also hampering the aggregation of data from different contexts. More importantly, difficulties in reproducing experiments prevent sharing and reuse by other researchers of costly experimental data.

Accordingly, with the advent of high-throughput robotic platforms, the bottleneck to efficient experimentation on a micro-scale has thus shifted from running a large number of parallel experiments to data management, model building, and experimental design, all of which currently rely on a considerable amount of human intervention which makes experiments barely reproducible. Only a proper data management system with standardized machine-actionable

data and automated metadata capture would allow an automatic flow of information through all stages of experimentation in bioprocess development and facilitate the use of machine learning models for decision-making in the face of uncertainty.

Let us consider scale-up as a representative example of the importance of metadata and experiment reproducibility. Miniaturized and versatile multi-bioreactor systems combined with LHS have the potential to significantly contribute to the practical generation of informative data to increase scale-up efficiency bearing in mind robustness to face the variability in operating conditions during strain selection at the initial stage. When transferring the acquired knowledge in the lab to the industrial scale, the remaining uncertainty in model predictions is significantly high due to insufficient data annotation and low levels of automation. Hence, critical decisions must be taken under high uncertainty, which imposes significant risks to most decisions taken throughout the bioprocess lifecycle.

The different stages of development cannot be treated in isolation. For example, the variability of operating conditions during strain selection directly influences the reproducibility of productivity levels in the scaled process. Hence, a promising route to faster development of innovative bioprocesses is a comprehensive automation of model building and experimental design across all stages of development. To drastically speed up the bioprocess development of innovative products, the ubiquitous use of automation in active learning from data and model building must be introduced in all stages, from product conceptualization to reproducible end-use properties. At any of these development stages, problem-solving and decision-making require building a model with enough predictive capability and a proper evaluation of its associated uncertainty.

In today's practice, model building and data collection depend heavily on manual tweaking and human intervention, which slows down the development effort and constitute a significant obstacle to lower costs and shorter times to market. Also, ML algorithms should be deployed with higher levels of autonomy to release the end-user from choosing alternatives for algorithms, hyperparameters, and problem representation incompatible with their understanding of the methods involved and underlying assumptions.

This review follows two crucial aims. On the one hand, the biotech and bioengineering community should be aware of the potential and, most importantly, some limitations of existing ML methods for their application in biotechnology and biopharma. On the other hand, it is essential to identify the missing links to enable the easy implementation of ML and Artificial Intelligence (AI) solutions in valuable solutions for the bio-community as end-users.

1.1. Decisions and Models

As shown in Fig. 1, model building is an important activity to assess alternatives and advance fast in the bioprocess lifecycle by making rational decisions that systematically reduce uncertainty. Model-based decision-making is widely used in the development lifecycle of different processes and products (e.g., electrical, chemical, aeronautics) for cost-effective design and improved operation

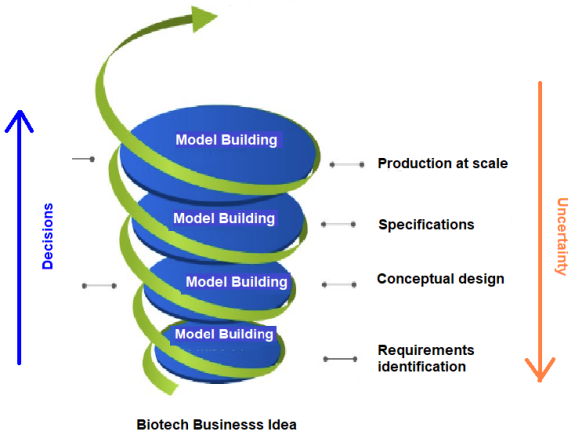


Figure 1: Reducing uncertainty in the bioprocess development lifecycle.

in the face of uncertainty. Mainly due to the so-called "small data problem," [6] bioprocess development has been an exception, though, with a significantly higher degree of empiric procedures, expert-based decisions, and strongly segmented design strategies and strain screening methods. The increased complexity of living organisms with thousands of intracellular biochemical reactions and uncomprehended responses in their metabolic activity due to regulatory mechanisms, combined with the difficulties in obtaining trustworthy observations, make it very difficult to build sound mathematical models since data collected from biological systems are inherently scarce and low-dimensional. However, similar dynamic behaviors within families of genetically modified microorganisms make enough room for transfer learning and meta-learning, using available data (with their metadata) to build predictive models for a new, unseen mutant. Based on such prior knowledge, experiments can be readily designed to gather informative data to increase the predictive power of models built to support decision-making effectively.

1.2. Automated Model-Building

Automation of the model-building cycle aims to assist experts and scientists in facilitating and transforming decision-making in the context of bioprocess engineering and biotechnology, not replacing them. Some model-building aspects are more difficult to automate because of technological challenges and involve open-ended questions and context-dependent tasks requiring human cognitive abilities. The most difficult challenge to model-building automation is that data sources in the development pipeline are diverse, distributed, and multi-structured. Moreover, not only the available data is highly heterogeneous, but they also may need to be more informative regarding the purpose of the model for a given stage. This fact contrasts sharply with the common assumption that sufficient amounts of high-quality data are available for model building.

Data collection methods for bioprocess development are descriptive of an inherently dynamic behavior since many process parameters are gathered online as continuous measurements. They are available and highly dependent on the quality of sensor devices, standard operating procedures, material and methods used, frequency of measurements (e.g., temperature, optical density, pH, oxygen, glucose, oxygen uptake rate, stirring speed), analytical sample processing, device calibration parameters, etc. Then, the optimal design of experiments for gathering information must be an integral part of the model-building cycle. As a result, the model-building at the different stages of bioprocess development will comprise automated procedures for actively seeking or generating highly informative data regarding the objective and type of decisions that must be taken in each stage in Fig. 1. It is related to the machine learning subfield known as active learning (AL). As an example, data that are informative for strain screening highlights the robustness of different strains to scale up effects where lack of enough aeration influences the physiological state of the bioreactor. However, these data are possibly poorly informative on optimizing the chosen strain's productivity after scale-up.

Faced with the choice of a large set of machine learning algorithms and an even larger space of hyperparameter settings, experts often must resort to costly experimentation in time and money to determine what combination works best for a given problem. Hence, automated model-building approaches must include automatic model selection, hyperparameter tuning, model training and model validation. If possible, models should be trained and validated with respect to the final properties of interest based on an end-to-end approach. This not only spares non-experts the time and effort of extensive, often onerous trial-and-error experimentation but also enables bioprocess engineers to obtain substantially better performance with fewer data and faster than possible without automation. In some applications of reinforcement learning to control and optimize, close-loop experimentation must be part and parcel of the model building cycle, making automation even more important. Hyperparameters, in this case, drive learning and define which data is gathered in the learning curve. Without a meta-learning level in model building, the initial setting of hyperparameters can easily prevent learning a predictive model that can make a real difference compared to not using a model at all for taking decisions. The use of ML for model-building automation can be seen as a way of introducing another level of abstraction that allows human experts to focus on higher level cognitive tasks for bioprocess development. First, probabilistic programming is used for the autonomous building of predictive models. Second, ML automatically assesses alternative decisions by planning experiments to test hypotheses and then planning and executing experiments to gather informative data that focus on model selection based on the uncertainty of model predictions. Therefore, ML methods can be seen as meta-algorithms for model-building tasks and automated data generation and hypothesis testing. Finally, the automated model-building uses algorithms that select and configure ML algorithms. That is, meta-meta-algorithms that can be understood as Bayesian machine experimenters [7] that can generate autonomously new data to transform a priori knowledge into ra-

tional decisions that further bioprocess development.

1.3. Present State of Data and Models in Bioprocess Development

At the initial stages of development (see 1), fundamental problems are addressed and key decisions are taken, such as strain screening which involves testing their robustness to alternative operating conditions, cultivation media, and bioreactor designs. The availability of Process Analytical Tools (PAT) [8, 9, 10] allows a deeper understanding of the processes and the technological advances of HT and LHS in robotic platforms [11, 12, 13] that can generate large amounts of experimental data to feed the model-building cycle. Yet, the bottleneck step of human-in-the-loop prevents a rapid transition toward design and operating decisions at larger scales. An essential link is missing toward model-based bioprocess systems engineering [14]: the conversion of automated experimental tasks and data (e.g., cultivation, sampling, analytics) into knowledge expressed in mathematical expressions. The large amounts of heterogeneous low quality data make manual treatment and model development almost impossible. Automating model-building using ML is envisioned as the alternative of choice to speed up automated bioprocess development while providing a setting for provenance and reproducibility to transform HT experimental data into information, information into knowledge, and to use this knowledge to understand, control, and optimize the bioprocess throughout its entire lifecycle.

Machine learning tools are already contributing to accelerated drug discovery [15] and have the potential also to speed up process development for biopharmaceuticals. When the ML tools are used in the actual production of pharmaceuticals, the requirements of regulatory bodies (e.g., FDA) for good manufacturing practices and process performance qualifications become an issue. In the context of software as a medical device (SaMD), the FDA published a paper on a proposed regulatory framework [16]. A database of FDA-approved SaMD applications until 2020 contained 64 medical applications based on ML/AI [17], but notably, only 29 of the items used machine learning or artificial intelligence-related terms in the official FDA documents. The FDA used to validate 'locked' algorithms only, that is, algorithms with parameters after training such that the same input would always map to the same output. Fortunately, the proposed regulatory framework shows that the FDA knows that many or perhaps the most relevant machine learning applications would be adaptive and continuously retrained on new data. Instead of a fixed input-output behavior, this requires a total product lifecycle regulatory approach, which determines how exactly models are retrained and validated. How far this approach will determine the FDA's behavior towards ML in manufacturing remains to be seen. The uncertainty that reigns until an explicit statement by the FDA and other regulatory bodies may, at present, deter companies from using ML in production. But, as we have seen, SaMD devices based on ML, which were not declared as such, have been validated by the FDA. The same may apply to process analytical components that are packaged as soft sensors but rely on ML.

As the developmental stages are more concerned with decisions related design and operating conditions, the model-building should focus on guaranteeing

physiological conditions that maximize productivity and product quality. For example, in the fed-batch cultivation phase, both overfeeding and underfeeding typically yield inferior results in cell growth and product formation [18]. Several studies have resorted to mechanistic models for (re)designing HT experiments of several fed-batch mini-bioreactors. The main challenges which are pending to be addressed are i) the use of impulsive control systems due to bolus-feeding for a miniaturized system, ii) ill-conditioned parameter estimation, and iii) low predictive power of mechanistic models.

In the work of [19, 12, 20], optimal experimental design problems were studied to maximize the information content for effective identification of a mechanistic model. Model predictive control using a mechanical model to maximize cell growth was implemented to an in silico system [21] and validated using an HT experiment [22].

However, due to its imperfect structure, a mechanistic model alone cannot significantly reduce the uncertainty related to operating and design decisions at more advanced stages. The latest trend clearly shows that machine learning techniques may give more room for more efficient utilization of available data and automate the generation of highly informative new data. Machine learning can provide viable and effective solution to the preceding problems. Over the past few decades, biotechnology has seen a significant shift from manual modeling to data-driven modeling, e.g., applying ML, partly due to a large amount of existing data for some biological systems [23, 24, 25]. It is an essential premise for integrating machine learning models that are built based well-informed biodata which are both FAIR and comprehensively annotated. Thus, data-driven models offer an appealing alternative for autonomous discovering in the field of bioprocessing [26, 27, 28].

Data-driven models are validated by their performance on the tasks for which they are trained. We should, however, bear in mind that unlike models built on first principles models learned from data will only extrapolate well to data coming from the same distribution. This may be the reason for a certain reluctance to adopt machine learning models, seen as black boxes without an interpretation. (It is, of course, possible to analyze or explain a machine learning model, once it is trained.) And yet, whenever no satisfactory first principles model is available, machine learning is the method of choice despite its lack of interpretability [29].

Machine learning has proved its effectiveness in many areas of biology: 3D structure of proteins [30, 31], up-downstream processes [32, 33, 34, 35], bioprocessing for chemical and biologic product manufacturing [25, 36], enzymes and cell growth [37, 38, 39], cell culture expression systems [40, 41, 42], and many others. However,According to [43], machine learning has not been as extensively used for bioprocess development as one might expect. This may be attributed to various reasons, of which some have already been mentioned. Data management and curation with an appropriate ontology for the metadata is one requirement not yet met. Furthermore the "small data problem" makes the off-the-shelf use of existing models problematic. Models have to be specially tailored to be expressive enough for the complexity of the investigated

phenomena, but constrained enough to be trainable with the available data. Overarching data management standards may also help to aggregate data and to tackle the small data problem from the opposite side. But even with appropriate models and well managed data, model selection, hyperparameter tuning, training and validation are still cognitive demanding tasks. Automation of this model building cycle is mandatory to increase the adoption of machine learning tools in bioprocess engineering. The selection of the most efficient algorithm and its parameters is based on many factors, including the transformation from a bioprocessing engineering problem into machine learning tasks, the quantity and quality of the data collected, the type of problem being solved (regression, classification,forecasting, control, etc.), the required overall accuracy and performance, availability of prior bioprocessing knowledge to control the hyperparameters tuning [44, 45]. As a result, a key challenge for model building automation is integrating a meta-learning layer for setting all hyper-parameters using techniques such as Bayesian optimization [46], which can take full advantage of cumulative data in the bioprocess lifecycle to systemically reduce uncertainty.

2. Elucidation of Machine Learning Strategies

2.1. Key Concepts

2.1.1. Brief Definition of Machine Learning in the Context of Bioprocess Engineering

Machine learning is a field of computer science and statistics that deals with data-driven modeling and algorithms. It is a new form of computational statistics applicable when no explicit mathematical description of relationships between data is known from theory. Machine learning has been particularly successful in domains where large amounts of data with a complex structure can be aggregated.

Machine learning approaches can be classified according to different criteria, here we mention the basic paradigms of supervised, unsupervised, and reinforcement learning, see an illustration in Figure 2. The review will mainly deal with supervised learning and reinforcement learning under a perspective of their applicability in biochemical engineering. Interested readers can refer to many complementary references on machine learning domains [47, 48, 49, 50].

The general principles of machine learning are best explained for supervised learning, that is, in fact, regression and classification. A data source would produce pairs of inputs $x \in \mathcal{X}$ and outputs $y \in \mathcal{Y}$ coming from some unknown distribution on $\mathcal{X} \times \mathcal{Y}$. Usually, x is a vector of features, and y is a vector of real numbers (regression) or class labels (classification).

The data obtained until some time would be collected in a dataset $D = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_N, y_N)\}$. The goal of supervised learning is to learn a predictive model from such a dataset, namely a map $f : \mathcal{X} \to \mathcal{Y}, x \mapsto f(x)$ that predicts or estimates y given x. Such maps usually are obtained by specifying the parameters of a parametrized family of maps $f_\theta : \mathcal{X} \to \mathcal{Y}$ for θ in some

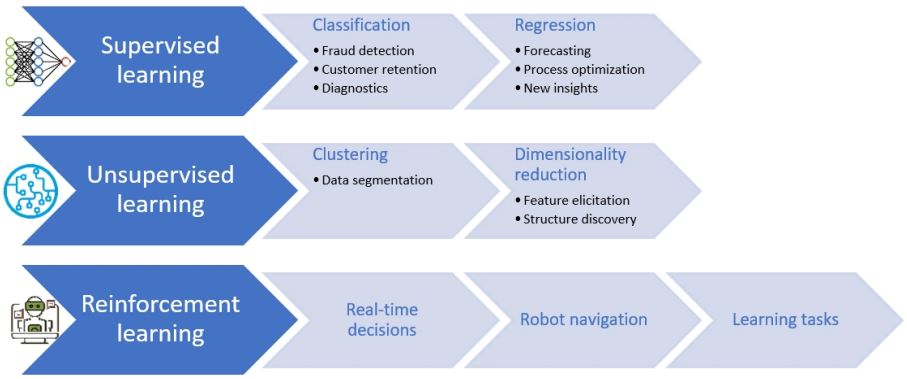


Figure 2: Sub-fields of machine learning.

parameter space. The quality of a prediction $\hat{y}$ is measured by a loss function $\ell(y, \hat{y})$.

Models are usually trained on a dataset by minimizing the so-called empirical risk, which is the average loss on a given 'training' dataset

$$\mathcal{L}(\theta) \stackrel{\text{def}}{=} \frac{1}{N} \sum_{i=1}^{N} \ell(y_i, f_{\boldsymbol{\theta}}(\mathbf{x}_i)) \ . \tag{1}$$

The minimization problem to be solved is

$$\hat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta}}{\operatorname{argmin}}\, \mathcal{L}(\boldsymbol{\theta}) = \underset{\boldsymbol{\theta}}{\operatorname{argmin}}\, \frac{1}{N} \sum_{i=1}^{N} \ell\left(y_i, f_{\boldsymbol{\theta}}(\mathbf{x}_i)\right) \ . \tag{2}$$

With a quadratic loss function (the negative log-likelihood of Gaussian noise on the y), this is the classical least squares fitting of the parameters of a parametric regression model.

The purpose of a predictive model is, of course to give reliable predictions for *new* inputs, that is to generalize well. The best model in the family would be the model with the lowest *expected prediction error* (EPE), that is the lowest average loss over all possible data pairs $\mathbb{E}_{x,y}\ell(y, f_\theta(x))$ in the limit for infinite sample size.

However, the empirical risk for the optimized parameters

$$\mathcal{L} \stackrel{\text{def}}{=} \frac{1}{N} \sum_{i=1}^{N} \ell(y_i, f_{\hat{\boldsymbol{\theta}}}(\mathbf{x}_i)) \tag{3}$$

on the training dataset is often much smaller than the expected prediction error, namely when the model is overfitting. A small empirical risk on a training set (or a good fit) is no reason to rely on a predictive model.

Overfitting can be an issue even in mechanistic models with only a few parameters and lots of data, but it always is in machine learning, as an ML model

would typically use a considerable number of parameters to be flexible enough for modeling an unknown relationship. Therefore it is essential to estimate the EPE reasonably to assess a model.

Cross-validation is the method of choice to obtain reliable estimates of the EPE under fairly general conditions. One divides the dataset into a training dataset and a test dataset $D = D_{\text{train}} \cup D_{\text{test}}$. The model is then trained on D_{train}, while the average loss is reported on D_{test}, that is on samples never seen during training. Cross-validation simulates applying the model to new data. That's the basic idea, although it is usually advisable to average this procedure over different splits (N-fold cross-validation); for more sophisticated versions of cross-validation and best practices of data partitioning, see [49, 51, 52].

Apparently, more interdisciplinary communication would be necessary to make the notion of model assessment by cross-validation well understood and accepted outside ML [53]. There are some caveats: With small amounts of available data, cross-validation may be unfeasible. Furthermore, the structure of datasets can make partitioning a complex task quite tricky. Different replicates of an experiment should, for example, all end up in the train or all in the test partition; otherwise, the test loss can underestimate the expected prediction error. Doing cross-validation right is of the essence and would usually require communication between a domain expert and a machine learning expert.

Machine learning models are constructed with different architectural choices and varying techniques of regularization to avoid overfitting, leading to a family of parametrized models instead of just one model. The family is parametrized by the so-called hyperparameters that control architecture, regularization, training, etc. Each model defined by a specific set of hyperparameters has trainable parameters to fit the available data.

We now encounter the classical task of model discrimination and selection in a new guise. Finding optimal hyperparameters means selecting the model in the family with the best predictive performance, that is the lowest expected prediction error. The estimation of the EPE used in model selection should rely on a validation dataset $D_{\text{validation}}$ different from the test set D_{test} used for reporting the EPE of the selected model, otherwise the latter may be grossly underestimated. Thus e would need to partition the dataset in three disjoint datasets D_{train}, $D_{\text{validation}}$ and D_{test}.

Hyperparameter tuning [54, 55] is an essential part of machine learning, models rarely work convincingly out of the box, which should not come as a surprise as even simple regularized regression methods like Ridge Regression and Lasso require tuning the regularization parameter in order to pay off. Cross-validation comes with a high computational burden, which cannot be avoided. But it would not require an additional mental effort from scientists that want to apply ML, once frameworks automate this procedure.

In machine learning, a baseline is any simple algorithm, with or without learnable parameters, for solving a task, usually based on a heuristic experience, randomization, or elementary summary statistics [56, 57]. It is an important reminder when tackling new domains with machine learning techniques, such as bioengineering and bioprocessing. Before attempting to develop more sophis-

ticated models, obtaining existing simple baselines is more critical. It takes a simple hypothesis that is consistent with the available data. All models and algorithms already established in the domain serve as baselines. A sophisticated time series forecasting model for a bioreactor must be measured against existing reactor models to assess it.

2.1.2. When to Use Machine Learning?

Machine learning is now extensively applied and is even a driving force of discovery all over science, but it is not a panacea. The notable successes come at the price of the less apparent failures. Quite a few things can go wrong if not heeded, leading to the risk of misinterpretations, over-optimistic results, and models that fail to generalize. Recommendations and best practices for the use of machine learning in science [58], or more specifically computational biology and biology [59, 60] can help to avoid these mistakes and to save time and money. When should we consider deploying and investing in machine learning?

When it is cost-effective. It isn't easy to know in advance when the application of machine learning will lead to a cost reduction in bioprocess engineering. Decisions for investments would ideally be based on comparing the cost of alternatives [61, 62]. Improved models and algorithms through machine learning may reduce the cost and experimental the burden of developing and scaling a bioprocess and possibly lead to more cost efficient control of bioreactors at the industrial scale. But applying machine learning would also create costs, namely for data management and curation, development of models, expensive hardware or cloud computing for training the models, building and running the infrastructure to deploy and monitor the complete machine learning project life cycle which includes further continuous monitoring of the model, collecting new data, and keeping the model up to date. However, we believe that given the range of problems that might be solved by machine learning investment in such infrastructure seems reasonable.

When needing regression or classification with enough data . Whenever a problem in the domain can be formalized as one of the fundamental supervised machine learning problems (regression, classification), and when there is a relatively large amount of aggregated legacy data or the acquirement of new data relatively cheap, machine learning can make valuable contributions. It depends on the model's quality already in use and whether a significant improvement is to be expected. So it is, above all, the expert's knowledge of the deficiencies of the models and algorithms they use that points to practical applications of machine learning.

As mentioned above in Section 2.1.1, good data management and curation is an enabler for machine learning. Data with complete metadata annotations allow for aggregation of data across different situations, e.g., bioprocess data for different strains, scales, and reactors. And that is the situation where adequate machine learning models are the most advantageous.

When the data consists of images or videos. Machine learning models based on convolutional neural networks have consistently beaten all previous methods for image classification, object detection, image segmentation, and other image-related tasks. Automatic analysis of images allows turning imaging devices into soft sensors. For example, microscopic images of samples from bacterial fermentation give information on the heterogeneity of the population, inclusion bodies, and shapes of bacteria. Building an automatic image analysis pipeline for this purpose is relatively easy, using an open-source library for bacterial image analysis [63]. Manually annotated training data would still be mandatory.

2.1.3. Machine Learning Project Life Cycle

Machine learning is implemented as a process containing chained stages: Data cleaning techniques, data transformation or normalization, hyperparameter optimization using cross-validation, model training, and validation, deployment, monitoring, and maintenance, which includes updating trained models (and possibly also the hyperparameters) when new data come in or possibly querying new data to improve the model (active learning) [64, 65, 66, 67].

When applying machine learning to bioprocess engineering the specific problem has to be defined and then be formalized as a machine learning task or possibly as a composition of several machine-learning tasks [68].

If, for example, an application problem can be framed as a supervised learning problem, we have to specify which output quantity should be inferred from which input quantity, what is the relevant loss function to evaluate model predictions, what kind and the amount of available data. It might also be necessary to specify requirements on train-test-splits.

After these specifications, machine learning pursues a well-defined goal. For supervised learning, the procedure would try to obtain a model with the lowest expected prediction error among all candidates. If the loss function indeed reflects the requirement of the engineers, the model should be helpful for them.

However, the goal of a bioprocess engineer is more general, namely, to achieve a technological objective with available resources. So the engineer has to care about the cost of lab work, monetary investment, and data collection necessary for a successful solution of the narrower machine learning task.

In general, the life cycle of a machine learning project, illustrated in Figure 3, consists of the following stages: 1) bioengineering, isolating a problem and rephrasing it as equivalent machine learning tasks, 2) data engineering, e.g., data collection and preparation, feature engineering, 3) machine learning engineering, e.g., model training, model evaluation and tuning, model deployment, 4) machine learning in production, e.g., model serving, model monitoring, and maintenance [69, 70, 71].

2.2. Active Learning

2.2.1. What is Active Learning?

ML models are usually trained on large corpora of data created by a potentially unknown process. As stated in Section 2.1.1, supervised machine learning

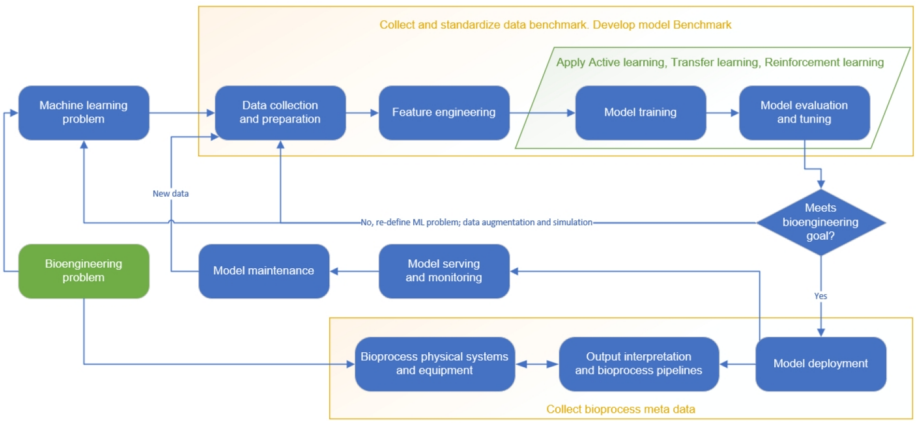


Figure 3: Machine learning bioengineering project life cycle.

solves a regression or classification problem that requires the data to be given or to be representable as predictors x and target values or labels y. In some contexts, for example, in image classification, the target values are also known as annotations. The term refers to a situation where a large data set of predictor data x_i is available or continuously generated, but human domain experts are needed to annotate, i.e. provide the label y_i for some of the data points x_i, an expensive and time-consuming procedure. In other contexts, the acquisition of new data is inherently expensive, for example, if the data are obtained by chemical, biochemical or biological experiments or complex computer simulations. In all cases the cost of data acquisition and a limited budget force ML practitioners to select which data should be acquired or annotated to be most informative for the model. This process is called Active Learning (AL).

The active learning task to query new data beneficially can be seen as a generalization of the classical problem of (sequential) optimal experimental design (OED). Experimental designs can be chosen optimally for different purposes, e.g., to discriminate model hypotheses, estimate model parameters, or predict at specific points.

Since active learning methods are incremental (selecting the next data point based on the current labeled data and model), they often require a so-called seed set. This is a small set of labeled data $(x_i, y_i)_{i \in I_l}$ used to train the initial model and calibrate the AL method. For a graphical overview of the AL cycle see Figure 4. The labeled set is initially comprised of the seed set. Each cycle adds one or more data points to the labeled pool.

2.2.2. Different Sampling Scenarios

One distinguishes three scenarios as to the way the next data point is sampled.

Stream-based Selective Sampling Data x_i is presented in a stream, e.g., images arriving from a camera, and a cost is incurred for acquiring target

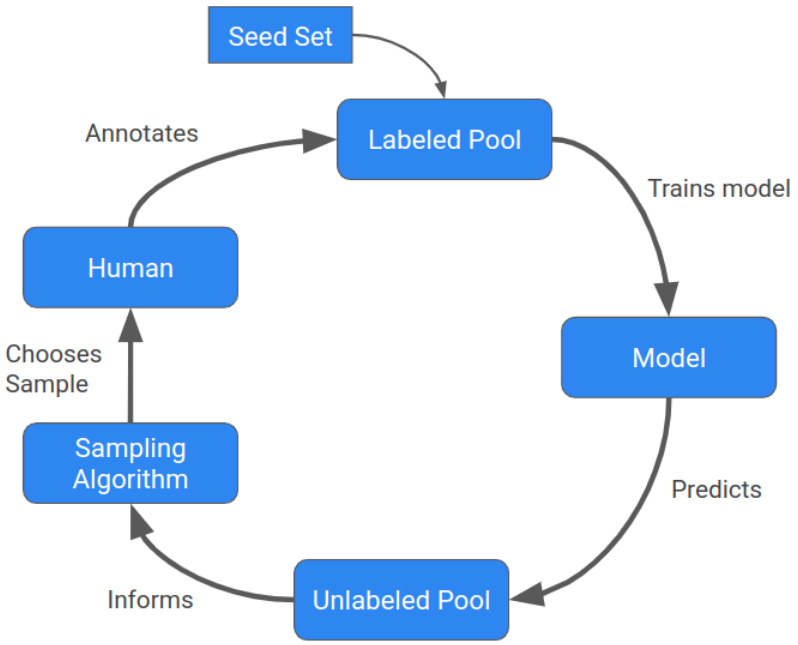


Figure 4: The active learning cycle. The seed set contains a small number of labeled samples. Each cycle adds one or more data points to the labeled pool. Repeated until some stopping criterion is met (usually a budget constraint or performance threshold).

values y_i, e.g., labels by a human expert. The active learning algorithm has to decide on a case-to-case basis if a sample is to be labelled or not.

Pool-based Sampling A large pool (or a subset thereof) of unlabelled instances $(x_i)_{i \in I_u}$ is given. The AL algorithm has to pick one or more data points from the pool, which are to be labelled y_i.

Query Synthesis The AL algorithm uses the current labelled data $(x_i, y_i)_I$ to synthesize new cases x_i for which the target value should be queried. This does not rely on existing unlabelled data as in the previous two scenarios but creates new data. These data might e.g. correspond to a real-world experiment described by parameters x_i, the outcome of which becomes the target value y_i.

For all three sampling scenarios, there are potential applications in chemical engineering and bio-engineering. For tasks like anomaly detection in processes, one would have a pool of legacy data with a partial annotation of anomaly, an incoming stream of new data without annotation and would choose the cases, for which to require an expert annotation [72]. However, in the query synthesis case, is arguably the most important in the biotechnological context: New experiments are designed in order to produce the most informative data.

One should be aware that the distribution of the queries can be expected to differ from the distribution of the ordinary data-generating process. If, for example, AL/OED is used to optimize a bioreactor model for later use in the control of the reactor, it is entirely possible that the regular operating regime is different from the data distribution that creates a strong predictive model.

2.2.3. Querying the Most Informative Data

A good intuition is that AL and OED will query the most informative data for the purpose at hand, although not all algorithms define 'most informative' in the same way.

The expected information gain (EIG), which is the expected reduction of entropy by the queries, is an ideal Bayesian utility function for AL to optimize, however estimating the EIG is computationally very challenging. Most of the AL methods below use a different objective, but a unified view ([73]) is possible, which explains their relation to the EIG. All the following methods use different proxies of EIG to select 'informative' queries.

Uncertainty Sampling Beginning with [74] this is a widely used class of methods with different underlying estimators of uncertainty. While working well in some instances, such algorithms can over-sample regions of the space $\mathcal{X}$ where noise dominates.

The most prevalent measurement of uncertainty is the Shannon entropy applied to the output of a classification model (see Fig. 5 (c)). If the model assigns a high probability to one class and low probability to all others, the entropy between those probabilities is low. If the model assigns an equal probability to all classes (the model is uncertain) the entropy is high. A sample is considered informative if it produces high entropy in the model's output. One might e.g. design a fermentation experiment with a feeding profile, for which a given model has highest uncertainty in its predictions.

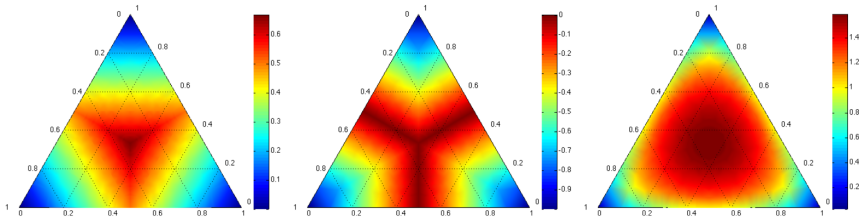


Figure 5: Different uncertainty measures applied to a 3-way classification problem (Ref. [75] Fig. 5). Each corner represents one class. Each point within the triangle represents the prediction of a model given an arbitrary data point. Each point in the triangle indicates the assigned probability to each class by its position. The color indicates the amount of estimated uncertainty, where red and blue indicate high and low uncertainty respectively.

Reducing the version space Several approaches can be described as reducing the space of hypotheses compatible with the data, the so-called version space. These approaches maintain an ensemble of many models rather than just one. Each model represents one hypothesis about the available data (see Fig. 6). An informative sample is considered one that produces high disagreement between the hypotheses/models, forcing wrong models to be dropped or updated. Repeating this process will push all models of the ensemble to converge to the "true" hypothesis. Algorithms that fall in this class are Query by committee and Query by disagreement. This method was implemented by [76] and applied to image classification tasks. The approach is related to a classical design of experiments for

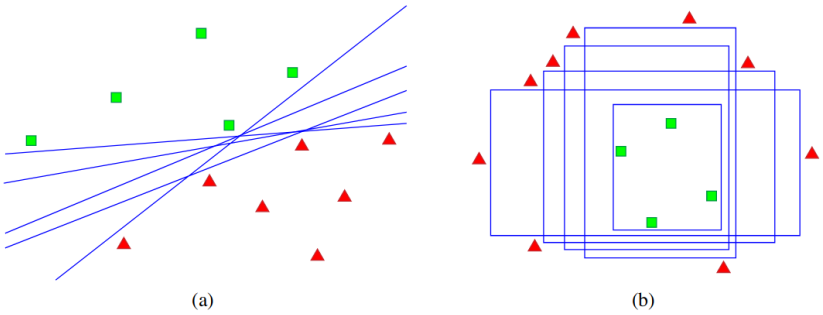


Figure 6: Different hypotheses of classification models for two types of classifiers (Ref. [75] Fig. 6). Each line or box respectively represents one correct hypothesis about the given data. An informative sample would be any new point that contradicts one or more of these hypotheses.

model discrimination.

Expected error reduction Another proxy of EIG is an estimate of the expected error after seeing a new query. Since all ML approaches rely on some error function to optimize their models, one can estimate the expected improvement in these functions given a selected data point from the unlabeled set. Points associated with a more considerable error reduction are considered more informative. This method has been successfully applied in classification ([77]). Since estimating the error reduction is computationally very expensive, [78] trained a regression model to predict the error reduction and applied it to 3D Electron Microscopy (Striatum) and MRI brain scans (BRATS).

Variance reduction That is where the classical 'alphabetical' frequentist OED criteria can be placed. In maximum likelihood estimation, the covariance of the parameter estimates is bounded below by the inverse of the Fisher information matrix (Cramér-Rao). Different criteria that control the eigenvalues of the Fisher information are used to find an optimal design (D-optimal determinant, A-optimal trace, etc.). Controlling the variance

of the parameter estimates has an impact on the predictive variance of the model but is still a different problem. A-optimal design, however, implies minimizing a lower bound on the predictive variance. Calculating and inverting the Fisher information metrics for all parameters of a large ANN is prohibitive, but recently this approach has been applied to just the last layer of an ANN [79]. It should be noted the Crámer-Rao lower bound may underestimate the true variance, and that the variance can be a poor descriptor for non-Gaussian distributions.

Minimizing the EIG Direct estimation of the EIG has usually been considered an intractable problem, but recently useful (sharp) upper and variational lower bounds have been discovered and exploited for Bayesian Optimal Experimental Design (BOED) ([80],[81],[82], [83], [84], [85]). These promising approaches still remain to be tested in the context of bioprocess engineering.

None of these active learning techniques natively distinguish between epistemic uncertainty, caused by modelling errors, and uncertainty caused by the variability of different sensors in the data collection process ('aleatoric'). Most machine learning models do not consider this difference during the modelling and training process. Both kinds of uncertainties are absorbed into the model's prediction output, which can lead to undesirable outcomes of the active learning strategies, e.g. uncertainty sampling can end up sampling cases again and again for which model predictions are irreducibly uncertain.

2.2.4. Learning How to Active(ly) Learn

There are three issues to raise with the previously mentioned methods.
(i) Most design criteria, even the theoretically sound ones, do not directly improve the utility of the predictions for the final purpose.
Increasing a model's information content or generalization capabilities is excellent, but the exact relation to a specific prediction task or decision problem is not apparent. Therefore it would be advisable to *learn* an active learning strategy from data for the final task, end-to-end. It directly connects a model's performance on a given task to the selected queries.
(ii) If a new set of queries or experimental designs are selected each time, a complex nonlinear optimization problem has to be solved. This might require more time. In a real-time setting where, i.e., experiments have to be redesigned based on new incoming data, the sampling process also needs to be fast (iii) All methods discussed so far rely on heuristics to select their samples. These heuristics only use a limited subset of the available information and do so in a static fashion that does not adapt to the presented data.

All three issues can be addressed when recasting the problem as reinforcement learning (RL) (see section 2.4) by parametrizing a policy that selects or generates the next samples. This policy is trained to maximize a reward function that should reflect the value of selected samples with respect to the downstream task of the model. The most common examples of such reward functions are

the induced increase in validation accuracy [86, 87], or negative validation loss [88] of the model after adding a selected point to the labelled set. Such models are typically trained (or at least pre-trained) off-line, either making use of a large amount of existing data or of a simulator. Crucially, the online application (and possibly re-training) of the policy to generate new data is straightforward (solving (ii)). In this setting, the policy is trained end-to-end concerning the final use of the prediction model, thus avoiding a possible mismatch between the optimization objective and the final use case (solving (i)). Finally, the policy is usually represented by an ANN, so it can incorporate large quantities of information and data and dynamically learn how to utilize them (solving (iii)). This includes summary statistics about the unlabeled pool ([86]) or additional information about the model prediction and confidence ([87]).

This makes 'learning to actively learn' one the most promising approaches for AL [87, 88, 86], [89]. Some of the recent BOED methods mentioned above ([80], [85]) are policy based, too.

If no sufficient legacy records of selected samples and improvement in model performance are available, one needs to employ more complex reinforcement learning approaches. The authors of [86] use model-based RL to solve the problem. The agent is primarily trained within a simulation of the AL process and further improved based on the limited real-world data.

The interested reader can refer to section 2.4 or directly to [90] for an introduction to reinforcement learning.

2.2.5. A Special Case

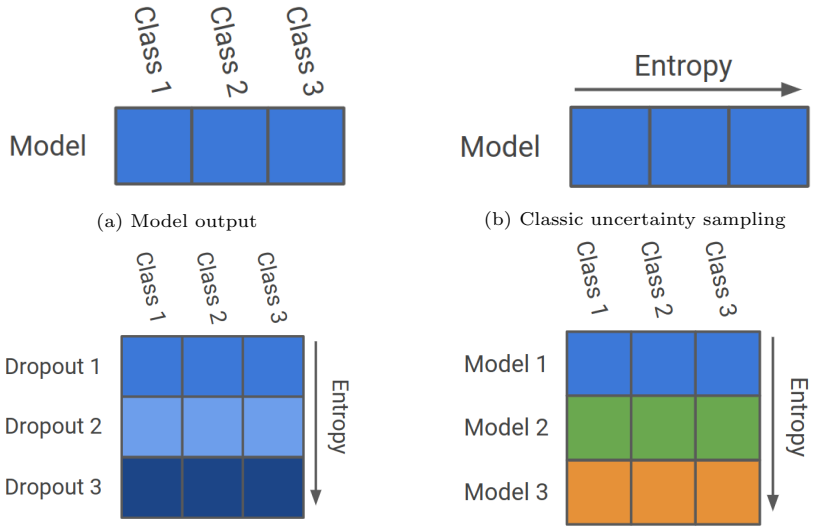
All previously described AL methods are done by analytical and probabilistic models. However, there are also discrete problems amenable to logical analysis in the application domain of this article. The most prominent example is the Robot Scientist Ada [91], an automatic system that designs experiments to determine the gene function of yeast using deletion mutations and auxotrophic growth experiments. The active learning strategy of Adam can be formally understood along the previously sketched lines as reducing the hypothesis space by minimizing a probabilistic objective function (expected cost [92]). However, the gene network to be deciphered is treated as a logical problem, and a central ingredient of the algorithm is automatic logical reasoning. This is an interesting case that recalls the ambiguous meaning of 'artificial intelligence, which can refer to logical reasoning systems and to statistical learning models alike.

In real-world scenarios, logical reasoning about complex encoded information and statistical learning on collected data can both play a role, though the great successes of machine learning of the latter kind have recently eclipsed the former.

2.2.6. Spotlight: Uncertainty Quantification

This section aims to deepen our understanding of uncertainty sampling, as it is the most straightforward and most used implementation of active learning. As stated above, uncertainty sampling aims to measure the model's confidence for a given prediction and uses this as a proxy for the EIG. The more uncertain a model is, the more informative this sample is considered, and following

that, the more useful this sample will be when meaningfully annotated. Figure 7 compares different setups for uncertainty sampling with entropy as an uncertainty measure. We will consider a 3-way classification problem so that the model will assign one probability for each class (subfigure (a)). The classic uncertainty sampling will compute the entropy across the three classes (subfigure (b)). Query-by-Committee was previously introduced as an alternative to un-



(a) Model output

(b) Classic uncertainty sampling

(c) Uncertainty sampling with MC Dropout

(d) Classic uncertainty with ensembles

Figure 7: Comparison of different setups for uncertainty sampling with entropy for a 3-way classification problem

certainty sampling since it was derived from a different theoretical motivation. However, the measurement of uncertainty in both frameworks is very similar. Since Query-by-Committee algorithms maintain an ensemble of many models, uncertainty can be measured on a per-class basis (across models) rather than per model (subfigure (d)). To assign a scalar value to each sample, the per-class uncertainties are usually summed ([76]). Since maintaining and updating many ANNs is computationally very expensive, some methods try to simulate an ensemble by using an approach called Monte-Carlo Dropout (MC Dropout) ([93]). For MC Dropout, only a single ANN with Dropout-Layers is trained. During prediction, where a single forward pass with a dropout rate of 0 is usually done, MC Dropout performs multiple forward passes with non-zero dropout, resulting in slightly different versions of the prediction. Treating each forward pass as a separate model in an ensemble, the same Query-by-Committee algorithm can be applied ([94], [76]) (subfigure (c)).

2.3. Transfer Learning

Humans do not learn to perform tasks independently but always use previously acquired knowledge and skills when dealing with a new task. It is an important challenge for machine learning to find ways that mimic this human connectivity of knowledge and allow it to reuse information from other contexts in a new one. Machine learning approaches that try to *reuse* (parts of) models trained for one task in a new task are known as *transfer learning* [95, 96, 97, 98, 99]. There are other notions to be distinguished from transfer learning, which also imply 'learning from other cases', namely *meta learning* and *multi-task learning*. Meta-learning applies when a task refers to datasets drawn from a distribution of datasets. For example, each dataset collects bioprocess data for a specific strain of *E.coli*. The goal is to devise models that train faster on a new dataset, using information from other datasets. Multi-task learning, on the other hand, is closely related to transfer learning, but it deals with *simultaneously* training models for different tasks instead of *reusing* pre-trained models.

Most applications of transfer learning refer to learning tasks where the inputs x have the same data type and can be assumed to be similar in some sense, e.g., images of a specific format, protein amino acid sequences, and fermentation data with the same observables in the same form. The outputs y, however, can be particular to the different tasks.

Reusing models trained for different tasks can be a very cheap way to overcome the 'small data problem'.

Neural networks for image classification trained on very large image datasets [100, 101] have led to the arguably most successful applications of transfer learning [102]. Such models process the original image through subsequent stages, each stage producing a new representation. These representations capture image features, some very general and helpful outside the original training context, some very specific to the initial training task. For a new task on a small dataset, one can use a part of the trained network as a component of a new model and then train the model on the new data. It has been successfully applied to many different image domains (medical histology, plant images, etc.) [103, 104, 105, 106, 107].

A biochemical and biotechnological use case of transfer learning that is slowly unfolding its potential is the prediction of protein properties from the underlying sequences. In the very large protein libraries, some protein properties are more frequently available than others. Most proteins are equipped with class labels in a protein classification, fractions of the proteins have 3d structures, enzymatic activities, physicochemical properties attached. A model trained for predicting some of the frequently available properties or for an unsupervised task on all available protein sequences may learn internal representations that are also useful for other tasks related to rarely available properties [108, 109, 110, 111, 112, 113]. A regression model on a low-dimensional representation may be trainable with only a few examples, whereas any model that directly works on the high-dimensional space of amino acid sequences need large

amounts of data. If protein properties prediction models are good enough at ranking potential protein variants they can speed up directed evolution [112].

Transfer learning has also been applied to modeling lutein production by microalgae [6]. For one microalga species, a comparatively large amount of published data was available, and there were less data for a second microalga with a similar growth behavior. Models were then trained on the data for one species and then transferred to models for the second species. Some data augmentation was done in both cases to improve training.

The particularities of transfer learning are presented in Figure 8.

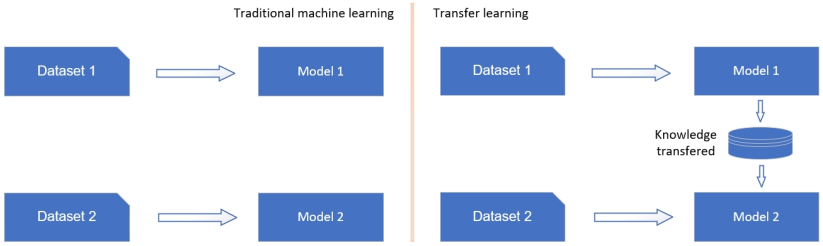


Figure 8: Difference between traditional machine learning and transfer learning.

When training a model with a pre-trained part one can decide which of the inherited parameters will be frozen and will be retrained with the new model. Thus the pre-trained model either serves as a feature extractor [114, 115], see Figure 10, or as an initializer [116], see Figure 9.

In transfer learning, the learning rate and number of training epochs correspond to a trade-off between the influence of the data from the original domain and the influence of the new data [117]. The optimal amount of training and the the best architecture for the task has, as always, to be determined by cross-validation.

2.4. Reinforcement Learning

Reinforcement learning (RL) is one of the main branches of machine learning. While the supervised and unsupervised learning methods learn the model from a given data set, RL methods learn to act. In other words, the outcome of the RL is the optimal decision rule for a given state, also referred to as 'policy' given an objective [90]. RL generally performs the following three-step procedure iteratively: data generation, performance evaluation, and policy improvement. By interacting with the dynamic systems according to the policy, the RL agent receives the data consisting of states, actions, and rewards. The data is used as a reinforcement signal that evaluates the performance of the policy. The policy is improved based on performance evaluation with various types of optimization methods. The procedure is usually designed to be stochastic, not only to act against the uncertain systems but also to add exploratory actions to the system to prevent trainable machine learning models from being overfitted [118]. It addresses the trade-off between exploration and exploration explicitly.

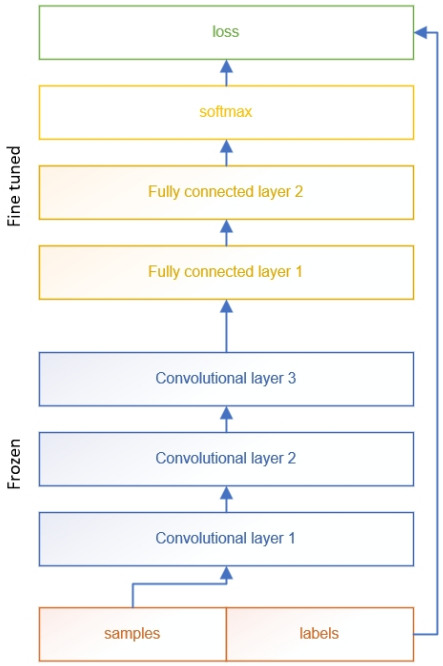


Figure 9: Frozen (no update during training) and fine-tuned (update during training) layers.

RL is deeply connected with process control in the sense that RL solves sequential decision-making problems [119]. RL has several potential advantages over the standard approaches of bioprocess control that use mechanistic models and mathematical programming. First, RL is flexible to work with varying levels of mechanical knowledge and structure of the systems [120, 121, 122]. Model-free is a special characteristic that distinguishes RL from other process control methods, and the reinforcement signal is solely used for policy improvement. Therefore, model-free RL can handle (1) hybrid systems consisting of mixed continuous and discrete states, actions, and events, (2) problems with various objective functions encompassing tracking control, economic optimization, and experimental design, and (3) model uncertainties that are not restricted to Gaussian distribution. This flexibility is an appealing characteristic for bioprocess control, and optimization [123], because biological models are often challenging to build, and biological systems have a considerable level of uncertainties. Recent advances in statistical machine learning enable feature analysis of the raw sensory-level data by using deep neural networks and the implementation of various information-theoretic techniques. Synthesis with a deep learning framework, deep RL (DRL) has successfully achieved a scale-up of RL methods to high-dimensional problems, showing remarkable performances in various applications such as process scheduling, reaction mechanism, fluid dynamics, robotics, autonomous driving, etc. [124, 125, 126, 127, 128, 129, 130].

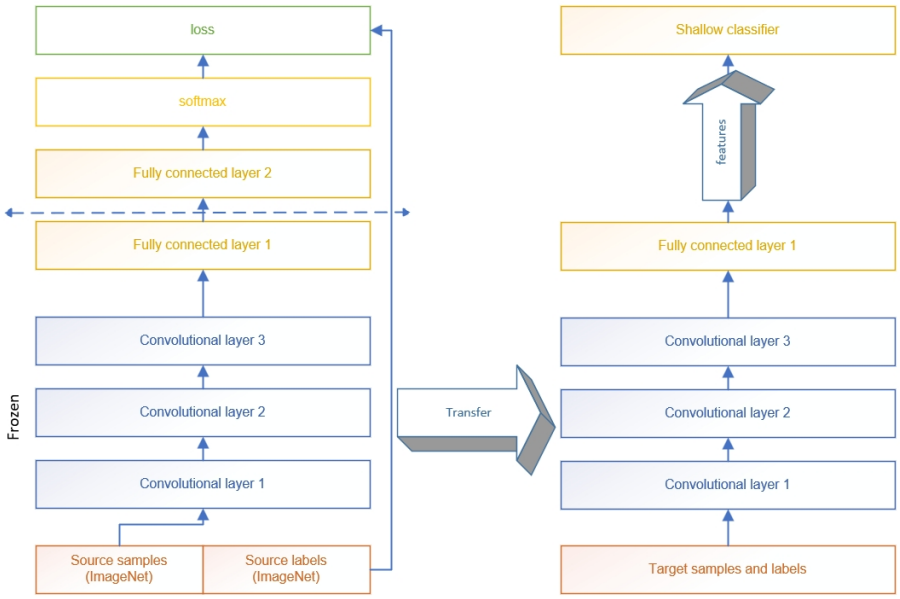


Figure 10: Initial layers as feature extractors.

The RL's second advantage is that most of the computation is done offline. In contrast, the conventional mathematical programming approaches need consistent re-planning, which can lead to exorbitant online computational demand. Because a single model cannot perfectly characterize the complexity of the metabolism, bioprocesses have to be operated in a closed-loop manner, adapting the model to the most recent experimental data [131]. However, mathematical programming-based approaches such as model predictive control (MPC) cannot match the online computation limit when the complexity is high due to the combination of the model, operating constraints, and uncertainties. Several researchers have focused on the RL framework as a complementary method [132, 133, 134]. It is the nature of RL that the policy is obtained, essentially the closed-loop feedback rule concerning states of the system. An end-to-end closed-loop operation can be achieved if the RL is applied to industrial bioprocesses using massive historical raw data in an offline environment.

Motivated by these advantages, several pioneering pieces of work for bioprocess control first appeared in [135, 136, 137]. In these studies, the RL methods use the lookup table that measures the optimality (e.g., 'cost-to-go' function or Q-function) with respect to the discretized state and action space. [135] used a fuzzy lookup-table guided by expert knowledge in the frame of a Q-learning algorithm. It showed that the RL could achieve near-optimal performance for batch process control. [137] combined the fuzzy rule with the Q-learning method to determine the gains of a PID controller for a tracking problem of the fed-batch bioreactor. [136] solved a free-end problem for a fed-batch bioreactor using ap-

proximate dynamic programming, an analogous algorithm to the RL. The RL algorithm was tested under different initial conditions and showed optimal performance without additional recomputation. It is the first work that recognizes the merit of RL for the closed-loop operation in the presence of disturbances.

Recent works incorporate DRL methods, which allow for an extension to the optimization under the continuous state and action space [138, 139, 36]. In [138], partially supervised RL was used to solve a tracking problem of a yeast fermentation problem. Neural networks that map the state and setpoint with the control input were trained and refined using RL. A DRL algorithm, asynchronous advantage actor-critic (A3C), was incorporated into the biomass maximization problem of a fed-batch bioreactor [139]. [36] utilized the policy gradient method for the optimization and recurrent neural networks (RNN) to approximate the policy function. The RL method was performed preliminary using offline data, and the policy was further trained in the online implementation.

The main drawback of model-fee RL is that it is notoriously difficult to use due to the sensitivity to hyperparameters, intractable data requirement, and optimistic estimation of the Q-function values [140, 141, 121]. Even for the optimal control of the most straightforward linear system with the quadratic objective function, model-free RL fails to achieve a reliable solution compared to the standard linear quadratic regulator algorithm [142]. It limits the actual applications to the control and optimization of the real bioprocesses. Model-based RL can help solve the issue by using the mechanistic model as a simulator for the offline training or utilizing the model equations' gradients to accelerate the training [143]. [144] suggested a two-stage optimal control for a closed-loop dynamic optimization of a fed-batch bioreactor. In the high-level optimizer, differential dynamic programming, a model-based RL that uses model gradient, is used for long-term planning with the economic objective of maximizing productivity. Whereas in the low-level controller, MPC is used for the short-term planning that tracks the high-level plan and, at the same time, rejects disturbances and model-plant mismatch. [145] proposed the integrated formulation of the MPC and RL, where the terminal cost function of the MPC is replaced by the value function obtained by the model-free method. The method was validated for the optimization of an industrial-scale penicillin bioreactor.

Another issue about RL is the consideration of critical process constraints for safety and keeping the operating condition within the valid domain [118]. A typical way to consider process constraints is to augment the amount of constraint violation as the penalization term to the objective. Using augmentation solely cannot always guarantee the feasibility of the exploration. In [146, 147], the probability of constraint violation was formulated as chance constraints, and an adaptive back-off approach was implemented to reduce the violation. Nevertheless, the trade-off between the original objective and constraint penalization is not uniquely determined, therefore adding another hyperparameter to the overall algorithm. It is not the case in conventional mathematical programming-based approaches such as MPC. In this regard, model-based RL can be helpful. [148] suggested Gaussian processes regression for the data-driven state-space

model and model-based RL for fed-batch fermentation processes. Mechanistic model-based RL approaches [144, 145] can naturally address constraints of fed-batch bioprocesses.

3. Current Integration of Machine Learning in Bioprocess Subfields

Machine learning has significantly contributed to the development of bioprocess engineering, but its application still needs to be improved, hampering the enormous potential for bioprocess automation. In this section, we summarize recent research across several important subfields of bioprocess systems, see Table 1, including bioreactor engineering [149], biodevices and biosensors [150, 151, 152, 153], biomaterials engineering [154, 155, 156], and metabolic engineering [24, 157, 158, 159]. Bioreactor engineering studies the correlation and effects between complex intrinsic factors that operate a bioreactor (e.g., contaminant concentrations, temperature, pH level, substrates, stirring and mixing duration, rate of nutrient inflow) and primary cellular metabolism (e.g., product synthesis and nutrient uptake). In this subfield of bioprocess engineering, machine learning has contributed to necessary research such as (1) estimating and predicting state variables at some points in the future (e.g., biomass concentration), (2) monitoring the factors that affect the bioreactor's performance, and (3) automating the bioprocess regarding safe operation and control purposes. The next subfield of bioprocess engineering is Biodevices and biosensors which machine learning implementation can be found in three primary areas: (1) optimization and control of microbial fuel cells, (2) development of soft and microfluidic sensors, and (3) chemical analysis of data collected from real-time measurements. Next, we also highlight the implementation of machine learning models to assist in the design and engineering of biomaterials in which biological engineers are interested in three primary research goals: (1) the efficient design and production of existing biological materials, (2) acceleration in developing new biological materials or improving the existing functions; and (3) quantification and automation of structural-functional relationships. The next subfield that the authors want to summarize in this review is metabolic engineering, in which the application of machine learning focuses on (1) completing the missing information to reconstruct the metabolic network, (2) identifying essential and influential enzymes and genes expression to product synthesis, and (3) exploiting the complex interactions between omics from fluxomics to genomics and growth kinetics of extracellular microorganisms.

As mentioned in sections earlier, we highlight the bioprocess tasks, experimental datasets, and machine-learning approaches within the subfields. Note that there are two fundamental goals when experts manipulate a bioprocess. The first goal is to make an accurate translation from bioprocess problems to appropriate machine learning tasks that can produce a correct prediction on the experimental datasets. The second goal is to ensure that anyone in the same laboratory or further researchers can reproduce the experiments. Therefore, we will also provide an in-depth investigation of the reproducibility capability of these mentioned research so that we either believe in the results or build

confidence in reproducing the whole experiments and improving further from there.

Many machine learning models have been utilized and integrated into bioprocess systems are support vector regression (SVR), partial least square regression (PLSR), multi-gene genetic programming (MGGP), artificial neural network (ANN), Gaussian process (GP), Convolutional neural network (CNN), nonlinear model predictive control (NMPC), hierarchical recurrent sensing network (HRSN), recurrent neural network (RNN), multilayer perceptron (MLP), relevant vector machine (RVM), accelerating genetic algorithm (AGA), K-nearest neighbors (KNN), support vector machine (SVM), convolutional neural network (CNN), and principal components analysis (PCA). The authors aim to introduce and explain only some of the above models again, which could be referred to many references [159, 160].

Subfield	Task	Dataset	Machine learning model	Reproducibility			Meta data	Ref.
				low	med.	high		
Bioreactor engineering	Predict the final antibody and lactate concentration.	Time series data of 134 temporal process parameters in four seed cultures (80L, 400L, 2000L, 12000L). Train-test ratio 90-10. 10-fold cross-validation. Data are not available.	SVR in LIBSVM. PLSR in SIMPLS	X			Yes	[161]
	Predict the performance of microbial fuel cell (MFC).	Data were taken from [162] Train-test ratio 80-20. Data are not available.	MGGP in MATLAB R2010b using GPTIPS software. SVR in MATLAB R2010b using LS-SVM toolbox. ANN in statistical software JMP version 9 (1 hidden layer, 2-9 neurons in hidden layer).	X			Yes	[163]
	Simulate lutein bioproduction process control and optimization.	4 sets of data, each containing 12 datapoints. 50 replications of artificial datasets were produced. Train-test ratio 3/4-1/4. Data are not available.	ANN in pybrain library (2 hidden layers, 20 neurons per hidden layer).	X			Yes	[164]
	Predict the evolution of multivariate states for lutein production process.	No clear information about experimented data. Data are not available.	GP. Compare with 1 and 2 hidden layers ANN. Neurons per layer in {3,5,10,15,20,25}.	X			No	[165]

Table 1 continued from previous page

Subfield	Task	Dataset	Machine learning model	Reproducibility			Meta data	Ref.
				low	med.	high		
	Simulate the fed-batch production process for cyanobacterial C-phycocyanin.	Biomass concentration, nitrate concentration, and phycocyanin productionwere measured every 8 hours. The original dataset consists of 135 data points, plus 100 artificially generated data points. Data are not available.	ANN, no configuration given.	X			No	[166]
	Simulate the algal biomass growth and bisabolene production.	Data were taken from 40 different experimental scenarios on different 120L photobioreactors. Each scenario contains 9000 data points. Train-test ratio 70-30. Data are not available.	CNN, 1 input layer with 7 neurons, 2 hidden layers containing convolutional blocks, 1 output layer with 3 neurons.	X			Yes	[167]
	Compare 6 different GP-based NMPC models for finite horizon control.	Simulated dataset. Data are not available.	GP, no configuration given.	X			No	[168]
Biodevices and biosensors	Characterize microfluidic soft sensor.	Data were collected from two soft pressure sensors. The processed data are available on Github.	HRSN based on RNN. The model is provided on Github.		X		Yes	[169]
	Process higher-throughput Raman spectroscopy and molecular images.	1.5 million hyperspectral Raman images. The processed data are available on Github.	ResUNet. Applied transfer learning technique. The model is provided on Github.		X		Yes	[170]

Table 1 continued from previous page

Subfield	Task	Dataset	Machine learning model	Reproducibility			Meta data	Ref.
				low	med.	high		
	Diagnose anatomical ex-vivo eye tissue segments in the usage of Raman spectroscopy.	Data were collected from 11 separate enucleated eyes, consisting of 88 spectra scans per tissue segment. The processed data are available on Github.	Self optimising Kohonen index network (SKiNET). The model is provided on Github.		X		Yes	[171]
	Classify traumatic brain injury severity via Raman spectroscopy of the retina.	14400 retina tissue samples were collected from adult male mice. Train-test ratio 80-20. 10-fold cross-validation. Data are not available.	Self optimising Kohonen index network (SKiNET). The model is provided on Github.	X			Yes	[172]
	Predict bioelectricity production in microbial fuel cells.	Train-test ratio 70-30. Data are not available.	MLP with {2,3,4,5} neurons. The model is not available.	X			Yes	[173]
	Optimize the operation of multi variable microbial fuel cells.	Data are not available.	Combination of uniform design, RVM, and AGA. No configuration given.	X			No	[174]
Biomaterials engineering	Predict phase of high-entropy alloys.	Data were taken from [175]. Data are not available.	KNN with k = {1,2,...,10}, SVM in MATLAB. ANN (1 input layer with 5 neurons, 3 hidden layers with 5 neurons each, 1 output layer with 3 neurons).	X			No	[176]

Table 1 continued from previous page

Subfield	Task	Dataset	Machine learning model	Reproducibility			Meta data	Ref.
				low	med.	high		
	Predict mechanical functionality of protein networks from confocal microscopy imaging.	Data were generated in MATLAB R2019 consisting of 26 calculated structural features of 37 protein networks. 5-fold cross-validation. Data are not available.	Gradient boosting, random forest. The models are provided on Github.	X			No	[177]
	Predict the performance of metal-organic frameworks (MOFs).	3385 MOFs containing 41 distinct network topologies. Processed data are available.	ANN (1 hidden layer with 30 neurons). Experimental reproducibility has given on a dedicated website [178].		X		Yes	[179]
	Predict injection of microparticles through hypodermic needles	Train-validation-test ratio 60-20-20. Raw data and codes can be provided upon reasonable request. Licensing fees might be applied.	ANN (10 hidden layers) in MATLAB Deep Learning Toolbox. No configuration given.	X			Yes	[180]
	Detect amino acids with nanoporous single-layer molybdenum disulfide.	Raw data and codes can be provided upon reasonable request.	KNN, random forest, logistic regression. No configuration given.	X			No	[181]
	Classify cell shape phenotypes.	Data are not available.	SVM in MATLAB. No configuration given.	X			Yes	[182]
	Detect scattering effect in light-based 3D printing.	300 mask-structure pairs plus 600 pairs augmented. Data are not available.	Neural network with 14-layer CNN architecture combined with U-Net skip connections. The model is not available.	X			Yes	[183]
Metabolic engineering	Fill gaps in a metabolic network.	BiGG database [184]. Data are available on BoostGapFill's Github.	BoostGapFill open source tool.			X	Yes	[185]

Table 1 continued from previous page

Subfield	Task	Dataset	Machine learning model	Reproducibility			Meta data	Ref.
				low	med.	high		
	Identify specific enzymes limiting production in a pathway.	Data are not available.	PCA in MATLAB.	X			Yes	[186]
	Predict the bacterial central metabolism.	Data collected from 100 C-metabolic flux analysis papers. Data are available.	MFlux web-based platform. Source codes are available. MFlux applies SVM, KNN, decision tree.			X	Yes	[187]
	Predict essential genes in Escherichia coli metabolism.	4094 metabolic reaction-gene pairs. Several additional datasets from private providers and E. coli Gene Expression Database. Data are not available.	SVM. No configuration given.	X			Yes	[188]
	Selecting substrates that best expand an enzyme's promiscuity.	BRENDA online enzyme database [189].	SVM. Apply active learning approach.		X		Yes	[190]
	Propose a real-time optimization for the control of co-cultures within the continuous bioreactors.	Simulated data of continuous bioreactor, 24h duration, measurement every 5 min. Data is not available.	Neural fitted Q-learning, reinforcement learning. The models are provided on Github.	X			No	[191]
	Predict xylose consumption, biomass and xylitol production.	Datasets were collected every 6 h interval resulting in 340 data points (27 runs). Data are not available.	ANN of 5-10-2 topology in MATLAB Deep Learning Toolbox.	X			Yes	[192]

Table 1 continued from previous page

Subfield	Task	Dataset	Machine learning model	Reproducibility			Meta data	Ref.
				low	med.	high		
	Explore the bioretrosynthesis space in synthetic pathway design.	Golden dataset of 20 manually curated experimental pathways, 152 metabolic engineering projects. Data are available.	Applied reinforcement learning. The open source solution is provided on Github.			X	Yes	[193]
	Predict protein expression from promoter sequences.	675,000 constitutive and 327,000 inducible promoter sequences. Data are available.	CNN, no configuration given. The model is available on Github.			X	Yes	[194]

Table 1: Application of machine learning in various subfields of bioprocess engineering.

Table 1 shows us a huge problem with the information needed to ensure the reliability and reproducibility of the experiment. When we sorted by problem requirements, data, and machine learning models from a machine learning perspective, we witnessed more clearly the inadequacies and inconsistencies in the information presented in those published studies. Metadata about curation, provenance, and aggregation needs to be clearly described. The infrastructure condition and data engineering pipeline should be mentioned. Those papers do not record the model construction process or justification for design decisions. It leads to the acceptance of simplifications and assumptions about the system design, environmental context, biological or biochemical features, and other artifacts. Note that the required information will help avoid unnecessary recreation and model version control. Several data and models in bioengineering are conducted from a simulation process. It is the best practice if all simulation inputs and applied methods, initial conditions, numerical integration algorithms, seed values, and other emerged data should be carefully recorded. If any parameters and hyperparameters are estimated, share the estimation algorithms and the value ranges. As the results of our mentioned issues, Table 1 is not consistent in how it describes the case studies as in some cases, the programming language used is given but not in all. All columns in this figure should contain the same information to allow the readers to compare between studies. How can we collect the required information if they still need to be provided? It explicitly confirms the reproducibility problem we want to discuss in the paper.

4. Challenges and Future Research Directions

4.1. Challenge 1: Reproducibility Crisis

Machine learning is, to a considerable extent, an experimental science. As a result, the reproducibility of computational pipelines is of significant concern [195, 196, 197]. Machine learning experts have highlighted that the reproducibility of scientific results is a key element of science and credibility of conclusions made to the extent that they explicitly encourage replicating the experimental results of any published study [198, 199, 200, 201, 202]. In the nine major machine learning conferences, including NeurIPS, ICML, ICLR, ACL-IJCNLP, EMNLP, CVPR, ICCV, AAAI, and IJCAI, the criterion of reproducibility has been highly required in every peer-reviewed process and published research paper [203, 204]. To establish which algorithm is better for a learning task, it is an essential rule that any computational experiment for algorithm assessment should be carried out on the same datasets representing the task. This dataset must be publicly available or published together with the first paper addressing this task. The evaluation metrics will be calculated using the same formulas as the first published paper. In the case of using a new set of formulas, it is necessary to re-test the model in the first publication, applying the methods of optimal search for the participants on this new set of formulas. Take an example as follows, we have two algorithms to compare. Algorithm A is our development, and algorithm B is proposed by previous research. The comparison results depend on how much documentation is publicly made available. For

example, if we only have access to the written documents as published articles, we have to self-implement algorithm B and test it on the data we collect ourselves. In fact, there is practically no way to verify that we have implemented and configured the algorithm in precisely the same way as the original authors, especially the values used for hyperparameters. Thus, the more literature (articles, algorithmic code, and data) provided by the original authors, the easier it is for independent researchers to reproduce and demonstrate the published results that the claims made are valid. We proceed with the problem further regarding the above algorithms A and B. Suppose we want to test algorithm A on the same published data set. In that case, the question is whether we have to test algorithm B again to verify the correctness or accept the results reported as comparative results. This is a relevant question because newly proposed algorithms are often compared with published models developed by third parties without re-testing. However, one scenario exists when algorithm B compares itself with many previous algorithms, but the code is not publicly available. And instead, later researchers often take reported results to compare and accept as proven facts. In addition, independent research experts have found it difficult to obtain similar results when re-implementing complete experiments reported in the scientific literature if the values for some important parameters and hyperparameters are not given. Computer science, specifically machine learning, is in a favorable situation where identical empirical procedures can be followed using the same data sets. Although in this case, the biggest challenge is the lab, different hardware, and software where the experiments are conducted. Reproducibility is best demonstrated by applying algorithms A and B on the same data but for different laboratory, configurations to produce similar results and arrive at the same conclusions. Interest has grown not only in the machine learning community but also in bioengineering [205, 206, 207], biomedical engineering [208], biology [209], and genome editing [210] regarding the reproducibility of published scientific results.

However, the reproducibility requirement for biological systems is much more difficult because data are extracted from living organisms, chemicals, and organic interactions, e.g., proteins and strains of cells. In systems biology modeling, the issue of reproducibility involves a combination of not having FAIR experimental data and difficult-to-reproduce model fitting strategies due to missing parameters, initial conditions, and inconsistent model structure [211]. Even the biomass collected during experiments in the same laboratory, on the same bioreactor, differed by the time of year or collected by different technicians. This complicates efforts to apply approaches from the field of machine learning, where data is more stable and redundant. Furthermore, increasingly sophisticated bioengineering tools are making cell biology experiments more complex. The time to conduct biological experiments is also longer, leading to more complex reproducibility. Thorough validation can take months or even years to complete. That makes it difficult for laboratories that are not equipped with modern equipment to reproduce experimental conditions that more qualified laboratories have done. Instead, the biological sciences depend on other less reliable techniques for reproducing experiments, resulting in publications that are

less conditional on comparison with previous studies which makes data difficult to share or reuse.

According to a Nature survey of 1576 researchers, M. Baker points out that the scientific community has a general view that there is an ongoing reproducibility crisis [212]. Surveys have shown that more than 70% of researchers have tried and failed to reproduce other scientists' experiments, more than 50% have been unable to replicate their own experiments, and more than 30% believe in published results even though they acknowledge that published results may be wrong. Another interesting survey published on Molecular Systems Biology[1] [211] that the authors investigated 455 kinetic models of various biological processes. The authors concluded that 49% of the models could not be reproduced using the information provided in the manuscripts. They even proceeded with an effort by contacting and asking the authors of 455 published papers. And surprisingly, only 12% out of 49% could be reproduced. The plausible reasons for non-reproducibility include inconsistency in model structure, missing parameter values, missing initial concentration, and even unknown reasons. Many bioengineering professionals reuse machine learning as a complete implementation on a particular computing platform. However, another study even concluded that a machine learning platform does not guarantee computational reproducibility and that the test results generated from a machine learning platform cannot be trusted entirely [213].

4.2. Proposed Research 1: Promote a Culture of Inferential Reproducibility in Bioengineering

In addition to the techniques and methodologies proposed in machine learning [214, 215, 216], we need to change the culture regarding research reproducibility. We must encourage the practice of reproducibility and help subsequent researchers to enforce it as a cornerstone of science [217]. Reproducible models confer essential benefits because they are easier to understand, trust, modify and reuse. This facilitates our collaboration better and is more open, thus, attracting follow-up studies to construct multi-scale models of larger, more complex systems from the current results. We need to fund and encourage individuals and research groups to confirm (or sometimes disprove) the findings of others with reproducible results. We should not criticize studies whose findings cannot be confirmed. In contrast, our work attempts to replicate highly reproducible studies, even if the results are not precisely the same. Journals can even create a new criteria category for assessing which research supports or integrates research reproducibility. The study replication levels can be found in Figure 11.

The lowest level of reproducibility requires a research article and possible supplementary where the researchers should describe how bioprocess experiments have been conducted. Other metadata should also be available. However, the experimental codes and datasets, or the executable scripts can be missing.

[1] `https://www.ebi.ac.uk/biomodels/reproducibility`

Hence, by fulfilling those mentioned missing codes, datasets, and scripts, the reproducibility is improved to the Medium level. The High level requires basic machine learning in production where the programming environment, a platform of development, hosting and metadata are presented.

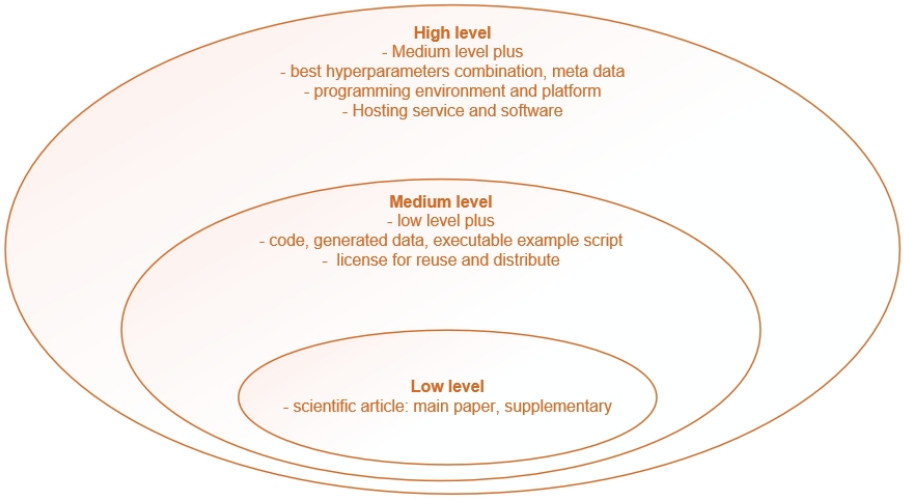


Figure 11: Reproducibility levels from the lowest, e.g., Low Level, to the highest, e.g., High Level. Each level requires some performance and proof.

4.3. Challenge 2: Benchmark Datasets and Evaluations of Bioengineering approaches

Within computer science, benchmarking is the development of guidelines and best practices. It contains three sub-fields: scientific machine learning benchmark, application benchmark, and system benchmark [218]. Application benchmark concerns the complete deployment of machine learning applications using various hardware and software settings. The benchmark evaluates the use of resources, e.g., file systems, software libraries and versions, hardware configuration, and scaling factor of computing capacity, that affect the time-to-solution of the application. System benchmark concentrates on the availability of a machine learning application in a broader environment. The system benchmark evaluates network throughput and the number of floating-point operations per second. These two benchmarking frameworks are not technically suitable for bioprocess engineering. This article focuses on the machine learning benchmark and how to promote it within bioprocess engineering research. The machine learning benchmark is much simpler and easy to implement as it requires two subjects: datasets and reference models Firstly, benchmark datasets which are the fundamental cornerstone of machine learning should be made available to the research community. The exact training, validation, and test sets are on which all the reference implementation must be based on. Secondly, proposed

approaches and state-of-the-art modeling applications will be developed over time and considered blueprints for further use on different benchmark datasets. Thirdly, an excellent overall design of the machine learning benchmark has fostered great boosters for research and discussion of corresponding areas: out-of-the-box downloading and usage, interoperability, and ease of customization [219, 220]. Bioprocess engineers must be able to understand the most suitable machine learning models by looking at the benchmarking performance on the equivalent datasets and types of investigated problems. More specifically, Bioprocess experts might refer a blossoming of benchmarks on neural networks and applications [221, 222, 223, 224, 225, 226], time series [227, 228, 229, 230, 231], image data [232, 101, 233], text-based source [234, 235, 236], via community competition [2], [3], and many others [237, 238].

4.4. Proposed Research 2: Comprehensive Construction of Bioprocess Engineering Benchmark

The development of standards for bioprocessing engineering is essential to accelerate its growth while also attracting the participation of experts from many other fields. As we discussed above and the lessons learned from the machine learning community for the necessity of an ideal benchmark. More specifically, the benchmark (1) should provide publicly available datasets, while also providing standard procedures on those public datasets like typical machine learning tasks, such as classification, regression, and prediction; and (2) must be generic enough and easily integrated to accommodate different bio-research engineering pipelines. However, an important point that makes the bioprocessing specification more prominent and specific to its field is the bioprocessing metadata [239, 240, 241, 242]. Take a look at the following example of experimental verification.

A laboratory, named A, performed a verification experiment of four optimally designed experiments (two were performed by a kinetic model and the others by an artificial neural network) [243]. These experiments were performed in a glass tubular photoreactor with a capacity of 1 L (length of 15.5 cm and diameter of 9.5 cm). A technician attaches an artificial light source to opposite sides of the reactor using 14 W TL 5 tungsten incandescent lamps, manufactured by Philip Co., China). The experiments started with two hyperparameters of biomass concentration and nitrate concentration set to 0.27 g/L and 9 mM, respectively. The experiments also set two other hyperparameters, the influential nitrate concentration, and the fixed culture temperature of 0.1 M and 35 °C for all experiments. Cultures were continuously aerated with 2.5% CO2 in air at 0.2 vvm and pH = 7.5 at a stirring rate of 300 rpm. The technician varied the nitrate feeding rate and light intensity daily throughout the experiment.

Let's assume that laboratory A releases the experimental datasets and reference model, e.g., an artificial neural network in this case. If the laboratory,

[2] `https://www.kaggle.com/competitions`
[3] `https://paperswithcode.com/`

named B, is interested in the experiments and wants to improve its current project with a similar verification experiment. Then laboratory B must know the exact experimental settings and configuration such as the equipment, chemical origin, spacial location of equipment installation, nitrate feeding rate log, and other necessary metadata. Hence, the bioprocess engineering benchmark should have the third component: bioprocess metadata as presented in Figure 12. Unfortunately, the bioprocess engineering literature witnesses not many the variety of benchmark-ready published articles and dedicated benchmarking [244, 245, 246, 247, 248].

Figure 12: Components of bioprocess engineering benchmark.

5. Conclusion

The potential of machine learning in bioprocess engineering is just beginning to unfold. High-throughput experimental facilities continuously evolve with increased automation and better analytics, generating more considerable amounts of high-quality data with less human intervention. Simultaneously we have seen the rise of machine learning offering new algorithms that can learn not only predictive models but also representations to do so and even "learn to learn" to drive data gathering. With the increased availability of computational power, automated model building based on machine learning is almost within reach.

In this work, we reviewed existing methods and hinted at potential applications of machine learning and artificial intelligence in bioprocess engineering from the perspective of making a faster and less costly development spiral. We have summarized machine learning fields and model classes that have great potential to automate model-building and reduce uncertainty in different development stages, even if this has been realized to a limited extent only at the time when this review was written. There are reasons to firmly believe that the combination of bioprocess engineering and machine learning will be a key the driver of progress in the coming decade, especially as it allows to build of predictive models using all available aggregated information.

Yet we are aware of essential obstacles to overcome, namely the still limited degree of automation in the workflows used to generate data and the lack of comprehensive implementation of FAIR principles (Findable, Accessible, Interoperable, and Reusable) in bioprocess development. For bioprocess engineering

to fruitfully meet machine learning, it is mandatory to implement end-to-end digitalization of experiments and achieve significantly higher levels of automation to generate informative experimental data with the required meta-data automatically attached.

6. Acknowledgments

The authors kindly appreciate the support of the German Federal Ministry of Education and Research through the Program "International Future Labs for Artificial Intelligence (Grant number 01DD20002A)". We acknowledge the Open Access Publication Fund of Technische Universität Berlin.

References

[1] J. Lücke, M. Bädeker, M. Hildinger, Medizinische biotechnologie in deutschland 2021, vfa-Publikationen, Deutschland, The Boston Consulting Group, München.

[2] A. Waldbaur, J. Kittelmann, C. P. Radtke, J. Hubbuch, B. E. Rapp, Microfluidics on liquid handling stations (μf-on-lhs): an industry compatible chip interface between microfluidics and automated liquid handling stations, Lab on a Chip 13 (12) (2013) 2337–2343.

[3] C. P. Radtke, M. Delbé, M. Wörner, J. Hubbuch, Photoinitiated miniemulsion polymerization in microfluidic chips on automated liquid handling stations: Proof of concept, Engineering in Life Sciences 16 (6) (2016) 505–514.

[4] K. Treier, S. Hansen, C. Richter, P. Diederich, J. Hubbuch, P. Lester, High-throughput methods for miniaturization and automation of monoclonal antibody purification processes, Biotechnology progress 28 (3) (2012) 723–732.

[5] J. A. Reuter, D. V. Spacek, M. P. Snyder, High-throughput sequencing technologies, Molecular cell 58 (4) (2015) 586–597.

[6] A. W. Rogers, F. Vega-Ramon, J. Yan, E. A. del Río-Chan ona, K. Jing, D. Zhang, A transfer learning approach for predictive modeling of bioprocesses using small data, Biotechnology and Bioengineering 119 (2) (2022) 411–422.

[7] F. Romero, J. Sprenger, Scientific self-correction: the bayesian way, Synthese 198 (23) (2021) 5803–5823.

[8] M. Känsäkoski, M. Kurkinen, N. von Weymarn, P. Niemelä, P. Neubauer, E. Juuso, T. Eerikäinen, S. Turunen, S. Aho, P. Suhonen, Process analytical technology (pat) needs and applications in the bioprocess industry, VTT Technical Research Centre of Finland 60 (2006) 99.

[9] J. Glassey, K. Gernaey, C. Clemens, T. W. Schulz, R. Oliveira, G. Striedner, C.-F. Mandenius, Process analytical technology (pat) for biopharmaceuticals, Biotechnology Journal 6 (4) (2011) 369–377.

[10] L. L. Simon, H. Pataki, G. Marosi, F. Meemken, K. Hungerbuhler, A. Baiker, S. Tummala, B. Glennon, M. Kuentz, G. Steele, et al., Assessment of recent process analytical technology (pat) trends: a multiauthor review, Organic Process Research & Development 19 (1) (2015) 3–62.

[11] P. Diederich, J. Hubbuch, High-throughput column chromatography performed on liquid handling stations, Preparative chromatography for separation of proteins 100 (2017) 293–332.

[12] T. Barz, A. Sommer, T. Wilms, P. Neubauer, M. N. C. Bournazou, Adaptive optimal operation of a parallel robotic liquid handling station, IFAC-PapersOnLine 51 (2) (2018) 765–770.

[13] S. Hans, M. Gimpel, F. Glauche, P. Neubauer, M. N. Cruz-Bournazou, Automated cell treatment for competence and transformation of escherichia coli in a high-throughput quasi-turbidostat using microtiter plates, Microorganisms 6 (3) (2018) 60.

[14] M. Koutinas, A. Kiparissides, E. N. Pistikopoulos, A. Mantalaris, Bioprocess systems engineering: transferring traditional process engineering principles to industrial biotechnology, Computational and structural biotechnology journal 3 (4) (2012) e201210022.

[15] S. Dara, S. Dhamercherla, S. S. Jadav, C. M. Babu, M. J. Ahsan, Machine Learning in Drug Discovery: A Review, Artif Intell Rev 55 (3) (2022) 1947–1999.

[16] Food, D. Administration, et al., Proposed regulatory framework for modifications to artificial intelligence/machine learning (ai/ml)-based software as a medical device (samd).

[17] S. Benjamens, P. Dhunnoo, B. Meskó, The state of artificial intelligence-based FDA-approved medical devices and algorithms: an online database, npj Digital Medicine 3 (1) (2020) 118.

[18] J. Lee, S. Y. Lee, S. Park, A. P. Middelberg, Control of fed-batch fermentations, Biotechnology advances 17 (1) (1999) 29–48.

[19] M. N. Cruz Bournazou, T. Barz, D. Nickel, D. Lopez Cárdenas, F. Glauche, A. Knepper, P. Neubauer, Online optimal experimental redesign in robotic parallel fed-batch cultivation facilities, Biotechnology and bioengineering 114 (3) (2017) 610–619.

[20] J. W. Kim, N. Krausch, J. Aizpuru, T. Barz, S. Lucia, E. C. Martínez, P. Neubauer, M. N. C. Bournazou, Model predictive control guided with optimal experimental design for pulse-based parallel cultivation, arXiv preprint arXiv:2112.10548.

[21] J. W. Kim, N. Krausch, J. Aizpuru, T. Barz, S. Lucia, P. Neubauer, M. N. C. Bournazou, Model predictive control and moving horizon estimation for adaptive optimal bolus feeding in high-throughput cultivation of E. coli, arXiv preprint arXiv:2203.07211.

[22] N. Krausch, J. W. Kim, T. Barz, S. Lucia, S. Groß, M. Huber, S. Schiller, P. Neubauer, M. C. Bournazou, High-throughput screening of optimal process conditions using model predictive control, Authorea Preprints.

[23] M. Mowbray, T. Savage, C. Wu, Z. Song, B. A. Cho, E. A. Del Rio-Chanona, D. Zhang, Machine learning for biochemical engineering: A review, Biochemical Engineering Journal 172 (2021) 108054.

[24] C. E. Lawson, J. M. Martí, T. Radivojevic, S. V. R. Jonnalagadda, R. Gentz, N. J. Hillson, S. Peisert, J. Kim, B. A. Simmons, C. J. Petzold, et al., Machine learning for metabolic engineering: A review, Metabolic Engineering 63 (2021) 34–60.

[25] T. Scheper, S. Beutel, N. McGuinness, S. Heiden, M. Oldiges, F. Lammers, K. F. Reardon, Digitalization and bioprocessing: Promises and challenges, Digital Twins (2020) 57–69.

[26] H. Narayanan, M. F. Luna, M. von Stosch, M. N. Cruz Bournazou, G. Polotti, M. Morbidelli, A. Butté, M. Sokolov, Bioprocessing in the digital age: the role of process models, Biotechnology journal 15 (1) (2020) 1900172.

[27] P. Neubauer, F. Glauche, M. N. Cruz-Bournazou, Bioprocess development in the era of digitalization, Engineering in Life Sciences 17 (11) (2017) 1140.

[28] P. Neubauer, E. Anane, S. Junne, M. N. Cruz Bournazou, Potential of integrating model-based design of experiments approaches and process analytical technologies for bioprocess scale-down, Digital Twins (2020) 1–28.

[29] T. Miller, Explanation in artificial intelligence: Insights from the social sciences, Artificial intelligence 267 (2019) 1–38.

[30] G.-W. Wei, Protein structure prediction beyond alphafold, Nature Machine Intelligence 1 (8) (2019) 336–337.

[31] J. Jumper, R. Evans, A. Pritzel, T. Green, M. Figurnov, O. Ronneberger, K. Tunyasuvunakool, R. Bates, A. Žídek, A. Potapenko, et al., Highly accurate protein structure prediction with alphafold, Nature 596 (7873) (2021) 583–589.

[32] L. Kaspersetz, S. Waldburger, M.-T. Schermeyer, S. L. Riedel, S. Gross, P. Neubauer, M.-N. Cruz-Bournazou, Automated bioprocess feedback operation in a high throughput facility via the integration of a mobile robotic lab assistant, bioRxiv.

[33] D. Schönberger, Deep copyright: up-and downstream questions related to artificial intelligence (ai) and machine learning (ml), SCHÖNBERGER Daniel, Deep Copyright: Up-and Downstream-Questions Related to Artificial Intelligence (AI) and Machine Learning (ML) in Droit d'auteur 4 (2018) 145–173.

[34] S. Haque, S. Khan, M. Wahid, S. A. Dar, N. Soni, R. K. Mandal, V. Singh, D. Tiwari, M. Lohani, M. Y. Areeshi, et al., Artificial intelligence vs. statistical modeling and optimization of continuous bead milling process for bacterial cell lysis, Frontiers in microbiology 7 (2016) 1852.

[35] C. Walther, M. Voigtmann, E. Bruna, A. Abusnina, A.-L. Tscheließnig, M. Allmer, H. Schuchnigg, C. Brocard, A. Föttinger-Vacha, G. Klima, Smart process development: Application of machine-learning and integrated process modeling for inclusion body purification processes, Biotechnology Progress (2022) e3249.

[36] P. Petsagkourakis, I. O. Sandoval, E. Bradford, D. Zhang, E. A. del Rio-Chanona, Reinforcement learning for batch bioprocess optimization, Computers & Chemical Engineering 133 (2020) 106649.

[37] S. Mazurenko, Z. Prokop, J. Damborsky, Machine learning in enzyme engineering, ACS Catalysis 10 (2) (2019) 1210–1223.

[38] D. Heckmann, C. J. Lloyd, N. Mih, Y. Ha, D. C. Zielinski, Z. B. Haiman, A. A. Desouki, M. J. Lercher, B. O. Palsson, Machine learning applied to enzyme turnover numbers reveals protein structural correlates and improves metabolic models, Nature communications 9 (1) (2018) 1–10.

[39] J.-X. Tan, H. Lv, F. Wang, F.-Y. Dao, W. Chen, H. Ding, A survey for predicting enzyme family classes using machine learning methods, Current drug targets 20 (5) (2019) 540–550.

[40] T. Barz, J. Kager, C. Herwig, P. Neubauer, M. N. C. Bournazou, F. Galvanin, Characterization of reactions and growth in automated continuous flow and bioreactor platforms—from linear doe to model-based approaches, in: Simulation and Optimization in Process Engineering, Elsevier, 2022, pp. 273–319.

[41] N. Borisov, V. Tkachev, I. Muchnik, A. Buzdin, Individual drug treatment prediction in oncology based on machine learning using cell culture gene expression data, in: Proceedings of the 2017 International Conference on Computational Biology and Bioinformatics, 2017, pp. 1–6.

[42] M. Ashraf, M. Khalilitousi, Z. Laksman, Applying machine learning to stem cell culture and differentiation, Current Protocols 1 (9) (2021) e261.

[43] V. Venkatasubramanian, The promise of artificial intelligence in chemical engineering: Is it here, finally, AIChE J 65 (2) (2019) 466–478.

[44] S. K. Niazi, J. L. Brown, Fundamentals of modern bioprocessing, CRC Press, 2017.

[45] P. Neubauer, M. N. Cruz-Bournazou, Continuous bioprocess development: methods for control and characterization of the biological system, Continuous Biomanufacturing: Innovative Technologies and Methods; John Wiley & Sons: Hoboken, NJ, USA.

[46] R. Garnett, Bayesian Optimization, Cambridge University Press, 2022, in preparation.

[47] F. Hutter, L. Kotthoff, J. Vanschoren, Automated machine learning: methods, systems, challenges, Springer Nature, 2019.

[48] M. Mohammed, M. B. Khan, E. B. M. Bashier, Machine learning: algorithms and applications, Crc Press, 2016.

[49] K. P. Murphy, Probabilistic machine learning: an introduction, MIT press, 2022.

[50] S. Sra, S. Nowozin, S. J. Wright, Optimization for machine learning, Mit Press, 2012.

[51] M. S. Mahmud, J. Z. Huang, S. Salloum, T. Z. Emara, K. Sadatdiynov, A survey of data partitioning and sampling methods to support big data analysis, Big Data Mining and Analytics 3 (2) (2020) 85–101.

[52] N. Bussola, A. Marcolini, V. Maggio, G. Jurman, C. Furlanello, Ai slipping on tiles: Data leakage in digital pathology, in: International Conference on Pattern Recognition, Springer, 2021, pp. 167–182.

[53] R. D. King, O. I. Orhobor, C. C. Taylor, Cross-validation is safe to use, Nat Mach Intell 3 (4) (2021) 276–276.

[54] M. Feurer, F. Hutter, Hyperparameter optimization, in: Automated machine learning, Springer, Cham, 2019, pp. 3–33.

[55] L. Yang, A. Shami, On hyperparameter optimization of machine learning algorithms: Theory and practice, Neurocomputing 415 (2020) 295–316.

[56] T. Chen, W. Zhang, Z. Jingyang, S. Chang, S. Liu, L. Amini, Z. Wang, Training stronger baselines for learning to optimize, Advances in Neural Information Processing Systems 33 (2020) 7332–7343.

[57] W. Chung, V. Thomas, M. C. Machado, N. Le Roux, Beyond variance reduction: Understanding the true impact of baselines on policy optimization, in: International Conference on Machine Learning, PMLR, 2021, pp. 1999–2009.

[58] P. Riley, Three pitfalls to avoid in machine learning (2019).

[59] J. G. Greener, S. M. Kandathil, L. Moffat, D. T. Jones, A guide to machine learning for biologists, Nature Reviews Molecular Cell Biology 23 (1) (2022) 40–55.

[60] D. Chicco, Ten quick tips for machine learning in computational biology, BioData mining 10 (1) (2017) 1–17.

[61] S. M. Wheelwright, Economic and cost factors of bioprocess engineering, in: Biotechnology and Biopharmaceutical Manufacturing, Processing, and Preservation, CRC Press, 2020, pp. 333–354.

[62] K. S. Ng, J. A. Smith, M. P. McAteer, B. E. Mead, J. Ware, F. O. Jackson, A. Carter, L. Ferreira, K. Bure, J. A. Rowley, et al., Bioprocess decision support tool for scalable manufacture of extracellular vesicles, Biotechnology and bioengineering 116 (2) (2019) 307–319.

[63] C. Spahn, E. Gómez-de Mariscal, R. F. Laine, P. M. Pereira, L. von Chamier, M. Conduit, M. G. Pinho, G. Jacquemet, S. Holden, M. Heilemann, R. Henriques, DeepBacs for multi-task bacterial image analysis using open-source deep learning approaches, Commun Biol 5 (1) (2022) 1–18, number: 1 Publisher: Nature Publishing Group.

[64] R. Ashmore, R. Calinescu, C. Paterson, Assuring the machine learning lifecycle: Desiderata, methods, and challenges, ACM Computing Surveys (CSUR) 54 (5) (2021) 1–39.

[65] F. Kumeno, Sofware engneering challenges for machine learning applications: A literature review, Intelligent Decision Technologies 13 (4) (2019) 463–476.

[66] H. Luu, Managing the machine learning life cycle, in: Beginning Apache Spark 3, Springer, 2021, pp. 395–429.

[67] M. Zaharia, A. Chen, A. Davidson, A. Ghodsi, S. A. Hong, A. Konwinski, S. Murching, T. Nykodym, P. Ogilvie, M. Parkhe, et al., Accelerating the machine learning lifecycle with mlflow., IEEE Data Eng. Bull. 41 (4) (2018) 39–45.

[68] A. S. Rathore, S. Mishra, S. Nikita, P. Priyanka, Bioprocess control: current progress and future perspectives, Life 11 (6) (2021) 557.

[69] N. Habibi, S. Z. M. Hashim, A. Norouzi, M. R. Samian, A review of machine learning methods to predict the solubility of overexpressed recombinant proteins in escherichia coli, BMC bioinformatics 15 (1) (2014) 1–16.

[70] F. Mey, J. Clauwaert, K. Van Huffel, W. Waegeman, M. De Mey, Improving the performance of machine learning models for biotechnology: The quest for deus ex machina, Biotechnology advances 53 (2021) 107858.

[71] S. Panjwani, I. Cui, K. Spetsieris, M. Mleczko, W. Wang, J. X. Zou, M. Anwaruzzaman, S. Liu, R. Canales, O. Hesse, Application of machine learning methods to pathogen safety evaluation in biological manufacturing processes, Biotechnology Progress 37 (3) (2021) e3135.

[72] L. Rychener, F. Montet, J. Hennebert, Architecture Proposal for Machine Learning Based Industrial Process Monitoring, Procedia Computer Science 170 (2020) 648–655.

[73] B. Settles, Active Learning, Synthesis Lectures on Artificial Intelligence and Machine Learning Series, Morgan & Claypool, 2012.

[74] D. D. Lewis, W. A. Gale, A Sequential Algorithm for Training Text Classifiers, in: B. W. Croft, C. J. van Rijsbergen (Eds.), SIGIR '94, Springer London, London, 1994, pp. 3–12.

[75] B. Settles, Active Learning Literature Survey 67.

[76] W. H. Beluch, T. Genewein, A. Nurnberger, J. M. Kohler, The Power of Ensembles for Active Learning in Image Classification, in: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, IEEE, Salt Lake City, UT, 2018, pp. 9368–9377.

[77] N. Roy, A. McCallum, Toward optimal active learning through sampling estimation of error reduction, in: ICML, 2001.

[78] K. Konyushkova, R. Sznitman, P. Fua, Learning Active Learning from Data 11.

[79] J. T. Ash, S. Goel, Gone Fishing: Neural Active Learning with Fisher Embeddings 13.

[80] A. Foster, D. R. Ivanova, I. Malik, T. Rainforth, Deep Adaptive Design: Amortizing Sequential Bayesian Experimental Design 12.

[81] A. Foster, M. Jankowiak, M. O'Meara, Y. W. Teh, T. Rainforth, A Unified Stochastic Gradient Approach to Designing Bayesian-Optimal Experiments 10.

[82] A. Foster, M. Jankowiak, E. Bingham, P. Horsfall, Y. W. Teh, T. Rainforth, N. Goodman, Variational Bayesian Optimal Experimental Design 12.

[83] S. Kleinegesse, M. U. Gutmann, Bayesian Experimental Design for Implicit Models by Mutual Information Neural Estimation, arXiv:2002.08129 [cs, stat]ArXiv: 2002.08129.

[84] S. Kleinegesse, M. Gutmann, Efficient Bayesian Experimental Design for Implicit Models, arXiv:1810.09912 [cs, stat]ArXiv: 1810.09912.

[85] D. R. Ivanova, A. Foster, S. Kleinegesse, M. U. Gutmann, T. Rainforth, Implicit Deep Adaptive Design: Policy-Based Experimental Design without Likelihoods, arXiv:2111.02329 [cs, stat].

[86] T.-T. Vu, M. Liu, D. Phung, G. Haffari, Learning How to Active Learn by Dreaming 11.

[87] M. Fang, Y. Li, T. Cohn, Learning how to Active Learn: A Deep Reinforcement Learning Approach, in: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Copenhagen, Denmark, 2017, pp. 595–605.

[88] M. Liu, W. Buntine, G. Haffari, Learning How to Actively Learn: A Deep Imitation Learning Approach, in: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, Melbourne, Australia, 2018, pp. 1874–1883.

[89] P. Bachman, A. Sordoni, A. Trischler, Learning Algorithms for Active Learning 10.

[90] R. S. Sutton, A. G. Barto, Reinforcement learning: An introduction, MIT press, 2018.

[91] R. D. King, K. E. Whelan, F. M. Jones, P. G. K. Reiser, C. H. Bryant, S. H. Muggleton, D. B. Kell, S. G. Oliver, Functional genomic hypothesis generation and experimentation by a robot scientist, Nature 427 (6971) (2004) 247–252.

[92] C. H. Bryant, S. H. Muggleton, S. G. Oliver, D. B. Kell, P. Reiser, R. D. King, Combining Inductive Logic Programming, Active Learning and Robotics to Discover the Function of Genes 45.

[93] Y. Gal, Uncertainty in deep learning, University of Cambridge.

[94] E. Tsymbalov, M. Panov, A. Shapeev, Dropout-based Active Learning for Regression, arXiv:1806.09856 [cs, stat] 11179 (2018) 247–258, arXiv:1806.09856.

[95] L. Torrey, J. Shavlik, Transfer learning, in: Handbook of Research on Machine Learning Applications and Trends: Algorithms, Methods, and Techniques, IGI Global, 2010, pp. 242–264.

[96] K. Weiss, T. M. Khoshgoftaar, D. Wang, A survey of transfer learning, Journal of Big Data 3 (1) (2016) 9.

[97] M. Rostami, Transfer Learning Through Embedding Spaces, CRC Press, 2021.

[98] M. Long, H. Zhu, J. Wang, M. I. Jordan, Deep transfer learning with joint adaptation networks, in: International conference on machine learning, PMLR, 2017, pp. 2208–2217.

[99] N. Duong-Trung, L.-D. Quach, C.-N. Nguyen, Learning deep transferability for several agricultural classification problems, International Journal of Advanced Computer Science and Applications 10 (1).

[100] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, L. Fei-Fei, Imagenet: A large-scale hierarchical image database.

[101] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al., Imagenet large scale visual recognition challenge, International journal of computer vision 115 (3) (2015) 211–252.

[102] M. Huh, P. Agrawal, A. A. Efros, What makes imagenet good for transfer learning?, arXiv preprint arXiv:1608.08614.

[103] M. Christopher, A. Belghith, C. Bowd, J. A. Proudfoot, M. H. Goldbaum, R. N. Weinreb, C. A. Girkin, J. M. Liebmann, L. M. Zangwill, Performance of deep learning architectures and transfer learning for detecting glaucomatous optic neuropathy in fundus photographs, Scientific reports 8 (1) (2018) 16685.

[104] N. Duong-Trung, L.-D. Quach, M.-H. Nguyen, C.-N. Nguyen, Classification of grain discoloration via transfer learning and convolutional neural networks, in: Proceedings of the 3rd International Conference on Machine Learning and Soft Computing, 2019, pp. 27–32.

[105] N. Duong-Trung, L.-D. Quach, M.-H. Nguyen, C.-N. Nguyen, A combination of transfer learning and deep learning for medicinal plant classification, in: Proceedings of the 2019 4th International Conference on Intelligent Information Technology, 2019, pp. 83–90.

[106] A. C. Tran, N. C. Tran, N. Duong-Trung, Recognition and quantity estimation of pastry images using pre-training deep convolutional networks, in: International Conference on Future Data and Security Engineering, Springer, 2020, pp. 200–214.

[107] N. Duong-Trung, D. N. Le Ha, H. X. Huynh, Classification-segmentation pipeline for mri via transfer learning and residual networks., in: International Conference on Research in Intelligent Computing in Engineering, Annals of Computer Science and Information Systems, 2021, pp. 39–43.

[108] E. C. Alley, G. Khimulya, S. Biswas, M. AlQuraishi, G. M. Church, Unified rational protein engineering with sequence-based deep representation learning, Nat Methods 16 (12) (2019) 1315–1322.

[109] T. Bepler, B. Berger, Learning protein sequence embeddings using information from structure, arXiv:1902.08661 [cs, q-bio, stat]ArXiv: 1902.08661.

[110] K. K. Yang, Z. Wu, C. N. Bedbrook, F. H. Arnold, Learned protein embeddings for machine learning, Bioinformatics 34 (15) (2018) 2642–2648.

[111] R. Rao, N. Bhattacharya, N. Thomas, Y. Duan, X. Chen, J. Canny, P. Abbeel, Y. S. Song, Evaluating Protein Transfer Learning with TAPE, arXiv:1906.08230 [cs, q-bio, stat]ArXiv: 1906.08230.

[112] B. J. Wittmann, Y. Yue, F. H. Arnold, Machine Learning-Assisted Directed Evolution Navigates a Combinatorial Epistatic Fitness Landscape with Minimal Screening Burden, preprint, Bioinformatics (Dec. 2020). URL http://biorxiv.org/lookup/doi/10.1101/2020.12.04.408955

[113] E. Fenoy, A. A. Edera, G. Stegmayer, Transfer learning in proteins: evaluating novel protein learned representations for bioinformatics tasks, Briefings in Bioinformatics 23 (4) (2022) bbac232.

[114] S. Mahajan, A. Raina, X.-Z. Gao, A. Kant Pandit, Plant recognition using morphological feature extraction and transfer learning over svm and adaboost, Symmetry 13 (2) (2021) 356.

[115] M. Izadpanahkakhk, S. M. Razavi, M. Taghipour-Gorjikolaie, S. H. Zahiri, A. Uncini, Deep region of interest and feature extraction models for palmprint verification using convolutional neural networks transfer learning, Applied Sciences 8 (7) (2018) 1210.

[116] B. Neyshabur, H. Sedghi, C. Zhang, What is being transferred in transfer learning?, Advances in neural information processing systems 33 (2020) 512–523.

[117] N. Duong-Trung, L.-D. Quach, C.-N. Nguyen, Towards classification of shrimp diseases using transferred convolutional neural networks, Advances in Science, Technology and Engineering Systems Journal 5 (4) (2020) 724–732.

[118] H. Yoo, H. E. Byun, D. Han, J. H. Lee, Reinforcement learning for batch process control: Review and perspectives, Annual Reviews in Control 52 (2021) 108–119.

[119] D. P. Bertsekas, Dynamic programming and optimal control, Vol. 1, Athena Scientific Belmont, MA, 2005.

[120] L. Busoniu, R. Babuska, B. De Schutter, D. Ernst, Reinforcement learning and dynamic programming using function approximators, CRC press, 2017.

[121] J. W. Kim, B. J. Park, H. Yoo, T. H. Oh, J. H. Lee, J. M. Lee, A model-based deep reinforcement learning method applied to finite-horizon optimal control of nonlinear control-affine system, Journal of Process Control 87 (2020) 166–178.

[122] J. W. Kim, T. H. Oh, S. H. Son, D. H. Jeong, J. M. Lee, Convergence analysis of the deep neural networks based globalized dual heuristic programming, Automatica 122 (2020) 109222.

[123] J. M. Lee, J. H. Lee, Approximate dynamic programming-based approaches for input–output data-driven control of nonlinear processes, Automatica 41 (7) (2005) 1281–1288.

[124] J. W. Kim, G. B. Choi, J. M. Lee, A POMDP framework for integrated scheduling of infrastructure maintenance and inspection, Computers & Chemical Engineering 112 (2018) 239–252.

[125] T. H. Oh, J. W. Kim, S. H. Son, H. Kim, K. Lee, J. M. Lee, Automatic control of simulated moving bed process with deep Q-network, Journal of Chromatography A 1647 (2021) 462073.

[126] J. Horwood, E. Noutahi, Molecular design in synthetically accessible chemical space via deep reinforcement learning, ACS omega 5 (51) (2020) 32984–32994.

[127] G. Novati, H. L. de Laroussilhe, P. Koumoutsakos, Automating turbulence modelling by multi-agent reinforcement learning, Nature Machine Intelligence 3 (1) (2021) 87–96.

[128] S. Levine, C. Finn, T. Darrell, P. Abbeel, End-to-end training of deep visuomotor policies, The Journal of Machine Learning Research 17 (1) (2016) 1334–1373.

[129] G. Williams, P. Drews, B. Goldfain, J. M. Rehg, E. A. Theodorou, Aggressive driving with model predictive path integral control, in: 2016 IEEE International Conference on Robotics and Automation (ICRA), IEEE, 2016, pp. 1433–1440.

[130] D. Silver, J. Schrittwieser, K. Simonyan, I. Antonoglou, A. Huang, A. Guez, T. Hubert, L. Baker, M. Lai, A. Bolton, et al., Mastering the game of Go without human knowledge, Nature 550 (7676) (2017) 354.

[131] S. Lucia, A. Tătulea-Codrean, C. Schoppmeyer, S. Engell, Rapid development of modular and sustainable nonlinear model predictive control solutions, Control Engineering Practice 60 (2017) 51–62.

[132] L. Buşoniu, T. de Bruin, D. Tolić, J. Kober, I. Palunko, Reinforcement learning for control: Performance, stability, and deep approximators, Annual Reviews in Control.

[133] J. M. Lee, N. S. Kaisare, J. H. Lee, Choice of approximator and design of penalty function for an approximate dynamic programming based control approach, Journal of Process Control 16 (2) (2006) 135–156.

[134] J. M. Lee, J. H. Lee, An approximate dynamic programming based approach to dual adaptive control, Journal of process control 19 (5) (2009) 859–864.

[135] J. Wilson, E. Martinez, Neuro-fuzzy modeling and control of a batch process involving simultaneous reaction and distillation, Computers & chemical engineering 21 (1997) S1233–S1238.

[136] C. V. Peroni, N. S. Kaisare, J. H. Lee, Optimal control of a fed-batch bioreactor using simulation-based approximate dynamic programming, IEEE Transactions on Control Systems Technology 13 (5) (2005) 786–790.

[137] D. Li, L. Qian, Q. Jin, T. Tan, Reinforcement learning control with adaptive gain for a Saccharomyces cerevisiae fermentation process, Applied Soft Computing 11 (8) (2011) 4488–4495.

[138] B. J. Pandian, M. M. Noel, Control of a bioreactor using a new partially supervised reinforcement learning algorithm, Journal of Process Control 69 (2018) 16–29.

[139] Y. Ma, D. A. Noreña-Caro, A. J. Adams, T. B. Brentzel, J. A. Romagnoli, M. G. Benton, Machine-learning-based simulation and fed-batch control of cyanobacterial-phycocyanin production in Plectonema by artificial neural network and deep reinforcement learning, Computers & Chemical Engineering 142 (2020) 107016.

[140] P. Henderson, R. Islam, P. Bachman, J. Pineau, D. Precup, D. Meger, Deep reinforcement learning that matters, arXiv preprint arXiv:1709.06560.

[141] S. Fujimoto, H. Van Hoof, D. Meger, Addressing function approximation error in actor-critic methods, arXiv preprint arXiv:1802.09477.

[142] B. Recht, A tour of reinforcement learning: The view from continuous control, Annual Review of Control, Robotics, and Autonomous Systems 2 (2019) 253–279.

[143] E. Langlois, S. Zhang, G. Zhang, P. Abbeel, J. Ba, Benchmarking model-based reinforcement learning, arXiv preprint arXiv:1907.02057.

[144] J. W. Kim, B. J. Park, T. H. Oh, J. M. Lee, Model-based reinforcement learning and predictive control for two-stage optimal control of fed-batch bioreactor, Computers & Chemical Engineering 154 (2021) 107465.

[145] T. H. Oh, H. M. Park, J. W. Kim, J. M. Lee, Integration of reinforcement learning and model predictive control to optimize semi-batch bioreactor, AIChE Journal (2022) e17658.

[146] E. Pan, P. Petsagkourakis, M. Mowbray, D. Zhang, E. A. del Rio-Chanona, Constrained model-free reinforcement learning for process optimization, Computers & Chemical Engineering 154 (2021) 107462.

[147] P. Petsagkourakis, I. O. Sandoval, E. Bradford, F. Galvanin, D. Zhang, E. A. del Rio-Chanona, Chance constrained policy optimization for process control and optimization, Journal of Process Control 111 (2022) 35–45.

[148] M. Mowbray, P. Petsagkourakis, E. A. del Rio-Chanona, D. Zhang, Safe chance constrained reinforcement learning for batch process control, Computers & Chemical Engineering 157 (2022) 107630.

[149] K. Xiao, S. Liang, X. Wang, C. Chen, X. Huang, Current state and challenges of full-scale membrane bioreactor applications: A critical review, Bioresource technology 271 (2019) 473–481.

[150] K. Sode, T. Yamazaki, I. Lee, T. Hanashi, W. Tsugawa, Biocapacitor: A novel principle for biosensors, Biosensors and Bioelectronics 76 (2016) 20–28.

[151] B. Dai, L. Wang, Y. Wang, G. Yu, X. Huang, Single-cell nanometric coating towards whole-cell-based biodevices and biosensors, ChemistrySelect 3 (25) (2018) 7208–7221.

[152] S. Pradhan, A. Brooks, V. Yadavalli, Nature-derived materials for the fabrication of functional biodevices, Materials Today Bio 7 (2020) 100065.

[153] P. Mehrotra, Biosensors and their applications–a review, Journal of oral biology and craniofacial research 6 (2) (2016) 153–159.

[154] J. Ong, M. R. Appleford, G. Mani, Introduction to biomaterials: basic theory with engineering applications, Cambridge University Press, 2014.

[155] M.-C. Tanzi, S. Farè, G. Candiani, Foundations of biomaterials engineering, Academic Press, 2019.

[156] V. Dos Santos, R. N. Brandalise, M. Savaris, Engineering of biomaterials, Springer, 2017.

[157] B. M. Woolston, S. Edgar, G. Stephanopoulos, Metabolic engineering: past and future, Annual review of chemical and biomolecular engineering 4 (2013) 259–288.

[158] T. U. Chae, S. Y. Choi, J. W. Kim, Y.-S. Ko, S. Y. Lee, Recent advances in systems metabolic engineering tools and strategies, Current opinion in biotechnology 47 (2017) 67–82.

[159] K. V. Presnell, H. S. Alper, Systems metabolic engineering meets machine learning: A new era for data-driven metabolic engineering, Biotechnology journal 14 (9) (2019) 1800416.

[160] M. Banner, H. Alosert, C. Spencer, M. Cheeks, S. S. Farid, M. Thomas, S. Goldrick, A decade in review: use of data analytics within the biopharmaceutical sector, Current Opinion in Chemical Engineering 34 (2021) 100758.

[161] H. Le, S. Kabbur, L. Pollastrini, Z. Sun, K. Mills, K. Johnson, G. Karypis, W.-S. Hu, Multivariate analysis of cell culture bioprocess data—lactate consumption as process indicator, Journal of biotechnology 162 (2-3) (2012) 210–223.

[162] L. Wei, Z. Yuan, M. Cui, H. Han, J. Shen, Study on electricity-generation characteristic of two-chambered microbial fuel cell in continuous flow mode, International journal of hydrogen energy 37 (1) (2012) 1067–1073.

[163] A. Garg, V. Vijayaraghavan, S. Mahapatra, K. Tai, C. Wong, Performance evaluation of microbial fuel cell by artificial intelligence methods, Expert systems with applications 41 (4) (2014) 1389–1399.

[164] E. A. del Rio-Chanona, F. Fiorelli, D. Zhang, N. R. Ahmed, K. Jing, N. Shah, An efficient model construction strategy to simulate microalgal lutein photo-production dynamic process, Biotechnology and Bioengineering 114 (11) (2017) 2518–2527.

[165] E. Bradford, A. M. Schweidtmann, D. Zhang, K. Jing, E. A. del Rio-Chanona, Dynamic modeling and optimization of sustainable algal production with uncertainty using multivariate gaussian processes, Computers & Chemical Engineering 118 (2018) 143–158.

[166] E. A. del Rio-Chanona, E. Manirafasha, D. Zhang, Q. Yue, K. Jing, Dynamic modeling and optimization of cyanobacterial c-phycocyanin production process by artificial neural network, Algal Research 13 (2016) 7–15.

[167] E. A. del Rio-Chanona, J. L. Wagner, H. Ali, F. Fiorelli, D. Zhang, K. Hellgardt, Deep learning-based surrogate modeling and optimization for microalgal biofuel production and photobioreactor design, AIChE Journal 65 (3) (2019) 915–923.

[168] E. Bradford, L. Imsland, D. Zhang, E. A. del Rio Chanona, Stochastic data-driven model predictive control using gaussian processes, Computers & Chemical Engineering 139 (2020) 106844.

[169] S. Han, T. Kim, D. Kim, Y.-L. Park, S. Jo, Use of deep learning for characterization of microfluidic soft sensors, IEEE Robotics and Automation Letters 3 (2) (2018) 873–880.

[170] C. C. Horgan, M. Jensen, A. Nagelkerke, J.-P. St-Pierre, T. Vercauteren, M. M. Stevens, M. S. Bergholt, High-throughput molecular imaging via deep-learning-enabled raman spectroscopy, Analytical chemistry 93 (48) (2021) 15850–15860.

[171] C. Banbury, R. Mason, I. Styles, N. Eisenstein, M. Clancy, A. Belli, A. Logan, P. Goldberg Oppenheimer, Development of the self optimising kohonen index network (skinet) for raman spectroscopy based detection of anatomical eye tissue, Scientific reports 9 (1) (2019) 1–9.

[172] C. Banbury, I. Styles, N. Eisenstein, E. R. Zanier, G. Vegliante, A. Belli, A. Logan, P. G. Oppenheimer, Spectroscopic detection of traumatic brain injury severity and biochemistry from the retina, Biomedical optics express 11 (11) (2020) 6249–6261.

[173] A. Tardast, M. Rahimnejad, G. Najafpour, A. Ghoreyshi, G. C. Premier, G. Bakeri, S.-E. Oh, Use of artificial neural network for the prediction of bioelectricity production in a membrane less microbial fuel cell, Fuel 117 (2014) 697–703.

[174] F. Fang, G.-L. Zang, M. Sun, H.-Q. Yu, Optimizing multi-variables of microbial fuel cell for electricity generation with an integrated modeling and experimental approach, Applied energy 110 (2013) 98–103.

[175] D. B. Miracle, O. N. Senkov, A critical review of high entropy alloys and related concepts, Acta Materialia 122 (2017) 448–511.

[176] W. Huang, P. Martin, H. L. Zhuang, Machine-learning phase prediction of high-entropy alloys, Acta Materialia 169 (2019) 225–236.

[177] P. Asgharzadeh, A. I. Birkhold, Z. Trivedi, B. Özdemir, R. Reski, O. Röhrle, A nanofe simulation-based surrogate machine learning model to predict mechanical functionality of protein networks from live confocal imaging, Computational and structural biotechnology journal 18 (2020) 2774–2788.

[178] Mof mechanical properties explorer: Adsorption advanced materials group, university of cambridge (2019).
URL `http://aam.ceb.cam.ac.uk/mof-explorer/mechanicalproperties/`

[179] P. Z. Moghadam, S. M. Rogge, A. Li, C.-M. Chow, J. Wieme, N. Moharrami, M. Aragones-Anglada, G. Conduit, D. A. Gomez-Gualdron, V. Van Speybroeck, et al., Structure-mechanical stability relations of metal-organic frameworks via machine learning, Matter 1 (1) (2019) 219–234.

[180] M. Sarmadi, A. M. Behrens, K. J. McHugh, H. T. Contreras, Z. L. Tochka, X. Lu, R. Langer, A. Jaklenec, Modeling, design, and machine learning-based framework for optimal injectability of microparticle-based drug formulations, Science advances 6 (28) (2020) eabb6594.

[181] A. B. Farimani, M. Heiranian, N. R. Aluru, Identification of amino acids with sensitive nanoporous mos2: towards machine learning-based prediction, Nat. 2D Mater 2.

[182] F. Tourlomousis, C. Jia, T. Karydis, A. Mershin, H. Wang, D. M. Kalyon, R. C. Chang, Machine learning metrology of cell confinement in melt electrowritten three-dimensional biomaterial substrates, Microsystems & nanoengineering 5 (1) (2019) 1–19.

[183] S. You, J. Guan, J. Alido, H. H. Hwang, R. Yu, L. Kwe, H. Su, S. Chen, Mitigating scattering effects in light-based three-dimensional printing using machine learning, Journal of Manufacturing Science and Engineering 142 (8) (2020) 081002.

[184] Z. A. King, J. Lu, A. Dräger, P. Miller, S. Federowicz, J. A. Lerman, A. Ebrahim, B. O. Palsson, N. E. Lewis, Bigg models: A platform for integrating, standardizing and sharing genome-scale models, Nucleic acids research 44 (D1) (2016) D515–D522.

[185] T. Oyetunde, M. Zhang, Y. Chen, Y. Tang, C. Lo, Boostgapfill: improving the fidelity of metabolic network reconstructions through integrated constraint and pattern-based methods, Bioinformatics 33 (4) (2017) 608–611.

[186] J. Alonso-Gutierrez, E.-M. Kim, T. S. Batth, N. Cho, Q. Hu, L. J. G. Chan, C. J. Petzold, N. J. Hillson, P. D. Adams, J. D. Keasling, et al., Principal component analysis of proteomics (pcap) as a tool to direct metabolic engineering, Metabolic engineering 28 (2015) 123–133.

[187] S. G. Wu, Y. Wang, W. Jiang, T. Oyetunde, R. Yao, X. Zhang, K. Shimizu, Y. J. Tang, F. S. Bao, Rapid prediction of bacterial heterotrophic fluxomics using machine learning and constraint programming, PLoS computational biology 12 (4) (2016) e1004838.

[188] S. Nandi, A. Subramanian, R. R. Sarkar, An integrative machine learning strategy for improved prediction of essential genes in escherichia coli metabolism using flux-coupled features, Molecular BioSystems 13 (8) (2017) 1584–1596.

[189] I. Schomburg, A. Chang, C. Ebeling, M. Gremse, C. Heldt, G. Huhn, D. Schomburg, Brenda, the enzyme database: updates and major new developments, Nucleic acids research 32 (suppl_1) (2004) D431–D433.

[190] R. Liu, M. C. Bassalo, R. I. Zeitoun, R. T. Gill, Genome scale engineering techniques for metabolic engineering, Metabolic engineering 32 (2015) 143–154.

[191] N. J. Treloar, A. J. Fedorec, B. Ingalls, C. P. Barnes, Deep reinforcement learning for the control of microbial co-cultures in bioreactors, PLoS computational biology 16 (4) (2020) e1007783.

[192] S. M. J. Pappu, S. N. Gummadi, Artificial neural network and regression coupled genetic algorithm to optimize parameters for enhanced xylitol

production by debaryomyces nepalensis in bioreactor, Biochemical engineering journal 120 (2017) 136–145.

[193] M. Koch, T. Duigou, J.-L. Faulon, Reinforcement learning for bioretrosynthesis, ACS Synthetic Biology 9 (1) (2019) 157–168.

[194] B. J. Kotopka, C. D. Smolke, Model-driven generation of artificial yeast promoters, Nature communications 11 (1) (2020) 1–13.

[195] M. Hutson, Artificial intelligence faces reproducibility crisis (2018).

[196] O. E. Gundersen, S. Kjensmo, State of the art: Reproducibility in artificial intelligence, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 32, 2018.

[197] B. Haibe-Kains, G. A. Adam, A. Hosny, F. Khodakarami, L. Waldron, B. Wang, C. McIntosh, A. Goldenberg, A. Kundaje, C. S. Greene, et al., Transparency and reproducibility in artificial intelligence, Nature 586 (7829) (2020) E14–E16.

[198] X. Bouthillier, C. Laurent, P. Vincent, Unreproducible research is reproducible, in: International Conference on Machine Learning, PMLR, 2019, pp. 725–734.

[199] J. Pineau, P. Vincent-Lamarre, K. Sinha, V. Larivière, A. Beygelzimer, F. d'Alché Buc, E. Fox, H. Larochelle, Improving reproducibility in machine learning research: a report from the neurips 2019 reproducibility program, Journal of Machine Learning Research 22.

[200] X. Bouthillier, G. Varoquaux, Survey of machine-learning experimental methods at neurips2019 and iclr2020, Ph.D. thesis, Inria Saclay Ile de France (2020).

[201] J. Leipzig, D. Nüst, C. T. Hoyt, K. Ram, J. Greenberg, The role of metadata in reproducible computational research, Patterns 2 (9) (2021) 100322.

[202] S. S. Alahmari, D. B. Goldgof, P. R. Mouton, L. O. Hall, Challenges for the repeatability of deep learning models, IEEE Access 8 (2020) 211860–211868.

[203] E. Raff, A step toward quantifying independently reproducible machine learning research, Advances in Neural Information Processing Systems 32.

[204] A. Sethi, A. Sankaran, N. Panwar, S. Khare, S. Mani, Dlpaper2code: Auto-generation of code from deep learning research papers, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 32, 2018.

[205] M. M. Jessop-Fabre, N. Sonnenschein, Improving reproducibility in synthetic biology, Frontiers in Bioengineering and Biotechnology 7 (2019) 18.

[206] A. Amanullah, J. M. Otero, M. Mikola, A. Hsu, J. Zhang, J. Aunins, H. B. Schreyer, J. A. Hope, A. P. Russo, Novel micro-bioreactor high throughput technology for cell culture process development: Reproducibility and scalability assessment of fed-batch cho cultures, Biotechnology and bioengineering 106 (1) (2010) 57–67.

[207] T. Fuchs, N. D. Arnold, D. Garbe, S. Deimel, J. Lorenzen, M. Masri, N. Mehlmer, D. Weuster-Botz, T. B. Brück, A newly designed automatically controlled, sterilizable flat panel photobioreactor for axenic algae culture, Frontiers in bioengineering and biotechnology 9 (2021) 566.

[208] M. P. Raphael, P. E. Sheehan, G. J. Vora, A controlled trial for reproducibility (2020).

[209] K. Roper, A. Abdel-Rehim, S. Hubbard, M. Carpenter, A. Rzhetsky, L. Soldatova, R. D. King, Testing the reproducibility and robustness of the cancer biology literature by robot, Journal of the Royal Society Interface 19 (189) (2022) 20210821.

[210] L. Teboul, Y. Herault, S. Wells, W. Qasim, G. Pavlovic, Variability in genome editing outcomes: challenges for research reproducibility and clinical safety, Molecular Therapy 28 (6) (2020) 1422–1431.

[211] K. Tiwari, S. Kananathan, M. G. Roberts, J. P. Meyer, M. U. Sharif Shohan, A. Xavier, M. Maire, A. Zyoud, J. Men, S. Ng, et al., Reproducibility in systems biology modelling, Molecular systems biology 17 (2) (2021) e9982.

[212] M. Baker, 1,500 scientists lift the lid on reproducibility, Nature 533 (7604).

[213] O. E. Gundersen, S. Shamsaliei, R. J. Isdahl, Do machine learning platforms provide out-of-the-box reproducibility?, Future Generation Computer Systems 126 (2022) 34–47.

[214] S. N. Goodman, D. Fanelli, J. P. Ioannidis, What does research reproducibility mean?, Science translational medicine 8 (341) (2016) 341ps12–341ps12.

[215] U. Dirnagl, Rethinking research reproducibility, The EMBO Journal 38 (2) (2019) e101117.

[216] R. Tatman, J. VanderPlas, S. Dane, A practical taxonomy of reproducibility for machine learning research.

[217] V. L. Porubsky, A. P. Goldberg, A. K. Rampadarath, D. P. Nickerson, J. R. Karr, H. M. Sauro, Best practices for making reproducible biochemical models, Cell systems 11 (2) (2020) 109–120.

[218] J. Thiyagalingam, M. Shankar, G. Fox, T. Hey, Scientific machine learning benchmarks, Nature Reviews Physics (2022) 1–8.

[219] M.-A. Zöller, M. F. Huber, Benchmark and survey of automated machine learning frameworks, Journal of artificial intelligence research 70 (2021) 409–472.

[220] E. Denton, A. Hanna, R. Amironesei, A. Smart, H. Nicole, M. K. Scheuerman, Bringing the people back in: Contesting benchmark machine learning datasets, arXiv preprint arXiv:2007.07399.

[221] S. Dong, D. Kaeli, Dnnmark: A deep neural network benchmark suite for gpus, in: Proceedings of the General Purpose GPUs, 2017, pp. 63–72.

[222] S. Alzahrani, B. Al-Bander, W. Al-Nuaimy, A comprehensive evaluation and benchmarking of convolutional neural networks for melanoma diagnosis, Cancers 13 (17) (2021) 4494.

[223] V. P. Dwivedi, C. K. Joshi, T. Laurent, Y. Bengio, X. Bresson, Benchmarking graph neural networks, arXiv preprint arXiv:2003.00982.

[224] Y. Hirose, N. Yoshinari, S. Shirakawa, Nas-hpo-bench-ii: A benchmark dataset on joint optimization of convolutional neural network architecture and training hyperparameters, in: Asian Conference on Machine Learning, PMLR, 2021, pp. 1349–1364.

[225] H. Zhu, M. Akrout, B. Zheng, A. Pelegris, A. Jayarajan, A. Phanishayee, B. Schroeder, G. Pekhimenko, Benchmarking and analyzing deep neural network training, in: 2018 IEEE International Symposium on Workload Characterization (IISWC), IEEE, 2018, pp. 88–100.

[226] R. V. Sharan, H. Xiong, S. Berkovsky, Benchmarking audio signal representation techniques for classification with convolutional neural networks, Sensors 21 (10) (2021) 3434.

[227] J. Xie, Q. Wang, Benchmark machine learning approaches with classical time series approaches on the blood glucose level prediction challenge, in: KHD@ IJCAI, 2018.

[228] A. Javed, B. S. Lee, D. M. Rizzo, A benchmark study on time series clustering, Machine Learning with Applications 1 (2020) 100001.

[229] K. Fauvel, V. Masson, E. Fromont, A performance-explainability framework to benchmark machine learning methods: application to multivariate time series classifiers, arXiv preprint arXiv:2005.14501.

[230] Y. Hao, X. Qin, Y. Chen, Y. Li, X. Sun, Y. Tao, X. Zhang, X. Du, Tsbenchmark: A benchmark for time series databases, in: 2021 IEEE 37th International Conference on Data Engineering (ICDE), IEEE, 2021, pp. 588–599.

[231] A. Bauer, M. Züfle, S. Eismann, J. Grohmann, N. Herbst, S. Kounev, Libra: A benchmark for time series forecasting methods, in: Proceedings of the ACM/SPEC International Conference on Performance Engineering, 2021, pp. 189–200.

[232] B. H. Menze, A. Jakab, S. Bauer, J. Kalpathy-Cramer, K. Farahani, J. Kirby, Y. Burren, N. Porz, J. Slotboom, R. Wiest, et al., The multimodal brain tumor image segmentation benchmark (brats), IEEE transactions on medical imaging 34 (10) (2014) 1993–2024.

[233] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, C. L. Zitnick, Microsoft coco: Common objects in context, in: European conference on computer vision, Springer, 2014, pp. 740–755.

[234] I. Partalas, A. Kosmopoulos, N. Baskiotis, T. Artieres, G. Paliouras, E. Gaussier, I. Androutsopoulos, M.-R. Amini, P. Galinari, Lshtc: A benchmark for large-scale text classification, arXiv preprint arXiv:1503.08581.

[235] O. Bojar, C. Buck, C. Federmann, B. Haddow, P. Koehn, J. Leveling, C. Monz, P. Pecina, M. Post, H. Saint-Amand, et al., Findings of the 2014 workshop on statistical machine translation, in: Proceedings of the ninth workshop on statistical machine translation, 2014, pp. 12–58.

[236] P. Rajpurkar, J. Zhang, K. Lopyrev, P. Liang, Squad: 100, 000+ questions for machine comprehension of text, in: EMNLP, 2016.

[237] R. S. Olson, W. La Cava, P. Orzechowski, R. J. Urbanowicz, J. H. Moore, Pmlb: a large benchmark suite for machine learning evaluation and comparison, BioData mining 10 (1) (2017) 1–13.

[238] J. D. Romano, T. T. Le, W. La Cava, J. T. Gregg, D. J. Goldberg, P. Chakraborty, N. L. Ray, D. Himmelstein, W. Fu, J. H. Moore, Pmlb v1. 0: an open-source dataset collection for benchmarking machine learning methods, Bioinformatics 38 (3) (2022) 878–880.

[239] S. Charaniya, W.-S. Hu, G. Karypis, Mining bioprocess data: opportunities and challenges, Trends in biotechnology 26 (12) (2008) 690–699.

[240] P. Grover, A. Shah, S. Sen, Mining and analysis of bioprocess data, in: Machine Learning and IoT, CRC Press, 2018, pp. 29–42.

[241] S. Rommel, A. Schuppert, Data mining for bioprocess optimization, Engineering in life sciences 4 (3) (2004) 266–270.

[242] J. S. Alford, Bioprocess control: Advances and challenges, Computers & Chemical Engineering 30 (10-12) (2006) 1464–1475.

[243] E. A. Del Rio-Chanona, N. R. Ahmed, J. Wagner, Y. Lu, D. Zhang, K. Jing, Comparison of physics-based and data-driven modelling techniques for dynamic optimisation of fed-batch bioprocesses, Biotechnology and Bioengineering 116 (11) (2019) 2971–2982.

[244] Z. Wu, B. Ramsundar, E. N. Feinberg, J. Gomes, C. Geniesse, A. S. Pappu, K. Leswing, V. Pande, Moleculenet: a benchmark for molecular machine learning, Chemical science 9 (2) (2018) 513–530.

[245] A. F. Villaverde, D. Henriques, K. Smallbone, S. Bongard, J. Schmid, D. Cicin-Sain, A. Crombach, J. Saez-Rodriguez, K. Mauch, E. Balsa-Canto, et al., Biopredyn-bench: a suite of benchmark problems for dynamic modelling in systems biology, BMC systems biology 9 (1) (2015) 1–15.

[246] A. F. Villaverde, F. Fröhlich, D. Weindl, J. Hasenauer, J. R. Banga, Benchmarking optimization methods for parameter estimation in large kinetic models, Bioinformatics 35 (5) (2019) 830–838.

[247] B. Ballnus, S. Hug, K. Hatz, L. Görlitz, J. Hasenauer, F. J. Theis, Comprehensive benchmarking of markov chain monte carlo methods for dynamical systems, BMC systems biology 11 (1) (2017) 1–18.

[248] J. Riordon, D. Sovilj, S. Sanner, D. Sinton, E. W. Young, Deep learning with microfluidics for biotechnology, Trends in biotechnology 37 (3) (2019) 310–324.