# Machine learning in physics: a short guide

Francisco A. Rodrigues

[1] *Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São, Carlos, São Paulo, Brazil*

**Abstract** – Machine learning is a rapidly growing field with the potential to revolutionize many areas of science, including physics. This review provides a brief overview of machine learning in physics, covering the main concepts of supervised, unsupervised, and reinforcement learning, as well as more specialized topics such as causal inference, symbolic regression, and deep learning. We present some of the principal applications of machine learning in physics and discuss the associated challenges and perspectives.

Ernest Rutherford once declared: "if your experiment needs statistics, you ought to have done a better experiment" [1]. His remark reflects his belief in the significance of well-controlled experiments and the need for experimental designs that minimize uncertainties and sources of errors. However, while Rutherford's statement may have merit in his time, it no longer applies in the modern scientific landscape. The growing complexity of experiments, the necessity to quantify uncertainty, the pivotal role of hypothesis testing and inference, the incorporation of statistical analysis in experimental design and power analysis, and the emergence of advanced data analysis techniques have rendered statistics an indispensable tool in contemporary scientific research [2,3]. During the last decades, statistics have empowered researchers to navigate complex data, derive valid conclusions, and make evidence-based decisions, playing an instrumental role in advancing scientific understanding [1,4,5].

In the last decade, machine learning (ML) methods have complemented the statistical analysis in Physics [2,6–9]. ML has been used in processing satellite data in atmospheric physics [10], in weather forecasts [11], predicting the behaviour of systems of many particles [6], discovering functional materials [12] and generating new organic molecules [13]. Indeed, recent works have shown that deep learning techniques outperform human-designed statistics [14], providing evidence of the power of ML for analysing experimental data. Moreover, ML can discover new physical laws and equations. For example, symbolic regression [15, 16] and sparse identification methods [17] have been used to derive physics equations from data.

Also, generative modelling offers a way to discern the most credible theory from various explanations for observational data. This is achieved solely through the data, without any predetermined understanding of the potential physical mechanisms operating within the studied system [18]. Therefore, the possibilities for using ML algorithms in physics range from experiments to theoretical analysis, opening up many opportunities.

Although ML can help to address fundamental problems in physics [9], most physicists still do not recognise the importance of these methods and how they can help to discover functional patterns in data. ML has roots in Statistics and Computer Science [19–21], with applications from image segmentation to medical diagnosis [22]. Only recently, with new methods, such as deep learning [21,23], the area has increased its potential, allowing it to work with massive data. These improvements have made ML fundamental for physics discoveries, mainly where data is present [2]. Also, many works have shown that Physics has the potential to develop new ML methods and help understand the "black boxes" algorithms, such as neural networks [24]. Despite its potential, it is hard for physicists to grasp the immense literature available. Also, machine learning methods must be used carefully since wrong modelling choices or assumptions can result in misleading or unreliable conclusions [3]. Machine learning and statistical models should always be interpreted cautiously, considering the limitations of the data and potential pitfalls of the algorithms to avoid drawing erroneous or misleading inferences. For example, the p-value has been a subject of debate and criticism in scientific research due to several

problems associated with its interpretation and use [25]. The primary objective of this review is to introduce machine learning methods to physicists, presenting both the fundamental concepts and concrete examples of their application.

This review delves into the fundamental concepts of ML, offers primary references, and outlines the steps to employ ML methods for physics discovery. We present the fundamental concepts, research papers, applications, and tools for utilizing ML in the field of physics while also exploring how physics can contribute to the development of ML.

**Basic concepts:** The main goal of ML is to find useful patterns in data [19, 20]. In Physics, ML has been used when we have complex problems and lots of data [3]. For example, in the Large Hadron Collider (LHC) experiments at CERN, the collisions of protons or ions in several places around the circular collider generate tens of thousands of PetaBytes per year [26]. However, since collisions are rare, it is necessary to classify the actual collisions as either interesting or uninteresting. Thus, ML methods filter the signal from the noise and help to search for new fundamental physics [9].

ML can be divided into three main categories [19, 20], namely: (i) supervised learning, (ii) unsupervised learning and (iii) reinforcement learning. Supervised learning involves learning from labelled data to make predictions or classifications [19], unsupervised learning discovers patterns and structures in unlabeled data [20], and reinforcement learning focuses on an agent learning optimal behaviours through interactions with an environment and rewards [27].

**Supervised learning:** Supervised learning involves training a model based on a given set of examples [3, 20]. Initially, we split the data set into two disjoint parts, one for training and another for testing. The training set is used to train a ML algorithm, which will learn the patterns in the data and then will be used to predict new observations on the test set. The data is composed of a set of examples, which are represented by the vectors $\{\mathbf{x}_i, y_i\}$, $i = 1, 2, \ldots, n$, where $\mathbf{x}_i$ is the set of attributes of the example $i$ and $y_i$ is the target variable we want to predict, which can be a single value or a vector. For example, to predict the ordered phase in the Ising model, $\mathbf{x}_i$, $i = 1, 2, \ldots, n$, can represent the spin configuration in a 2D lattice and $y_i$ is the phase, i.e., $y_i = 1$ represents an ordered phase and $y_i = 0$ a disordered one [28]. When the target variable $y \in \mathbb{R}$, it indicates a regression problem. Conversely, if $y$ represents a class or label ($y \in \{C_1, C_2 \ldots, C_k\}$, where $C_j$ is a class), it becomes a classification problem. Predicting the ordered phase in the Ising model involves classification while estimating the time required to reach the ordered phase corresponds to a regression. Figure 1 summarizes the supervised learning process.

ML algorithms aim to fit a model $f$ that maps the attributes to the target variable [29]. To elaborate, it is possible to fine-tune the function $f$ using input attributes

denoted as $\mathbf{x}$ along with the target variable $y$. This fine-tuning process, performed during training, enables the derivation of the model's parameter set $\theta$, thus facilitating accurate predictions of $y$. Mathematically,

$$y = f(\mathbf{x}, \theta) + \epsilon, \tag{1}$$

where $\epsilon$ is the random error ($\langle \epsilon \rangle = 0$ and $\langle \epsilon_i \epsilon_j \rangle = 0$, $i, j = 1, 2, \ldots, n$, where $n$ is the number of observations in the training set). We can refer to $f(\mathbf{x}, \theta)$ as the function that generates the data, and the goal of the ML algorithm is to estimate its set of parameters $\theta$ so that we can predict new observations as accurately as possible.

Generally, optimization methods are used to adjust a supervised learning model [30]. In this case, we have to define a training loss function that will be minimized. This function can have different forms and depend on the type of learning process [22]. In general, the loss function $L$ compares the values of the target variable $y$ and the respective predictions $\hat{y}$. The average loss of the predictor on the training set is called the cost function, or empirical risk in decision theory [22],

$$C(\hat{y}, y) = \frac{1}{n} \sum_{i=1}^{n} L(y_i, \hat{y}_i). \tag{2}$$

In the case of classification, the choice of cost functions depends on the specific algorithm and objective [19, 22, 31, 32]. Typical examples include the 0-1 loss function and cross-entropy [22]. The 0-1 loss function measures the misclassification rate on the training set,

$$C(\hat{y}, y) = \frac{1}{n} \sum_{i=1}^{n} I(\hat{y}_i \neq y_i), \tag{3}$$

where $I$ is the indicator function which returns one if and only if $y = \hat{y}$.

In the case of regression, typical cost functions use the mean squared error loss [22]:

$$\text{MSE}(y, \hat{y}) = \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2, \tag{4}$$

or the mean absolute error loss [22]. Other functions are also possible (e.g. [22, 32]).

Thus, the model training process aims to discover a configuration of model parameters denoted as $\theta$, which serves to minimize the cost function when applied to the training dataset, i.e.,

$$\theta = \text{argmin}_\theta \frac{1}{n} \sum_{i=1}^{n} L(y_i, \hat{y}_i). \tag{5}$$

Different algorithms can be applied to minimize the loss function, and their effectiveness depends on the problem and the algorithm's characteristics [33]. Indeed, the "No Free Lunch" theorem states that, on average, no algorithm
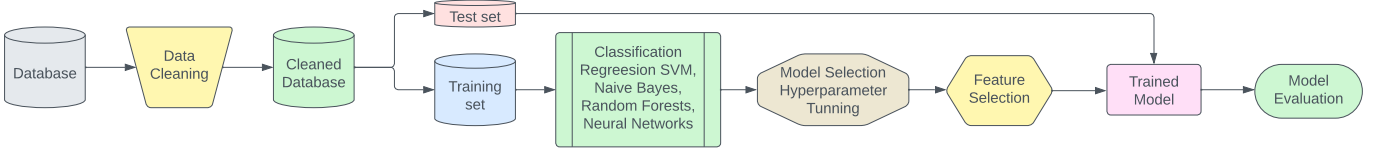
Fig. 1: The supervised learning pipeline. Typically, the data is divided into two sets. About 80% of the data is allocated to the training set, where the model learns patterns and relationships from the data. The remaining 20% forms the test set, which is an independent evaluation to gauge the model's performance and generalization abilities.

performs better than any other when considering all possible problems [34]. It implies that the effectiveness of an algorithm is highly dependent on the specific characteristics of the problem at hand [34]. These algorithms are based on different concepts [20], such as probability theory (e.g. naive Bayes and logistic regression), decision trees, optimization techniques (e.g. support vector machines), neural networks and ensemble methods (e.g. random forests). All these methods can be used both for classification and regression.

Naive Bayes is a probabilistic classification algorithm based on Bayes' theorem [35]. Given the class label, it assumes that the features are conditionally independent, hence the "naive assumption". It is computationally efficient and works well with high-dimensional data. On the other hand, logistic regression is a linear classification algorithm that models the relationship between the input variables and the probability of belonging to a specific class [36]. It is commonly used when the target variable is binary or categorical. Logistic regression can handle both numerical and categorical input features, and it provides interpretable coefficients that indicate the influence of each feature on the prediction [22]. Support Vector Machines (SVMs) are a powerful supervised learning algorithm for classification and regression tasks [37, 38]. SVMs aim to find an optimal hyperplane that separates the data points of different classes with the largest margin [37, 38]. They can handle both linear and non-linear relationships using different kernel functions. Also based on optimization, neural networks, inspired by the structure and function of biological neural networks, are versatile models capable of learning complex patterns and relationships from data [39]. They consist of interconnected layers of artificial neurons that process information. Hebbian learning is used in neural networks to strengthen the connections between activated neurons, allowing the network to learn patterns from data [40]. Deep neural networks with many hidden layers have led to breakthroughs in various domains, including computer vision, natural language processing, and speech recognition [21, 23].

Finally, random forests (RF) are an ensemble learning method that combines multiple decision trees to make predictions [41]. Using a random subset of the data and features, RF reduces overfitting, which occurs when a model becomes excessively fine-tuned to the training data, capturing noise and anomalies rather than the underlying patterns, leading to poor performance on unseen data. Random forests handle complex relationships, missing values, and identifying essential features, acting as a feature ranking algorithm [20].

We can employ various metrics to evaluate the performance of machine learning algorithms [32]. While accuracy, which returns the fraction of correct predictions, is a commonly used measure for classification, caution is needed when dealing with unbalanced data as it can be misleading [42, 43]. Additional vital metrics for binary classification include precision (proportion of true positive predictions out of all positive predictions), recall (proportion of true positive predictions out of all actual positive instances), F1 score (harmonic mean of precision and recall), and the area under the ROC (Receiver Operating Characteristic) curve [32]. In regression tasks, commonly used metrics include mean squared error (equation (4)), mean absolute error, and the coefficient of determination ($R^2$) [35], which measures the proportion of the variance in the target variable explained by the model. It is calculated as the square of the Pearson correlation coefficient between the actual and predicted values [35].

Unlike traditional methods used in Physics to analyze data, ML does not use curve fitting [35]. Curve fitting aims to find the best fit for the existing data points, focusing on data approximation. However, minimizing the error in the training data is not the goal of ML algorithms. Training in machine learning refers to optimizing a model's parameters to predict unseen data by learning from labelled training examples. Curve fitting is related to overfitting [19], which occurs when a model excessively performs well on the training data but fails to generalize to unseen data. On the other hand, when the model is too simple and lacks the flexibility to capture the underlying patterns in the data, we have an underfitting [39]. The goal of machine learning algorithms is to find the balance between underfitting (high bias) and overfitting (high variance) in modelling data [35]. Bias refers to the error introduced by approximating a real-world problem with a simplified model. Variance, on the other hand, represents the variability of model predictions for different training datasets. The bias-variance tradeoff arises because reducing bias often increases variance, and vice versa [35]. Finding the optimal tradeoff is crucial for building models that generalize well to new data. The objective is to strike a balance where the model is complex enough to capture the un-

derlying patterns but not overly complex to fit noise or irrelevant details.

There are many techniques to avoid overfitting [44]. Cross-validation is a resampling technique that can decrease variance and assist in model selection and adjustment of hyperparameters [32]. Many ML models present several hyperparameters, which are values provided by the user. For instance, the number of layers in a neural network or the number of trees in the random forest algorithm are examples of hyperparameters. One common approach is the k-fold cross-validation. This method divides the dataset into k subsets or folds of approximately equal size. The model is then trained and evaluated k times, using a different fold as the validation set and the remaining folds as the training set. The performance metrics, such as accuracy or mean squared error, are recorded for each fold. The final performance measure is usually computed by averaging the results across the k iterations. Cross-validation helps decrease variance by providing a more robust estimate of the model's performance. It reduces the dependency on a single training-test split, which can be biased due to the randomness present in the data. By training and evaluating the model on multiple subsets of the data, cross-validation provides a more representative evaluation of the model's performance on unseen data.

Regularization techniques, such as L1 (Lasso) and L2 (Ridge) regression, can also avoid overfitting [22], which adds a penalty term to the loss function during model training. This penalty discourages excessive complexity in the model by shrinking the magnitude of the coefficients. Regularization helps prevent overfitting by promoting simpler models that generalize well to unseen data. Dropout is a regularization technique commonly used in neural networks [21]. It involves randomly disabling a fraction of the neurons during each training iteration. This helps prevent the network from relying too heavily on specific neurons or memorizing noise in the training data, reducing overfitting and promoting more robust generalization. Other methods to avoid overfitting include dropout in neural networks, early stopping, feature selection and dimensionality reduction, ensemble methods and data augmentation, which are techniques to artificially expand the training dataset by applying transformations, such as rotation, scaling, or flipping, to the existing data. By addressing overfitting, machine learning models can achieve better performance, improve robustness, and provide more accurate predictions in real-world applications.

Supervised learning methods in Physics are mainly used in problems that involve classification, regression and time series forecasting [6]. For example, supervised learning algorithms have been used to classify particles in particle physics experiments, predict the Higgs boson's mass, and weather and climate forecasting [3, 6, 9].

**Unsupervised learning:** Unlike supervised learning, unsupervised learning is a type of machine learning where the algorithm learns patterns or structures from unlabeled data without any explicit target variable [19, 20]. The goal is to discover hidden patterns, relationships, or clusters within the data. Examples of unsupervised learning methods include (i) clustering algorithms like k-means and hierarchical clustering; (ii) dimensionality reduction techniques, such as principal component analysis (PCA) and t-SNE; (iii) anomaly detection approaches, like one-class SVM and Isolation Forest; (iv) association rule learning, which are used to discover associations or patterns in large datasets, often applied to market basket analysis or recommendation systems; (v) generative models, like Gaussian Mixture Models (GMM) and Variational Autoencoders (VAEs), which learn the underlying data distribution and can generate new samples similar to the training data; (vi) topic modelling, which can be used to discover hidden topics or themes within a collection of documents; and (vii) density estimation, which can estimate the underlying probability distribution of a dataset.

Unsupervised learning methods can be applied in various ways in physics. Principal component and clustering analysis can be used to identify phases and phase transitions of many-body systems [45]. Dimensionality reduction methods can assist in visualizing and understanding high-dimensional experimental or simulation data [3]. Anomaly detection algorithms can identify rare or unexpected events, outliers, or anomalies in experimental data, helping physicists to find potential errors, instrument malfunctions, or novel phenomena [6]. Generative models can generate synthetic data that matches the statistical properties of experimental observations, enabling model validation and exploration of new scenarios [18].

Although with great potential to be applied in physics, data clustering presents several challenges [46]. Firstly, the absence of ground truth labels makes evaluating clustering results subjective and dependent on heuristic measures. Secondly, determining the optimal number of clusters and selecting appropriate algorithms can be subjective and context-dependent [46]. Additionally, the curse of dimensionality poses difficulties as higher-dimensional spaces make it harder to distinguish meaningful clusters [46]. Complex and irregular data structures, scalability issues, and the interpretation of clustering results further contribute to the challenges.

Overcoming these hurdles requires careful consideration of dataset characteristics, algorithm selection, parameter tuning, and the incorporation of domain knowledge [46]. Ongoing research focuses on developing robust and scalable clustering methods to address the complexities of real-world data. The evaluation is also problematic since metrics like cluster purity, silhouette coefficient or adjusted rand index struggle to capture complex cluster structures, such as overlapping clusters, varying cluster densities, or non-spherical shapes [20]. Also, different clustering metrics may yield inconsistent results, making it challenging to compare and interpret the performance of different algorithms. Developing more robust and versatile clustering evaluation metrics is an ongoing research area, also for physicists [24].

**Semi-supervised learning (SSL):** Unlike traditional supervised learning, where a large labelled dataset is required for training, semi-supervised learning leverages a combination of labelled and unlabeled data. SSL can be a powerful tool for improving the accuracy of machine learning models. This is because the unlabeled data can be used to regularize the model, preventing overfitting. In Physics, SSL has been used, for example, to classify materials synthesis procedures [47] and detect distinct events in a large dataset of in tokamak discharges [48]. The main algorithms for SSL [49] are (i) self-training, which can take any supervised method for classification or regression and modify it to work in a semi-supervised manner, taking advantage of labeled and unlabeled data; (ii) transductive SVM, which is a variation of support vector machines (SVMs) that is specifically designed for semi-supervised learning; (iii) label propagation, which assigns labels to unlabeled data by propagating labels from labeled data points to unlabeled data points that are similar to them; and (iv) ensemble methods, which combines multiple semi-supervised learning algorithms on different subsets of the data, and then their predictions are combined to make a final prediction.

**Reinforcement learning (RL):** In contrast to conventional machine learning techniques, reinforcement learning (RL) enables the extraction of knowledge from real-world experiences, surpassing the limitations of training data alone [27]. RL focuses on training agents to make sequential decisions in an interactive environment. Through trial and error, agents explore and exploit the environment, receiving rewards or penalties based on their actions. The goal is to learn an optimal policy that maximizes cumulative rewards. RL is used in scenarios without labelled training data and has applications in robotics, game-playing, and recommendation systems. For example, a RL system can master the game of chess solely from its rules, devoid of any preceding knowledge. By engaging in matches against adversaries or even self-play, the system progressively learns and hones its skills [27].

The main elements in RL are an agent and an environment it interacts with [49]. The environment provides the agent with information and feedback according to the agent's actions. The agent's primary goal is to maximize the obtained rewards the environment provides. Thus, the RL algorithms aim to learn an optimal policy maximising rewards. Given the observations, this policy defines the actions to take, thereby defining the agent's strategy. In this case, Markov Decision Process is a mathematical framework used to solve these decision-making problems [27,50]. The main RL algorithms use Q-learning, which continuously learns the optimal action-value function regardless of the policy followed during the training. This algorithm has many versions and can be implemented in neural networks [50]. In physics, reinforcement learning is utilized for tasks such as control of quantum systems [51], create new experiments [52], and discovering novel materials [53].

**Deep Learning:** While classic supervised learning relies on manual feature engineering and simpler models, deep learning leverages deep neural networks to automatically learn representations from the data, making it capable of capturing complex patterns and achieving state-of-the-art performance in various domains [21,23]. A neural network is a computational model composed of interconnected nodes, called neurons, organized into layers. Each neuron processes incoming information and produces an output, which becomes the input for other neurons in subsequent layers [39]. Through training, neural networks learn to adjust the weights of connections between neurons to recognize patterns and relationships in complex data effectively. This learning process enables them to make predictions, classify data, and solve various problems, such as image and speech recognition, natural language processing, and game playing [23].

With their ability to learn from examples and generalize from the data, neural networks have become a cornerstone of modern AI applications, driving remarkable advancements and innovations across numerous domains [23]. Many libraries are available to use in deep learning, including (i) TensorFlow [54], which is an open-source software library powered by Google Brain, (ii) PyTorch [55], which Facebook originally developed; (iii) Scikit-learn [56], which provides a wide range of algorithms for supervised, unsupervised, and reinforcement learning. Deep learning methods can be used for all the tasks discussed previously.

Deep neural networks can also predict time series [57]. The most popular approaches include (i) recurrent neural networks, (ii) convolutional neural networks and (iii) transformers. Mainly, transformers, although developed for natural language processing tasks, treat the time series data as a sequence of words or characters. These models have been used to predict dynamical systems representative of physical phenomena [58].

**Physics informed machine learning:** Deep neural networks have also been used to solve partial differential equations, which enable scientific prediction and discovery from incomplete models and incomplete data [59]. In this approach, called physics-informed machine learning, models are trained on both data and physical principles. More specifically, the cost function is changed to include a term that penalizes the model for violating the physical principles [60]. For example, if we are trying to model the behaviour of a fluid, we might add the Navier-Stokes equations as a constraint to the cost function. This ensures that the machine learning model is consistent with the known laws of physics, which can help to improve the accuracy of the model. Therefore, this approach combines deep learning with prior knowledge about the fundamental laws and principles of physics to create more accurate and reliable models of the physical world [59,60].

**Physical discovery:** Machine learning methods have also been used for physics discovery. Mainly, symbolic regression searches the space of mathematical expressions to find the model that best fits a given dataset [17,61]. No

particular model is provided as a starting point for symbolic regression. For example, when applied to 100 equations from the Feynman Lectures on Physics, symbolic regression discovered all of them [61]. Most methods for symbolic regression are based on genetic algorithms [17].

**Causal inference:** Most machine learning methods do not consider the causal relationships between variables. Only recently, causal machine learning methods have been designed to identify the causal relationships between variables and to use this information to make better predictions (e.g. [62]). Causal inference methods can make causal claims about the world, even with confounding variables. In physics, causality can be used to infer the connection between variables. For instance, in complex systems, causality methods have been used to infer the structure of the underlying system, like in the brain [63] and climate systems [62].

**Perspectives** While ML has demonstrated successful applications in the realm of Physics, several challenges and adjustments still must be addressed [3,6,50]. A recurring hurdle is the scarcity of data suitable for training ML models, often accompanied by the predicament of imbalanced data—where significant class imparities exist, hampering model accuracy. Furthermore, scalability emerges as a concern, given the computationally intensive nature of training and deploying ML models, particularly for vast datasets [3]. The inherent opacity of ML model decision-making poses an additional obstacle in comprehending their predictions [24]. In these scenarios, physicists can rise to the occasion, devising strategies to surmount these obstacles and tailoring approaches for enhanced knowledge extraction within physical systems [24]. Particularly, statistical physics can shed some light on these problems, contributing to the development of ML [24,64].

**Additional material:** We made available a Jupyter notebook to put in practice most of the concepts presented here, mainly associated to supervised and unsupervised learning. The code can be used as a starting guide for analysing data in Python. The reader can access the code in the following link:
`https://github.com/franciscorodrigues-usp/MLP`

\* \* \*

REFERENCES

[1] David L Vaux. Basic statistics in cell biology. *Annual review of cell and developmental biology*, 30:23–37, 2014.
[2] Lenka Zdeborová. New tool in the box. *Nature Physics*, 13(5):420–421, 2017.
[3] Pankaj Mehta, Marin Bukov, Ching-Hao Wang, Alexandre GR Day, Clint Richardson, Charles K Fisher, and David J Schwab. A high-bias, low-variance introduction to machine learning for physicists. *Physics Reports*, 810:1–124, 2019.
[4] Roger J Barlow. *Statistics: a guide to the use of statistical methods in the physical sciences*, volume 29. John Wiley & Sons, 1993.
[5] Robert D Cousins. Why isn't every physicist a bayesian? *American Journal of Physics*, 63(5):398–410, 1995.
[6] Giuseppe Carleo, Ignacio Cirac, Kyle Cranmer, Laurent Daudet, Maria Schuld, Naftali Tishby, Leslie Vogt-Maranto, and Lenka Zdeborová. Machine learning and the physical sciences. *Reviews of Modern Physics*, 91(4):045002, 2019.
[7] Roger Guimerà, Ignasi Reichardt, Antoni Aguilar-Mogas, Francesco A Massucci, Manuel Miranda, Jordi Pallarès, and Marta Sales-Pardo. A bayesian machine scientist to aid in the solution of challenging scientific problems. *Science advances*, 6(5):eaav6971, 2020.
[8] Elena Agliari, Adriano Barra, Peter Sollich, and Lenka Zdeborová. Machine learning and statistical physics: preface. *Journal of Physics A: Mathematical and Theoretical*, 53:500401, 2020.
[9] Georgia Karagiorgi, Gregor Kasieczka, Scott Kravitz, Benjamin Nachman, and David Shih. Machine learning in the search for new fundamental physics. *Nature Reviews Physics*, 4(6):399–412, 2022.
[10] Anuj Karpatne, Imme Ebert-Uphoff, Sai Ravela, Hassan Ali Babaie, and Vipin Kumar. Machine learning for the geosciences: Challenges and opportunities. *IEEE Transactions on Knowledge and Data Engineering*, 31(8):1544–1554, 2018.
[11] Nicola Jones. How machine learning could help to improve climate forecasts. *Nature*, 548(7668), 2017.
[12] Prasanna V Balachandran. Machine learning guided design of functional materials with targeted properties. *Computational Materials Science*, 164:82–90, 2019.
[13] Keith T Butler, Daniel W Davies, Hugh Cartwright, Olexandr Isayev, and Aron Walsh. Machine learning for molecular and materials science. *Nature*, 559(7715):547–555, 2018.
[14] Zoltán Haiman. Learning from the machine. *Nature Astronomy*, 3(1):18–19, 2019.
[15] Silviu-Marian Udrescu and Max Tegmark. Ai feynman: A physics-inspired method for symbolic regression. *Science Advances*, 6(16):eaay2631, 2020.
[16] Ziming Liu and Max Tegmark. Machine learning conservation laws from trajectories. *Physical Review Letters*, 126(18):180604, 2021.
[17] Steven L Brunton, Joshua L Proctor, and J Nathan Kutz. Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *Proceedings of the National Academy of Sciences*, 113(15):3932–3937, 2016.
[18] Kevin Schawinski, M Dennis Turp, and Ce Zhang. Exploring galaxy evolution with generative models. *Astronomy & Astrophysics*, 616:L16, 2018.
[19] Christopher M Bishop and Nasser M Nasrabadi. *Pattern recognition and machine learning*, volume 4. Springer, 2006.
[20] Kevin P Murphy. *Machine learning: a probabilistic perspective*. MIT Press, 2012.
[21] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.

[22] Kevin P Murphy. *Probabilistic machine learning: an introduction.* MIT press, 2022.

[23] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015.

[24] Lenka Zdeborová. Understanding deep learning is also a job for physicists. *Nature Physics*, 16(6):602–604, 2020.

[25] Jeffrey T Leek and Roger D Peng. Statistics: P values are just the tip of the iceberg. *Nature*, 520(7549):612–612, 2015.

[26] Andy Buckley, Alistair Shilton, and MJ White. Fast supersymmetry phenomenology at the large hadron collider using machine learning techniques. *Computer Physics Communications*, 183(4):960–970, 2012.

[27] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction.* MIT press, 2018.

[28] Juan Carrasquilla and Roger G Melko. Machine learning phases of matter. *Nature Physics*, 13(5):431–434, 2017.

[29] Pedro Domingos. A few useful things to know about machine learning. *Communications of the ACM*, 55(10):78–87, 2012.

[30] Ajay Shrestha and Ausif Mahmood. Review of deep learning algorithms and architectures. *IEEE access*, 7:53040–53065, 2019.

[31] Lorenzo Rosasco, Ernesto De Vito, Andrea Caponnetto, Michele Piana, and Alessandro Verri. Are loss functions all the same? *Neural computation*, 16(5):1063–1076, 2004.

[32] Sebastian Raschka. Model evaluation, model selection, and algorithm selection in machine learning. *arXiv preprint arXiv:1811.12808*, 2018.

[33] Ruoyu Sun. Optimization for deep learning: theory and algorithms. *ArXiv preprint arXiv:1912.08957*, 2019.

[34] Stavros P Adam, Stamatios-Aggelos N Alexandropoulos, Panos M Pardalos, and Michael N Vrahatis. No free lunch theorem: A review. *Approximation and Optimization: Algorithms, Complexity and Applications*, pages 57–82, 2019.

[35] James Gareth, Witten Daniela, Hastie Trevor, and Tibshirani Robert. *An introduction to statistical learning: with applications in R.* Spinger, 2013.

[36] Sandro Sperandei. Understanding logistic regression analysis. *Biochemia medica*, 24(1):12–18, 2014.

[37] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20:273–297, 1995.

[38] William S Noble. What is a support vector machine? *Nature biotechnology*, 24(12):1565–1567, 2006.

[39] Christopher M Bishop et al. *Neural networks for pattern recognition.* Oxford university press, 1995.

[40] Francesco Alemanno, Miriam Aquaro, Ido Kanter, Adriano Barra, and Elena Agliari. Supervised hebbian learning. *Europhysics Letters*, 141(1):11001, 2023.

[41] Leo Breiman. Random forests. *Machine learning*, 45:5–32, 2001.

[42] Le Wang, Meng Han, Xiaojuan Li, Ni Zhang, and Haodong Cheng. Review of classification methods on unbalanced data sets. *IEEE Access*, 9:64606–64628, 2021.

[43] Pradeep Kumar, Roheet Bhatnagar, Kuntal Gaur, and Anurag Bhatnagar. Classification of imbalanced data: review of methods and applications. In *IOP conference series: materials science and engineering*, volume 1099, page 012077. IOP Publishing, 2021.

[44] Douglas M Hawkins. The problem of overfitting. *Journal of chemical information and computer sciences*, 44(1):1–12, 2004.

[45] Lei Wang. Discovering phase transitions with unsupervised learning. *Physical Review B*, 94(19):195105, 2016.

[46] Mayra Z Rodriguez, Cesar H Comin, Dalcimar Casanova, Odemir M Bruno, Diego R Amancio, Luciano da F Costa, and Francisco A Rodrigues. Clustering algorithms: A comparative approach. *PloS one*, 14(1):e0210236, 2019.

[47] Haoyan Huo, Ziqin Rong, Olga Kononova, Wenhao Sun, Tiago Botari, Tanjin He, Vahe Tshitoyan, and Gerbrand Ceder. Semi-supervised machine-learning classification of materials synthesis procedures. *Npj Computational Materials*, 5(1):62, 2019.

[48] Kevin Joseph Montes, Cristina Rea, Roy Alexander Tinguely, Ryan Sweeney, Jinxiang Zhu, and RS Granetz. A semi-supervised machine learning detector for physics events in tokamak discharges. *Nuclear Fusion*, 61(2):026022, 2021.

[49] YCA Padmanabha Reddy, P Viswanath, and B Eswara Reddy. Semi-supervised learning: a brief review. *International Journal of Engineering & Technology*, 7(1.8):81, 2018.

[50] Anna Dawid, Julian Arnold, Borja Requena, Alexander Gresch, Marcin Płodzień, Kaelan Donatella, Kim A Nicoli, Paolo Stornati, Rouven Koch, Miriam Büttner, et al. Modern applications of machine learning in quantum sciences. *arXiv preprint arXiv:2204.04198*, 2022.

[51] Thomas Fösel, Petru Tighineanu, Talitha Weiss, and Florian Marquardt. Reinforcement learning with neural networks for quantum feedback. *Physical Review X*, 8(3):031084, 2018.

[52] Alexey A Melnikov, Hendrik Poulsen Nautrup, Mario Krenn, Vedran Dunjko, Markus Tiersch, Anton Zeilinger, and Hans J Briegel. Active learning machine learns to create new quantum experiments. *Proceedings of the National Academy of Sciences*, 115(6):1221–1226, 2018.

[53] Seyed Mohamad Moosavi, Kevin Maik Jablonka, and Berend Smit. The role of machine learning in the understanding and design of materials. *Journal of the American Chemical Society*, 142(48):20273–20287, 2020.

[54] Bo Pang, Erik Nijkamp, and Ying Nian Wu. Deep learning with tensorflow: A review. *Journal of Educational and Behavioral Statistics*, 45(2):227–248, 2020.

[55] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

[56] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *the Journal of machine Learning research*, 12:2825–2830, 2011.

[57] Bryan Lim and Stefan Zohren. Time-series forecasting with deep learning: a survey. *Philosophical Transactions of the Royal Society A*, 379(2194):20200209, 2021.

[58] Nicholas Geneva and Nicholas Zabaras. Transformers for modeling physical systems. *Neural Networks*, 146:272–289, 2022.

[59] Maziar Raissi, Paris Perdikaris, and George E Karniadakis. Physics-informed neural networks: A deep learn-

ing framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational physics*, 378:686–707, 2019.

[60] George Em Karniadakis, Ioannis G Kevrekidis, Lu Lu, Paris Perdikaris, Sifan Wang, and Liu Yang. Physics-informed machine learning. *Nature Reviews Physics*, 3(6):422–440, 2021.

[61] Silviu-Marian Udrescu and Max Tegmark. Ai feynman: A physics-inspired method for symbolic regression. *Science Advances*, 6(16):eaay2631, 2020.

[62] Jakob Runge, Vladimir Petoukhov, Jonathan F Donges, Jaroslav Hlinka, Nikola Jajcay, Martin Vejmelka, David Hartman, Norbert Marwan, Milan Paluš, and Jürgen Kurths. Identifying causal gateways and mediators in complex spatio-temporal systems. *Nature Communications*, 6(1):8502, 2015.

[63] Caroline L Alves, Thaise GL de O Toutain, Patricia de Carvalho Aguiar, Aruane M Pineda, Kirstin Roster, Christiane Thielemann, Joel Augusto Moura Porto, and Francisco A Rodrigues. Diagnosis of autism spectrum disorder based on functional brain networks and machine learning. *Scientific Reports*, 13(1):8072, 2023.

[64] Elena Agliari, Linda Albanese, Adriano Barra, and Gabriele Ottaviani. Replica symmetry breaking in neural networks: a few steps toward rigorous results. *Journal of Physics A: Mathematical and Theoretical*, 53(41):415005, 2020.