entropy

MDPI

Article

Why Do Big Data and Machine Learning Entail the Fractional Dynamics?

Haoyu Niu [1,†], YangQuan Chen [2,*,†] and Bruce J. West [3]

[1] Electrical Engineering and Computer Science Department, University of California, Merced, CA 95340, USA; hniu2@ucmerced.edu
[2] Mechanical Engineering Department, University of California, Merced, CA 95340, USA
[3] Office of the Director, Army Research Office, Research Triangle Park, NC 27709, USA; brucejwest213@gmail.com
* Correspondence: ychen53@ucmerced.edu; Tel.: +1-209-2284672
† MESA Lab address: Room 22, 4225 Hospital Road, Atwater, CA 95301, USA.

Abstract: Fractional-order calculus is about the differentiation and integration of non-integer orders. Fractional calculus (FC) is based on fractional-order thinking (FOT) and has been shown to help us to understand complex systems better, improve the processing of complex signals, enhance the control of complex systems, increase the performance of optimization, and even extend the enabling of the potential for creativity. In this article, the authors discuss the fractional dynamics, FOT and rich fractional stochastic models. First, the use of fractional dynamics in big data analytics for quantifying big data variability stemming from the generation of complex systems is justified. Second, we show why fractional dynamics is needed in machine learning and optimal randomness when asking: "is there a more optimal way to optimize?". Third, an optimal randomness case study for a stochastic configuration network (SCN) machine-learning method with heavy-tailed distributions is discussed. Finally, views on big data and (physics-informed) machine learning with fractional dynamics for future research are presented with concluding remarks.

Keywords: fractional calculus; fractional dynamics; fractional-order thinking; heavytailedness; big data; machine learning; variability; diversity

check for updates

Citation: Niu, H.; Chen, Y.; West, B.J. Why Do Big Data and Machine Learning Entail the Fractional Dynamics? *Entropy* **2021**, *23*, 297. https://doi.org/10.3390/e23030297

Academic Editor: Jose A. Tenreiro Machado

Received: 2 February 2021
Accepted: 24 February 2021
Published: 28 February 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Fractional Calculus (FC) and Fractional-Order Thinking (FOT)

Fractional calculus (FC) is the quantitative analysis of functions using non-integer-order integration and differentiation, where the order can be a real number, a complex number or even the function of a variable. The first recorded query regarding the meaning of a non-integer order differentiation appeared in a letter written in 1695 by Guillaume de l'Hôpital to Gottfried Wilhelm Leibniz, who at the same time as Isaac Newton, but independently of him, co-invented the infinitesimal calculus [1]. Numerous contributors have provided definitions for fractional derivatives and integrals [2] since then, and the theory along with the applications of FC have been expanded greatly over the centuries [3–5]. In more recent decades, the concept of **fractional dynamics** has merged and gained followers in the statistical and chemical physics communities [6–8]. For example, optimal image processing has improved through the use of fractional-order differentiation and fractional-order partial differential equations as summarized in Chen et al. [9–11]. Anomalous diffusion was described using fractional-diffusion equations in [12,13], and Metzler et al. used fractional Langevin equations to model viscoelastic materials [14].

Today, big data and machine learning (ML) are two of the hottest topics of applied scientific research, and they are closely related to one another. To better understand them, we also need fractional dynamics, as well as fractional-order thinking (FOT). Section 2 is devoted to the discussion of the relationships between big data, variability, and fractional dynamics, as well as to fractional-order data analytics (FODA) [15]. The topics touched

on in this section include the Hurst parameter [16,17], fractional Gaussian noise (fGn), fractional Brownian motion (fBm), the fractional autoregressive integrated moving average (FARIMA) [18], the formalism of continuous time random walk (CTRW) [19], unmanned aerial vehicles (UAVs) and precision agriculture (PA) [20]. In Section 3, how to learn efficiently (optimally) for ML algorithms is investigated. The key to developing an efficient learning process is the method of optimization. Thus, it is important to design an efficient or perhaps optimal optimization method. The derivative-free methods, and the gradient-based methods, such as the Nesterov accelerated gradient descent (NAGD) [21], are both discussed. Furthermore, the authors propose designing and analyzing the ML algorithms in an S or Z transform domain in Section 3.3. FC is used in optimal randomness in the methods of stochastic gradient descent (SGD) [22] and random search, and in implementing the internal model principle (IMP) [23].

FOT is a way of thinking using FC. For example, there are non-integers between the integers; between logic 0 and logic 1, there is the fuzzy logic [24]; compared with integer-order splines, there are fractional-order splines [25]; between the high-order integer moments, there are non-integer-order moments, etc. FOT has been entailed by many research areas, for example, self-similar [26,27], scale-free or scale-invariant, power-law, long-range-dependence (LRD) [28,29], and $1/f^{\alpha}$ noise [30,31]. The terms porous media, particulate, granular, lossy, anomaly, disorder, soil, tissue, electrodes, biology [32], nano, network, transport, diffusion, and soft matters are also intimately related to FOT. However, in the present section, we mainly discuss **complexity and inverse power laws (IPL)**.

1.1. Complexity and Inverse Power Laws

When studying complexity, it is fair to ask, what does it mean to be complex? When do investigators begin identifying a system, network or phenomenon as complex [33,34]? There is an agreement among a significant fraction of the scientific community that when the distribution of the data associated with the process of interest obeys an IPL, the phenomenon is complex; see Figure 1. On the left side of the figure, the complexity "bow tie" [35–38] is the phenomenon of interest, thought to be a complex system. On the right side of the figure is the spectrum of system properties associated with IPL probability density functions (PDFs): the system has one or more of the properties of being scale-free, having a heavy tail, having a long-range dependence, and/or having a long memory [39,40]. In the book by West and Grigolini [41], there is a table listing a sample of the empirical power laws and IPLs uncovered in the past two centuries. For example, in scale-free networks, the degree distributions follow an IPL in connectivity [42,43]; in the processing of signals containing pink noise, the power spectrum follows an IPL [29]. For other examples, such as the probability density function (PDF), the autocorrelation function (ACF) [44], allometry ($Y = aX^b$) [45], anomalous relaxation (evolving over time) [46], anomalous diffusion (mean squared dissipation versus time) [13], and self-similarity can all be described by the IPL "bow tie" depicted in Figure 1.

The power law is usually described as:

$$f(x) = ax^k, \tag{1}$$

when k is negative, $f(x)$ is an IPL. One important characteristic of this power law is scale invariance [47] determined by:

$$f(cx) = a(cx)^k = c^k f(x) \propto f(x). \tag{2}$$

Note that when x is the time, the scaling depicts a property of the system dynamics. However, when the system is stochastic, the scaling is a property of the PDF (or correlation structure) and is a constraint on the collective properties of the system.

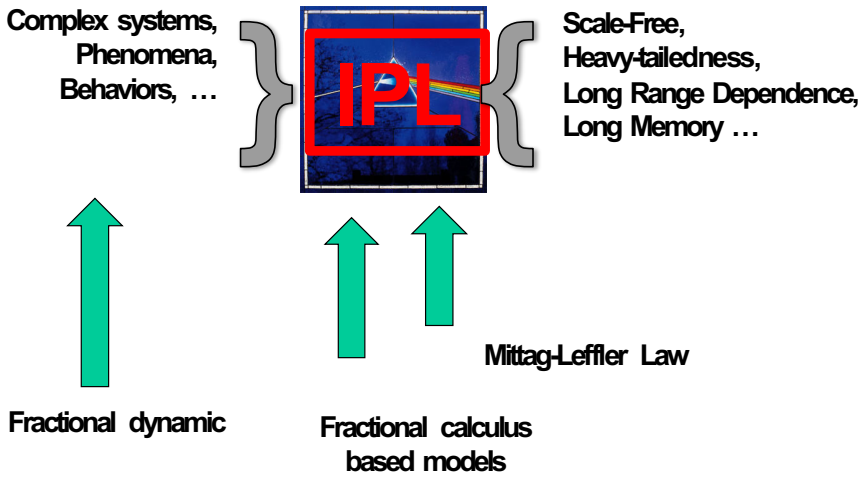


Figure 1. Inverse power law (complexity "bow tie"): On the left are the systems of interest that are thought to be complex. In the center panel, an aspect of the empirical data is characterized by an inverse power law (IPL). The right panel lists the potential properties associated with systems with data that have been processed and yield an IPL property. See text for more details.

FC is entailed by complexity, since an observable phenomenon represented by a fractal function has integer-order derivatives that diverge. Consequently, for the complexity characterization and regulation, we ought to use the fractional dynamics point of view because the fractional derivative of a fractal function is finite. Thus, complex phenomena, no matter whether they are natural or carefully engineered, ought to be described by fractional dynamics. Phenomena in complex systems in many cases should be analyzed using FC-based models, where mathematically, the IPL is actually the "Mittag–Leffler law" (MLL), which asymptotically becomes an IPL (Figure 2), known to have heavy-tail behavior.

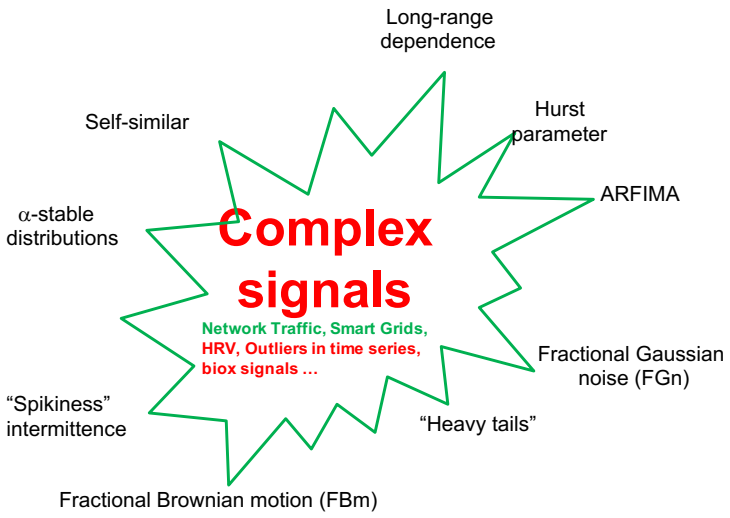


Figure 2. Complex signals (IPL): Here, the signal generated by a complex system is depicted. Exemplars of the systems are given as are the potential properties arising from the systems' complexity.

When an IPL results from processing data, one should think about how the phenomena can be connected to the FC. In [48], Gorenflo et al. explained the role of the FC in generating

stable PDFs by generalizing the diffusion equation to one of fractional order. For the Cauchy problem, they considered the space-fractional diffusion equation:

$$\frac{\partial u}{\partial t} = D(\alpha)\frac{\partial^{\alpha} u}{\partial |x|^{\alpha}}, \tag{3}$$

where $-\infty < x < \infty$, $t \geq 0$ with $u(x,0) = \delta(x)$, $0 < \alpha \leq 2$, and $D(\alpha)$ is a suitable diffusion coefficient. The fractional derivative in the diffusion variable is of the Riesz–Feller form, defined by its Fourier transform to be $|k|^{a}$. For the signalling problem, they considered the so-called time-fractional diffusion equation [49]:

$$\frac{\partial^{2\beta} u}{\partial t^{2\beta}} = D(\beta)\frac{\partial^{2} u}{\partial x^{2}}, \tag{4}$$

where $x \geq 0$, $t \geq 0$ with $u(0,t) = \delta(t)$, $0 < \beta < 1$, and $D(\beta)$ is a suitable diffusion coefficient. Equation (4) has also been investigated in [50–52]. Here, the Caputo fractional derivative in time is used.

There are rich forms in stochasticity [22], for example, heavytailedness, which corresponds to fractional-order master equations [53]. In Section 1.2, heavy-tailed distributions are discussed.

1.2. Heavy-Tailed Distributions

In probability theory, heavy-tailed distributions are PDFs whose tails do not decay exponentially [54]. Consequently, they have more weight in their tails than does an exponential distribution. In many applications, it is the right tail of the distribution that is of interest, but a distribution may have a heavy left tail, or both tails may be heavy. Heavy-tailed distributions are widely used for modeling in different disciplines, such as finance [55], insurance [56], and medicine [57]. The distribution of a real-valued random variable X is said to have a heavy right tail if the tail probabilities $P(X > x)$ decay more slowly than those of any exponential distribution:

$$\lim_{x\to\infty}\left(\frac{P(X > x)}{e^{-\lambda x}}\right) = \infty, \tag{5}$$

for every $\lambda > 0$ [58]. For the heavy left tail, an analogous definition can be constructed [59]. Typically, there are three important subclasses of heavy-tailed distributions: fat-tailed, long-tailed and subexponential distributions.

1.2.1. Lévy Distribution

A Lévy distribution, named after the French mathematician Paul Lévy, can be generated by a random walk whose steps have a probability of having a length determined by a heavy-tailed distribution [60]. As a fractional-order stochastic process with heavy-tailed distributions, a Lévy distribution has better computational characteristics [61]. A Lévy distribution is stable and has a PDF that can be expressed analytically, although not always in closed form. The PDF of Lévy flight [62] is:

$$p(x,\mu,\gamma) = \begin{cases} \frac{\sqrt{\frac{\gamma}{2\pi}}}{e^{\frac{\gamma}{2(x-\mu)}}(x-\mu)^{3/2}}, & x > \mu, \\ 0, & x \leq \mu, \end{cases} \tag{6}$$

where μ is the location parameter and γ is the scale parameter. In practice, the Lévy distribution is updated by

$$Lévy(\beta) = \frac{u}{|\nu|^{1/\beta}}, \tag{7}$$

where u and ν are random numbers generated from a normal distribution with a mean of 0 and standard deviation of 1 [63]. The stability index β ranges from 0 to 2. Moreover, it is

interesting to point out that the well-known Gaussian and Cauchy distributions are special cases of the Lévy PDF when the stability index is set to 2 and 1, respectively.

1.2.2. Mittag–Leffler PDF

The Mittag–Leffler PDF [64] for the time interval between events can be written as a mixture of exponentials with a known PDF for the exponential rates:

$$E_\theta(-t^\theta) = \int_0^\infty exp(-\mu t)g(\mu)d\mu, \tag{8}$$

with a weight for the rates given by:

$$g(\mu) = \frac{1}{\pi}\frac{\sin(\theta\pi)}{\mu^{1+\theta} + 2\cos(\theta\pi)\mu + \mu^{1-\theta}}. \tag{9}$$

The most convenient expression for the random time interval was proposed by [65]:

$$\tau_\theta = -\gamma_t(\ln u\frac{\sin(\theta\pi)}{\tan(\theta\pi v)} - \cos(\theta\pi))^{1/\theta}, \tag{10}$$

where $u, v \in (0,1)$ are independent uniform random numbers, γ_t is the scale parameter, and τ_θ is the Mittag–Leffler random number. In [66], Wei et al. used the Mittag–Leffer distribution for improving the Cuckoo Search algorithm, which did show an improved performance.

1.2.3. Weibull Distribution

A random variable is described by a Weibull distribution function F:

$$F(x) = e^{-(x/k)^{\lambda_w}}, \tag{11}$$

where $k > 0$ is the scale parameter, and $\lambda_w > 0$ is the shape parameter [67]. If the shape parameter is $\lambda_w < 1$, the Weibull distribution is determined to be heavy tailed.

1.2.4. Cauchy Distribution

A random variable is described by a Cauchy PDF if its cumulative distribution is [68,69]:

$$F(x) = \frac{1}{\pi}\arctan(\frac{2(x-\mu_c)}{\sigma}) + \frac{1}{2}, \tag{12}$$

where μ_c is the location parameter and σ is the scale parameter. Cauchy distributions are examples of fat-tailed distributions, which have been empirically encountered in a variety of areas including physics, earth sciences, economics and political science [70]. Fat-tailed distributions include those whose tails decay like an IPL, which is a common point of reference in their use in the scientific literature [71].

1.2.5. Pareto Distribution

A random variable is said to be described by a Pareto PDF if its cumulative distribution function is

$$F(x) = \begin{cases} 1 - (\frac{b}{x})^a, x \geq b, \\ 0, x < b, \end{cases} \tag{13}$$

where $b > 0$ is the scale parameter and $a > 0$ is the shape parameter (Pareto's index of inequality) [72] (Figure 3).

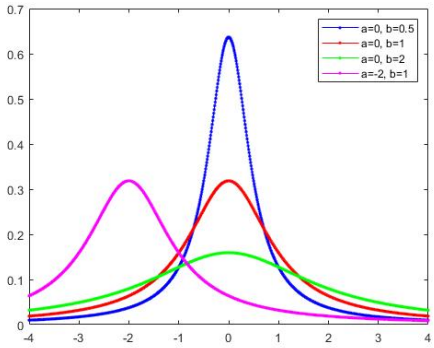


Figure 3. Cauchy distributions are examples of fat-tailed distributions. The parameter a is the location parameter; the parameter b is the scale parameter.

1.2.6. The α-Stable Distribution

A PDF is said to be stable if a linear combination of two independent random variables, each with the same distribution, has the same distribution for the conjoined variable. This PDF is also called the Lévy α-stable distribution [73,74]. Since the normal distribution, Cauchy distribution and Lévy distribution all have the above property, one can consider them to be special cases of stable distributions. Stable distributions have $0 < \alpha \leq 2$, with the upper bound corresponding to the normal distribution, and $\alpha = 1$, to the Cauchy distribution (Figure 4). The PDFs have undefined variances for $\alpha < 2$, and undefined means for $\alpha \leq 1$. Although their PDFs do not admit a closed-form formula in general, except in special cases, they decay with an IPL tail and the IPL index determines the behavior of the PDF. As the IPL index gets smaller, the PDF acquires a heavier tail. An example of an IPL index analysis is given in Section 1.4.

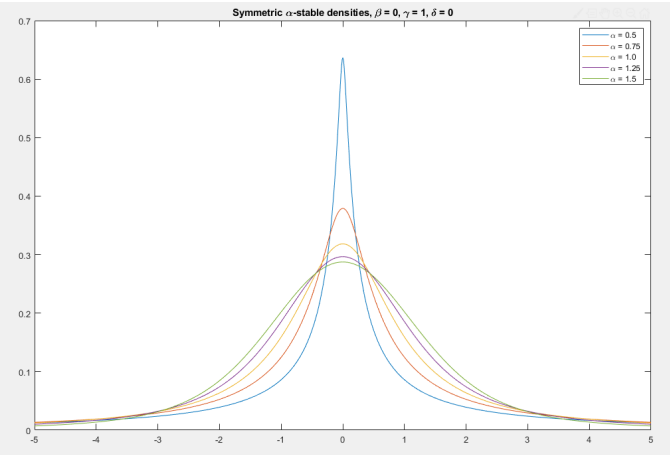


Figure 4. Symmetric α-stable distributions with unit scale factor. The most narrow PDF shown has the smallest IPL index and, consequently, the most weight in the tail regions.

1.3. Mixture Distributions

A mixture distribution is derived from a collection of other random variables. First, a random variable is selected by chance from the collection according to given probabilities of selection. Then, the value of the selected random variable is realized. The mixture PDFs are complicated in terms of simpler PDFs, which provide a good model for certain datasets. The different subsets of the data can exhibit different characteristics. Therefore, the mixed PDFs can effectively characterize the complex PDFs of certain real-world datasets. In [75], a robust stochastic configuration network (SCN) based on a mixture of Gaussian and Laplace PDFs was proposed. Thus, Gaussian and Laplace distributions are mentioned in this section for comparison purposes.

1.3.1. Gaussian Distribution

A random variable X has a Gaussian distribution with the mean μ_G and variance σ_G^2 ($-\infty < \mu_G < \infty$ and $\sigma_G > 0$) if X has a continuous distribution for which the PDF is as follows [76]:

$$f(x|\mu_G, \sigma_G^2) = \frac{1}{(2\pi)^{1/2}\sigma_G} e^{-\frac{1}{2}(\frac{x-\mu_G}{\sigma_G})^2}, for \ -\infty < x < \infty. \tag{14}$$

1.3.2. Laplace Distribution

The PDF of the Laplace distribution can be written as follows [75]:

$$F(x|\mu_l, \eta) = \frac{1}{(2\eta^2)^{1/2}} e^{(-\frac{\sqrt{2}|x-\mu_l|}{\eta})}, \tag{15}$$

where μ_l and η represent the location and scale parameters, respectively.

1.4. IPL Tail-Index Analysis

There are two approaches to the problem of the IPL tail-index estimation: the parametric [77] and the nonparametric [78]. To estimate the tail index using the parametric approach, some researchers employ a generalized extreme value (GEV) distribution [79] or Pareto distribution, and they may apply the maximum-likelihood estimator (MLE).

The stochastic gradient descent (SGD) has been widely used in deep learning with great success because of the computational efficiency [80,81]. The gradient noise (GN) in the SGD algorithm is often considered to be Gaussian in the large data regime by assuming that the classical central limit theorem (CLT) kicks in. The machine-learning tasks are usually considered as solving the following optimization problem:

$$w^* = argmin\{f(w) \triangleq \frac{1}{n}\sum_{i=1}^{n} f^{(i)}(w)\}, \tag{16}$$

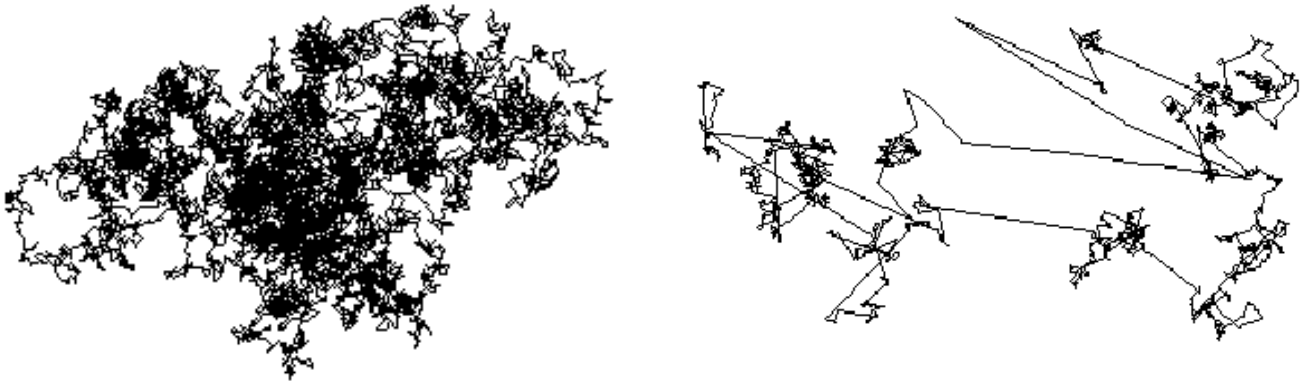
where w denotes the weights of the neural network, f denotes the loss function, and n denotes the total number of instances. Then, the SGD is calculated based on the following iterative scheme:

$$w_{k+1} = w_k - \eta \nabla f_k(w_k), \tag{17}$$

where k means the iteration number, and $\nabla f_k(w_k)$ denotes the stochastic gradient at iteration k.

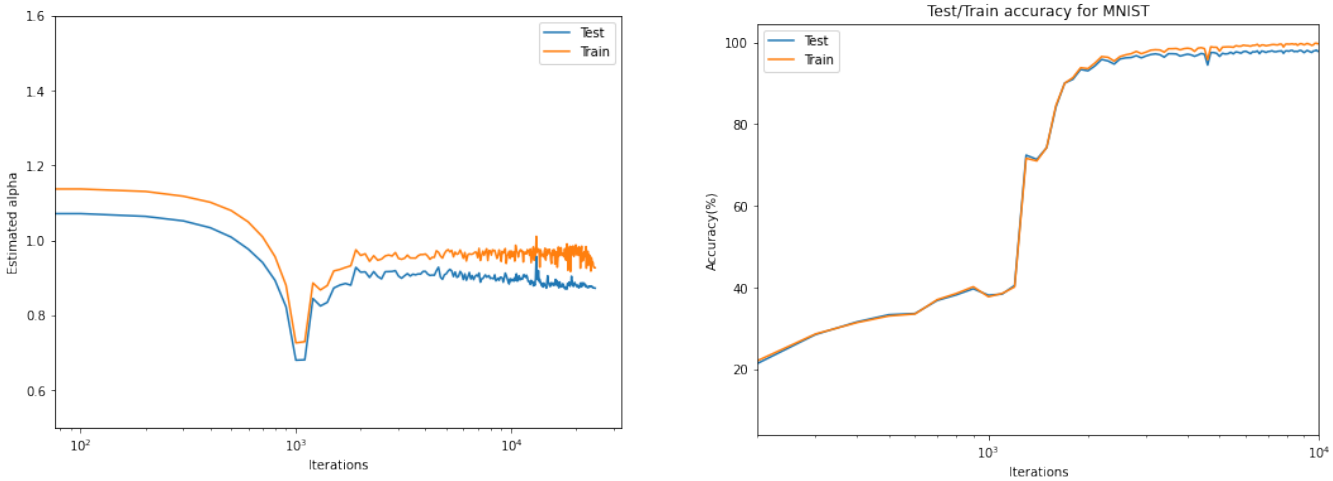
Since the gradient noise might not be Gaussian, the use of Brownian motion would not be appropriate to represent its behavior. Therefore, Şimşekli et al. replaced the gradient noise with the α-stable Lévy motion [82], whose increments have an α-stable distribution [83]. Because of the heavy-tailed nature of the α-stable distribution, the Lévy motion might incur large, discontinuous jumps [84], and therefore, it would exhibit a fundamentally different behavior than would Brownian motion (Figure 5):

Figure 6 shows that there are two distinct phases of SGD (in this configuration, before and after iteration 1000). At first, the loss decreases very slowly, the accuracy slightly increases, and more interestingly, α rapidly decreases. When α reaches its lowest level, which means a longer tail distribution, there is a significant jump, which causes a sudden decrease in accuracy. Beyond this point, the process recovers again, and we see stationary behavior in α and an increasing behavior in the accuracy.



(**a**) (**b**)

Figure 5. (**a**) Brownian motion; (**b**) Lévy motion. Note that both figures are at the same size scale.



(**a**) (**b**)

Figure 6. (**a**) The behavior of tail-index α during the iterations; (**b**) The training and testing accuracy. At first, the α decreases very slowly; when α reaches its lowest level, which means longer tail distribution, there is a significant jump, which causes a sudden decrease in accuracy. Beyond this point, the process recovers again, and we see stationary behavior in α and an increasing behavior in the accuracy.

2. Big Data, Variability and FC

The term "big data" started showing up in the early 1990s. The world's technological per capita capacity to store information has roughly doubled every 40 months since the 1980s [85]. Since 2012, there have been 2.5 exabytes (2.5×2^{60} bytes) of data generated every day [86]. According to data report predictions, there will be 163 zettabytes of data by 2025 [87]. Firican proposed, in [88], ten characteristics (properties) of big data to prepare for both the challenges and advantages of big data initiatives (Table 1). In this article, **variability** is the most important characteristic being discussed. Variability refers to several properties of big data. First, it refers to the number of inconsistencies in the data, which need to be understood by using anomaly- and outlier-detection methods for any meaningful analytics to be performed. Second, variability can also refer to diversity [89,90], resulting from disparate data types and sources, for example, healthy or unhealthy [91,92]. Finally, variability can refer to multiple research topics (Table 2).

Considering variability, Xunzi (312 BC–230 BC), who was a Confucian philosopher, made a useful observation: "Throughout a thousand acts and ten thousand changes, his way remains one and the same" [93]. Therefore, we ask: what is the "one and the same" for big data? This is the **variability**, which refers to the behavior of the dynamic system. The ancient Greek philosopher Heraclitus (535 BC–475 BC) also realized the importance

of variability, prompting him to say: "The only thing that is constant is change"; "It is in changing that we find purpose"; "Nothing endures but change"; "No man ever steps in the same river twice, for it is not the same river and he is not the same man".

Heraclitus actually recognized the (fractional-order) dynamics of the river without modern scientific knowledge (in nature). After two thousand years, the integer-order calculus was invented by Sir Issac Newton and Gottfried Wilhelm Leibniz, whose main purpose was to quantify that change [94,95]. From then, scientists started using integer-order calculus to depict dynamic systems, differential equations, modelling, etc. In the 1950s, Scott Blair, who first introduced the FC into rheology, pointed out that the integer-order dynamic view of change is only for our own "convenience" (a little bit selfish). In other words, denying fractional calculus is equivalent to denying the existence of non-integers between the integers!

Table 1. The 10 Vs of big data.

Characteristics	Description
1. Volume	Best known characteristic of big data; more than 90 percent of the whole data were created in the past couple of years.
2. Velocity	The speed at which data are being generated.
3. Variety	Processing structured, unstructured and semistructured data.
4. Variability	Inconsistent speed of data loading, multitude of data dimensions, and number of inconsistencies.
5. Veracity	Confidence or trust in the data.
6. Validity	Refers to how accurate and correct the data are.
7. Vulnerability	Security concerns, data breaches.
8. Volatility	Design policy for data currency, availability, and rapid retrieval of information when required.
9. Visualization	Develop new tools considering the complex relationships between the above properties.
10. Value	The most important of the 10 Vs; substantial value must be found.

Table 2. Variability in multiple research topics.

Topics	Description
1. Climate variability	Changes in the components of the climate system and their interactions.
2. Genetic variability	Measurements of the tendencies of individual genotypes between regions.
3. Heart rate variability	Physiological phenomenon where the time interval between heart beats varies.
4. Human variability	Measurements of the characteristics, physical or mental, of human beings.
5. Spatial variability	Measurements at different spatial points exhibit different values.
6. Statistical variability	A measure of dispersion in statistics.

Blair said [96]: "We may express our concepts in Newtonian terms if we find this convenient but, if we do so, we must realize that we have made a translation into a language which is foreign to the system which we are studying (1950)".

Therefore, variability exists in big data. However, how do we realize the modeling, analysis and design (MAD) for the variability in big data within complex systems? We need fractional calculus! In other words, big data are at the nexus of complexity and FC. Thus, we first proposed fractional-order data analytics (FODA) in 2015. Metrics based on

using the fractional-order signal processing techniques should be used for quantifying the generating dynamics of observed or perceived variability [15].

2.1. Hurst Parameter, fGn, and fBm

The Hurst parameter or Hurst exponent (H) was proposed for the analysis of the long-term memory of time series. It was originally developed to quantify the long-term storage capacity of reservoirs for the Nile river's volatile rain and drought conditions more than a half century ago [16,17]. To date, the Hurst parameter has also been used to measure the intensity of long range dependence (LRD) in time series [97], which requires accurate modeling and forecasting. The self-similarity and the estimation of the statistical parameters of LRD have commonly been investigated recently [98]. The Hurst parameter has also been used for characterizing the LRD process [97,99]. A LRD time series is defined as a stationary process that has long-range correlations if its covariance function $C(n)$ decays slowly as:

$$\lim_{n\to\infty} \frac{C(n)}{n^{-\alpha}} = c, \tag{18}$$

where $0 < \alpha < 1$, which relates to the Hurst parameter according to $\alpha = 2 - 2H$ [100,101]. The parameter c is a finite, positive constant. When the value of n is large, $C(n)$ behaves as the IPL c/n^{α} [102]. Another definition for an LRD process is that the weakly stationary time-series $X(t)$ is said to be LRD if its power spectral density (PSD) follows:

$$f(\lambda) \sim C_f|\lambda|^{-\beta}, \tag{19}$$

as $\lambda \to 0$, for a given $C_f > 0$ and a given real parameter $\beta \in (0,1)$, which corresponds to $H = (1+\beta)/2$ [103]. When $0 < H < 0.5$, it indicates that the time intervals constitute a negatively correlated process. When $0.5 < H < 1$, it indicates that time intervals constitute a positively correlated process. When $H = 0.5$, it indicates that the process is uncorrelated.

Two of the most common LRD processes are fBm [104] and fGn [105]. The fBm process with $H(0 < H < 1)$ is defined as:

$$B_H(t) = \frac{1}{\Gamma(H+1/2)}\{\int_{-\infty}^{0}[(t-s)^{H-1/2} - (-s)^{H-1/2}]dW(s) + \int_{0}^{t}(t-s)^{H-1/2}dW(s)\}, \tag{20}$$

where W denotes a Wiener process defined on $(-\infty, \infty)$ [106]. The fGn process is the increment sequences of the fBm process, defined as:

$$X_k = Y(k+1) - Y(k), \tag{21}$$

where $Y(k)$ is a fBm process [107].

2.2. Fractional Lower-Order Moments (FLOMs)

The FLOM is based on α-stable PDFs. The PDFs of an α-stable distribution decay in the tails more slowly than a Gaussian PDF does. Therefore, for sharp spikes or occasional bursts in signals, an α-stable PDF can be used for characterizing signals more frequently than Gauss-distributed signals [108]. Thus, the FLOM plays an important role in impulsive processes [109], equivalent to the role played by the mean and variance in a Gaussian processes. When $0 < \alpha \leq 1$, the α-stable processes have no finite first- or higher-order moments; when $1 < \alpha < 2$, the α-stable processes have a finite first-order moment and all the FLOMs with moments of fractional order that is less than 1. The correlation between the FC and FLOM was investigated in [110,111]. For the Fourier-transform pair $p(x)$ and $\phi(\mu)$, the latter is the characteristic function and is the Fourier transform of the PDF; a complex FLOM can have complex fractional lower orders [110,111]. A FLOM-based fractional power spectrum includes a covariation spectrum and a fractional low-order covariance spectrum [112]. FLOM-based fractional power spectrum techniques have been successfully used in time-delay estimation [112].

2.3. Fractional Autoregressive Integrated Moving Average (FARIMA) and Gegenbauer Autoregressive Moving Average (GARMA)

A continuous-time linear time-invariant (LTI) system can be characterized using a linear difference equation, which is known as an autoregression and moving average (ARMA) model [113,114]. The process X_t of ARMA(p, q) is defined as:

$$\Phi(B)X_t = \Theta(B)\epsilon_t, \tag{22}$$

where ϵ_t is white Gaussian noise (wGn), and B is the backshift operator. However, the ARMA model can only describe a short-range dependence (SRD) property. Therefore, based on the Hurst parameter analysis, more suitable models, such as FARIMA [115,116] and fractional integral generalized autoregressive conditional heteroscedasticity (FIGARCH) [117], were designed to more accurately analyze the LRD processes. The most important feature of these models is the long memory characteristic. The FARIMA and FIGARCH can capture both the short- and the long-memory nature of time series. For example, the FARIMA process X_t is usually defined as [118]:

$$\Phi(B)(1-B)^d X_t = \Theta(B)\epsilon_t, \tag{23}$$

where $d \in (-0.5, 0.5)$, and $(1-B)^d$ is a fractional-order difference operator. The locally stationary long-memory FARIMA model has the same equation as that of Equation (23), except that d becomes d_t, which is a time-varying parameter [119]. The locally stationary long-memory FARIMA model captures the local self-similarity of the system.

The generalized locally stationary long-memory process FARIMA model was investigated in [119]. For example, a generalized FARIMA model, which is called the Gegenbauer autoregressive moving average (GARMA), was introduced in [120]. The GARMA model is defined as:

$$\Phi(B)(1-2uB+B^2)^d X_t = \Theta(B)\epsilon_t, \tag{24}$$

where $u \in [-1, 1]$, which is a parameter that can control the frequency at which the long memory occurs. The parameter d controls the rate of decay of the autocovariance function. The GARMA model can also be extended to the so-called "k-factor GARMA model", which allows for long-memory behaviors to be associated with each of k frequencies (Gegenbauer frequencies) in the interval [0, 0.5] [121].

2.4. Continuous Time Random Walk (CTRW)

The CTRW model was proposed by Montroll and Weiss as a generalization of diffusion processes to describe the phenomenon of anomalous diffusion [19]. The basic idea is to calculate the PDF for the diffusion process by replacing the discrete steps with continuous time, along with a PDF for step lengths and a waiting-time PDF for the time intervals between steps. Montroll and Weiss applied random intervals between the successive steps in the walking process to account for local structure in the environment, such as traps [122]. The CTRW has been used for modeling multiple complex phenomena, such as chaotic dynamic networks [123]. The correlation between CTRW and diffusion equations with fractional time derivatives has also been established [124]. Meanwhile, time-space fractional diffusion equations can be treated as CTRWs with continuously distributed jumps or continuum approximations of CTRWs on lattices [125].

2.5. Unmanned Aerial Vehicles (UAVs) and Precision Agriculture

As a new remote-sensing platform, researchers are more and more interested in the potential of small UAVs for precision agriculture [126–136], especially for heterogeneous crops, such as vineyards and orchards [137,138]. Mounted on UAVs, lightweight sensors, such as RGB cameras, multispectral cameras and thermal infrared cameras, can be used to collect high-resolution images. The higher temporal and spatial resolutions of the images, relatively low operational costs, and nearly real-time image acquisition make

the UAVs an ideal platform for mapping and monitoring the variability of crops and trees. UAVs can create big data and demand the FODA due to the "complexity" and, thus, variability inherent in the life process. For example, Figure 7 shows the normalized difference vegetation index (NDVI) mapping of a pomegranate orchard at a USDA ARS experimental field. Under different irrigation levels, the individual trees can show strong variability during the analysis of water stress. Life is complex! Thus, it entails variability, which as discussed above, in turn, entails fractional calculus. UAVs can then become "Tractor 2.0" for farmers in precision agriculture.

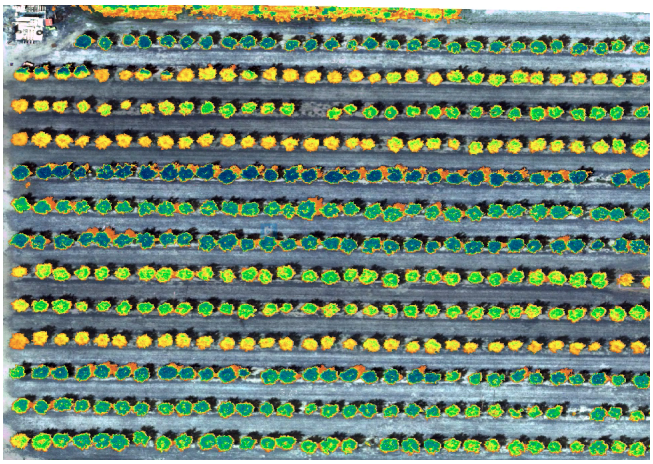


Figure 7. Normalized difference vegetation index (NDVI) mapping of pomegranate trees.

3. Optimal Machine Learning and Optimal Randomness

Machine learning (ML) is the science (and art) of programming computers so they can learn from data [139]. A more engineering-oriented definition was given by Tom Mitchell in 1997 [140], "A computer program is said to learn from experience E with respect to some task T and some performance measure P, if its performance on T, as measured by P, improves with experience E".

Most ML algorithms perform training by solving optimization problems that rely on first-order derivatives (Jacobians), which decide whether to increase or decrease weights. For huge speed boosts, faster optimizers are being used instead of the regular gradient descent optimizer. For example, the most popular boosters are momentum optimization [141], Nesterov acelerated gradient [21], AdaGrad [142], RMSProp [143] and Adam optimization [144]. The second-order (Hessian) optimization methods usually find the solutions with faster rates of convergence but with higher computational costs. Therefore, the answer to the following question is important: what is a more optimal ML algorithm? What if the derivative is fractional order instead of integer order? In this section, we discuss some applications of fractional-order gradients to optimization methods in machine-learning algorithms and investigate the accuracy and convergence rates.

As mentioned in the big data section, there is a huge amount of data in human society and nature. During the learning process of ML, we care not only about the speed, but also the accuracy of the data the machine is learning (Figure 8). The learning algorithm is important; otherwise, the data labeling and other labor costs will exhaust people beyond their abilities. When applying the accoladed artificial intelligence (AI) to an algorithm, a strong emphasis is on artificial, only followed weakly by intelligence. Therefore, the key to ML is what optimization methods are being applied. The convergence rate and global searching are two important parts of the optimization method.

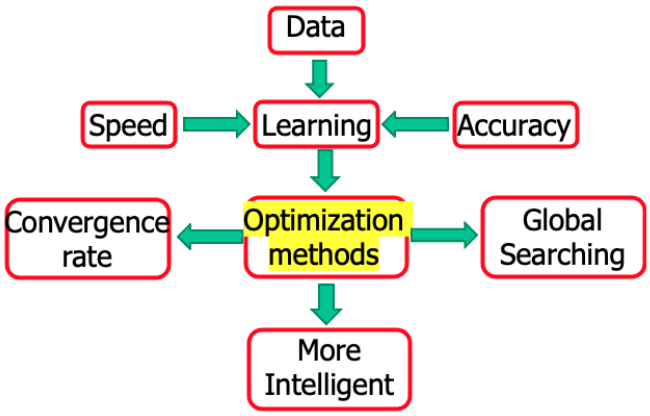


Figure 8. Data analysis in nature.

Reflection: ML is, today, a hot research topic and will probably remain so into the near future. How a machine can learn efficiently (optimally) is always important. The key for the learning process is the optimization method. Thus, in designing an efficient optimization method, it is necessary to answer the following three questions:

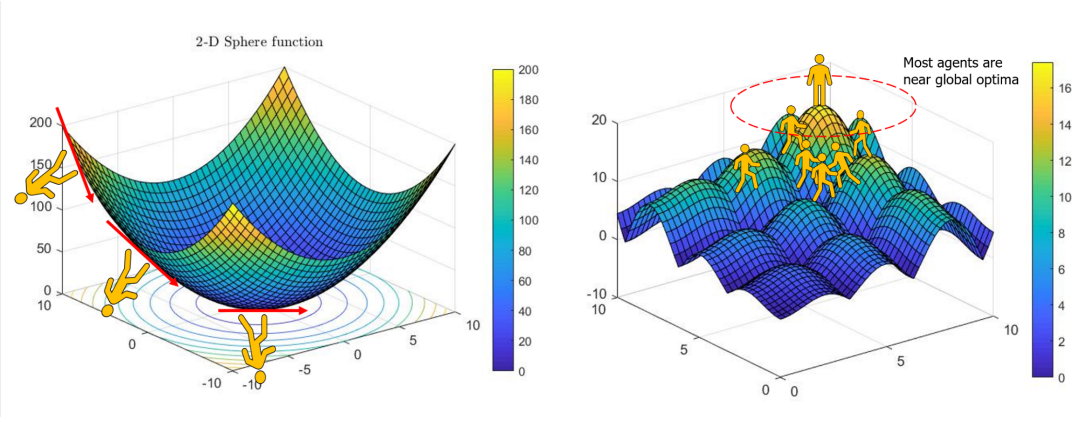
- What is the optimal way to optimize?
- What is the **more optimal** way to optimize?
- Can we demand **"more optimal machine learning"**, for example, deep learning with the minimum/smallest labeled data)?

Optimal randomness: In the section on the Lévy PDF, the Lévy flight is the search strategy for food the albatross has developed over millions of years of evolution. Admittedly, this is a slow optimization procedure [84]. From this perspective, we should call "Lévy distribution" an optimized or learned randomness used by albatrosses for searching for food. Therefore, we pose the question: "can the search strategy be more optimal than Lévy flight?" The answer is yes if one adopts the FC [145]! Optimization is a very complex area of study. However, a few studies have investigated using FC to obtain a better optimization strategy.

Theoretically, there are two broad optimization categories; these are derivative-free and gradient-based. For the derivative-free methods, there are the direct-search methods, consisting of particle swarm optimization (PSO) [146,147], etc. For the gradient-based methods, there are gradient descent and its variants. Both of the two categories have shown better performance when using the FC as demonstrated below.

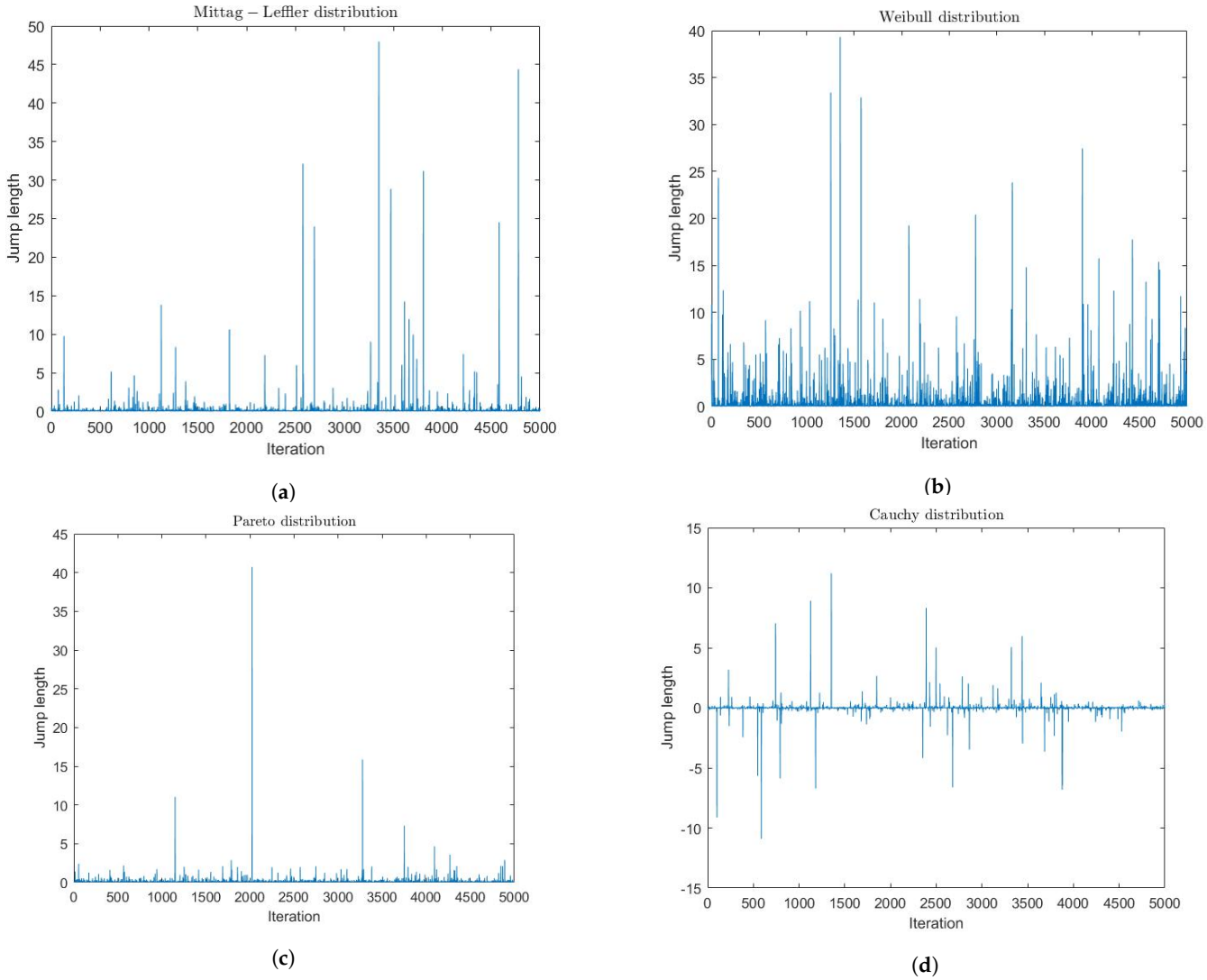
3.1. Derivative-Free Methods

For derivative-free methods, there are single agent search and swarm-based search methods (Figure 9). Exploration is often achieved by randomness or random numbers in terms of some predefined PDFs. Exploitation uses local information such as gradients to search local regions more intensively, and such intensification can enhance the rate of convergence. Thus, a question was posed: what is the optimal randomness? Wei et al. [148] investigated the optimal randomness in a swarm-based search. Four heavy-tailed PDFs have been used for sample path analysis (Figure 10). Based on the experimental results, the randomness-enhanced cuckoo search (CS) algorithms [66,149,150] can identify the unknown specific parameters of a fractional-order system with better effectiveness and robustness. The randomness-enhanced CS algorithms can be considered as a promising tool for solving real-world complex optimization problems. The reason is that optimal randomness is applied with fractional-order noise during the exploration, which is more optimal than the "optimized PSO", CS. The fractional-order noise refers to the stable PDFs [48]. In other words, when we are discussing optimal randomness, we are discussing fractional calculus!



(**a**) (**b**)

Figure 9. The 2-D Alpine function for derivative-free methods; there are (**a**) single agent search and (**b**) swarm-based search methods.



(**a**) (**b**)

(**c**) (**d**)

Figure 10. Sample paths. Wei et al. [148] investigated the optimal randomness in a swarm-based search. Four heavy-tailed PDFs were used for sample path analysis; there are (**a**) Mittag-Leffler distribution, (**b**) Weibull distribution, (**c**) Pareto distribution, and (**d**) Cauchy distribution. The Long steps, referring to the jump length, frequently happened for all distributions, which showed strong heavy-tailed performance. For more details, please refer to [148].

3.2. The Gradient-Based Methods

The gradient descent (GD) is a very common optimization algorithm, which can find the optimal solutions by iteratively tweaking parameters to minimize the cost function. The stochastic gradient descent (SGD) randomly selects times during the training process. Therefore, the cost function bounces up and down, decreasing on average, which is good for escape from local optima. Sometimes, noise is added into the GD method, and usually, such noise follows a Gaussian PDF in the literature. We ask, "why not heavy-tailed PDFs"? The answer to this question could lead to interesting future research.

Nesterov Accelerated Gradient Descent (NAGD)

There are many variants of GD analysis as suggested in Figure 11. One of the most popular methods is the NAGD [21]:

$$\begin{cases} y_{k+1} = ay_k - \mu \nabla f(x_k), \\ x_{k+1} = x_k + y_{k+1} + by_k, \end{cases} \tag{25}$$

where by setting $b = -a/(1+a)$, one can derive the NAGD. When $b = 0$, one can derive the momentum GD. The NAGD can also be formulated as:

$$\begin{cases} x_k = y_{k-1} - \mu \nabla f(y_{k-1}), \\ y_k = x_k + \frac{k-1}{k+2}(x_k - x_{k-1}). \end{cases} \tag{26}$$

Set $t = k\sqrt{\mu}$, and one can, in the continuous limit, derive the corresponding differential equation:

$$\ddot{X} + \frac{3}{t}\dot{X} + \nabla f(X) = 0. \tag{27}$$

The main idea of Jordan's work [151] is to analyze the iteration algorithm in the continuous-time domain. For differential equations, one can use the Laypunov or variational method to analyze the properties; for example, the convergence rate is $O(\frac{1}{t^2})$. One can also use the variational method to derive the master differential equation for an optimization method, such as the least action principle [152], Hamilton's variational principle [153] and the quantum-mechanical path integral approach [154]. Wilson et al. [151] built a Euler–Lagrange function to derive the following equation:

$$\ddot{X}_t + 2\gamma \dot{X}_t + \frac{\gamma^2}{\mu}\nabla f(X_t) = 0. \tag{28}$$

which is in the same form as the master differential equation of NAGD.

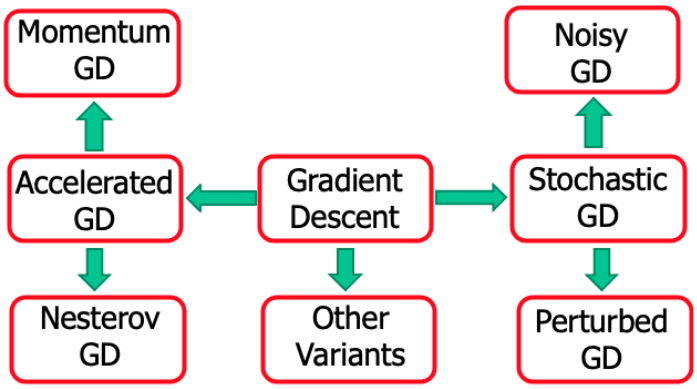


Figure 11. Gradient descent and its variants.

Jordan's work revealed that one can transform an iterative (optimization) algorithm to its continuous-time limit case, which can simplify the analysis (Laypunov methods). One can directly design a differential equation of motion (EOM) and then discretize it to derive an iterative algorithm (variational method). The key is to find a suitable Laypunov functional to analyze the stability and convergent rate. The new exciting fact established by Jordan is that optimization algorithms can be systematically synthesized using Lagrangian mechanics (Euler–Lagrange) through EOMs.

Thus, is there an optimal way to optimize using optimization algorithms stemming from Equation (28)? Obviously, why not an equation such as Equation (28) of fractional order? Considering the $\dot{X}_t$ as $X_t^{(\alpha)}$, it will provide us with more research possibilities, such as the fractional-order calculus of variation (FOCV) and fractional-order Euler–Lagrange (FOEL) equation. For the SGD, optimal randomness using the fractional-order noises can also offer better than the best performance, similarly shown by Wei et al. [148].

3.3. What Can the Control Community Offer to ML?

In the IFAC 2020 World Congress Pre-conference Workshop, Eric Kerrigan proposed "The Three Musketeers" that the control community can contribute to ML [155]. These three are the IMP [23], the Nu-Gap metric [156] and model discrimination [157]. Herein, we focused on the IMP. Kashima et al. [158] transferred the convergence problem of numerical algorithms into a stability problem of a discrete-time system. An et al. [159] explained that the commonly used SGD-momentum algorithm in ML is a PI controller and designed a PID algorithm. Motivated by [159] but differently from M. Jordan's work, we proposed designing and analyzing the algorithms in the S or Z domain. Remember that GD is a first-order algorithm:

$$x_{k+1} = x_k - \mu \nabla f(x_k), \tag{29}$$

where $\mu > 0$ is the step size (or learning rate). Using the Z transform, one can achieve:

$$X(z) = \frac{\mu}{z-1}[-\nabla f(x_k)]_z. \tag{30}$$

Approximate the gradient around the extreme point x^*, and one can obtain:

$$\nabla f(x_k) \approx A(x_k - x^*), \; with \; A = \nabla^2 f(x^*). \tag{31}$$

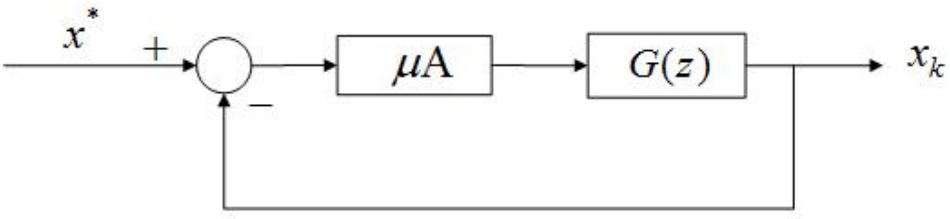


Figure 12. The integrator model (embedded in $G(z)$). The integrator in the forward loop eliminates the tracking steady-state error for a constant reference signal (internal model principle (IMP)).

For the plain GD in Figure 12, we have $G(z) = 1/(z-1)$, which is an integrator. For fractional-order GD (FOGD), the updating term of x_k in Equation (29) can be treated as a filtered gradient signal. In [160], Fan et al. shared similar thoughts: "Accelerating the convergence of the moment method for the Boltzmann equation using filters". The integrator in the forward loop eliminates the tracking error for a constant reference signal according to the internal model principle (IMP). Similarly, the GD momentum (GDM) designed to accelerate the conventional GD, which is popularly used in ML, can be analyzed using Figure 12 by:

$$\begin{cases} y_{k+1} = \alpha y_k - \mu \nabla f(x_k), \\ x_{k+1} = x_k + y_{k+1}, \end{cases} \tag{32}$$

where y_k is the accumulation of the history gradient and $\alpha \in (0,1)$ is the rate of the moving average decay. Using the Z transform for the update rule, one can derive:

$$\begin{cases} zY(z) = \alpha Y(z) - \mu[\nabla f(x_k)]_z, \\ zX(z) = X(z) + zY(z). \end{cases} \tag{33}$$

Then, after some algebra, one obtains the following equation:

$$X(z) = \frac{\mu z}{(z-1)(z-\alpha)}[-\nabla f(x_k)]_z. \tag{34}$$

For the GD momentum, we have $G(z) = \frac{z}{(z-1)(z-\alpha)}$ in Figure 12, with an integrator in the forward loop. The GD momentum is a second-order ($G(z)$) algorithm with an additional pole at $z = \alpha$ and one zero at $z = 0$. The "second-order" refers to the order of $G(z)$, which makes it different from the algorithm using the *Hessian* matrix information. Moreover, NAGD can be simplified as:

$$\begin{cases} y_{k+1} = x_k - \mu \nabla f(x_k), \\ x_{k+1} = (1-\lambda) y_{k+1} + \lambda y_k, \end{cases} \tag{35}$$

where μ is the step size and λ is a weighting coefficient. Using the Z transform for the update rule, one can derive:

$$\begin{cases} zY(z) = X(z) - \mu[\nabla f(x_k)]_z, \\ zX(z) = (1-\lambda) zY(z) + \lambda Y(z). \end{cases} \tag{36}$$

Different from the GD momentum, and after some algebra, one obtains:

$$X(z) = \frac{-(1-\lambda)z - \lambda}{(z-1)(z+\lambda)}\mu[\nabla f(x_k)]_z = \frac{z + \frac{\lambda}{1-\lambda}}{(z-1)(z+\lambda)}\mu(1-\lambda)[-\nabla f(x_k)]_z. \tag{37}$$

For NAGD, we have $G(z) = \frac{z+\frac{\lambda}{1-\lambda}}{(z-1)(z+\lambda)}$, again, with an integrator in the forward loop (Figure 12). NAGD is a second-order algorithm with an additional pole at $z = -\lambda$ and a zero at $z = \frac{-\lambda}{1-\lambda}$.

"Can $G(z)$ be of higher order or fractional order"? Of course it can! As shown in Figure 12, a necessary condition for the stability of an algorithm is that all the poles of the closed-loop system are within the unit disc. If the Lipschitz continuous gradient constant L is given, one can replace A with L, and then, the condition is sufficient. For each $G(z)$, there is a corresponding iterative optimization algorithm. $G(z)$ can be a third- or higher-order system. Apparently, $G(z)$ can also be a fractional-order system. Considering a general second-order discrete system:

$$G(z) = \frac{z+b}{(z-1)(z-a)}, \tag{38}$$

the corresponding iterative algorithm is Equation (25). As mentioned earlier, when setting $b = -a/(1+a)$, one can derive the NAGD. When $b = 0$, one can derive the momentum GD. The iterative algorithm can be viewed as a state-space realization of the corresponding system. Thus, it may have many different realizations (all are equivalent). Since two parameters a and b are introduced for a general second-order algorithm design, we used the integral squared error (ISE) as the criterion to optimize the parameters. This is because for different target functions $f(x)$, the Lipschitz continuous gradient constant is different. Thus, the loop forward gain is defined as $\rho := \mu A$.

Table 3. General second-order algorithm design. The parameter ρ is the loop forward gain; see text for more details.

ρ	**0.4**	**0.8**	**1.2**	**1.6**	**2.0**	**2.4**
a	-0.6	-0.2	0.2	0.6	1	1.4
b	1.5	0.25	-0.1667	-0.3750	-0.5	-0.5833

According to the experimental results (Table 3), interestingly, it is found that the optimal a and b satisfy $b = -a/(1+a)$, which is the same design as NAGD. Other criteria such as the IAE and ITAE were used to find other optimal parameters, but the results are the same as for the ISE. Differently from for NAGD, the parameters were determined by search optimization rather than by mathematical design, which can be extended to more general cases. The algorithms were then tested using the MNIST dataset (Figure 13). It is obvious that for different zeros and poles, the performance of the algorithms is different. One finds that both the $b = -0.25$ and $b = -0.5$ cases perform better than does the SGD momentum. Additionally, both $b = 0.25$ and $b = 0.5$ perform worse. It is also shown that an additional zero can improve the performance, if adjusted properly. It is interesting to observe that both the method and the Nesterov method give an optimal choice of the zero, which is closely related to the pole ($b = -a/(1+a)$).

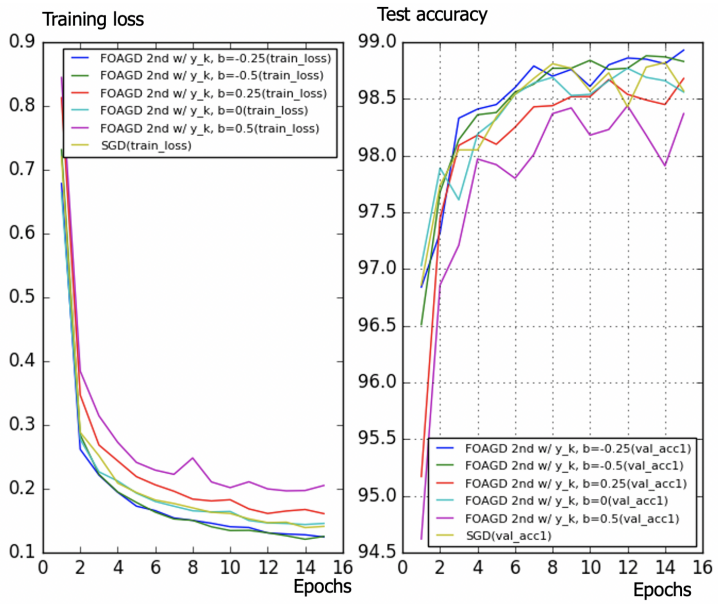


Figure 13. Training loss (**left**); test accuracy (**right**). It is obvious that for different zeros and poles, the performance of the algorithms is different. One finds that both the $b = -0.25$ and $b = -0.5$ cases perform better than does the stochastic gradient descent (SGD) momentum. Additionally, both $b = 0.25$ and $b = 0.5$ perform worse. It is also shown that an additional zero can improve the performance, if adjusted carefully.

Now, let us consider a general third-order discrete system:

$$G(z) = \frac{z^2 + cz + d}{(z-1)(z^2 + az + b)}. \tag{39}$$

Set $b = d = 0$; it will reduce to the second-order algorithm discussed above. Compared with the second-order case, the poles can now be complex numbers. More generally, a higher-order system can contain more internal models. If all the poles are real, then:

$$G(z) = \frac{1}{(z-1)} \frac{(z-c)}{(z-a)} \frac{(z-d)}{(z-b)}, \tag{40}$$

whose corresponding iterative optimization algorithm is

$$\begin{cases} y_{k+1} = y_k - \mu \nabla f(x_k), \\ z_{k+1} = a z_k + y_{k+1} - c y_k, \\ x_{k+1} = b x_k + z_{k+1} - d z_k. \end{cases} \tag{41}$$

Table 4. General third-order algorithm design, with parameters defined by Equation (41).

ρ	**0.4**	**0.8**	**1.2**	**1.6**	**2.0**	**2.4**
a	0.6439	0.5247	−0.4097	−0.5955	−1.0364	−1.4629
b	0.0263	0.0649	0.0419	−0.0398	0.0364	0.0880
c	1.5439	0.5747	−0.3763	−0.3705	−0.5364	−0.6462
d	0.0658	0.0812	0.0350	−0.0408	0.0182	0.0367

After some experiments (Table 4), it was found that since the ISE was used for tracking a step signal (it is quite simple), the optimal poles and zeros are the same as for the second-order case with a pole-zero cancellation. This is an interesting discovery. In this optimization result, all the poles and zeros are real, and the resulting performance is not very good, as expected. Compare this with the second-order case; the only difference is that in the latter, complex poles can possibly appear. Thus, the question arises: "how do complex poles play a role in the design?" The answer is obvious: by fractional calculus!

Inspired by M. Jordan's idea in the frequency domain, a continuous time fractional-order system was designed:

$$G(s) = \frac{1}{s(s^\alpha + \beta)}, \tag{42}$$

where $\alpha \in (0, 2)$, $\beta \in (0, 20]$ at first. It was then found that the optimal parameters were obtained by searching using the ISE criterion (Table 5).

Table 5. The continuous time fractional-order system.

ρ	**0.3**	**0.5**	**0.7**	**0.9**
α	1.8494	1.6899	1.5319	1.2284
β	20	20	20	20

Equation (42) encapsulates the continuous-time design, and one can use the numerical inverse Laplace transform (NILP) [161] and Matlab command **stmcb( )** [162] to derive its discrete form. After the complex poles are included, one can have:

$$G(z) = \frac{(z+c)}{(z-1)}\left(\frac{1}{z-a+jb} + \frac{1}{z-a-jb}\right) \tag{43}$$

whose corresponding iterative algorithm is:

$$\begin{cases} y_{k+1} = a y_k - b z_k - \mu \nabla f(x_k), \\ z_{k+1} = a z_k + b y_k, \\ x_{k+1} = x_k + y_{k+1} + c y_k. \end{cases} \tag{44}$$

Then, the algorithms were tested again using the MNIST dataset, and the results were compared with the SGD's. For the fractional order, $\rho = 0.9$ was used, $a = 0.6786$, $b = 0.1354$, and different values for zero c were used. When $c = 0$, the result was similar to that for the second-order SGD. When c was not equal to 0, the result was similar to that for the second-order NAGD. For the SGD, α was set to be 0.9, and the learning rate was 0.1

(Figure 14). Both $c = 0$ and $c = 0.283$ perform better than the SGD momentum; generally, with appropriate values of c, better performance can be achieved than in the second-order cases. The simulation results demonstrate that fractional calculus (complex poles) can potentially improve the performance, which is closely related to the learning rate.

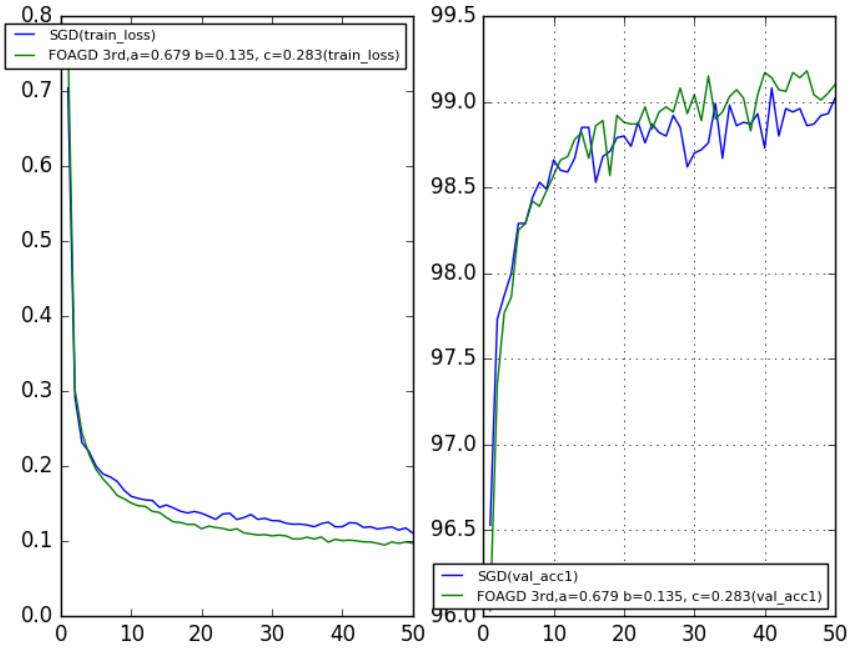


Figure 14. Training loss (**left**); test accuracy (**right**).

In general, M. Jordan asked the question: "is there an optimal way to optimize?". Our answer is a resounding yes, by limiting dynamics analysis and discretization and SGD with other randomness, such as Langevin motion. Herein, the question posed was: "is there a more optimal way to optimize?". Again, the answer is yes, but it requires the fractional calculus to be used to optimize the randomness in SGD, random search and the IMP. There is more potential for further investigations along this line of ideas.

4. A Case Study of Machine Learning with Fractional Calculus: A Stochastic Configuration Network with Heavytailedness

4.1. Stochastic Configuration Network (SCN)

The SCN model is generated incrementally by using stochastic configuration (SC) algorithms [163]. Compared with the existing randomized learning algorithms for single-layer feed-forward neural networks (SLFNNs) [164], the SCN can randomly assign the input weights (w) and biases (b) of the hidden nodes in a supervisory mechanism, which is selecting random parameters with an inequality constraint and assigning the scope of the random parameters adaptively. It can ensure that the built randomized learner models have a universal approximation property. Then, the output weights are analytically evaluated in either a constructive or selective manner [163]. In contrast with the known randomized learning algorithms, such as the randomized radial basis function (RRBF) networks [165] and the random vector functional link (RVFL) [166], the SCN can provide good generalization performance at a faster speed. Concretely, there are three types of SCN algorithms, which are labeled for convenience as SC-I, SC-II and SC-III.

The SC-I algorithm uses a constructive scheme to evaluate the output weights only for the newly added hidden node [167]. All of the previously obtained output weights are kept the same. The SC-II algorithm recalculates part of the current output weights by analyzing a local-least-squares problem with a user-defined shifting window size. The SC-III algorithm finds all the output weights together by solving a global-least-squares problem. The SCN

has better performance than other randomized neural networks in terms of fast learning, the scope of the random parameters, and the required human intervention. Therefore, it has already been used in many data-processing projects, such as [134,168,169].

4.2. SCN with Heavy-Tailed PDFs

For the original SCN algorithms, weights and biases are randomly generated using a uniform PDF. Randomness plays a significant role in both exploration and exploitation. A good neural network architecture with randomly assigned weights can easily outperform a more deficient architecture with finely tuned weights [170]. Therefore, it is critical to discuss the optimal randomness for the weights and biases in SCN algorithms. Heavy-tailed PDFs have shown optimal randomness for finding targets [171,172], which plays a significant role in exploration and exploitation [148]. Therefore, herein, heavy-tailed PDFs were used to randomly update the weights and biases in the hidden layers to determine if the SCN models display improved performance. Some of the key parameters of the SCN models are listed in Table 6. For example, the maximum times of random configuration T_{max} are set as 200. The scale factor lambda in the activation function, which directly determines the range for the random parameters, was examined by using different settings (0.5–200). The tolerance was set as 0.05. Most of the parameters for the SCN with heavy-tailed PDFs were kept the same with the original SCN algorithms for comparison purposes. For more details, please refer to [163] and Appendix A.

Table 6. Stochastic configuration networks (SCNs) with key parameters.

Properties	Values
Name:	"Stochastic Configuration Networks"
Version:	"1.0 beta"
L:	hidden node number
W:	input weight matrix
b:	hidden layer bias vector
Beta:	output weight vector
r:	regularization parameter
tol:	tolerance
Lambda:	random weight range
L_{max}:	maximum number of hidden neurons
T_{max}:	maximum times of random configurations
nB:	number of node being added in one loop

4.3. A Regression Model and Parameter Tuning

The dataset of the regression model was generated by a real-valued function [173]:

$$f(x) = 0.2e^{-(10x-4)^2} + 0.5e^{-(80x-40)^2} + 0.3e^{-(80x-20)^2}, \quad (45)$$

where x ∈ [0, 1]. There were 1000 points randomly generated from the uniform distribution on the unit interval [0, 1] in the training dataset. The test set had 300 points generated from a regularly spaced grid on [0, 1]. The input and output attributes were normalized into [0, 1], and all the results reported in this research represent averages over 1000 independent trials. The settings of the parameters were similar to for the SCN in [163].

Heavy-tailed PDF algorithms have user-defined parameters, for example, the power-law index for SCN-Lévy, and location and scale parameters for SCN-Cauchy and SCN-Weibull, respectively. Thus, to illustrate the effect of parameters on the optimization results and to offer reference values for the proposed SCN algorithms, parameter analysis was conducted, and corresponding experiments were performed. Based on the experimental results, for the SCN-Lévy algorithm, the most optimal power-law index is 1.1 for achieving the minimum number of hidden nodes. For the SCN-Weibull algorithm, the optimal location parameter α and scale parameter β for the minimum number of hidden nodes are 1.9

and 0.2, respectively. For the SCN-Cauchy algorithm, the optimal location parameter α and scale parameter β for the minimum number of hidden nodes are 0.9 and 0.1, respectively.

Performance Comparison among SCNs with Heavy-Tailed PDFs

In Table 7, the performance of SCN, SCN-Lévy, SCN-Cauchy, SCN-Weibull and SCN-Mixture are shown, in which mean values are reported based on 1000 independent trials. Wang et al. [163] used time cost to evaluate the SCN algorithms' performance. In the present study, the authors used the mean hidden node numbers to evaluate the performance. The number of hidden nodes is associated with modeling accuracy. Therefore, herein, the analysis determined if an SCN with heavy-tailed PDFs used fewer hidden nodes to generate high performance, which would make the NNs less complex. According to the numerical results, the SCN-Cauchy used the lowest number of mean hidden nodes, 59, with an root mean squared error (RMSE) of 0.0057. The SCN-Weibull had a mean number of 63 hidden nodes, with an RMSE of 0.0037. The SCN-Mixture had a mean number of 70 hidden nodes, with an RMSE of 0.0020. The mean number of hidden nodes for SCN-Lévy was also 70. The original SCN model had a mean number of 75 hidden nodes. A more detailed training process is shown in Figure 15. With fewer hidden node numbers, the SCN models with heavy-tailed PDFs can be faster than the original SCN model. The neural network structure is also less complicated than the SCN. Our numerical results for the regression task demonstrate remarkable improvements in modeling performance compared with the current SCN model results.

Table 7. Performance comparison of SCN models for regression problem.

Models	Mean Hidden Node Number	RMSE
SCN	75 ± 5	0.0025
SCN-Lévy	70 ± 6	0.0010
SCN-Cauchy	59 ± 3	0.0057
SCN-Weibull	63 ± 4	0.0037
SCN-Mixture	70 ± 5	0.0020

4.4. MNIST Handwritten Digit Classification

The handwritten digit dataset contains 4000 training examples and 1000 testing examples, a subset of the MNIST handwritten digit dataset. Each image is a 20-by-20-pixel grayscale image of the digit (Figure 16). Each pixel is represented by a number indicating the grayscale intensity at that location. The 20-by-20 grid of pixels is "unrolled" into a 400-dimensional vector.

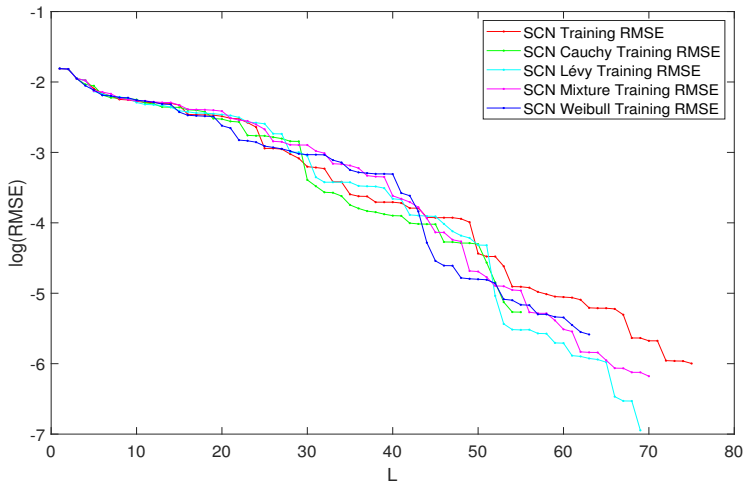


Figure 15. Performance of SCN, SCN-Lévy, SCN-Weibull, SCN-Cauchy and SCN-Mixture. The parameter L is the hidden node number.

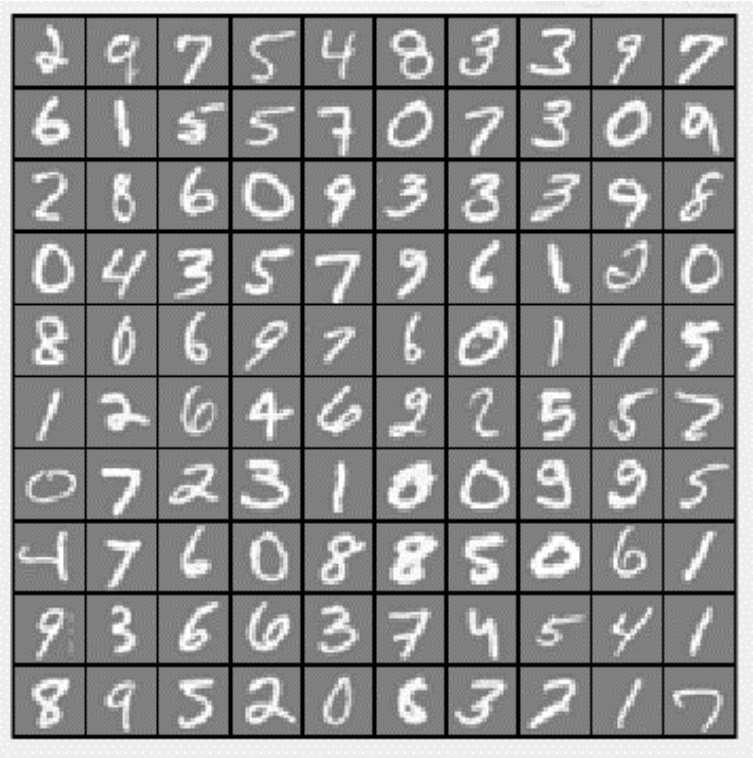


Figure 16. The handwritten digit dataset example.

Similar to the parameter tuning for the regression model, parameter analysis was conducted to illustrate the impact of parameters on the optimization results and to offer reference values for the MNIST handwritten digit classification SCN algorithms. Corresponding experiments were performed. According to the experimental results, for the SCN-Lévy algorithm, the most optimal power law index is 1.6 for achieving the best RMSE performance. For the SCN-Cauchy algorithm, the optimal location parameter α and scale parameter β for the lowest RMSE are 0.2 and 0.3, respectively.

Performance Comparison among SCNs on MNIST

The performance of the SCN, SCN-Lévy, SCN-Cauchy and SCN-Mixture are shown in Table 8. Based on the experimental results, the SCN-Cauchy, SCN-Lévy and SCN-Mixture have better performance in training and test accuracy, compared with the original SCN model. A detailed training process is shown in Figure 17. Within around 100 hidden nodes, the SCN models with heavy-tailed PDFs perform similarly to the original SCN model. When the number of the hidden nodes is greater than 100, the SCN models with heavy-tailed PDFs have lower RMSEs. Since more parameters for weights and biases are initialized in heavy-tailed PDFs, this may cause an SCN with heavy-tailed PDFs to converge to the optimal values at a faster speed. The experimental results for the MNIST handwritten classification problem demonstrate improvements in modeling performance. They also show that SCN models with heavy-tailed PDFs have a better search ability for achieving lower RMSEs.

Table 8. Performance comparison of SCNs.

Models	Training Accuracy	Test Accuracy
SCN	$94.0 \pm 1.9\%$	$91.2 \pm 6.2\%$
SCN-Lévy	$94.9 \pm 0.8\%$	$91.7 \pm 4.5\%$
SCN-Cauchy	$95.4 \pm 1.3\%$	$92.4 \pm 5.5\%$
SCN-Mixture	$94.7 \pm 1.1\%$	$91.5 \pm 5.3\%$

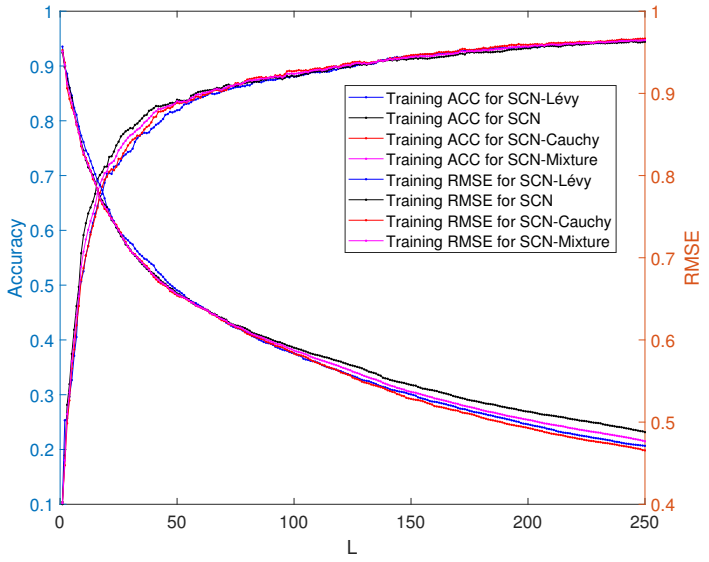


Figure 17. Classification performance of SCNs.

5. Take-Home Messages and Looking into the Future: Fractional Calculus Is Physics Informed

Big data and machine learning (ML) are two of the hottest topics of applied scientific research, and they are closely related to one another. To better understand them, in this article, we advocate fractional calculus (FC), as well as fractional-order thinking (FOT), for big data and ML analysis and applications. In Section 2, we discussed the relationships between big data, variability and FC, as well as why fractional-order data analytics (FODA) should be used and what it is. The topics included the Hurst parameter, fractional Gaussian noise (fGn), fractional Brownian motion (fBm), the fractional autoregressive integrated moving average (FARIMA), the formalism of continuous time random walk (CTRW), unmanned aerial vehicles (UAVs) and precision agriculture (PA).

In Section 3, how to learn efficiently (optimally) for ML algorithms is discussed. The key to developing an efficient learning process is the method of optimization. Thus, it is important to design an efficient optimization method. The derivative-free methods, as well as the gradient-based methods, such as the Nesterov accelerated gradient descent (NAGD), are discussed. Furthermore, it is shown to be possible, following the internal model principle (IMP), to design and analyze the ML algorithms in the S or Z transform domain in Section 3.3. FC is used in optimal randomness in the methods of stochastic gradient descent (SGD) and random search. Nonlocal models have commonly been used to describe physical systems and/or processes that cannot be accurately described by classical approaches [174]. For example, fractional nonlocal Maxwell's equations and the corresponding fractional wave equations were applied in [175] for fractional vector calculus [176]. The nonlocal differential operators [177], including nonlocal analogs of the gradient/Hessian, are the key of these nonlocal models, which could lead to very interesting research with FC in the near future.

Fractional dynamics is a response to the need for a more advanced characterization of our complex world to capture structure at very small or very large scales that had previously been smoothed over. If one wishes to obtain results that are better than the best possible using integer-order calculus-based methods, or are "more optimal", we advocate applying FOT and going fractional! In this era of big data, decision and control need FC, such as fractional-order signals, systems and controls. The future of ML should be physics-informed, scientific (cause–effect embedded or cause–effect discovery) and involving the use of FC, where the modeling is closer to nature. Laozi (unknown, around the 6th century to 4th century BC), the ancient Chinese philosopher, is said to have written

a short book *Dao De Jing*(*Tao Te Ching*), in which he observed: "The Tao that can be told is not the eternal Tao" [178]. People over thousands of years have shared different understandings of the meaning of the Tao. Our best understanding of the Tao is nature, whose rules of complexity can be explained in a non-normal way. Fractional dynamics, FC and heavytailedness may well be that non-normal way (Figure 18), at least for the not-too-distant future.

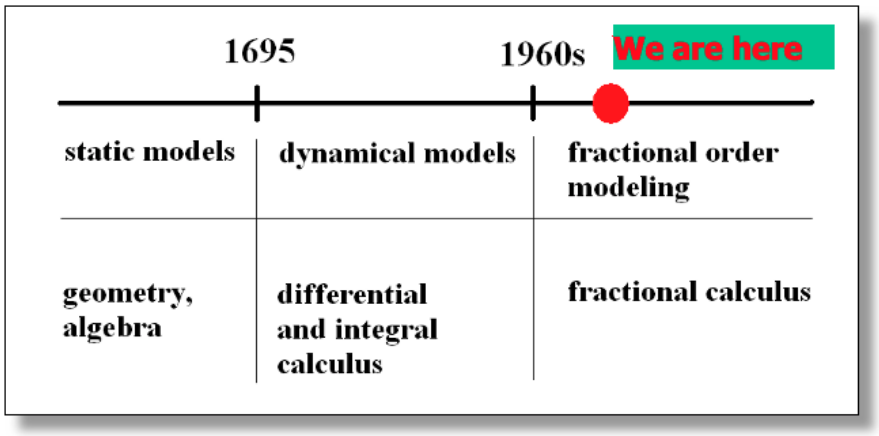


Figure 18. Timeline of FC (courtesy of Professor Igor Podlubny).

Author Contributions: H.N. drafted the original manuscript based on numerous talks/discussions with Y.C. in the past several years plus a seminar (http://mechatronics.ucmerced.edu/news/2020/why-big-data-and-machine-learning-must-meet-fractional-calculus) (accessed on 2 February 2021). Y.C. and B.J.W. contributed to the result interpretation, discussions and editing of the manuscript. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: Thanks go to Jiamin Wei, Yuquan Chen, Guoxiang Zhang, Tiebiao Zhao, Lihong Guo, Zhenlong Wu, Yanan Wang, Panpan Gu, Jairo Viola, Jie Yuan, etc., for walks, chats and tea/coffee breaks at Castle, Atwater, CA, before the COVID-19 era. In particular, Yuquan Chen performed computation in various IMP-based GD schemes, and Jiamin Wei performed the computation in cuckoo searches using four different heavy-tailed randomnesses. YangQuan Chen would like to thank Justin Dianhui Wang for many fruitful discussions in the past years on SCN, in particular, and machine learning in general. We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan X Pascal GPU used for this research. Last but not least, we thank the helpful reviewers for constructive comments.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

ACF	Auto-Correlation Function
AI	Artificial Intelligence
ARMA	Autoregression and Moving Average
CLT	Classical Central Limit Theorem
CS	Cuckoo Search
CTRW	Continuous Time Random Work

EOM	Equation of Motion
fBm	Fractional Brownian Motion
fGn	Fractional Gaussian Noise
FARIMA	Fractional Autoregressive Integrated Moving Average
FC	Fractional Calculus
FIGARCH	Fractional Integral Generalized Autoregressive Conditional Heteroscedasticity
FLOM	Fractional Lower-Order Moments
FOCV	Fractional-Order Calculus of Variation
FODA	Fractional-Order Data Analytics
FOEL	Fractional-Order Euler–Lagrange
FOT	Fractional-Order Thinking
GARMA	Gegenbauer Autoregressive Moving Average
GD	Gradient Descent
GDM	Gradient Descent Momentum
GEV	Generalized Extreme Value
IMP	Internal Model Principle
IPL	Inverse Power Law
ISE	Integral Squared Error
LGD	Long Range Dependence
LTI	Linear Time Invariant
MAD	Modeling, Analysis and Design
ML	Machine Learning
MLL	Mittag–Leffler Law
MNIST	Modified National Institute of Standards and Technology Database
NAGD	Nesterov Accelerated Gradient Descent
NDVI	Normalized Difference Vegetation Index
NILT	Numerical Inverse Laplace Transform
NN	Neural Networks
PA	Precision Agriculture
PDF	Probability Density Function
PID	Proportional, Integral, Derivative
PSO	Particle Swarm Optimization
RBF	Randomized Radial Basis Function (RBF) Networks
RGB	Red, Green, Blue
RMSE	Root Mean Squared Error
RVFL	Random Vector Functional Link
RW-FNN	Feed-Forward Networks with Random Weights
SCN	Stochastic Configuration Network
SGD	Stochastic Gradient Descent
SLFNNs	Single-Layer Feed-Forward Neural Networks
UAVs	Unmanned Aerial Vehicles
USDA	United States Department of Agriculture
wGn	White Gaussian Noise

Appendix A. SCN Codes

The Matlab and Python codes can be found at https://github.com/niuhaoyu16/StochasticConfigurationNetwork (accessed on 2 February 2021).

References

1. Vinagre, B.M.; Chen, Y. Lecture notes on fractional calculus applications in automatic control and robotics. In Proceedings of the 41st IEEE CDC Tutorial Workshop, Las Vegas, NV, USA, 9 December 2002; pp. 1–310.
2. Valério, D.; Machado, J.; Kiryakova, V. Some Pioneers of the Applications of Fractional Calculus. *Fract. Calc. Appl. Anal.* **2014**, *17*, 552–578. [CrossRef]
3. Abel, N. Solution of a Couple of Problems by Means of Definite Integrals. *Mag. Naturvidenskaberne* **1823**, *2*, 2.
4. Podlubny, I.; Magin, R.L.; Trymorush, I. Niels Henrik Abel and the Birth of Fractional Calculus. *Fract. Calc. Appl. Anal.* **2017**, *20*, 1068–1075. [CrossRef]
5. Ross, B. The Development of Fractional Calculus 1695–1900. *Hist. Math.* **1977**, *4*, 75–89. [CrossRef]

6. Tarasov, V.E. *Fractional Dynamics: Applications of Fractional Calculus to Dynamics of Particles, Fields and Media*; Springer Science & Business Media: Berlin, Germany, 2011.
7. Klafter, J.; Lim, S.; Metzler, R. *Fractional Dynamics: Recent Advances*; World Scientific: Singapore, 2012.
8. Pramukkul, P.; Svenkeson, A.; Grigolini, P.; Bologna, M.; West, B. Complexity and the Fractional Calculus. *Adv. Math. Phys.* **2013**, *2013*, 498789. [CrossRef]
9. Chen, D.; Xue, D.; Chen, Y. More optimal image processing by fractional order differentiation and fractional order partial differential equations. In Proceedings of the International Symposium on Fractional PDEs, Newport, RI, USA, 3–5 June 2013.
10. Chen, D.; Sun, S.; Zhang, C.; Chen, Y.; Xue, D. Fractional-order TV-L 2 Model for Image Denoising. *Cent. Eur. J. Phys.* **2013**, *11*, 1414–1422. [CrossRef]
11. Yang, Q.; Chen, D.; Zhao, T.; Chen, Y. Fractional Calculus in Image Processing: A Review. *Fract. Calc. Appl. Anal.* **2016**, *19*, 1222–1249. [CrossRef]
12. Seshadri, V.; West, B.J. Fractal dimensionality of Lévy processes. *Proc. Natl. Acad. Sci. USA* **1982**, *79*, 4501. [CrossRef] [PubMed]
13. Metzler, R.; Klafter, J. The Random Walk's Guide to Anomalous Diffusion: A Fractional Dynamics Approach. *Phys. Rep.* **2000**, *339*, 1–77. [CrossRef]
14. Metzler, R.; Glöckle, W.G.; Nonnenmacher, T.F. Fractional Model Equation for Anomalous Diffusion. *Phys. A Stat. Mech. Appl.* **1994**, *211*, 13–24. [CrossRef]
15. Sheng, H.; Chen, Y.; Qiu, T. *Fractional Processes and Fractional-Order Signal Processing: Techniques and Applications*; Springer Science & Business Media: Berlin, Germany, 2011.
16. Mandelbrot, B.B.; Wallis, J.R. Robustness of the Rescaled Range R/S in the Measurement of Noncyclic Long Run Statistical Dependence. *Water Resour. Res.* **1969**, *5*, 967–988. [CrossRef]
17. Geweke, J.; Porter-Hudak, S. The Estimation and Application of Long Memory Time Series Models. *J. Time Ser. Anal.* **1983**, *4*, 221–238. [CrossRef]
18. Liu, K.; Chen, Y.; Zhang, X. An Evaluation of ARFIMA (Autoregressive Fractional Integral Moving Average) Programs. *Axioms* **2017**, *6*, 16. [CrossRef]
19. Montroll, E.W.; Weiss, G.H. Random Walks on Lattices. II. *J. Math. Phys.* **1965**, *6*, 167–181. [CrossRef]
20. Liakos, K.G.; Busato, P.; Moshou, D.; Pearson, S.; Bochtis, D. Machine Learning in Agriculture: A Review. *Sensors* **2018**, *18*, 2674. [CrossRef] [PubMed]
21. Nesterov, Y. A Method for Unconstrained Convex Minimization Problem with the Rate of Convergence O $(1/k^2)$. *Doklady an Ussr* **1983**, *269*, 543–547.
22. Montroll, E.W.; West, B.J. On An Enriched Collection of Stochastic Processes. *Fluct. Phenom.* **1979**, *66*, 61.
23. Francis, B.A.; Wonham, W.M. The Internal Model Principle of Control Theory. *Automatica* **1976**, *12*, 457–465. [CrossRef]
24. Zadeh, L.A. Fuzzy Logic. *Computer* **1988**, *21*, 83–93. [CrossRef]
25. Unser, M.; Blu, T. Fractional Splines and Wavelets. *SIAM Rev.* **2000**, *42*, 43–67. [CrossRef]
26. Samoradnitsky, G. *Stable Non-Gaussian Random Processes: Stochastic Models with Infinite Variance*; Routledge: Oxford, UK, 2017.
27. Crovella, M.E.; Bestavros, A. Self-similarity in World Wide Web Traffic: Evidence and Possible Causes. *IEEE/ACM Trans. Netw.* **1997**, *5*, 835–846. [CrossRef]
28. Burnecki, K.; Weron, A. Levy Stable Processes. From Stationary to Self-similar Dynamics and Back. An Application to Finance. *Acta Phys. Pol. Ser. B* **2004**, *35*, 1343–1358.
29. Pesquet-Popescu, B.; Pesquet, J.C. Synthesis of Bidimensional α-stable Models with Long-range Dependence. *Signal Process.* **2002**, *82*, 1927–1940. [CrossRef]
30. Hartley, T.T.; Lorenzo, C.F. Fractional-order System Identification Based on Continuous Order-distributions. *Signal Process.* **2003**, *83*, 2287–2300. [CrossRef]
31. Wolpert, R.L.; Taqqu, M.S. Fractional Ornstein–Uhlenbeck Lévy Processes and the Telecom Process: Upstairs and Downstairs. *Signal Process.* **2005**, *85*, 1523–1545. [CrossRef]
32. Bahg, G.; Evans, D.G.; Galdo, M.; Turner, B.M. Gaussian process linking functions for mind, brain, and behavior. *Proc. Natl. Acad. Sci. USA* **2020**, *117*, 29398–29406. [CrossRef]
33. West, B.J.; Geneston, E.L.; Grigolini, P. Maximizing Information Exchange between Complex Networks. *Phys. Rep.* **2008**, *468*, 1–99. [CrossRef]
34. West, B.J. Sir Isaac Newton Stranger in a Strange Land. *Entropy* **2020**, *22*, 1204. [CrossRef] [PubMed]
35. Csete, M.; Doyle, J. Bow Ties, Metabolism and Disease. *Trends Biotechnol.* **2004**, *22*, 446–450. [CrossRef]
36. Zhao, J.; Yu, H.; Luo, J.H.; Cao, Z.W.; Li, Y.X. Hierarchical Modularity of Nested Bow-ties in Metabolic Networks. *BMC Bioinform.* **2006**, *7*, 1–16. [CrossRef] [PubMed]
37. Doyle, J. Universal Laws and Architectures.Available online: http://www.ieeecss-oll.org/lecture/universal-laws-and-architectures. (accessed on 2 February 2021).
38. Doyle, J.C.; Csete, M. Architecture, Constraints, and Behavior. *Proc. Natl. Acad. Sci. USA* **2011**, *108*, 15624–15630. [CrossRef]
39. Sheng, H.; Chen, Y.Q.; Qiu, T. Heavy-tailed Distribution and Local Long Memory in Time Series of Molecular Motion on the Cell Membrane. *Fluct. Noise Lett.* **2011**, *10*, 93–119. [CrossRef]
40. Graves, T.; Gramacy, R.; Watkins, N.; Franzke, C. A Brief History of Long Memory: Hurst, Mandelbrot and the Road to ARFIMA, 1951–1980. *Entropy* **2017**, *19*, 437. [CrossRef]

41. West, B.J.; Grigolini, P. *Complex Webs: Anticipating the Improbable*; Cambridge University Press: Cambridge, UK, 2010.
42. Barabási, A.L.; Albert, R. Emergence of Scaling in Random Networks. *Science* **1999**, *286*, 509–512. [CrossRef]
43. Sun, W.; Li, Y.; Li, C.; Chen, Y. Convergence Speed of a Fractional Order Consensus Algorithm over Undirected Scale-free Networks. *Asian J. Control* **2011**, *13*, 936–946. [CrossRef]
44. Li, M. Modeling Autocorrelation Functions of Long-range Dependent Teletraffic Series Based on Optimal Approximation in Hilbert Space—A Further Study. *Appl. Math. Model.* **2007**, *31*, 625–631. [CrossRef]
45. Zhao, Z.; Guo, Q.; Li, C. A Fractional Model for the Allometric Scaling Laws. *Open Appl. Math. J.* **2008**, *2*, 26–30. [CrossRef]
46. Sun, H.; Chen, Y.; Chen, W. Random-order Fractional Differential Equation Models. *Signal Process.* **2011**, *91*, 525–530. [CrossRef]
47. Kello, C.T.; Brown, G.D.; Ferrer-i Cancho, R.; Holden, J.G.; Linkenkaer-Hansen, K.; Rhodes, T.; Van Orden, G.C. Scaling Laws in Cognitive Sciences. *Trends Cogn. Sci.* **2010**, *14*, 223–232. [CrossRef] [PubMed]
48. Gorenflo, R.; Mainardi, F. Fractional Calculus and Stable Probability Distributions. *Arch. Mech.* **1998**, *50*, 377–388.
49. Mainardi, F. The Fundamental Solutions for the Fractional Diffusion-wave Equation. *Appl. Math. Lett.* **1996**, *9*, 23–28. [CrossRef]
50. Luchko, Y.; Mainardi, F.; Povstenko, Y. Propagation Speed of the Maximum of the Fundamental Solution to the Fractional Diffusion–wave Equation. *Comput. Math. Appl.* **2013**, *66*, 774–784. [CrossRef]
51. Luchko, Y.; Mainardi, F. Some Properties of the Fundamental Solution to the Signalling Problem for the Fractional Diffusion-wave Equation. *Open Phys.* **2013**, *11*, 666–675. [CrossRef]
52. Luchko, Y.; Mainardi, F. Cauchy and Signaling Problems for the Time-fractional Diffusion-wave Equation. *J. Vib. Acoust.* **2014**, *136*, 050904. [CrossRef]
53. Li, Z.; Liu, L.; Dehghan, S.; Chen, Y.; Xue, D. A Review and Evaluation of Numerical Tools for Fractional Calculus and Fractional Order Controls. *Int. J. Control* **2017**, *90*, 1165–1181. [CrossRef]
54. Asmussen, S. Steady-state Properties of GI/G/1. In *Applied Probability and Queues*; Springer: New York, NY, USA, 2003; pp. 266–301.
55. Bernardi, M.; Petrella, L. Interconnected Risk Contributions: A Heavy-tail Approach to Analyze US Financial Sectors. *J. Risk Financ. Manag.* **2015**, *8*, 198–226. [CrossRef]
56. Ahn, S.; Kim, J.H.; Ramaswami, V. A New Class of Models for Heavy Tailed Distributions in Finance and Insurance Risk. *Insur. Math. Econ.* **2012**, *51*, 43–52. [CrossRef]
57. Resnick, S.I. *Heavy-tail Phenomena: Probabilistic and Statistical Modeling*; Springer Science & Business Media: Berlin, Germany, 2007.
58. Rolski, T.; Schmidli, H.; Schmidt, V.; Teugels, J.L. *Stochastic Processes for Insurance and Finance*; John Wiley & Sons: Hoboken, NJ, USA, 2009; Volume 505.
59. Foss, S.; Korshunov, D.; Zachary, S. *An Introduction to Heavy-Tailed and Subexponential Distributions*; Springer: Berlin, Germany, 2011; Volume 6.
60. Niu, H.; Chen, Y.; Chen, Y. Fractional-order extreme learning machine with Mittag-Leffler distribution. In Proceedings of ASME 2019 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, Anaheim, CA, USA, 18–21 August 2019.
61. Hariya, Y.; Kurihara, T.; Shindo, T.; Jin'no, K. Lévy flight PSO. In Proceedings of the IEEE Congress on Evolutionary Computation (CEC), Sendai, Japan, 25–28 May 2015.
62. Yang, X.S. *Nature-Inspired Metaheuristic Algorithms*; Luniver Press: London, UK, 2010.
63. Yang, X.S.; Deb, S. Engineering Optimisation by Cuckoo Search. *Int. J. Math. Model. Numer. Optim.* **2010**, *1*, 330–343. [CrossRef]
64. Haubold, H.J.; Mathai, A.M.; Saxena, R.K. Mittag-Leffler Functions and Their Applications. *J. Appl. Math.* **2011**, *2011*, 298628. [CrossRef]
65. Jayakumar, K. Mittag-Leffler Process. *Math. Comput. Model.* **2003**, *37*, 1427–1434. [CrossRef]
66. Wei, J.; Chen, Y.; Yu, Y.; Chen, Y. Improving cuckoo search algorithm with Mittag-Leffler distribution. In Proceedings of the ASME 2019 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, Anaheim, CA, USA, 18–21 August 2019.
67. Rinne, H. *The Weibull Distribution: A Handbook*; CRC Press: Boca Raton, FL, USA, 2008.
68. Johnson, N.L.; Kotz, S.; Balakrishnan, N. *Continuous Univariate Distributions*; John Wiley & Sons, Ltd.: Hoboken, NJ, USA, 1995.
69. Feller, W. *An Introduction to Probability Theory and Its Application Vol II*; John Wiley & Sons: Hoboken, NJ, USA, 1971.
70. Liu, T.; Zhang, P.; Dai, W.S.; Xie, M. An Intermediate Distribution Between Gaussian and Cauchy Distributions. *Phys. A Stat. Mech. Appl.* **2012**, *391*, 5411–5421. [CrossRef]
71. Bahat, D.; Rabinovitch, A.; Frid, V. *Tensile Fracturing in Rocks*; Springer: Berlin, Germany, 2005.
72. Geerolf, F. A Theory of Pareto Distributions. Available online: https://fgeerolf.com/geerolf-pareto.pdf (accessed on 2 February 2021).
73. Mandelbrot, B. The Pareto-Levy Law and the Distribution of Income. *Int. Econ. Rev.* **1960**, *1*, 79–106. [CrossRef]
74. Levy, M.; Solomon, S. New Evidence for the Power-law Distribution of Wealth. *Phys. A Stat. Mech. Appl.* **1997**, *242*, 90–94. [CrossRef]
75. Lu, J.; Ding, J. Mixed-Distribution-Based Robust Stochastic Configuration Networks for Prediction Interval Construction. *IEEE Trans. Ind. Inform.* **2019**, *16*, 5099–5109. [CrossRef]
76. Spiegel, M.R.; Schiller, J.J.; Srinivasan, R. *Probability and Statistics*; McGraw-Hill: New York, NY, USA, 2013.

77. Embrechts, P.; Klüppelberg, C.; Mikosch, T. *Modelling Extremal Events: For Insurance and Finance*; Springer Science & Business Media: Berlin, Germany, 2013; Volume 33.
78. Novak, S.Y. *Extreme Value Methods with Applications to Finance*; CRC Press: Boca Raton, FL, USA, 2011.
79. De Haan, L.; Ferreira, A. *Extreme Value Theory: An Introduction*; Springer Science & Business Media: Berlin, Germany, 2007.
80. Bottou, L.; Bousquet, O. The Tradeoffs of Large Scale Learning. *Adv. Neural Inf. Process. Syst.* **2007**, *20*, 161–168.
81. Bottou, L. Large-scale machine learning with stochastic gradient descent. In *Proceedings of the COMPSTAT*; Springer: Berlin, Germany, 2010; pp. 177–186.
82. Simsekli, U.; Sagun, L.; Gurbuzbalaban, M. A tail-index analysis of stochastic gradient noise in deep neural networks. *arXiv* **2019**, arXiv:1901.06053.
83. Yanovsky, V.; Chechkin, A.; Schertzer, D.; Tur, A. Lévy Anomalous Diffusion and Fractional Fokker–Planck Equation. *Phys. A Stat. Mech. Appl.* **2000**, *282*, 13–34. [CrossRef]
84. Viswanathan, G.M.; Afanasyev, V.; Buldyrev, S.; Murphy, E.; Prince, P.; Stanley, H.E. Lévy Flight Search Patterns of Wandering Albatrosses. *Nature* **1996**, *381*, 413–415. [CrossRef]
85. Hilbert, M.; López, P. The World's Technological Capacity to Store, Communicate, and Compute Information. *Science* **2011**, *332*, 60–65. [CrossRef] [PubMed]
86. Ward, J.S.; Barker, A. Undefined by data: A survey of big data definitions. *arXiv* **2013**, arXiv:1309.5821.
87. Reinsel, D.; Gantz, J.; Rydning, J. Data Age 2025: The Evolution of Data to Life-critical. *Don't Focus Big Data* **2017**, *2*, 2–24.
88. Firican, G. The 10 Vs of Big Data. 2017. Available online: https://tdwi.org/articles/2017/02/08/10-vs-of-big-data.aspx (accessed on 2 February 2021).
89. Nakahira, Y.; Liu, Q.; Sejnowski, T.J.; Doyle, J.C. Diversity-enabled sweet spots in layered architectures and speed-accuracy trade-offs in sensorimotor control. *arXiv* **2019**, arXiv:1909.08601.
90. Arabas, J.; Opara, K. Population Diversity of Non-elitist Evolutionary Algorithms in the Exploration Phase. *IEEE Trans. Evol. Comput.* **2019**, *24*, 1050–1062. [CrossRef]
91. Ko, M.; Stark, B.; Barbadillo, M.; Chen, Y. An Evaluation of Three Approaches Using Hurst Estimation to Differentiate Between Normal and Abnormal HRV. In Proceedings of the ASME 2015 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, Boston, MA, USA, 2–5 August 2015.
92. Li, N.; Cruz, J.; Chien, C.S.; Sojoudi, S.; Recht, B.; Stone, D.; Csete, M.; Bahmiller, D.; Doyle, J.C. Robust Efficiency and Actuator Saturation Explain Healthy Heart Rate Control and Variability. *Proc. Natl. Acad. Sci. USA* **2014**, *111*, E3476–E3485. [CrossRef] [PubMed]
93. Hutton, E.L. *Xunzi: The Complete Text*; Princeton University Press: Princeton, NJ, USA, 2014.
94. Boyer, C.B. *The History of the Calculus and Its Conceptual Development: (The Concepts of the Calculus)*; Courier Corporation: Chelmsford, MA, USA, 1959.
95. Bardi, J.S. *The Calculus Wars: Newton, Leibniz, and the Greatest Mathematical Clash of All Time*; Hachette UK: Paris, France, 2009.
96. Tanner, R.I.; Walters, K. *Rheology: An Historical Perspective*; Elsevier: Amsterdam, The Netherlands, 1998.
97. Chen, Y.; Sun, R.; Zhou, A. An Improved Hurst Parameter Estimator Based on Fractional Fourier Transform. *Telecommun. Syst.* **2010**, *43*, 197–206. [CrossRef]
98. Sheng, H.; Sun, H.; Chen, Y.; Qiu, T. Synthesis of Multifractional Gaussian Noises Based on Variable-order Fractional Operators. *Signal Process.* **2011**, *91*, 1645–1650. [CrossRef]
99. Sun, R.; Chen, Y.; Zaveri, N.; Zhou, A. Local analysis of long range dependence based on fractional Fourier transform. In Proceedings of the IEEE Mountain Workshop on Adaptive and Learning Systems, Logan, UT, USA, 24–26 July 2006; pp. 13–18.
100. Pipiras, V.; Taqqu, M.S. *Long-Range Dependence and Self-Similarity*; Cambridge University Press: Cambridge, UK, 2017; Volume 45.
101. Samorodnitsky, G. Long Range Dependence. Available online: https://onlinelibrary.wiley.com/doi/abs/10.1002/9781118445112.stat04569 (accessed on 2 February 2021).
102. Gubner, J.A. *Probability and Random Processes for Electrical and Computer Engineers*; Cambridge University Press: Cambridge, UK, 2006.
103. Clegg, R.G. A practical guide to measuring the Hurst parameter. *arXiv* **2006**, arXiv:math/0610756.
104. Decreusefond, L. Stochastic Analysis of the Fractional Brownian Motion. *Potential Anal.* **1999**, *10*, 177–214. [CrossRef]
105. Koutsoyiannis, D. The Hurst Phenomenon and Fractional Gaussian Noise Made Easy. *Hydrol. Sci. J.* **2002**, *47*, 573–595. [CrossRef]
106. Mandelbrot, B.B.; Van Ness, J.W. Fractional Brownian Motions, Fractional Noises and Applications. *SIAM Rev.* **1968**, *10*, 422–437. [CrossRef]
107. Ortigueira, M.D.; Batista, A.G. On the Relation between the Fractional Brownian Motion and the Fractional Derivatives. *Phys. Lett. A* **2008**, *372*, 958–968. [CrossRef]
108. Chen, Y.; Sun, R.; Zhou, A. An overview of fractional order signal processing (FOSP) techniques. In Proceedings of the International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, Las Vegas, NV, USA, 4–7 September 2007.
109. Liu, K.; Domański, P.D.; Chen, Y. Control performance assessment with fractional lower order moments. In Proceedings of the 2020 7th International Conference on Control, Decision and Information Technologies (CoDIT), Prague, Czech Republic, 29 June–2 July 2020.

110. Cottone, G.; Di Paola, M. On the Use of Fractional Calculus for the Probabilistic Characterization of Random Variables. *Probabilistic Eng. Mech.* **2009**, *24*, 321–330. [CrossRef]
111. Cottone, G.; Di Paola, M.; Metzler, R. Fractional Calculus Approach to the Statistical Characterization of Random Variables and Vectors. *Phys. A Stat. Mech. Appl.* **2010**, *389*, 909–920. [CrossRef]
112. Ma, X.; Nikias, C.L. Joint Estimation of Time Delay and Frequency Delay in Impulsive Noise Using Fractional Lower Order Statistics. *IEEE Trans. Signal Process.* **1996**, *44*, 2669–2687.
113. RongHua, F. Modeling and Application of Theory Based on Time Series ARMA. *Sci. Technol. Inf.* **2012**, *2012*, 153.
114. Shalalfeh, L.; Bogdan, P.; Jonckheere, E. Fractional Dynamics of PMU Data. *IEEE Trans. Smart Grid* **2020**. [CrossRef]
115. Harmantzis, F. Heavy network traffic modeling and simulation using stable FARIMA processes. In Proceedings of the 19th International Teletraffic Congress (ITC19), Beijing, China, 29 August–2 September 2005.
116. Sheng, H.; Chen, Y. FARIMA with Stable Innovations Model of Great Salt Lake Elevation Time Series. *Signal Process.* **2011**, *91*, 553–561. [CrossRef]
117. Li, Q.; Tricaud, C.; Sun, R.; Chen, Y. Great Salt Lake surface level forecasting using FIGARCH model. In Proceedings of the International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, Las Vegas, NV, USA, 4–7 September 2007; pp. 1361–1370.
118. Brockwell, P.J.; Davis, R.A.; Fienberg, S.E. *Time Series: Theory and Methods*; Springer Science & Business Media: Berlin, Germany, 1991.
119. Boutahar, M.; Dufrénot, G.; Péguin-Feissolle, A. A Simple Fractionally Integrated Model with a Time-varying Long Memory Parameter d_t. *Comput. Econ.* **2008**, *31*, 225–241. [CrossRef]
120. Gray, H.L.; Zhang, N.F.; Woodward, W.A. On Generalized Fractional Processes. *J. Time Ser. Anal.* **1989**, *10*, 233–257. [CrossRef]
121. Woodward, W.A.; Cheng, Q.C.; Gray, H.L. A k-factor GARMA Long-memory Model. *J. Time Ser. Anal.* **1998**, *19*, 485–504. [CrossRef]
122. West, B.J. *Fractional Calculus View of Complexity: Tomorrow's Science*; CRC Press: Boca Raton, FL, USA, 2016.
123. Zaslavsky, G.M.; Sagdeev, R.; Usikov, D.; Chernikov, A. *Weak Chaos and Quasi-Regular Patterns*; Cambridge University Press: Cambridge, UK, 1992.
124. Hilfer, R.; Anton, L. Fractional Master Equations and Fractal Time Random Walks. *Phys. Rev. E* **1995**, *51*, R848. [CrossRef]
125. Gorenflo, R.; Mainardi, F.; Vivoli, A. Continuous-time Random Walk and Parametric Subordination in Fractional Diffusion. *Chaos Solitons Fractals* **2007**, *34*, 87–103. [CrossRef]
126. Niu, H.; Hollenbeck, D.; Zhao, T.; Wang, D.; Chen, Y. Evapotranspiration Estimation with Small UAVs in Precision Agriculture. *Sensors* **2020**, *20*, 6427. [CrossRef]
127. Díaz-Varela, R.; de la Rosa, R.; León, L.; Zarco-Tejada, P. High-resolution Airborne UAV Imagery to Assess Olive Tree Crown Parameters Using 3D Photo Reconstruction: Application in Breeding Trials. *Remote Sens.* **2015**, *7*, 4213–4232. [CrossRef]
128. Gonzalez-Dugo, V.; Goldhamer, D.; Zarco-Tejada, P.J.; Fereres, E. Improving the Precision of Irrigation in a Pistachio Farm Using an Unmanned Airborne Thermal System. *Irrig. Sci.* **2015**, *33*, 43–52. [CrossRef]
129. Swain, K.C.; Thomson, S.J.; Jayasuriya, H.P. Adoption of an Unmanned Helicopter for Low-altitude Remote Sensing to Estimate Yield and Total Biomass of a Rice Crop. *Trans. ASABE* **2010**, *53*, 21–27. [CrossRef]
130. Zarco-Tejada, P.J.; González-Dugo, V.; Williams, L.; Suárez, L.; Berni, J.A.; Goldhamer, D.; Fereres, E. A PRI-based Water Stress Index Combining Structural and Chlorophyll Effects: Assessment Using Diurnal Narrow-band Airborne Imagery and the CWSI Thermal Index. *Remote Sens. Environ.* **2013**, *138*, 38–50. [CrossRef]
131. Niu, H.; Zhao, T.; Wang, D.; Chen, Y. Estimating evapotranspiration with UAVs in agriculture: A review. In Proceedings of the ASABE Annual International Meeting, Boston, MA, USA, 7–10 July 2019.
132. Niu, H.; Zhao, T.; Wang, D.; Chen, Y. A UAV resolution and waveband aware path planning for onion irrigation treatments inference. In Proceedings of the 2019 International Conference on Unmanned Aircraft Systems (ICUAS), Atlanta, GA, USA, 11–14 June 2019; pp. 808–812.
133. Niu, H.; Wang, D.; Chen, Y. Estimating crop coefficients using linear and deep stochastic configuration networks models and UAV-based Normalized Difference Vegetation Index (NDVI). In Proceedings of the International Conference on Unmanned Aircraft Systems (ICUAS), Athens, Greece, 1–4 September 2020.
134. Niu, H.; Wang, D.; Chen, Y. Estimating actual crop evapotranspiration using deep stochastic configuration networks model and UAV-based crop coefficients in a pomegranate orchard. In Proceedings of the Autonomous Air and Ground Sensing Systems for Agricultural Optimization and Phenotyping V. International Society for Optics and Photonics, 27 April–8 May 2020, held online.
135. Che, Y.; Wang, Q.; Xie, Z.; Zhou, L.; Li, S.; Hui, F.; Wang, X.; Li, B.; Ma, Y. Estimation of Maize Plant Height and Leaf Area Index Dynamic Using Unmanned Aerial Vehicle with Oblique and Nadir Photography. *Ann. Bot.* **2020**, *126*, 765–773. [CrossRef]
136. Deng, R.; Jiang, Y.; Tao, M.; Huang, X.; Bangura, K.; Liu, C.; Lin, J.; Qi, L. Deep Learning-based Automatic Detection of Productive Tillers in Rice. *Comput. Electron. Agric.* **2020**, *177*, 105703. [CrossRef]
137. Zhao, T.; Chen, Y.; Ray, A.; Doll, D. Quantifying almond water stress using unmanned aerial vehicles (UAVs): Correlation of stem water potential and higher order moments of non-normalized canopy distribution. In Proceedings of the ASME 2017 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, Cleveland, OH, USA, 6–9 August 2017.

138. Zhao, T.; Niu, H.; de la Rosa, E.; Doll, D.; Wang, D.; Chen, Y. Tree canopy differentiation using instance-aware semantic segmentation. In Proceedings of the 2018 ASABE Annual International Meeting, Detroit, MI, USA, 29 July–1 August 2018.
139. Géron, A. *Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*; O'Reilly Media: Sevastopol, CA, USA, 2019.
140. Mitchell, T.M. *Machine Learning*; McGraw-Hill: New York, NY, USA, 1997.
141. Polyak, B.T. Some Methods of Speeding up the Convergence of Iteration Methods. *USSR Comput. Math. Math. Phys.* **1964**, *4*, 1–17. [CrossRef]
142. Duchi, J.; Hazan, E.; Singer, Y. Adaptive Subgradient Methods for Online Learning and Stochastic Optimization. *J. Mach. Learn. Res.* **2011**, *12*, 2121–2159.
143. Hinton, G.; Tieleman, T. Slide 29 in Lecture 6. 2012. Available online: http://www.cs.toronto.edu/~tijmen/csc321/slides/lecture_slides_lec6.pdf (accessed on 2 February 2021).
144. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980.
145. Zeng, C.; Chen, Y. Optimal Random Search, Fractional Dynamics and Fractional Calculus. *Fract. Calc. Appl. Anal.* **2014**, *17*, 321–332. [CrossRef]
146. Wei, J.; Yu, Y.; Wang, S. Parameter Estimation for Noisy Chaotic Systems Based on an Improved Particle Swarm Optimization Algorithm. *J. Appl. Anal. Comput.* **2015**, *5*, 232–242.
147. Wei, J.; Yu, Y.; Cai, D. Identification of Uncertain Incommensurate Fractional-order Chaotic Systems Using an Improved Quantum-behaved Particle Swarm Optimization Algorithm. *J. Comput. Nonlinear Dyn.* **2018**, *13*, 051004. [CrossRef]
148. Wei, J.; Chen, Y.; Yu, Y.; Chen, Y. Optimal Randomness in Swarm-based Search. *Mathematics* **2019**, *7*, 828. [CrossRef]
149. Wei, J.; Yu, Y. A Novel Cuckoo Search Algorithm under Adaptive Parameter Control for Global Numerical Optimization. *Soft Comput.* **2019**, *24*, 4917–4940. [CrossRef]
150. Wei, J.; Yu, Y. An adaptive cuckoo search algorithm with optional external archive for global numerical optimization. In Proceedings of the International Conference on Fractional Differentiation and its Applications (ICFDA), Amman, Jordan, 16–18 July 2018.
151. Wilson, A.C.; Recht, B.; Jordan, M.I. A Lyapunov analysis of momentum methods in optimization. *arXiv* **2016**, arXiv:1611.02635.
152. Feynman, R.P. The Principle of Least Action in Quantum Mechanics. In *Feynman's Thesis—A New Approach to Quantum Theory*; World Scientific: Singapore, 2005; pp. 1–69.
153. Hamilton, S.W.R. *On a General Method in Dynamics*; Richard Taylor, 1834. Available online: http://www.kurims.kyoto-u.ac.jp/EMIS/classics/Hamilton/GenMeth.pdf (accessed on 2 February 2021).
154. Hawking, S.W. The Path-integral Approach to Quantum Gravity. In *General Relativity*; World Scientific: Singapore, 1979.
155. Kerrigan, E. What the Machine Should Learn about Models for Control. 2020. Available online: https://www.ifac2020.org/program/workshops/machine-learning-meets-model-based-control (accessed on 2 February 2021).
156. Vinnicombe, G. *Uncertainty and Feedback: H_∞ Loop-Shaping and the ν-Gap Metric*; World Scientific: Singapore, 2001.
157. Viola, J.; Chen, Y.; Wang, J. Information-based model discrimination for digital twin behavioral matching. In Proceedings of the International Conference on Industrial Artificial Intelligence (IAI), Shenyang, China, 23–25 October 2020; pp. 1–6.
158. Kashima, K.; Yamamoto, Y. System Theory for Numerical Analysis. *Automatica* **2007**, *43*, 1156–1164. [CrossRef]
159. An, W.; Wang, H.; Sun, Q.; Xu, J.; Dai, Q.; Zhang, L. A PID controller approach for stochastic optimization of deep networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 8522–8531.
160. Fan, Y.; Koellermeier, J. Accelerating the Convergence of the Moment Method for the Boltzmann Equation Using Filters. *J. Sci. Comput.* **2020**, *84*, 1–28. [CrossRef]
161. Kuhlman, K.L. Review of Inverse Laplace Transform Algorithms for Laplace-space Numerical Approaches. *Numer. Algorithms* **2013**, *63*, 339–355. [CrossRef]
162. Xue, D.; Chen, Y. *Solving Applied Mathematical Problems with MATLAB*; CRC Press: Boca Raton, FL, USA, 2009.
163. Wang, D.; Li, M. Stochastic Configuration Networks: Fundamentals and Algorithms. *IEEE Trans. Cybern.* **2017**, *47*, 3466–3479. [CrossRef] [PubMed]
164. Bebis, G.; Georgiopoulos, M. Feed-forward Neural Networks. *IEEE Potentials* **1994**, *13*, 27–31. [CrossRef]
165. Broomhead, D.; Lowe, D. Multivariable Functional Interpolation and Adaptive Networks. *Complex Syst.* **1988**, *2*, 321–355.
166. Pao, Y.H.; Takefuji, Y. Functional-link Net Computing: Theory, System Architecture, and Functionalities. *Computer* **1992**, *25*, 76–79. [CrossRef]
167. Wang, D.; Li, M. Deep stochastic configuration networks with universal approximation property. In Proceedings of the International Joint Conference on Neural Networks (IJCNN), Rio de Janeiro, Brazil, 8–13 July 2018; pp. 1–8.
168. Li, M.; Wang, D. 2-D Stochastic Configuration Networks for Image Data Analytics. *IEEE Trans. Cybern.* **2019**, 51, 359–372. [CrossRef] [PubMed]
169. Huang, C.; Huang, Q.; Wang, D. Stochastic Configuration Networks Based Adaptive Storage Replica Management for Power Big Data Processing. *IEEE Trans. Ind. Inf.* **2019**, *16*, 373–383. [CrossRef]
170. Scardapane, S.; Wang, D. Randomness in Neural Networks: An Overview. *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.* **2017**, *7*, e1200. [CrossRef]

171. Wei, J. Research on Swarm Intelligence Optimization Algorithms and Their Applications to Parameter Identification of Fractional-Order Systems. Ph.D. Thesis, Beijing Jiaotong University, Beijing, China, 2020.
172. Chen, Y. Fundamental Principles for Fractional Order Gradient Methods. Ph.D. Thesis, University of Science and Technology of China, Hefei, China, 2020.
173. Tyukin, I.Y.; Prokhorov, D.V. Feasibility of random basis function approximators for modeling and control. In Proceedings of the IEEE Control Applications, (CCA) & Intelligent Control, (ISIC), St. Petersburg, Russia, 8–10 July 2009.
174. Nagaraj, S. Optimization and learning with nonlocal calculus. *arXiv* **2020**, arXiv:2012.07013.
175. Tarasov, V.E. Fractional Vector Calculus and Fractional Maxwell's Equations. *Ann. Phys.* **2008**, *323*, 2756–2778. [CrossRef]
176. Ortigueira, M.; Machado, J. On Fractional Vectorial Calculus. *Bull. Pol. Acad. Sci. Tech. Sci.* **2018**, *66*, 389–402.
177. Feliu-Faba, J.; Fan, Y.; Ying, L. Meta-learning Pseudo-differential Operators with Deep Neural Networks. *J. Comput. Phys.* **2020**, *408*, 109309. [CrossRef]
178. Hall, D.L. *Dao De Jing: A Philosophical Translation*; Ballantine Books: New York, NY, USA, 2010.