ABC to Machine Learning

Supervisor: Dr Gareth Conduit[1]

[1]*Department of Physics, Cambridge University, Cavendish Laboratory, 19 JJ Thomson Ave, Cambridge CB3 0HE*

This literature review introduces popular supervised machine learning methods from regression to the physics-informed neural network and their perspective strength and limitations. Then two case studies of applications of machine learning in various fields of physics research such as material science and astronomy are discussed.

I. INTRODUCTION

Machine learning is an intriguing field of study that empowers computer systems to learn from data without being explicitly programmed. Its applications encompass various functions such as prediction, classification, and pattern detection[1]. Machine learning can be subdivided into three categories: supervised, unsupervised, and reinforcement learning. This review will primarily focus on supervised learning, as it is the most prevalent machine learning technique within research for making predictions. Some of the most commonly seen supervised machine learning methods include regression, neural work/deep neural network, decision tree, gaussian process, and physics-informed neural network. Their strengths and limitations, along with their applications will be discussed in this review.

II. SUPERVISED LEARNING

Supervised learning is a machine learning technique that uses labeled data to train a model to make predictions. The model is trained by comparing the predicted output with the true output and then adjusting the parameters of the model to minimize the error. The model is then tested on a test set to evaluate its performance[2]. Supervised learning breaks down into two different types of problems: regression and classification. Regression is the task of predicting a continuous value, such as a certain physical property of a material, while classification is the task of predicting a discrete value, such as the classification of astronomical objects into galaxies and stars. Due to the nature of their output, regression and classification have different methods of evaluation. For regression, the most common evaluation metric is the mean squared error (MSE), which is the average of the squared difference between the predicted value and the true value. For classification, the most common evaluation metric is accuracy, which is the fraction of the data points that are correctly classified.

III. SUPERVISED MACHINE LEARNING METHODS

There is a list of various machine learning methods. In this section, we will discuss some of the most common machine learning techniques used in research, including regression, neural network, decision tree, gaussian process and finally, physics-informed machine learning.

A. Regression Methods

Although the idea of machine learning was first introduced in 1959 [3], the most fundamental technique involved in machine learning dates way back to the 18th century when the first curve-fitting technique was introduced. Curve fitting is a process of construction a mathematics function, that has the best fit to a series of data points. It remains one of the most theoretically challenging parts of machine learning[4].

a. Linear regression The most basic and commonly seen fitting technique is a first-order polynomial equation, also known as linear regression:

$$y = ax + b \tag{1}$$

which is a straight line that connects two points with distinct x coordinates.

b. Polynomial Regression and Taylor Theorem With more than two data points to fit, we could always add a term of higher power of x, to make it a quadratic equation

$$y = ax^2 + bx + c \tag{2}$$

or another term to construct a cubic regression:

$$y = ax^3 + bx^2 + cx + d \tag{3}$$

In general, many data are not related by a simple linear relationship, we can construct a polynomial regression of any order, by adding more terms to the equation. This method is identical to Taylor expansion around a certain point

$$f(x) = \sum_{n=0}^{\infty} a_n (x-b)^n$$

The find the value of the parameters, we mini mize the ordinary least squares:

$$\sum y_i - (kx_i + c)^2 \tag{4}$$

The limitation of Taylor expansion comes when the x becomes infinitely large, which will cause the magnitude of y to become infinitely large, which may not reflect the datasets properly.

c. Padé approximant As a further improvement to Taylor expansion, Padé approximant was developed by Henri Padé around the year 1890 to fit curves using the ratio of two polynomials[5]. An $[N/M]$ Pade approximant is formed of a Nth-degree polynomial on the numerator and an Mth-degree polynomial on the denominator:

$$P(x) = \frac{a_0 + a_1 x + a_2 x^2 + ...a_N x^N}{b_0 + b_1 x + b_2 x^2 + ...b_M x^M} \tag{5}$$

[6] Padé approximant $\frac{ax^2+bx^3+...}{c+dx+...+x^6}$ making sure that $f(x)$ does not tend to infinity at large x, in this case, tend to $1/x$.
Padé approximant is superior to the Taylor series when describing a function that contains poles. Also, by dividing a polynomial by another, the Pade approximant prevents the function from diverging by letting $N <= M$. However, the Padé approximant is limited to low dimensional data and little noise[7]. Pade Approximant inspired the development of neural network[8], a systematic method to introduce non-linearity into the curve-fitting model, which also is the next topic we will discuss.

B. Neural network

In the previous cases, we discussed regression techniques for a single variable function, while in physics, a function is usually higher dimensional. The limitation comes in when the number of independent variables becomes more than 1. For example, y is now a function of x_1 and x_2. i.e. $y(x_1, x_2)$. In this case, we would have to include a term such as $x_1 x_2$ and $x_1 x_2$, which means that the number of coefficients we used now grows exponentially to the number of independent variables. This is where the neural network comes in and solves the problem. A neural network consists of a series of nodes, each with its own weight and bias, and the results are fed into a non-linear activation function, such as a sigmoid function, to produce the output. A model is built with one layer of nodes and its weights and bias is usually randomly assigned or set to zero. After assigning weights and biases in the model, the next step is to reduce the loss function in the neural network. Back-propagation is a way of propagating the total loss back into the neural network to know how much of the loss every node is responsible for, and subsequently updating the weights in a way that minimizes the loss by giving the nodes with higher error rates lower weights, and vice versa. This process is repeated until the loss function is minimized

a. Deep Neural Network Neural networks (NN) have a large number of tunable parameters, all of which fall into two categories[9]. The first type is internal parameters, weights w_i and bias b. The weights and biases are selected via an optimization routine (e.g., gradient descent) over a chosen metric (e.g., mean squared error). These two parameters are discussed in the previous section. The hyperparameters include width, which is the number of nodes per layer, and depth, which is the number of layers stacked to form the network. The depth and width of neural networks are usually pre-set by developers, though a few combinations can be tried to obtain the best model. A deeper and wider neural network generally has a better performance than a shadower and narrower neural network, however, it is also more expensive to train. [10] In a deep neural network, each node takes some number of inputs x_i, combines them, feeds them through an activation function, and the result is then passed on to some other nodes in the next layer. The combination of inputs is usually multiplied by an individual weight,w_i, then added along with a bias term,b. The activation function is a predetermined function, such as ReLU Function, Binary Step Function, and hyperbolic tangent function. It takes in a real value and spits out a value between (0 to 1) to add non-linearity to the network[11]. The limitation of neural networks includes their requirement for a very large amount of data to demonstrate significantly better performance compared to other techniques. It is also extremely expensive to train due to complex data models. Another limitation of the neural network is that the model itself is a black box and it is hard to interpret the model.

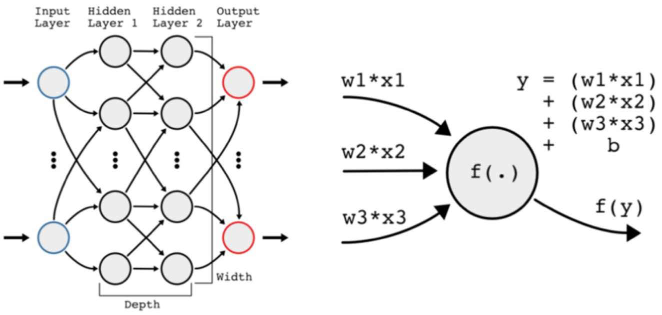


FIG. 1. Left: Schematic for a simple artificial neural network with a depth of two and unspecified width. Right: Schematic for a simple, generic node. Here three inputs x_i are scaled by weights w_i, summed, and biased b before being fed through an activation function $f(y)$[10].

C. Decision Trees

Apart from the most commonly seen neural network, there are a few other methods that are more compatible under certain circumstances. One of them is a decision tree. A decision tree is a popular method for classification problems in supervised machine learning. The decision tree model predicts a value of a target variable, usually categorical, based on input variables. [12] The decision tree is presented as a tree-like structure, where each internal node represents a test on an attribute, each branch represents the test outcome, and each leaf node represents a class label. At each node, the tree is split into branches based on the value of the attribute. The tree is built recursively until the stopping criteria are met. The stopping criteria can be the maximum depth of the tree, and the minimum number of samples in a leaf node. At each node, the algorithm selected the correct attributes by lowering the Gini score or reducing entropy. [13] The fundamental benefit of the decision tree lies in its easy-to-understand and interpretable nature. Furthermore, it has a minimum data requirement: the data do not require cleaning or normalization before being fed to the model.[14] Significant drawbacks of the decision tree method include its inaccuracy in extrapolating data and its limitation in predicting continuous values[15] [16].

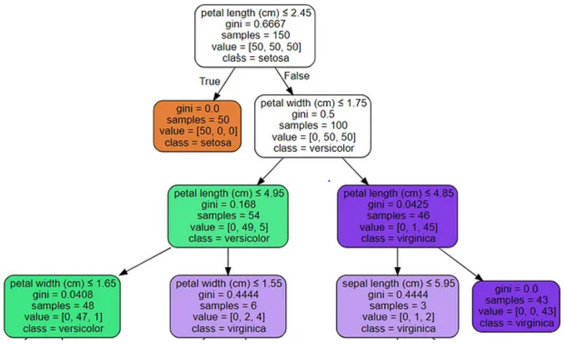


FIG. 2. Example of a Decision Tree model on the categorization of iris

D. Gaussian Process

Gaussian Process is a non-parametric method that can be used for regression and classification problems. The function is assumed to be drawn from a Gaussian process, which is a collection of random variables, any finite number of which have a joint Gaussian distribution. The Gaussian process is defined by a mean function and a covariance function. The mean function is a function that describes the average value of the function $f(x)$, and the covariance function is a function that describes the covariance between two points x_1 and x_2[17][18].

The distinctive feature of the Gaussian process is that it provides a probabilistic interpretation of the model instead of a point of estimation in other methods. This probabilistic representation allows for the quantification of uncertainty in the predictions, making GPs particularly useful in cases where uncertainty needs to be taken into account.

Another strength of the Gaussian Process is that it works with small sets of data very well, making them particularly useful in cases where data is limited or expensive to collect.

Examples of GPR implementation can be found in the Scikit-learn library for Python[19].

E. Physics informed machine-learning

The method introduced in the previous section works independently of any physical formula. However, in many situations, we know partially the physical constraints in our system. These constraints can be integrated into our model to improve performance. Physics-informed neural network (PINN) is a fast-growing field in machine learning. Usually, based on the amount of available data and knowledge in physics, we can subcategorize a machine-learning problem into three cases.

1. In the case of a well-developed understanding of the physical laws behind the system, i.e. the governing partial equations, initial condition, and boundary condition, we do not require a data-intensive approach to have a good understanding of the system.
2. On the contrary, in the case of little knowledge of the physical law, we can use a general machine learning method to find patterns and insights from the data, provided that a big set of experimental data is available.
3. In the last case, we have some physics and some data, probably missing some data or values for parameters in the partial differential equations. We can use neural networks that are specifically made to respect the physical laws, and PINN is best suited to solve this problem. [20]

Introducing a physics-informed neural network (PINN) with observational, inductive or learning biases will allow the model to learn bearing the physics constraints of the problem[21] [22]. Physics-informed neural network (PINN) is an ML model designed to embed the PDEs into the loss function of a neural network using automatic differentiation. We will use viscous Burgers' equations as an example:

$$\frac{\partial u}{\partial t}+u\frac{\partial u}{\partial x}=v\frac{\partial^2 u}{\partial x^2}$$

with a suitable initial condition and Dirichlet boundary conditions. In the figure, the left (physics-uninformed) network represents the surrogate of the PDE solution $u(x,t)$, while the right (physics-informed) network describes the PDE residual $\frac{\partial u}{\partial t}+u\frac{\partial u}{\partial x}-v\frac{\partial^2 u}{\partial x^2}$. The loss function includes a supervised loss of data measurements of u from the initial and boundary conditions and an unsupervised loss of PDE:

$$\mathcal{L}=w_{\text{data}}\,\mathcal{L}_{\text{data}}\;+w_{\text{PDE}}\mathcal{L}_{\text{PDE}}$$

where

$$\mathcal{L}_{\text{data}} = \frac{1}{N_{\text{data}}} \sum_{i=1}^{N_{\text{data}}} \left(u\left(x_i, t_i\right) - u_i\right)^2 \quad \text{and}$$

$$\mathcal{L}_{\text{PDE}} = \frac{1}{N_{\text{PDE}}} \sum_{j=1}^{N_{\text{PDE}}} \left(\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} - v\frac{\partial^2 u}{\partial x^2} \right)^2 \Bigg|_{(x_j,t_j)}$$

Here $\{(x_i, t_i)\}$ and $\{(x_j, t_j)\}$ are two sets of points sampled at the initial/boundary locations and in the entire domain, respectively, and u_i are values of u at (x_i, t_i); w_{data} and w_{PDE} are the weights used to balance the interplay between the two loss term. PINN is one of the more recent topics introduced in machine learning toolboxes. Its implementation is more complicated as there is currently no well-documented official library yet. In the future, we would expect to see more development in this method.

IV. CHOOSING ML TECHNIQUES

The most suitable ML model is task-specific, the factors that should be considered when choosing a model are:

1. The nature of output the problem, i.e. whether it is a classification or regression problem.
2. The amount of data and resources available for training.
3. The interpretability and complexity of the model.

A summary of the strength and limitations of Machine learning techniques is shown in Table 1 below.

ML Method	Requirement for data	Computational resource	Interpretibility	Other advantage/limitation
Decision Tree	Little requirement	Relatively few	Good with a nice visualisation (tree diagram)	Limited capability for regression,
Linear Regression	Average	Relatively few	Good	Could not handle non-linear cases. Less prone to overfitting
Neural Network	Perform well with Large amount of data points	Computationally Expensive	Black-box, hard to interpret result	Prone to over-fitting
Deep Neural Network	Perform well with Large amount of data points	Even more Computationally Expensive	Black-box, hard to interpret result	Prone to over-fitting
Gaussian Process	Perform well with limited amount of data points	Computationally expensive	Black-box, hard to interpret result	Gives uncertainty estimate

TABLE I. Summary for the strength and limitations of machine learning techniques.

It is also a common practice to use more than one model, or the same model with various hyperparameters can be used to compare the performance of the model. In the following section, I will give two case studies in two distinctive fields of physics research to demonstrate the application of machine learning in addressing regression and classification problems respectively.

V. CASE STUDY 1: USING A NEURAL NETWORK TO DESIGN NEW MATERIAL

This case study is an example of using a neural network to solve a material design problem [23].

In this research, a new polycrystalline nickel-base superalloy was proposed by the mode that has the optimal combination of cost, density, γ' phase content and solvus, phase stability, fatigue life, yield stress, ultimate tensile strength, stress rupture, oxidation resistance, and tensile elongation.

The construction and validation of the model are done in the following order: Firstly, predictive models are constructed for each property. Then, these models are used to calculate the probability that a proposed composition fulfills a target specification. Finally, the model searches through the composition space for an alloy that would most likely to fulfill all the specifications.

With a proposed alloy from the model, an experiment is then carried out to test the physical properties of the new alloy. Comparisons are made between the prediction from the neural network model and experimental data. The results are shown in the figure 3, demonstrating a successful application of machine learning in material design. The model in this case is validated by an experiment instead of cross-validation, which adds more credibility to the model

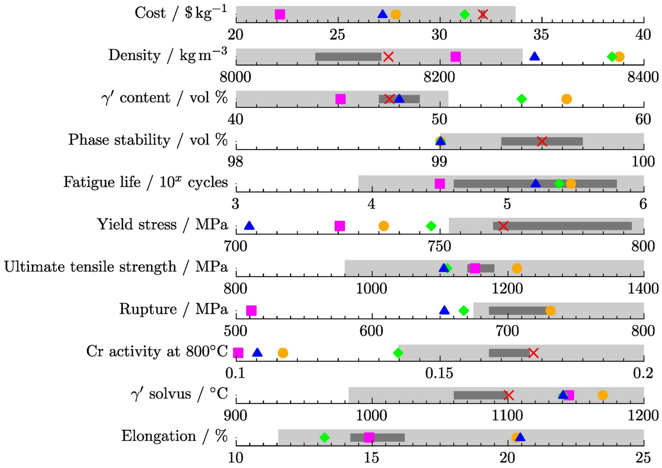
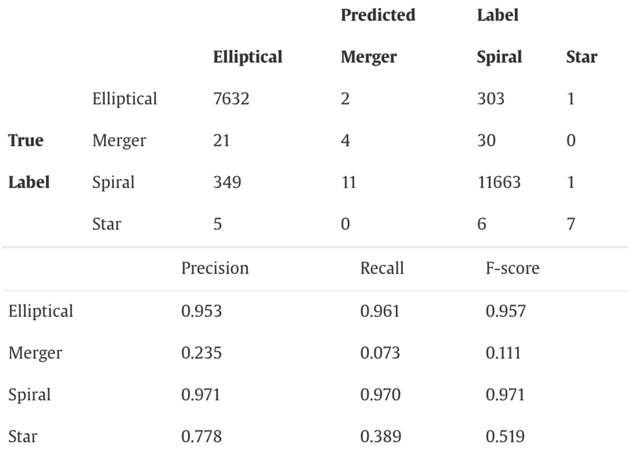


FIG. 3. Experimental results of the proposed alloy. For each listed property the gray box refers to the acceptable target properties, and the dark gray is the three-sigma uncertainty on the theoretical prediction. The points refer to experimentally measured values with V210A where measured

VI. CASE STUDY 2: MACHINE LEARNING IN GALAXY MORPHOLOGY CLASSIFICATION

This case study is chosen as it is a good example of using machine learning to replace human data labeling in classification problems in astrophysics[24]. The morphological classification of galaxies provides crucial astronomical insights and helps to understand how galaxies form and evolve. In this research, scientists classify samples into four categories: spirals, ellipticals, mergers and stars[25]. The researcher tested 5 various models (Decision Tree, Random Forest, Extra Trees, K-nearest Neighbours, Artificial Neural Network) using data from Sloan Digital Sky Survey and labeled data from Galaxy Zoo[26]. The results from the decision tree method are shown in Figure 4. The results are summarised in a confusion matrix, which shows the performance of the model in classifying each class. The diagonal elements of the matrix show the number of correct classifications, while the off-diagonal elements show the number of misclassification. The results show that the model is relatively good at classifying spirals and ellipticals, but not for mergers and stars. The results in Figure 5 shows a relatively similar performance of all five models, with all the model performing relatively well in classifying spirals and ellipticals, but not so well in classifying mergers and stars. This is a good example of a class imbalance problem in machine learning when there are many more instances of a certain class (i.e.elliptical and spiral galaxies) than other classes (i.e.mergers and stars). This case study demonstrated that machine learning is a powerful tool for classification problems which is a rather labor-intensive task had it done by human labeling. Various methods could produce similar results, but the choice of method is task-specific. However, machine learning is far from being a perfect solution as certain problems such as class imbalance problems would need more fine-tuning or perfection of data.



			Predicted	**Label**	
		Elliptical	**Merger**	**Spiral**	**Star**
	Elliptical	7632	2	303	1
True	Merger	21	4	30	0
Label	Spiral	349	11	11663	1
	Star	5	0	6	7

	Precision	Recall	F-score
Elliptical	0.953	0.961	0.957
Merger	0.235	0.073	0.111
Spiral	0.971	0.970	0.971
Star	0.778	0.389	0.519

FIG. 4. Confusion Matrix showing the performance of Decision Tree model

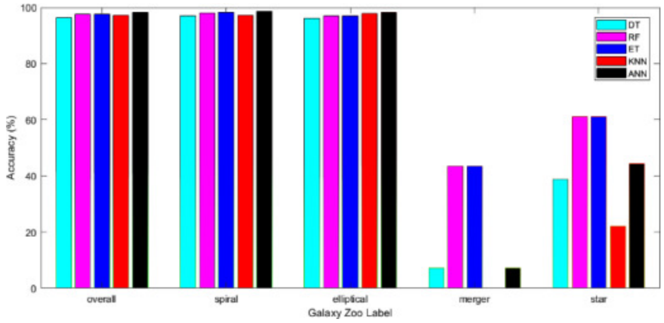


FIG. 5. Bar chart showing overall accuracy and classwise recall using different ML algorithms

VII. CONCLUSION

Machine learning is a powerful tool that can be applied in physics, given a mass quantity of data we can train the model and predict the outcome of the system. One should carefully choose a model based on the amount of data points available, the complexity and the interpretability of the model. Machine learning and its black-box nature result in a lack of interpretability, as well as a good reflection of the physical property of the system. However, the performance of the model can be further improved by incorporating existing physical laws into the model, or by recognizing the relevant symmetry in the model itself. There are already attempts to integrate physical law. In the future, we may see an increasing number of physics-informed machine-learning models.

[1] Victor Zhou. Machine Learning for Beginners: An Introduction to Neural Networks — towardsdatascience.com. [Accessed 26-Apr-2023].

[2] What is Supervised Learning? — IBM — ibm.com. https://www.ibm.com/topics/supervised-learning. [Accessed 26-Apr-2023].

[3] Arthur L Samuel. Some studies in machine learning using the game of checkers. *IBM Journal of research and development*, 3(3):210–229, 1959.

[4] Juras Juršėnas. A Deep Dive into Curve Fitting for ML — towardsdatascience.com. https://towardsdatascience.com/a-deep-dive-into-curve-fitting-for-ml-7aeef64755d2. [Accessed 23-Apr-2023].

[5] Padé approximant - Wikipedia — en.wikipedia.org. https://en.wikipedia.org/wiki/Pad%C3%A9_

approximant. [Accessed 23-Apr-2023].

[6] Padé Approximant – from Wolfram MathWorld — mathworld.wolfram.com. https://mathworld.wolfram.com/PadeApproximant.html. [Accessed 23-Apr-2023].

[7] Volker L Deringer, Albert P Bartók, Noam Bernstein, David M Wilkins, Michele Ceriotti, and Gábor Csányi. Gaussian process regression for materials and molecules. *Chemical Reviews*, 121(16):10073–10141, 2021.

[8] Francis Benistant. Taylor Series, Pade Approximants, and Neural Networks — 2020machinelearning. https://medium.com/@2020machinelearning.

[9] Daniel George and E.A. Huerta. Deep learning for real-time gravitational wave detection and parameter estimation: Results with advanced LIGO data. *Physics Letters B*, 778:64–70, mar 2018.

[10] Trisha A. Hinners, Kevin Tat, and Rachel Thorp. Machine learning techniques for stellar light curve classification. *The Astronomical Journal*, 156(1):7, jun 2018.

[11] Activation Functions in Neural Networks [12 Types Use Cases] — v7labs.com. https://www.v7labs.com/blog/neural-networks-activation-functions. [Accessed 23-Apr-2023].

[12] Lior Rokach and Oded Maimon. Data mining with decision trees - theory and applications. 2nd edition. In *Series in Machine Perception and Artificial Intelligence*, 2014.

[13] Shai Shalev-Shwartz and Shai Ben-David. *Understanding machine learning: From theory to algorithms*. Cambridge university press, 2014.

[14] J. Ross Quinlan. Induction of decision trees. *Machine learning*, 1:81–106, 1986.

[15] What is a Decision Tree — IBM — ibm.com. https://www.ibm.com/uk-en/topics/decision-trees. [Accessed 23-Apr-2023].

[16] 1.10. Decision Trees — scikit-learn.org. https://scikit-learn.org/stable/modules/tree.html#:~:text=Decision%20Trees%20(DTs)%20are%20a,as%20a%20piecewise%20constant%20approximation. [Accessed 23-Apr-2023].

[17] Manuel Perger, Guillem Anglada-Escudé, Ignasi Ribas, Albert Rosich, Enrique Herrero, and Juan Carlos Morales. Auto-correlation functions of astrophysical processes, and their relation to gaussian processes-application to radial velocities of different starspot configurations. *Astronomy & Astrophysics*, 645:A58, 2021.

[18] Yun Yuan, Zhao Zhang, Xianfeng Terry Yang, and Shandian Zhe. Macroscopic traffic flow modeling with physics regularized Gaussian process: A new insight into machine learning applications in transportation. *Transportation Research Part B: Methodological*, 146(C):88–110, 2021.

[19] 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.

[20] M. Raissi, P. Perdikaris, and G. E. Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, 378:686–707, February 2019.

[21] Me. So, what is a physics-informed neural network? - Ben Moseley — benmoseley.blog. https://benmoseley.blog/my-research/so-what-is-a-physics-informed-neural-network/. [Accessed 23-Apr-2023].

[22] 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.

[23] BD Conduit, Nick G Jones, Howard J Stone, and Gareth John Conduit. Design of a nickel-base superalloy using a neural network. *Materials & Design*, 131:358–365, 2017.

[24] José-Víctor Rodríguez, Ignacio Rodríguez-Rodríguez, and Wai Lok Woo. On the application of machine learning in astronomy and astrophysics: A text-mining-based scientometric analysis. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 12, 08 2022.

[25] Moonzarin Reza. Galaxy morphology classification using automated machine learning. *Astronomy and Computing*, 37:100492, 2021.

[26] Zooniverse — zooniverse.org. https://www.zooniverse.org/projects/zookeeper/galaxy-zoo/. [Accessed 25-Apr-2023].