Algorithms in Future Capital Markets

Adriano Koshiyama, Nick Firoozye and Philip Treleaven
Department of Computer Science
University College London
Gower Street, London WC1E 6BT

Abstract
This paper reviews Artificial Intelligence (AI), Machine Learning (ML) and associated algorithms in future Capital Markets. New AI algorithms are constantly emerging, with each 'strain' mimicking a new form of human learning, reasoning, knowledge, and decision-making. The current main disrupting forms of learning include Deep Learning, Adversarial Learning, Transfer and Meta Learning. Albeit these modes of learning have been in the AI/ML field more than a decade, they now are more applicable due to the availability of data, computing power and infrastructure. These forms of learning have produced new models (e.g., Long Short-Term Memory, Generative Adversarial Networks) and leverage important applications (e.g., Natural Language Processing, Adversarial Examples, Deep Fakes, etc.). These new models and applications will drive changes in future Capital Markets, so it is important to understand their computational strengths and weaknesses.

Since ML algorithms effectively self-program and evolve dynamically, financial institutions and regulators are becoming increasingly concerned with ensuring there remains a modicum of human control, focusing on Algorithmic *Interpretability/Explainability, Robustness* and *Legality*. For example, the concern is that, in the future, an ecology of trading algorithms across different institutions may 'conspire' and become unintentionally *fraudulent* (cf. LIBOR) or subject to subversion through compromised datasets (e.g. Microsoft Tay). New and unique forms of systemic risks can emerge, potentially coming from excessive algorithmic complexity.

The contribution of this paper is to review AI, ML and associated algorithms, their computational strengths and weaknesses, and discuss their future impact on the Capital Markets.

1. Introduction

Machine learning has offered great promise for applications in market-making and automated trading in capital markets but has enjoyed uneven success across the different asset classes. It is clear that in the data-rich and relatively idiosyncratic exchange traded classes like equities and futures, all areas of Computational Statistics, AI and ML have been well-applied. In the broader and more homogenous macroeconomic-driven classes such as fixed income and forex, the applications have been far less deep in AI and ML, with Computational Statistics approaches more likely to have been successful. Finally, for central banks and regulators, the use of Complex Systems (agent-based financial networks) have been an important tool to analyse and evaluate potential systemic risks across financial institutions.

To some extent the methods are driven by the increasing availability of 'Big' data. The abundance of diverse datasets in equities has led to an explosion of interest and an emphasis on data-rich methods in this area, while the relative sparsity in fixed income has led to less development as a whole but as well, more emphasis on novel approaches to dealing with limited data. While we cover the gamut of applications from Computational Statistics and Complex Systems to ML and AI in financial markets, we note that different asset classes have spurred interest in somewhat different methods, from the more flexible data-rich methods to the more robust and explainable.

The latest trend is algorithms that continuously evolve 'virus-like' mutating to their Capital Markets' ecosystem (ref). Influential algorithms include *Long-Short Term Memory* (LSTMs) – a type of deep recurrent neural network capable of learning arbitrary long-term dependencies; *Generative Adversarial Networks (*GANs*)* – an architecture comprised of two networks, pitting one against the other (thus the 'adversarial'); and *Transfer* and *Meta learning families* – paradigms to reuse experience gained by solving predecessor problems as well as fine-tuning them for unseen tasks.

Naturally, each algorithm has strengths and weaknesses for specific financial applications, leading to combinations of algorithms in a practical system. For example, future self-optimising trading systems might use GANs to deal with the data scarcity issue, LSTMs for trading, and Transfer Learning to provide a macro coordination and knowledge sharing across trading systems and markets. Other applications fuelled by these models include natural language understanding and sentiment analysis, risk management, portfolio optimisation, algorithmic trading, fraud detection, compliance and regulation.

We envisage not an incremental upgrade, but a Cambrian explosion of new use-cases that will reshape current Capital Markets. The supply forces driving these changes can be named: a) accessibility to vast amounts of (Alternative) data; b) availability of "unlimited' (Cloud) computing infrastructure; c) technology maturity and open-access to the state-of-art in AI/ML algorithm libraries. The developments have led to a scramble for talent across the Investment Banking world, with Data Scientists poached from Tech and retail companies; and AI Labs are being set up inside banks.

In the next sections we discuss the main driving forces of these changes, the current debate and what is coming to the future Capital Markets, starting with a taxonomy of algorithms.

2. AI, ML and associated Algorithms

For completeness, this section unpacks algorithms across three domains: Computational Statistics (e.g. Monte Carlo methods), AI and ML (e.g. Artificial Neural Networks), and Complex Systems (e.g. Agent-Based systems). See Figure 1a. While there may be some debate over the terminology, we find the classification helpful to distinguish between relatively well-established methods and more cutting-edge technologies.

- **Computational Statistics** - computationally intensive statistical methods.
- **AI Algorithms** - mimicking a new form of human learning, reasoning, knowledge, and decision-making
  - Knowledge or rule-based systems
  - Evolutionary algorithms
  - Machine learning
- **Complex Systems** - system featuring a large number of interacting components whose aggregate activity is nonlinear.

Figure 1a: Algorithm domains

Computational Statistics

Computational Statistics models refers to computationally intensive statistical methods including Resampling methods (e.g., Bootstrap and Cross-Validation), Monte Carlo methods, Kernel Density estimation and other Semi and Non-Parametric methods, and Generalized Additive Models (Efron and Hastie, 2016; Wood, 2017). Examples include: a) **Resampling methods** - a variety of methods for doing one of the following: i) estimating the precision of sample statistics using subsets of data (e.g. jack-knifing) or drawn randomly from a set of data points (e.g. bootstrapping); ii) exchanging labels on data points when performing significance tests (e.g. permutation tests); iii) validating models by using random subsets (e.g. repeated cross-validation); b) **Monte Carlo methods** - a broad class of computational algorithms that rely on repeated random sampling to approximate integrals, particularly used to compute expected values (e.g. options payoff) including those meant for inference and estimation (e.g., Bayesian estimation, simulated method of moments); c) **Kernel Density**

estimation - are a set of methods used to approximate multivariate density functions from a set of datapoints; it is largely applied to generate smooth functions, reduce outliers effects and improve joint density estimations, sampling, and to derive non-linear fits; and d) **Generalized Additive Models** – a large class of nonlinear models widely used for inference and predictive modelling (e.g. time series forecasting, curve-fitting, etc.). e) Regularisation Methods – Regularisation methods are increasingly used as an alternative to traditional hypothesis testing and criteria-based methods, for allowing better quality forecasts with a large number of features.

AI and Machine Learning

This AI continuum of epistemological models spans three main communities: a) **Knowledge-based** or heuristic algorithms (e.g. rule-based) - where knowledge is explicitly represented as ontologies or IF-THEN rules rather than implicitly via code (Giarratano and Riley, 1998); b) **Evolutionary** or metaheuristics algorithms - a family of algorithms for global optimization inspired by biological evolution, using population-based trial and error problem solvers with a metaheuristic or stochastic optimization character (e.g. Genetic Algorithms, Genetic Programming, etc.) (Poli et al., 2008; Brownlee, 2011); and c) **Machine Learning** algorithms - a type of AI program with the ability to learn without explicit programming, and can change when exposed to new data; mainly comprising *Supervised* (e.g. Support Vector Machines, Random Forest, etc.), *Unsupervised* (e.g. K-Means, Independent Component Analysis, etc.), and *Reinforcement Learning* (e.g. Q-Learning, Temporal Differences, Gradient Policy Search, etc.) (Hastie et al., 2009; Sutton and Barto, 2018). Russell and Norvig (2016) provide an in-depth view of different aspects of AI.

Complex Systems

Lastly, a complex system is any system featuring a large number of interacting components (e.g. agents, processes, etc.) whose aggregate activity is nonlinear (not derivable from the summations of the activity of individual components) and typically exhibit hierarchical self-organization under selective pressures (Taylor, 2014; Barabási, 2016). Examples include: a) **Cellular automata** - a collection of cells arranged in a grid, such that each cell changes state as a function of time according to a defined set of rules that includes the states of neighbouring cells; b) **Agent-based models** - a class of computational models for simulating the actions and interactions of autonomous agents (individual or collective entities such as organizations or groups) with a view to assessing their effects on the system as a whole; c) **Network-based models** - a complex network is a graph (network) with non-trivial topological features - features that do not occur in simple networks such as lattices or random graphs but often occur in graphs modelling of real systems; and d) **Multi-Agent systems** – this subarea focus on formulating cooperative-competitive policies to a multitude of agents with the aim to achieve a given goal; this topic has significant overlap with Reinforcement Learning and Agent-based models.

As an illustration of this landscape of algorithms and research, Figure 1 present a non-exhaustive list of references that links each class of algorithms to applications in different areas of Capital Markets. Reschenhofer et al. (2019) provide an evaluation of current research on stock return predictability, particularly focusing on Computational Statistics and more traditional technical indicators widely used in finance.

	NLP & Sentiment Analysis	**Risk management**	**Portfolio optimisation**	**Systematic trading**	**Fraud detection**	**Compliance/ regulation**
Computational Statistics	(Cambria et al., 2013)	(McNeil et al., 2005)	(Kolm et al., 2014)	(Acar and Satchell, 2002)	(Juszcak et al., 2008)	(Yang and Koshiyama, 2019)
Machine Learning	(Kolchyna et al., 2015)	(Aziz and Dowling, 2019)	(Heaton et al., 2017)	(De Prado, 2018)	(Adewumi and Akinyelu, 2017)	(Van Liebergen, 2017)
Complex Systems	(Batrinca and Treleaven, 2015)	(Caccioli et al., 2018)	Hüttner et al. (2018)	(Pozzi et al., 2013)	(Xu and Chen, 2005)	(May et al., 2008)

Figure 1: Landscape of Algorithms and Research.

3. Machine learning paradigms

The great computational strength of ML algorithms is their ability to 'learn' without explicit programming. Understanding computational 'learning' is likely to have a profound effect on future science, in both artificial and natural (biological) systems.

As illustrated by examples in Figure 2, the driving forces of new ML algorithms are broadly a combination of the classical *trio* of Supervised, Unsupervised and Reinforcement Learning, with the *disruptors*: Deep Learning, Adversarial Learning, Transfer and Meta Learning. This interaction constantly yields new models (e.g., Long Short-Term Memory, Generative Adversarial Networks) and applications (e.g., Natural Language Processing, Object Recognition, Forecasting etc.).

	Supervised	**Unsupervised**	**Reinforcement**
Deep Learning	Deep Convolutional Neural Networks, Deep Recurrent Neural Networks (LeCun et al., 1989; Hochreiter and Schmidhuber, 1997; Krizhevsky et al., 2012; Goodfellow et al., 2016)	Deep Autoencoders (Goodfellow et al., 2016), Deep clustering (Caron et al., 2018)	Deep Q-Learning, Trust Region Policy Optimization, Asynchronous Advantage Actor Critic (Arulkumaran et al., 2017)
Adversarial Learning	Adversarial Semi-supervised Learning (Miyato et al., 2016); Adversarial Robustness in Supervised Learning (Nicolae et al., 2018)	Adversarial Autoencoders (Makhzani et al., 2015); Adversarial Representation Learning (Chen et al., 2016), Generative Adversarial Networks (Goodfellow, 2014)	Adversarial Policies (Gleave et al., 2019); Robust Adversarial Reinforcement Learning (Pinto et al., 2017)
Transfer/Meta Learning	OPEN-GPT (Radford et al., 2019); BERT (Devlin et al., 2018); MedicalNet (Chen et al., 2019)	Bayesian Unsupervised One-Shot learning (Fei-Fei, 2003); Embeddings from Language Model (Siddhant et al., 2018)	Darla (Higgins et al., 2017); Deep Transfer Reinforcement Learning for Text Summarization (Keneshloo et al., 2019)

Figure 2: Algorithms emerging by interaction between different learning paradigms.

Supervised, Unsupervised, Reinforcement

ML firstly subdivides into:

- **Supervised learning**: Given a set of inputs/independent variables/predictors $\mathbf{x}$ and outputs/dependent variables/targets $\mathbf{y}$, the goal is to learn a function $f(\mathbf{x})$ that approximates $\mathbf{y}$. This is accomplished by supervising $f(\mathbf{x})$, that is, providing it with examples ($\mathbf{x}_1$, $\mathbf{y}_1$), …, ($\mathbf{x}_\mathrm{n}$, $\mathbf{y}_n$) and feedback whenever it makes mistakes or accurate predictions.
- **Unsupervised learning:** Given several objects/samples/transactions $\mathbf{x}_1, \dots, \mathbf{x}_\mathrm{n}$, the goal is to learn a hidden map $h(\mathbf{x})$ that can uncover a hidden structure in the data. This hidden map can be used to 'compress' $\mathbf{x}$ (aka dimensionality reduction) or to assign to every $\mathbf{x}_\mathrm{i}$ a group c_k (aka clustering or topic modelling).
- **Reinforcement learning**: Given an environment formed by several states $\boldsymbol{s}_1, \boldsymbol{s}_2, \dots, \boldsymbol{s}_n$, an agent, and a reward function, the goal is to learn a policy π that will guide an agent actions $\boldsymbol{a}_1, \boldsymbol{a}_2, \dots, \boldsymbol{a}_k$ through the state space so as to maximize occasional rewards.

Figure 3 provides an illustration of these key learning paradigms. Suppose a database of financial reports is available. If some of them have been historically labelled as positive and negative, we can leverage this to automatically tag future documents. This can be accomplished by training a Learner in a Supervised fashion. If these documents were unstructured, and spotting relations or topics is the goal (political events, economic data, etc.), a Learner trained in an Unsupervised manner can help uncover these hidden structures. Also, these documents can characterise the current state of the capital markets. Using that, a Learner can decide which actions should be taken in order to maximize profits, hedge against certain risks, etc. By interacting and gaining feedback from the environment (Markets), the Learner can Reinforce some behaviours so to avoid future losses or inaccurate decisions.

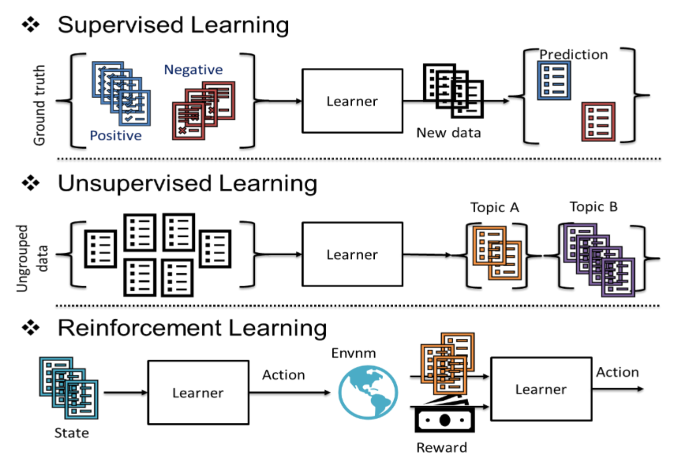


Figure 3: Main learning paradigms of Machine Learning.

Deep Learning, Adversarial Learning, Transfer/Meta Learning

These new forms of learning are 'disrupting' the current models prevalent in Supervised, Unsupervised and Reinforcement learning. They are not only powering new solutions and applications (e.g. driverless vehicles, smart-speakers, etc.) but they are making the resolution of previous problems cheaper, faster and more scalable. The second subdivision is:

- **Deep Learning** - deep learning algorithms attempt to model high-level abstractions in data by using multiple processing layers, with complex structures or otherwise, composed of multiple non-linear transformations. Hence, the mapping function we are attempting to learn can be break down in several compositional operations $f(\mathbf{x}) = f_1 \circ f_2 \circ f_3 \circ \ldots \circ f_n(\mathbf{x})$. Various deep learning architectures such as deep neural networks, convolutional deep neural networks, deep belief networks and recurrent neural networks have been applied to fields like computer vision, automatic speech recognition, natural language processing, audio recognition and bioinformatics where they have been shown to produce state-of-the-art results on various tasks (Goodfellow, et al., 2016; Chollet, 2017).

- **Adversarial Learning** - adversarial machine learning is a technique employed in the field of machine learning which attempts to 'fool' models through malicious input. More formally, assume a given input $\mathbf{x}$ associated to a label $\mathbf{c}$ and a machine learning model f such that $f(\mathbf{x}) = \mathbf{c}$, that is, f can perfectly classify $\mathbf{x}$. We consider $\mathbf{x}^*$ an adversarial example if $\mathbf{x}^*$ is indistinguishable from $\mathbf{x}$ and $f(\mathbf{x}) \neq \mathbf{c}$. Since they are automatically crafted, these adversarial examples tend to be misclassified more often than is true of examples which are perturbed by noise (Szegedy, 2013; Kurakin et al., 2016). Adversarial examples can be introduced during the training of models, making them more robust to attacks from adversarial agents. Typical applications involve increasing robustness in neural networks, spam filtering, information security applications, etc. (Huang et al., 2011).

- **Transfer/Meta Learning** – these two learning paradigms are tightly connected, as their main goal is to encapsulate knowledge learned across many tasks and transfer it to new, unseen ones. Knowledge transfer can help speed up training and prevent overfitting and can therefore improve the obtainable final performance. In Transfer learning, knowledge is transfer from a trained model (or a set thereof) to a new model by encouraging the new model to have similar parameters. The trained model(s) from which knowledge is transferred is not trained with this transfer in mind, and hence the task it was trained on must be very general for it to encode useful knowledge with respect to other tasks. In Meta Learning the learning method (learning rule, initialization, architecture etc.) is abstracted and shared across tasks, and meta-learned explicitly with transfer in mind, such that the learning method generalize to an unseen task. Concretely, often in Transfer learning a pre-trained model is moved to a new task (Devlin et al., 2018; Radford et al., 2019), whilst in Meta learning a pre-trained optimizer is transferred across problems (Andrychowicz et al., 2016; Finn et al., 2017; Flennerhag et al., 2018). In both cases, the usual approach is to learn a Deep Neural Network that can be reused later, usually by stripping some of its terminal layers and creating an encoder-decoder to match the input and output for a task.

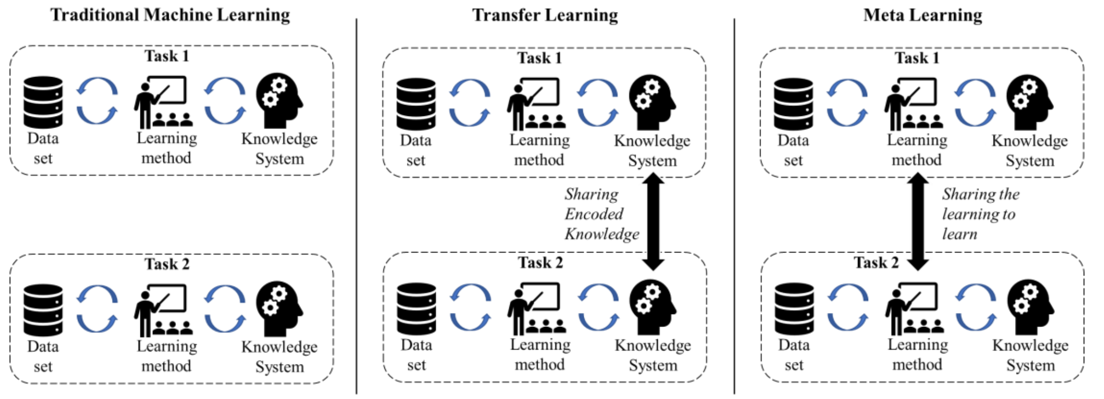


Figure 4: Traditional ML versus Transfer Learning versus Meta Learning.

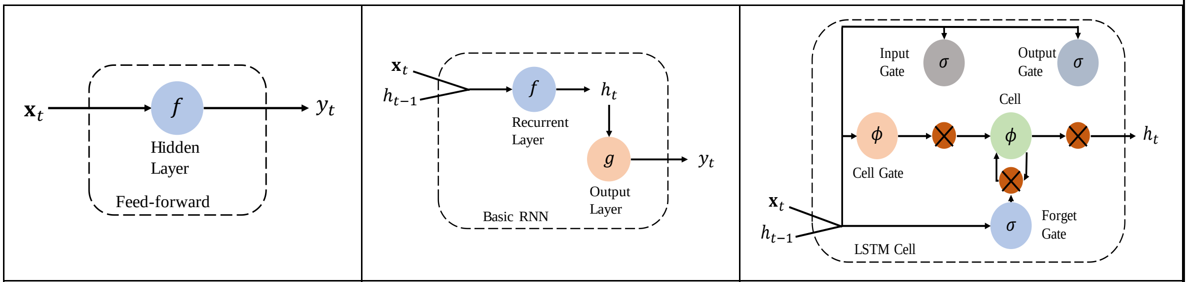
Next we look at the new ‘disruptive’ models and their application.

4. Machine learning Models and Applications

This section introduces LSTMs, GANs, Transfer and Meta Learning, their typical applications and potential uses across Capital Markets.

Long Short-Term Memory (LSTM)

LSTM networks (Hochreiter and Schmidhuber, 1997) are a type of recurrent neural network (RNN) capable of learning order dependence in sequence prediction problems, by keep information about past inputs for an amount of time that is not fixed a priori, but rather depends on its weights, number of stacked layers and on the input data. Whereas a simple feedforward neural network treats its inputs as independent, an RNN uses previous input sources within the calculations to recognize a data's sequential characteristics. Figure 5 illustrates the difference between a typical feed-forward, a recurrent neural and a LSTM network.



$y_t = f(\mathbf{x}_t)$	$h_t = f(h_{t-1}, \mathbf{x}_t)$ $y_t = g(h_t)$	$\mathrm{i}_t = \sigma(\mathbf{x}_t, h_{t-1})$ $\mathrm{f}_t = \sigma(\mathbf{x}_t, h_{t-1})$ $\mathrm{g}_t = \tanh(\mathbf{x}_t, h_{t-1})$ $\mathrm{o}_t = \sigma(\mathbf{x}_t, h_{t-1})$ $\mathrm{c}_t = \mathrm{f}_t * \mathrm{c}_{t-1} + \mathrm{i}_t * \mathrm{g}_t$ $h_t = \mathrm{o}_t * \tanh(\mathrm{c}_t)$ $y_t = g(\mathrm{h}_t)$
a) Feed-forward Network	b) Recurrent Network	c) LSTM Network

Figure 5: Feed-Forward, Recurrent and LSTM Networks.

More formally, we can express their distinctions by their mathematical steps to produce an output:

- **Feed-forward NN**
  - $y_t = f(\mathbf{x}_t)$,

with $\mathbf{x}_t, y_t$ as input and output at time t respectively, and $f(\mathbf{x}) = f_1 \circ f_2 \circ f_3 \circ \ldots \circ f_n(\mathbf{x})$ as similar as a deep neural network computation.

- **Basic RNN**
  - $h_t = f(h_{t-1}, \mathbf{x}_t)$
  - $y_t = g(h_t)$

with g representing a mapping from the hidden state h_t back to the 'visible' output state y_t at time t; this state can be broadly understood as a compressed representation of the sequence being historically observed so far.

- **LSTM**
  - $\mathrm{i}_t = \sigma(\mathbf{x}_t, h_{t-1})$
  - $\mathrm{f}_t = \sigma(\mathbf{x}_t, h_{t-1})$
  - $\mathrm{g}_t = \tanh(\mathbf{x}_t, h_{t-1})$
  - $\mathrm{o}_t = \sigma(\mathbf{x}_t, h_{t-1})$
  - $\mathrm{c}_t = \mathrm{f}_t * \mathrm{c}_{t-1} + \mathrm{i}_t * \mathrm{g}_t$
  - $h_t = \mathrm{o}_t * \tanh(\mathrm{c}_t)$
  - $y_t = g(\mathrm{h}_t)$

with $\mathrm{i}_t, \mathrm{f}_t, \mathrm{g}_t, \mathrm{o}_t$ denoting the input, forget, cell and output gates, respectively; σ the sigmoid function, $\tanh$ the hyperbolic tangent function, and $*$ the Hadamard product. The input, forget, and output gates are responsible for the transfer of information across the architecture, whilst the cell c_t accumulate the information processed across these gates. As their name imply, the input gate decides how much the time t input and previous hidden state still matters for the current moment; the forget gate acts as a 'reset', zeroing the accumulated information stored in the cell; the output gate modulates what part of the current cell state make it to the final hidden state.

Overall, Basic RNNs are a network of neuron-like nodes organized into successive 'layers.' Each node in a given layer connected with a directed (one-way) connection to every other node in the next successive layer. In summary, sequential information (e.g. financial time series) is preserved in the recurrent network's hidden state, which manages to span many time steps as it cascades forward to affect the processing of each new example. LSTMs are designed to overcome one of the drawbacks to Basic RNNs, called the vanishing gradient problem (Goodfellow et al., 2016), in which performance of the neural network suffers because it can't be trained properly. LSTM units categorize data into short-term and long-term memory cells. This enables identification of which data is important, should be remembered, and looped back into the network, and what data can be forgotten.

Traditional use cases: The LSTM model has been found highly successful in many applications, such as unconstrained handwriting recognition (Graves et al., 2009), speech recognition (Graves et al., 2013; Graves and Jaitly, 2014),handwriting generation (Graves, 2013), machine translation (Sutskever et al., 2014), image captioning (Kiros et al., 2014b; Xu et al., 2015), and parsing (Vinyals et al., 2014a).

Applications in capital markets: Heaton et al. (2016) demonstrates LSTMs networks as useful for asset returns movements and new ways to model volatility. It also has been used for trading (Zhang et al., 2019), particularly coupling it with Reinforcement Learning methods. Fischer and Krauss (2018) applied LSTMs to predict assets directional movement, benchmarking it against Random Forest and Logistic Regression. A popular application of LSTM has been on NLP: Hiew et al. (2019) combined BERT (a topic that we discuss ahead) with LSTMs to build a Financial Sentiment Index; RNNs have also been used to read financial news article (Vargas et al., 2017). LSTMs have also been used as the underlying model for model-based Reinforcement learning (Lu, 2017). Some other applications using Feed-forward nets, such as for hedging (Buehler et al., 2019) and to calibrate stochastic volatility models (Bayer et al., 2019), are other promising applications that can be potentially enhanced by LSTM model. More generally, the reader interested in a literature review on financial time series forecasting with Deep Learning should refer to the work of Sezer et al. (2019).

Generative Adversarial Networks (GANs)

Generative Adversarial Networks (Goodfellow et al., 2014) is a modelling strategy that employ two Neural Networks: a Generator (*G*) and a Discriminator (*D*) – Figure 6. The Generator is responsible to produce a rich, high dimensional vector attempting to replicate a given data generation process; the Discriminator acts to separate the input created by the Generator and of the real/observed data generation process. They are trained jointly, with *G* benefiting from *D* incapability to recognise true from generated data, whilst *D* loss is minimized when it is able to classify correctly inputs coming from *G* as fake and the dataset as true. Competition drive both networks to improve their performance until the genuine data is indistinguishable from the generated one.

Traditional use cases: Overall, GANs have been successfully applied to image and text generation (Creswell et al., 2018); BigGAN is a very successful example of using GANs to create high fidelity natural image synthesis and representation learning (Brock et al., 2018) – for a practical examination of it, check this demo.

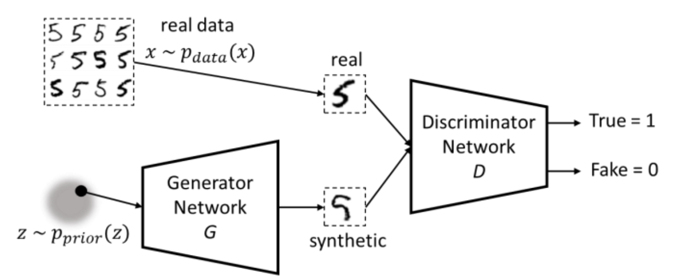


Figure 6: General scheme of a Generative Adversarial Network.

Applications in capital markets: different formulations of GANs are being applied across different domains of capital markets: Fiore et al. (2017) applied GANs to deal with the problem of imbalanced classification, focusing on fraud detection; Mariani et al. (2019) used GANs to perform portfolio analysis, whilst Marti (2019) applied GANs to sample realistic correlation matrices of financial time series. A particular area that have received a concentrated focus has been financial time series modelling using GANs (Koshiyama et al., 2019; Wiese et al., 2019a; Wiese et al., 2019b; Takahashi et al., 2019; Da Silva and Shi, 2019). In this case, a Conditional GAN (Mirza and Osindero, 2014) is often used as the modelling strategy to handle dependent data generation.

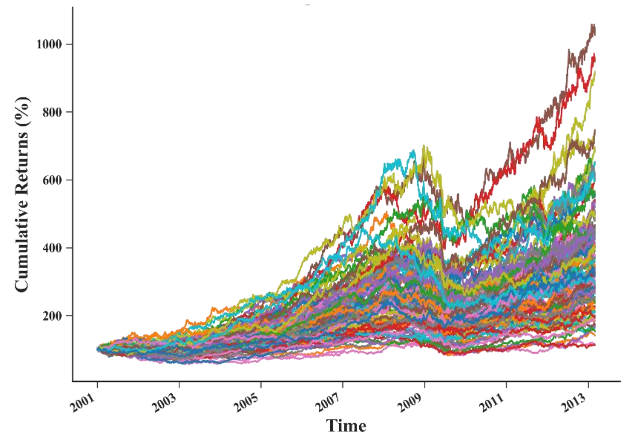


Figure 7: Samples generated by a Conditional GAN for SPX Index. Source: Koshiyama et al. (2019).

As the name implies, conditional GANs are an extension of a traditional GAN, when both *G* and *D* decision is based not only in noise or generated inputs but include an additional information set. For example, this set can represent a class label, a certain categorical feature, or even a current/expected market condition; hence, Conditional GAN attempts to learn an implicit conditional generative model. Such application is more appropriate in cases where the data follows a sequence (time series, text, etc.) or when the user wants to build 'what-if' scenarios (given that S\&P 500 has fallen 1\%, how much should I expect in basis points change of a US 10-year treasury?). Figure 7 depicts samples draw from a Conditional GAN for SPX Index. Using these samples, Koshiyama et al. (2019) managed to improve model validation and combination across trading strategies developed on different asset classes.

Pre-trained Deep Bidirectional Transformer (BERT)

One of the key challenges in applying predictive models to real tasks is the lack of training data. This happens in different areas of Capital Markets, particularly for NLP and Sentiment Analysis. Because NLP is a diversified field with many distinct tasks, most asset-specific datasets contain only a few hundreds or thousands human-labelled training examples. However, modern deep learning-based NLP models typically require much larger amounts of data, improving when trained on millions, or billions, of annotated training examples.

To help close this gap in data, researchers have developed a variety of techniques for training general purpose language representation models using the enormous amount of unannotated text on the web (known as pre-training). The pre-trained model can then be fine-tuned on small-data NLP tasks like sentiment analysis for a specific stock or commodity, resulting in substantial accuracy improvements compared to training on these datasets from scratch.

BERT, Bidirectional Encoder Representations from Transformers (Devlin et al., 2018), is a pre-trained model that can be reused to train state-of-the-art question answering or sentiment analysis systems. BERT is pre-trained on a large corpus of unlabelled text including the entire Wikipedia (roughly 2,500 million words) and Book Corpus (around 800 million words). BERT pre-trained representations are contextual and bidirectional. Figure 8 presents how BERT captures context from both the left and right directions.

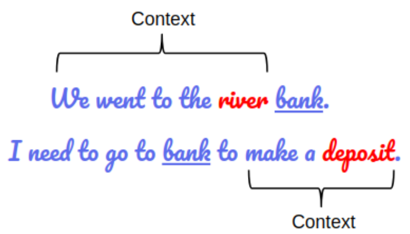


Figure 8: BERT model captures context from both the left and right directions. (Rizvi, 2019)

In the sentence 'We went to the river bank' a unidirectional contextual model would perform reasonably well, by using the previous context 'We went to the river' to predict the word 'bank'. In 'I need to go to bank to make a deposit', such procedure would fail in performing the same inference. However, BERT represents 'bank' using both its previous and next context — 'I need to go ... make a deposit' — starting from the very bottom of a deep neural network, making it deeply bidirectional.

Finally, we can fine-tune BERT by adding just a couple of additional output layers to create state-of-the-art models for a variety of NLP tasks. Figure 9 illustrates the overall pre-training and fine-tuning procedures for BERT. Apart from output layers, the same architectures are used in both pre-training and fine-tuning. The same pre-trained model parameters are used to initialize models for different down-stream tasks.

Traditional use cases: though very recent, BERT has been used as a basis for many research projects and applications of natural language processing (Nogueira and Cho, 2019; Goldberg, 2019; Lee et al., 2019). Google expects BERT systems to factor into about one in ten search results - a significant proportion for a technology published as a research paper less than a year ago. Similar pre-trained language models developed by other AI labs, such as Facebook AI Research's roBERTa and OpenAI's GPT, also received substantial interest and generated different applications (Radford et al., 2018; Liu et al., 2019). In Healthcare, Tencent's MedicalNet project (Chen et al., 2019) uses transfer learning for 3D medical image analysis to aggregate the dataset with diverse modalities, target organs, and pathologies to build relatively large datasets. Based on this dataset, a series of 3D-ResNet pre-trained models and corresponding transfer-learning training code are provided. Tencent's MedicalNet project provides a series of 3D-ResNet pre-trained models and relative code.

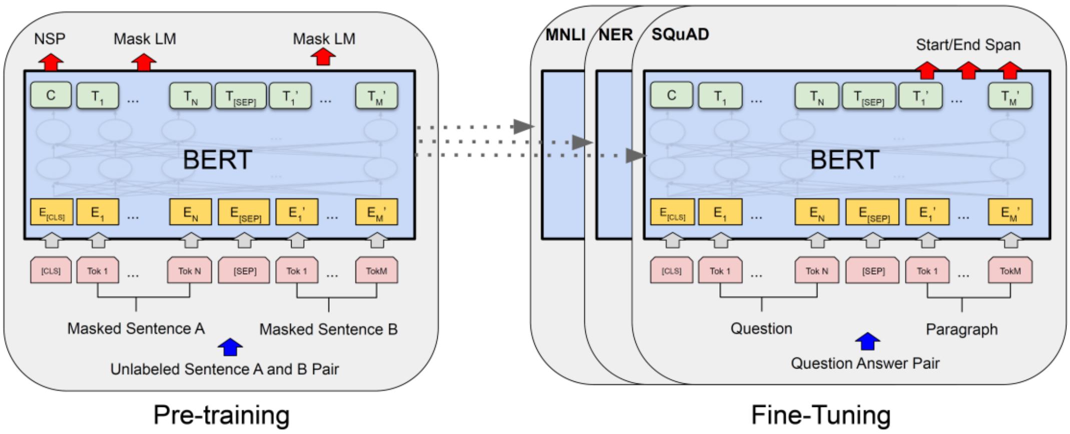


Figure 9: BERT pre-training and fine-tuning procedures (Devlin et al., 2018)

Applications in capital markets: pre-trained models are a very recent area, with very few applications beyond NLP and sentiment analysis. One of such applications is FinBERT (Araci, 2019), a variation of BERT specialized to financial sentiment analysis; it has obtained state-of-the-art results on FiQA sentiment scoring and Financial PhraseBank benchmaks. Hiew et al. (2019) provide a similar application but feeding the sentiment analysis index generated by BERT in a LSTM-based trading strategy to predict stock returns. Apart from Sentiment Analysis, we envisage applications of pre-trained language models to process and classify Compliance and Regulation files and Financial Contracts; pre-trained predictive models to enhance systematic trading and fraud detection; and learning to learn procedures to provide new ways for portfolio optimization and risk management.

5. Algorithms for Market Monitoring

A key driver of algorithms for the Capital Markets is market monitoring and the availability of huge and increasingly comprehensive data sets; especially so-called alternative data. The key data terms are:

- **'Big' data** - extremely large data sets analyzed computationally to reveal patterns, trends, and associations, especially relating to human behaviour and interactions.
- **Financial, Economic and Social Media sources** – here a computer program 'scrapes' data from online sources or extracts data from human-readable output coming from another program.

Company filings, Federal Reserve Bank of St. Louis, Twitter, and Instagram are example sources.

- **Alternative data sources** - data gathered from outside of traditional financial and economic sources. Alternative data in finance refers typically to data used to obtain insight into the investment process.

- **Multiple data sources** - data integration involving combining data coming from multiple disparate information sources and repositories to providing users with a unified view of these data.

Having listed the trends in 'Big' data, next we examine the important algorithm applications:

- **Natural Language Processing (NLP)** - the application of computational techniques to the analysis and synthesis of natural language and speech.

- **Sentiment Analysis** - the process of computationally identifying and categorizing opinions expressed in a piece of text, especially in order to determine whether the writer's attitude towards a particular topic, product, etc. is positive, negative, or neutral.

- **Behavioral Analytics** - an area of data analytics that focuses on providing insight into the actions of people, usually regarding online purchasing, interaction or social behaviour.

- **Predictive Analytics** - the practice of extracting information from existing data sets in order to determine patterns and predict future outcomes and trends; an example being prediction of crimes.

Next, we look at NLP in financial applications

NLP in Finance

The traditional application of NLP in Finance is to process news and tweets, evaluate their polarity/sentiment, and use this as an input for market monitoring and trading strategies (Kolchyna et al., 2015). These sentiment construction in finance relies heavily on the dictionary-based approach, with a few exceptions using simple machine learning techniques such as Naive Bayes classifier. Pre-trained models, such as BERT are starting to seep in this topic, with FinBERT (Araci, 2019) and BERT-LSTM (Hiew et al., 2019) providing a similar application but obtaining better results than the context-free approach of using dictionaries or word embedding representations.

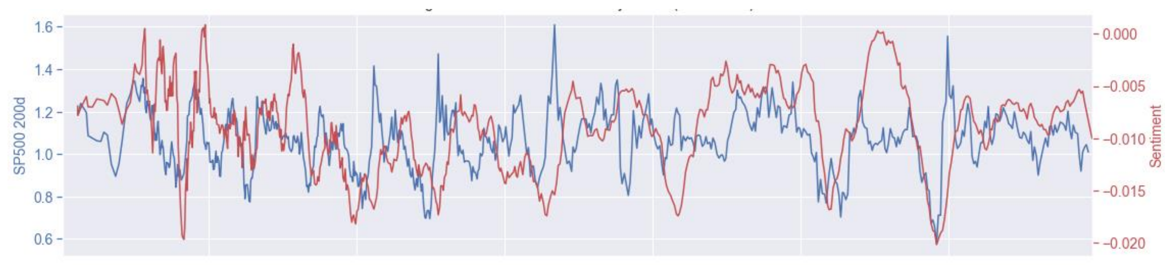


Figure 10: Relationship between sentiment analysis of Federal Open Market Committee minutes and SP500 200d average returns. Source: Bai (2018)

In Bai (2018) the author outlines a dictionary-based approach to read Federal Open Market Committee (FOMC) minutes. By enhancing it via topic modelling, they can breakdown the sentiment per issue (Interest Rate, Labour/Domestic Market, etc.). The main issue is that these FOMC minutes are released with different level of details in several intervals -- statement on the day, three weeks later minutes and full transcript after five years --, making it hard to use them as trading signal. Another application is to create "lie detectors" and sentiment analysis of earning calls (Loughran et al., 2016). For example, a manager in an earnings call might unintentionally use more weak modal words (e.g., may, could, and might), possibly signalling trouble for the firm. Since an analyst, especially during earnings season, can only be on so many calls physically, interesting details emerge when earnings calls are examined in bulk. Comparing nuances like sentiment across all the executives in an industry, or measuring an executive's changing attitude over time, can yield interesting insights. And finally, using it to parse derivatives contracts (Clark and McGonagle, 2019) to automate payments and deliveries.

6. Algorithms for Trading Systems

A major source of algorithm innovation in the capital markets has been trading systems. As illustrated by Figure 10, the major application areas include:

- **Back testing** – is the general method for assesses the viability or measuring how well an algorithmic model or trading strategy would have done ex-post using historical data. If back testing works, traders and analysts may have the confidence to employ it going forward.
- **Forecasting** – in trading a financial forecast uses historical as an estimate of future financial outcomes for an asset or company.
- **Portfolio optimization** –the process of selecting the best portfolio (asset distribution), from the set of all portfolios, to maximizes factors such as expected return, and minimizes costs like financial risk.
- **Trade execution** - the efficient completion of a buy or sell order. Smart order routing (SOR) trade execution is an automated process of handling orders, aimed at taking the best available opportunity throughout a range of different trading venues.

Classic Systematic Systems

Classic trading systems process (as illustrated by Figure 11) may be divided into five stages:

- **Data access/cleaning** – obtaining and cleaning (financial, economic, social, alternative) data that will drive algorithmic trading
- **Pre-trade analysis** – analyses properties of assets to identify trading opportunities using market data or financial news etc. (data analysis).
- **Trading signal generation** – identifies the portfolio of assets to be accumulated, based on the pre-trade analysis (what & when to trade).
- **Trade execution** – executing orders for the selected asset (how to trade).
- **Post-trade analysis** - analyses the results of the trading activity, such as P&L, the difference between the price when a buy/sell decision was made and the final execution price (slippage and transaction costs analysis), and overall return profiles for live and back-tested trading systems.

The Alpha model forecasts the best potential returns; the Risk model evaluates the ‘risk’ associated with a specific investment; and the Transactional Cost model calculates the cost associated with all assets potential sold or purchased. Traditional examples of Alpha models are ‘fixed’ momentum, and mean reversion algorithms. For high frequency trading, given millisecond decision time, often only knowledge-based algorithms are employed for the Alpha model and Portfolio Construction model.

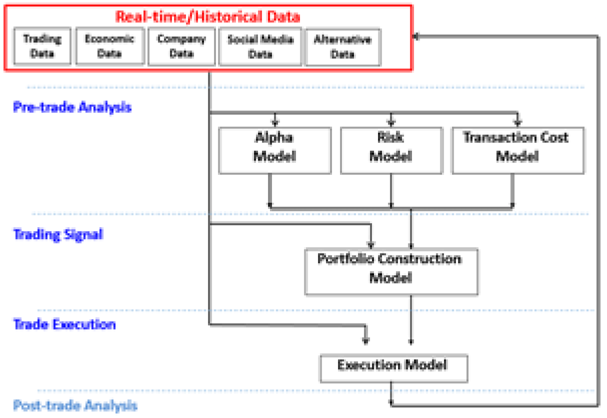


Figure 11: Traditional Algorithmic Trading System

Future Dynamic Systems

The two key aspects of future systematic or algorithmic trading systems are firstly the combination of multiple machine learning and computational statistics algorithms; and secondly systems that dynamically evolve to track the asset class and market. The distinguishing feature from traditional trading systems are the Data optimisation and Model optimisation components, which use multiple (competing) models and select different models over time based on the behaviour of the market.

As illustrated in Figure 12, on the left we have optionally a ‘traditional’ asset optimisation and dimensionality reduction system for selecting the most profitable assets. However, if the system is dedicated to a single commodity, such as oil or coffee, this may be little more than excluding certain countries. Next the system accesses ‘Big’ data, which is fed to the data optimisation component may use multiple models to select and ‘weight’ each variable that may contribute to the forecast.

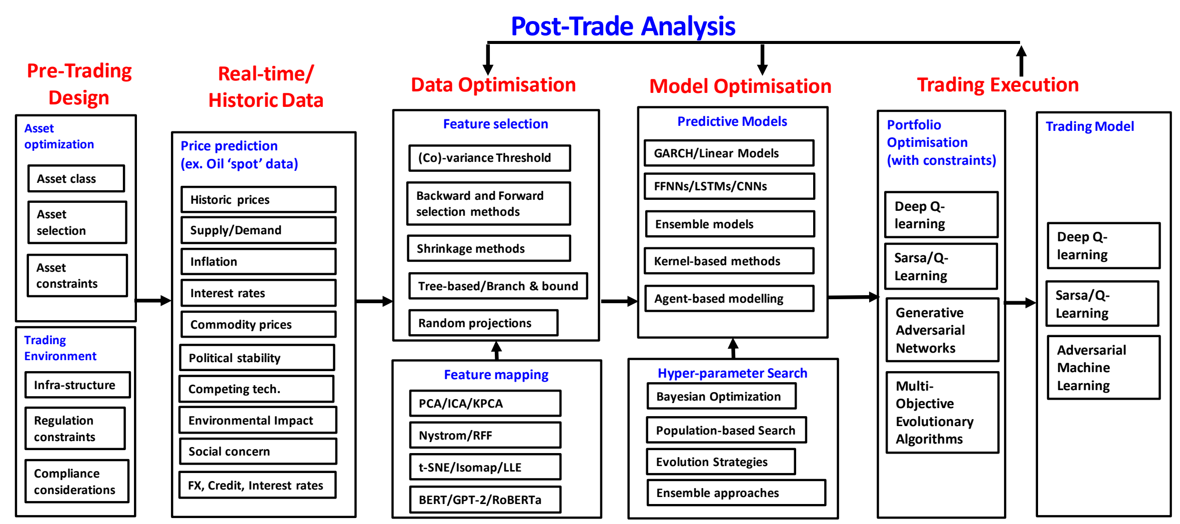


Figure 12: Self-evolving Algorithmic Trading Systems (with example algorithms)

Having examined the two important capital markets application areas, namely market monitoring and trading, we next examine the practical problems raised by the increasingly complex nature of algorithms, especially machine learning.

7. Governance of Algorithms

The major challenges in compiling this review is vast amount of material relevant to future capital markets, such as the practicalities of *algorithm selection* (see Figure 13), the growing requirement of *algorithm collaboration* and the increasing influence of Big data. The result, due to space, is merely to 'flag' and reference key issues.

Here we use the term *governance of algorithms* as a 'catch all' for the rules, practices and processes by which an institution directs and controls algorithms and data. This is increasingly important since ML algorithms effectively self-program and evolve dynamically. Hence, financial institutions and regulators are becoming increasingly concerned with issues of Algorithmic *Interpretability/Explainability* and Data *governance*.

Algorithm Selection

Users face a number of challenges when selecting algorithms for their application:

- **Backtest overfitting** - where many variations of a (trading) strategy are tried on the same dataset, and, as a result, strategies looking good on paper often perform poorly when presented with new data. Currently, there is an increasing quest for devising a set of procedures to deal with this issue; refer to Koshiyama and Firoozye (2019) for a review of the current literature and a few solutions to Backtest overfitting.

- **Feature engineering** - is augmentation of data; the process of going from raw data to data that is ready for modeling. Strategies and associated algorithms include: a) reduce data redundancy/dimensionality (e.g. PCA); b) capturing complex relationships (e.g. NNs); c) rescaling variables (e.g. standardizing or normalizing) etc.

- **Data scarcity** – means too few data points (to train a model) often because it is difficult to get data or the data is small as compared to the amount needed. Whereas *Data sparsity* means data distributed sparsely over the available feature space.

- **Data sensitivity** – data owners need to contribute data to collaborative analytics, while not wishing to 'share' extremely valuable and sensitive raw data. An important solution discussed below is Federated Learning.

- **Hyperparameter optimization** - is the problem of choosing a set of optimal input variables (i.e. hyperparameters) for a learning algorithm. A hyperparameter is a parameter whose value is used to control the learning process, in contrast to other parameters (typically node weights) that are 'learnt'.

- **Interpretability/Explainability** - in machine learning explainability and interpretability are often used interchangeably, but: a) *Interpretability* is about the extent to which a cause and effect can be observed within a system; b) *Explainability* is the extent to which the internal mechanics of an algorithm can be explained in human terms.

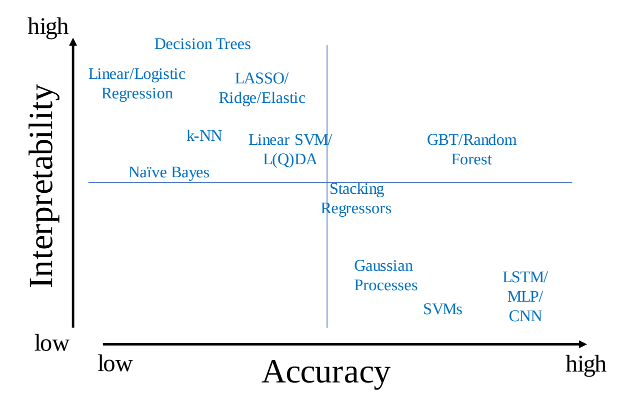


Figure 13: Algorithm selection trade-offs: model-specific Interpretability vs Accuracy.

Examples of potential solutions to model selection challenges are:

- **Pre-trained models** - a model created to solve a similar problem, often on a large data set, is used as a starting point instead of building a model from scratch. This is the basis of Transfer learning.

- **AutoML** - automated machine learning (AutoML) is the process of automating end-to-end the process of applying machine learning to real-world problems. Companies such as h2o.ai, Datarobot, Amazon etc. have created *AutoML*-like systems.

Algorithm Collaboration

Traditionally, financial systems have deployed single algorithms for market monitoring and trading. However, as discussed future systems will increasingly involve the collaboration of multiple algorithms and multiple data sources. (Closely related is *Edge computing* with client data processed at the periphery of the network, at or as close to the originating data source as possible).

Examples of algorithm collaboration are:

- **Multiple models** - a system that combines several base algorithms in order to produce one optimal predictive model. An example is ensemble methods (). In statistics and machine learning, ensemble methods use multiple learning algorithms to obtain better predictive performance than could be obtained from any of the constituent learning algorithms alone.

- **Dynamic systems** - systems comprising firstly multiple AI and computational statistics algorithms; and secondly ML systems that dynamically evolve to track the asset class and market (see Figure 12).

- **Federated Learning** –- a machine learning technique that trains an algorithm across multiple decentralized data sources without sharing sensitive 'raw' data; only locally analysed anonymous results returned (cf. taking the algorithm to the data).

Federated Learning is an important emerging technique, given the value and sensitivity of data (e.g. financial, business, social, alternative and regulatory). With Federated Learning, an algorithm at each client independently computes an update to the current model based on its local data, and communicates this update to a central algorithm, where the client-side updates are aggregated to compute a new global model.

Interpretability/Explainability

In the context of AI and ML, Explainability and Interpretability are often used interchangeably. Algorithmic *Interpretability* is about the extent to which a cause and effect can be observed within a system, and the extent an observer is able to predict what will happen, for a given set of input or algorithm parameters. Algorithmic *Explainability* is the extent to which the internal mechanics of a ML (deep learning) system is explainable in human terms. In simple terms, Interpretability is about understanding the algorithm mechanics (without necessarily knowing why); Explainability is being able to explain what is happening in the algorithm.

There are multiple forms to generate and provide explanations based on an algorithmic decision-making system. Figure 14 presents the types and levels of Explainability: model-specific and agnostic, global and local (Hall and Gill, 2018; Molnar, 2019). Below we unwrap these concepts, as well as outline some technical solutions:

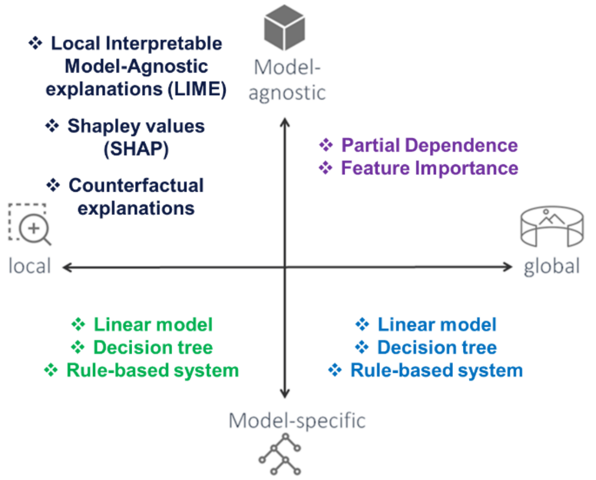


Figure 14: Types and levels of Explainability.

Model-specific: With model specific explainability, a model is designed and developed in such a way that it is fully transparent and explainable by design. In other words, an additional explainability technique is not required to be overlaid on the model in order to be able to fully explain its workings and outputs. In general, explainable models are simpler than non-explainable models and as such their performance in terms of accuracy is relatively diminished. Explainable models include linear regression, decision trees, k-nearest neighbours, and rule-based systems.

Model-agnostic: With model-agnostic explainability, a mathematical technique is applied to the outputs of any algorithm including very complex and opaque models, in order to provide an interpretation of the decision drivers for those models. A few of the most popular approaches include Shapley Explanations (Lundberg and Lee, 2017) and Local Interpretable Model-Agnostic Interpretation (Ribeiro et al., 2016). However, a general limitation of all model-agnostic explainability techniques is that it entails running an additional model on top of an already complex model. The explainability technique will never be 100% accurate, and therefore a layer of additional inaccuracy is introduced into an already inaccurate model, and the output becomes one step further removed from the reality.

Global: this facet focuses on understanding the algorithm's behaviour at a high/dataset/populational level. The usual techniques to provide these explanations are Feature Importance and Partial Dependence. Overall, these methods quantify the weight of a feature in the model's performance and predictions, usually by experimenting with small changes in the data. Apart from a few models (like Decision Trees), both techniques are computationally expensive; it will take time to vary each feature in order to approximate an accurate interpretation of the model, particularly with big datasets. The typical user of Feature Importance and Partial Dependence are researchers and designer of algorithms, since they tend to be more interested with the general insights and knowledge discovery that the model produce, rather than specific individual cases.

Local: this facet focuses on understanding the algorithm's behaviour at a low/subset/individual level. A variety of methods have been developed to help to interpret why a model decided for a particular data point. Three of the most popular tools are: Local Interpretable Model-Agnostic Interpretations (LIME); Shapley explanations (SHAP); Counterfactual explanations (CE) (Wachter et al., 2018). In a nutshell (i) LIME: samples individual data points and weighs them according to similarity to the individual data point that is to be explained; (ii) SHAP: trains a model with each individual feature, computes the result, repeats with the other features, and then adds features one by one into the model in order to identify the true importance of each feature – this is usually approximated via Monte Carlo sampling; and (iii) CE: this is a computationally expensive technique which considers how the model would behave if some features had different values, allowing an explanation to be built up of individual decision factors, and enabling potential recourse and a more clear understanding by the individual under analysis. The typical user of local explanations are individuals being targeted by an algorithm, as well as members of the judiciary and regulators trying to make a case about potential discrimination.

Robustness

Algorithmic robustness is characterized by how effectively an algorithm can be deemed as safe and secure, not vulnerable to tampering or compromising of the data they are trained on. We can rate an algorithm's robustness using four key criteria (EU-HLEG, 2019):

- **Resilience to attack and security**: AI systems, like all software systems, should be protected against vulnerabilities that can allow them to be exploited by adversaries, such as data poisoning, model leakage or the infrastructure, both software and hardware. This concept is linked with the mathematical concept of ***Adversarial Robustness*** (Carlini et al., 2019), that is, how the algorithm performed in the worst-case scenario? (e.g. how the algorithm would react during the 2008 Financial Crisis?).

- **Fallback plan and general safety**: AI systems should have safeguards that enable a fallback plan in case of problems. Also, the level of safety measures required depends on the magnitude of the risk posed by an AI system. This notion is strongly associated with the technical concept of ***Formal Verification*** (Qin et al., 2019), which in broad terms means: does the algorithm attends the problem specifications and constraints? (e.g. respect physical laws).

- **Accuracy**: pertains to an AI system's ability to make correct judgements, for example to correctly classify information into the proper categories, or its ability to make correct predictions, recommendations, or decisions based on data or models. Accuracy as a general concept can be quantified by estimating the ***Expected Generalization Performance*** (Arlot and Celisse, 2010), which means that in general, how well the algorithm works? (e.g. in 7 out of 10 cases, the algorithm makes the right decision).

- **Reliability and Reproducibility**: a reliable AI system is one that works properly with a range of inputs and in a range of situations, whilst reproducibility describes whether an AI experiment exhibits the same behaviour when repeated under the same conditions. This idea is tied with the software engineering concept of ***Continuous Integration*** (Meyer, 2014), that is, is the algorithm auditable? (e.g. reliably reproduce its decisions).

In practice, each technical criteria embodies a number of technical solutions (Figure 15). These technical solutions can aid the analyst in measuring and having systems in place to assess and make systems more robust before deployment stage.

Criteria	**Technical Solution**
Expected generalization performance	▪ **Cross-validation**: k-fold cv, bootstrap, etc. ▪ **Covariance-penalty**: Mallow's C_p, Stein Unbiased Risk Estimator, bootstrap approximation, etc.
Adversarial robustness	▪ **Evasion attacks**: fast gradient sign method, DeepFool, etc. ▪ **Defence**: label smoothing, variance minimization, etc.
Formal verification	▪ **Complete**: Satisfiability Modulo Theory, Mixed Prog, etc. ▪ **Incomplete**: Propagating bounds, Convex Optimization, etc.
Reliability and reproducibility	▪ **Code versioning**: Git (Github), Mercurial (BitBucket), etc. ▪ **Reproducible analysis**: Binder, Docker, etc. ▪ **Automated testing**: Travis CI, Scrutinizer CI, etc.

Figure 15: Mapping technical criteria and solutions for Algorithmic Robustness.

Risk

Algorithms, especially so-called black box trading algorithms, amplifies systemic risk for a number of reasons: a) *Intensifying Volatility* - algorithms can react instantaneously to market conditions and during volatile markets may greatly widen their bid-ask spreads, or temporarily stop trading thereby diminishing liquidity; b) *Flash crash* - increased algorithm and market integration means a meltdown in a major market or asset class often has a ripples effect across other markets; c) *Uncertainty* – algorithm opaqueness stokes investor uncertainty; d) *Rogue algorithms* – due to speed and lack of transparency one errant or faulty algorithm can rack up millions in losses in a very short period (e.g. Knight Capital lost $440 million in a 45-minute period on August 1, 2012); e) *Algorithm uniformity* – a lack of diversity in (trading) algorithms could reduce robustness in a market (cf. Irish potato famine).

Legality and Ethics

Increasingly, ML algorithms self-program and evolve dynamically, raising concerns about explainability of financial decisions (e.g. for mortgages, loans); discriminatory, unethical and illegal behaviour (e.g. CV/Resume 'sifting' recruitment systems); and unintentional fraudulent systems (e.g. systematic trading systems and market manipulation). Naturally, financial institutions and regulators are becoming increasingly concerned with ensuring there remains a modicum of human control.

Compliance and Regulation

Compliance departments are increasingly using AI algorithms to automate procedures and monitor behaviour of employees. Now they face the increasing challenge of self-programming algorithms being discriminatory, unethical and illegal; exposing the institutions to (unintentional) reputational damage, financial loss and potentially large fines.

Financial regulators are also increasing using AI algorithms to automate monitoring and reporting (Treleaven, 2016). To underpin this automation and leverage AI algorithms, leading financial regulators are seeking to encode regulatory rules as computer-executable code, allowing compliance and regulation to be fully automated, and operate in real time and across multiple jurisdictions.

Traditionally, Regulators have faced the challenge of regulating institutions and individuals, and the processing the 'tsunami' of reporting data. Going forward, Regulators have the additional challenge of regulating algorithm behaviour.

Legal Status of Algorithms

Finally, there is the growing discussion in the Judiciary concerning the 'status of algorithms in Law'. In Law, as we know, companies have the rights and obligations of a person. Algorithms are rapidly emerging as artificial persons: a legal entity that is not a human being but for certain purposes is legally considered to be a natural person (Treleaven et al., 2019). The argument is that since algorithms are creating agencies with humans, companies and even other algorithms they also need to have the status of an artificial person in Law.

Alternative Data

However, although AI and algorithms received all the publicity, many people believe that we have yet to experience the full extent of the so-called data revolution, and especially the use by the capital markets of alternative data. For example, investment funds are buying anonymised real-time credit card data, and therefore can 'see' what is going through a retailer's tills but can see it expansively across the whole industry sector. As a definition, alternative data (in finance) refers to data used to obtain insight into the investment process; sources such as financial transactions, retail data, sensors, mobile devices, satellites, public records, and the Internet (wiki). Surprisingly companies that produces alternative data (e.g. credit card, retail, telecoms, transportation etc.) generally overlook the value of their data to financial institutions. Hence, these data sets are often less readily accessible and less well-structured than traditional sources of data. Refer to the Deven and Amen (2020) book for a comprehensive introduction and review in this topic.

8. Conclusions

This paper reviews AI, ML and associated algorithms, and discuss their future impact on the Capital Markets. These technologies are already having a substantial impact in industry and academia. In industry, these developments have led to a scramble for talent across the Investment Banking world, with Data Scientists poached from Tech companies; and AI Labs are being set up inside banks. In academia, we can perceive a shift in the mainstream research from a more theory-driven and pure stochastic calculus-based research to a data-driven and machine-learning-based focus. These signals are traced across important venues, particularly the main mathematical finance conferences and journals.

The impact of algorithms in future Capital Markets are influenced by a number of research 'drivers'. Firstly, self-programming machine learning algorithms that dynamically adjust to their target markets. Secondly, collaboration of algorithms from computational statistics, AI and complex systems. Thirdly, the increasing use of Big data and novel 'alternative' datasets. Fourthly, the use of federated learning for privacy-preserving data access. Fifthly, linked to the dynamic nature of machine learning algorithms, issues of interpretability, ethics and legality of algorithms. Finally, it is important to note that foundational algorithm research is being pioneered increasingly in leading financial institutions for competitiveness.

9. Acknowledgements

The authors would like to thank the following reviewers for their suggestions for improving the manuscript: Sebastian Flennerhag, Tharsis Souza, Gerald Rushton, Janet Adams, Bertrand Nortier, etc. Adriano Koshiyama would like to acknowledge the National Research Council of Brazil for his PhD scholarship.

10. References

Acar, E., & Satchell, S. (Eds.). (2002). Advanced trading rules. Butterworth-Heinemann.

Adewumi, A. O., & Akinyelu, A. A. (2017). A survey of machine-learning and nature-inspired based credit card fraud detection techniques. International Journal of System Assurance Engineering and Management, 8(2), 937-953.

Andrychowicz, Marcin, Misha Denil, Sergio Gomez, Matthew W. Hoffman, David Pfau, Tom Schaul, Brendan Shillingford, and Nando De Freitas. "Learning to learn by gradient descent by gradient descent." In Advances in neural information processing systems, pp. 3981-3989. 2016.

Araci, D. (2019). FinBERT: Financial Sentiment Analysis with Pre-trained Language Models. arXiv preprint arXiv:1908.10063.
Arlot, Sylvain, and Alain Celisse. "A survey of cross-validation procedures for model selection." Statistics surveys 4 (2010): 40-79.

Arulkumaran, K., Deisenroth, M. P., Brundage, M., & Bharath, A. A. (2017). Deep reinforcement learning: A brief survey. IEEE Signal Processing Magazine, 34(6), 26-38.

Aziz, S., & Dowling, M. (2019). Machine Learning and AI for Risk Management. In Disrupting Finance (pp. 33-50). Palgrave Pivot, Cham.

Bäck, T., Fogel, D. B., & Michalewicz, Z. (1997). Handbook of evolutionary computation. CRC Press.

Bai, C. (2018). Trading signal from Fed minutes? Can fed minutes be a trading signal even after 3 weeks of delay? Link: https://towardsdatascience.com/trading-signal-from-fed-minutes-4c2def891134

Barabási, A. L. (2016). Network science. Cambridge university press.

Batrinca, B., & Treleaven, P. C. (2015). Social media analytics: a survey of techniques, tools and platforms. Ai & Society, 30(1), 89-116.

Bayer, C., Horvath, B., Muguruza, A., Stemper, B., & Tomas, M. (2019). On deep calibration of (rough) stochastic volatility models. arXiv preprint arXiv:1908.08806.

Brock, A., Donahue, J., & Simonyan, K. (2018). Large scale gan training for high fidelity natural image synthesis. arXiv preprint arXiv:1809.11096.
Brownlee, J. (2011). Clever algorithms: nature-inspired programming recipes. Jason Brownlee.

Buehler, H., Gonon, L., Teichmann, J., & Wood, B. (2019). Deep hedging. Quantitative Finance, 1-21.

Caccioli, F., Barucca, P., & Kobayashi, T. (2018). Network models of financial systemic risk: A review. Journal of Computational Social Science, 1(1), 81-114.

Cambria, E., Schuller, B., Liu, B., Wang, H., & Havasi, C. (2013). Statistical approaches to concept-level sentiment analysis. IEEE Intelligent Systems, 28(3), 6-9.

Carlini, Nicholas, Anish Athalye, Nicolas Papernot, Wieland Brendel, Jonas Rauber, Dimitris Tsipras, Ian Goodfellow, and Aleksander Madry. "On evaluating adversarial robustness." arXiv preprint arXiv:1902.06705 (2019).

Caron, M., Bojanowski, P., Joulin, A., & Douze, M. (2018). Deep clustering for unsupervised learning of visual features. In Proceedings of the European Conference on Computer Vision (ECCV) (pp. 132-149).

Chen, S., Ma, K., & Zheng, Y. (2019). Med3D: Transfer Learning for 3D Medical Image Analysis. arXiv preprint arXiv:1904.00625.

Chen, X., Duan, Y., Houthooft, R., Schulman, J., Sutskever, I., & Abbeel, P. (2016). Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In Advances in neural information processing systems (pp. 2172-2180).

Clack, C. D., & McGonagle, C. (2019). Smart Derivatives Contracts: the ISDA Master Agreement and the automation of payments and deliveries. arXiv preprint arXiv:1904.01461.

Chollet, F. (2017). Deep Learning with Python. Manning Publications co.

Creswell, A., White, T., Dumoulin, V., Arulkumaran, K., Sengupta, B., & Bharath, A. A. (2018). Generative adversarial networks: An overview. IEEE Signal Processing Magazine, 35(1), 53-65.

Da Silva, B., & Shi, S. S. (2019). Style Transfer with Time Series: Generating Synthetic Financial Data. arXiv preprint arXiv:1906.03232.

Deven, A., Amen, A. (2020). The book of alternative data: a guide to investors, traders and risk managers. Wiley.

Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.

Dixon, M. F., & Halperin, I. (2019). The Four Horsemen of Machine Learning in Finance. Available at SSRN 3453564.

Do Prado, M. L. (2018). Advances in financial machine learning. John Wiley & Sons.

Efron, B., & Hastie, T. (2016). Computer age statistical inference (Vol. 5). Cambridge University Press.

EU-HLEG. (2019). Ethics guidelines for trustworthy AI. https://ec.europa.eu/digital-single-market/en/news/ethics-guidelines-trustworthy-ai

Fei-Fei, L. (2003, October). A Bayesian approach to unsupervised one-shot learning of object categories. In Proceedings Ninth IEEE International Conference on Computer Vision (pp. 1134-1141). IEEE.

Finn, C., Abbeel, P., & Levine, S. (2017, August). Model-agnostic meta-learning for fast adaptation of deep networks. In Proceedings of the 34th International Conference on Machine Learning-Volume 70 (pp. 1126-1135). JMLR. org.

Fiore, U., De Santis, A., Perla, F., Zanetti, P., & Palmieri, F. (2017). Using generative adversarial networks for improving classification effectiveness in credit card fraud detection. Information Sciences.

Fischer, T., & Krauss, C. (2018). Deep learning with long short-term memory networks for financial market predictions. European Journal of Operational Research, 270(2), 654-669.

Flennerhag, S., Moreno, P. G., Lawrence, N. D., & Damianou, A. (2018). Transferring knowledge across learning processes. arXiv preprint arXiv:1812.01054.

Giarratano, J. C., & Riley, G. (1998). Expert systems. PWS publishing co..

Gleave, A., Dennis, M., Kant, N., Wild, C., Levine, S., & Russell, S. (2019). Adversarial Policies: Attacking Deep Reinforcement Learning. arXiv preprint arXiv:1905.10615.

Goldberg, Y. (2019). Assessing BERT's Syntactic Abilities. arXiv preprint arXiv:1901.05287.

Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep learning. MIT press.

Goodfellow, Ian, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. "Generative adversarial nets." In Advances in neural information processing systems, pp. 2672-2680. 2014.

Graves, A. (2013). Generating sequences with recurrent neural networks. Technical report,arXiv:1308.0850.

Graves, A. and Jaitly, N. (2014). Towards end-to-end speech recognition with recurrentneural networks. In ICML’2014

Graves, A., Liwicki, M., Fernández, S., Bertolami, R., Bunke, H., and Schmidhuber, J.(2009). A novel connectionist system for unconstrained handwriting recognition. PatternAnalysis and Machine Intelligence, IEEE Transactions on, 31(5), 855–868.

Graves, A., Mohamed, A., and Hinton, G. (2013). Speech recognition with deep recurrentneural networks. In ICASSP’2013 , pages 6645–6649.

Hall, Patrick, and Navdeep Gill. Introduction to Machine Learning Interpretability. O'Reilly Media, Incorporated, 2018.

Heaton, J. B., Polson, N. G., & Witte, J. H. (2017). Deep learning for finance: deep portfolios. Applied Stochastic Models in Business and Industry, 33(1), 3-12.

Hiew, J. Z. G., Huang, X., Mou, H., Li, D., Wu, Q., & Xu, Y. (2019). BERT-based Financial Sentiment Index and LSTM-based Stock Return Predictability. arXiv preprint arXiv:1906.09024.

Hiew, Joshua Zoen Git, Xin Huang, Hao Mou, Duan Li, Qi Wu, and Yabo Xu. "BERT-based Financial Sentiment Index and LSTM-based Stock Return Predictability." arXiv preprint arXiv:1906.09024 (2019).

Higgins, Irina, Arka Pal, Andrei Rusu, Loic Matthey, Christopher Burgess, Alexander Pritzel, Matthew Botvinick, Charles Blundell, and Alexander Lerchner. "Darla: Improving zero-shot transfer in reinforcement learning." In Proceedings of the 34th International Conference on Machine Learning-Volume 70, pp. 1480-1490. JMLR. org, 2017.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. Neural computation, 9(8), 1735-1780.

Hoi, S. C., Sahoo, D., Lu, J., & Zhao, P. (2018). Online Learning: A Comprehensive Survey, https://arxiv.org/abs/1802.02871

Huang, L., Joseph, A. D., Nelson, B., Rubinstein, B. I., & Tygar, J. D. (2011, October). Adversarial machine learning. In Proceedings of the 4th ACM workshop on Security and artificial intelligence (pp. 43-58). ACM.

Hüttner, A., Mai, J. F., & Mineo, S. (2018). Portfolio selection based on graphs: Does it align with Markowitz-optimal portfolios?. Dependence Modeling, 6(1), 63-87.

Juszczak, P., Adams, N. M., Hand, D. J., Whitrow, C., & Weston, D. J. (2008). Off-the-peg and bespoke classifiers for fraud detection. Computational Statistics & Data Analysis, 52(9), 4521-4532.

Keneshloo, Y., Ramakrishnan, N., & Reddy, C. K. (2019, May). Deep Transfer Reinforcement Learning for Text Summarization. In Proceedings of the 2019 SIAM International Conference on Data Mining (pp. 675-683). Society for Industrial and Applied Mathematics.

Kidger, P., Bonnier, P., Arribas, I. P., Salvi, C., & Lyons, T. (2019). Deep Signature Transforms. In Advances in Neural Information Processing Systems (pp. 3099-3109).

Kiros, R., Salakhutdinov, R., and Zemel, R. (2014b). Unifying visual-semantic embeddingswith multimodal neural language models. arXiv:1411.2539

Kolchyna, O., Souza, T. T., Treleaven, P., & Aste, T. (2015). Twitter sentiment analysis: Lexicon method, machine learning method and their combination. arXiv preprint arXiv:1507.00955.

Kolm, P. N., Tütüncü, R., & Fabozzi, F. J. (2014). 60 Years of portfolio optimization: Practical challenges and current trends. European Journal of Operational Research, 234(2), 356-371.

Koshiyama, A. S., Firoozye, N., & Treleaven, P. (2018). A Machine Learning-based Recommendation System for Swaptions Strategies. arXiv preprint arXiv:1810.02125.

Koshiyama, A., & Firoozye, N. (2019). Avoiding Backtesting Overfitting by Covariance-Penalties: an empirical investigation of the ordinary and total least squares cases. The Journal of Financial Data Science, jfds-2019.

Koshiyama, A., Firoozye, N., & Treleaven, P. (2019). Generative Adversarial Networks for Financial Trading Strategies Fine-Tuning and Combination. arXiv preprint arXiv:1901.01751.

Koshiyama, A., Firoozye, N., & Treleaven, P. (2019). Generative Adversarial Networks for Financial Trading Strategies Fine-Tuning and Combination. arXiv preprint arXiv:1901.01751.

Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. "Imagenet classification with deep convolutional neural networks." Advances in neural information processing systems. 2012.

Kurakin, A., Goodfellow, I., & Bengio, S. (2016). Adversarial machine learning at scale. arXiv preprint arXiv:1611.01236.

LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., & Jackel, L. D. (1989). Backpropagation applied to handwritten zip code recognition. Neural computation, 1(4), 541-551.

Lee, J., Yoon, W., Kim, S., Kim, D., Kim, S., So, C. H., & Kang, J. (2019). Biobert: pre-trained biomedical language representation model for biomedical text mining. arXiv preprint arXiv:1901.08746.

Li, Bin, and Steven CH Hoi. "Online portfolio selection: A survey." ACM Computing Surveys (CSUR) 46, no. 3 (2014): 35., https://dl.acm.org/citation.cfm?id=2512962

Liu, Yinhan, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. "Roberta: A robustly optimized bert pretraining approach." arXiv preprint arXiv:1907.11692 (2019).

Loughran, T., & McDonald, B. (2016). Textual analysis in accounting and finance: A survey. Journal of Accounting Research, 54(4), 1187-1230.

Lu, D. W. (2017). Agent inspired trading using recurrent reinforcement learning and lstm neural networks. arXiv preprint arXiv:1707.07338.

Lundberg, S. M., & Lee, S. I. (2017). A unified approach to interpreting model predictions. In Advances in Neural Information Processing Systems (pp. 4765-4774).

Makhzani, A., Shlens, J., Jaitly, N., Goodfellow, I., & Frey, B. (2015). Adversarial autoencoders. arXiv preprint arXiv:1511.05644.

Mariani, G., Zhu, Y., Li, J., Scheidegger, F., Istrate, R., Bekas, C., & Malossi, A. C. I. (2019). PAGAN: Portfolio Analysis with Generative Adversarial Networks. arXiv preprint arXiv:1909.10578.

Marti, G. (2019). CorrGAN: Sampling Realistic Financial Correlation Matrices Using Generative Adversarial Networks. arXiv: 1910.09504

May, R. M., Levin, S. A., & Sugihara, G. (2008). Complex systems: Ecology for bankers. Nature, 451(7181), 893.

McNeil, A. J., Frey, R., & Embrechts, P. (2005). Quantitative risk management: Concepts, techniques and tools (Vol. 3). Princeton: Princeton university press.

Meyer, M. (2014). Continuous integration and its tools. IEEE software, 31(3), 14-16.

Mirza, M., & Osindero, S. (2014). Conditional generative adversarial nets. arXiv preprint arXiv:1411.1784.

Miyato, T., Dai, A. M., & Goodfellow, I. (2016). Adversarial training methods for semi-supervised text classification. arXiv preprint arXiv:1605.07725.

Molnar, C. (2019). Interpretable machine learning. Lulu.com https://christophm.github.io/interpretable-ml-book/

Nicolae, Maria-Irina, Mathieu Sinn, Minh Ngoc Tran, Ambrish Rawat, Martin Wistuba, Valentina Zantedeschi, Nathalie Baracaldo et al. "Adversarial Robustness Toolbox v0. 4.0." arXiv preprint arXiv:1807.01069 (2018).

Nogueira, R., & Cho, K. (2019). Passage Re-ranking with BERT. arXiv preprint arXiv:1901.04085.

Nuti, G., Mirghaemi, M., Treleaven, P., & Yingsaeree, C. (2011). Algorithmic trading. Computer, 44(11), 61-69.

P. Treleaven, J. Barnett and A. Koshiyama, "Algorithms: Law and Regulation," in Computer, vol. 52, no. 2, pp. 32-40, Feb. 2019.

Pinto, L., Davidson, J., Sukthankar, R., & Gupta, A. (2017, August). Robust adversarial reinforcement learning. In Proceedings of the 34th International Conference on Machine Learning-Volume 70 (pp. 2817-2826). JMLR. org.

Poli, R., Langdon, W. B., McPhee, N. F., & Koza, J. R. (2008). A field guide to genetic programming. Lulu. com.

Pozzi, F., Di Matteo, T., & Aste, T. (2013). Spread of risk across financial markets: better to invest in the peripheries. Scientific reports, 3, 1665.

Qin, Chongli, Brendan O'Donoghue, Rudy Bunel, Robert Stanforth, Sven Gowal, Jonathan Uesato, Grzegorz Swirszcz, and Pushmeet Kohli. "Verification of non-linear specifications for neural networks." arXiv preprint arXiv:1902.09592 (2019).

Radford, A., Narasimhan, K., Salimans, T., & Sutskever, I. (2018). Improving language understanding with unsupervised learning. Technical report, OpenAI.

Radford, A., Wu, J., Amodei, D., Amodei, D., Clark, J., Brundage, M., & Sutskever, I. (2019). Better language models and their implications. OpenAI, 2018b. URL https://openai. com/blog/better-language-models.

Radford, A., Wu, J., Amodei, D., Amodei, D., Clark, J., Brundage, M., & Sutskever, I. (2019). Better language models and their implications. OpenAI, 2018b. URL https://openai. com/blog/better-language-models.

Reschenhofer, E., Mangat, M. K., Zwatz, C., & Guzmics, S. (2018). Evaluation of current research on stock return predictability. Journal of Forecasting, 1-18. https://doi.org/10.1002/for.2629

Ribeiro, Marco Tulio, Sameer Singh, and Carlos Guestrin. "Why should i trust you?: Explaining the predictions of any classifier." Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM, 2016.

Rizvi, M. S. Z. (2019) Demystifying BERT: A Comprehensive Guide to the Groundbreaking NLP Framework. Link: https://www.analyticsvidhya.com/blog/2019/09/demystifying-bert-groundbreaking-nlp-framework/

Russell, S. J., & Norvig, P. (2016). Artificial intelligence: a modern approach. Malaysia; Pearson Education Limited.

Sarkar, D., Bali, R., & Ghosh, T. (2018). Hands-On Transfer Learning with Python: Implement advanced deep learning and neural network models using TensorFlow and Keras. Packt Publishing Ltd.

Sezer, O. B., Gudelek, M. U., & Ozbayoglu, A. M. (2019). Financial Time Series Forecasting with Deep Learning: A Systematic Literature Review: 2005-2019. arXiv preprint arXiv:1911.13288.

Siddhant, A., Goyal, A., & Metallinou, A. (2019, July). Unsupervised transfer learning for spoken language understanding in intelligent agents. In Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 33, pp. 4959-4966).

Sutskever, I., Vinyals, O., and Le, Q. V. (2014). Sequence to sequence learning withneural networks. In NIPS'2014, arXiv:1409.3215 .

Sutton, R. S., & Barto, A. G. (2018). Reinforcement learning: An introduction. MIT press.

Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., & Fergus, R. (2013). Intriguing properties of neural networks. arXiv preprint arXiv:1312.6199.

Takahashi, S., Chen, Y., & Tanaka-Ishii, K. (2019). Modeling financial time-series with generative adversarial networks. Physica A: Statistical Mechanics and its Applications, 527, 121261.
Taylor, S. (Ed.). (2014). Agent-based modeling and simulation. Springer.

Torrey, L., & Shavlik, J. (2010). Transfer learning. In Handbook of research on machine learning applications and trends: algorithms, methods, and techniques (pp. 242-264). IGI Global.

Treleaven, P., Galas, M., & Lalchand, V. (2013). Algorithmic trading review. Communications of the ACM, 56(11), 76-85.

Trevor Hastie, Robert Tibshirani, and Jerome Friedman (2009). The Elements of Statistical Learning (2nd edition). Springer-Verlag. 763 pages.

Van Liebergen, B. (2017). Machine learning: A revolution in risk management and compliance?. Journal of Financial Transformation, 45, 60-67.

Vargas, M. R., De Lima, B. S., & Evsukoff, A. G. (2017, June). Deep learning for stock market prediction from financial news articles. In 2017 IEEE International Conference on Computational Intelligence and Virtual Environments for Measurement Systems and Applications (CIVEMSA) (pp. 60-65). IEEE.

Vinyals, O., Kaiser, L., Koo, T., Petrov, S., Sutskever, I., and Hinton, G. (2014a).Grammar as a foreign language. Technical report, arXiv:1412.7449.

Wachter, Sandra, Brent Mittelstadt, and Chris Russell. "Counterfactual explanations without opening the black box: Automated decisions and the GDPR." Harvard Journal of Law & Technology 31, no. 2 (2017): 2018.

Wiese, M., Knobloch, R., Korn, R., & Kretschmer, P. (2019a). Quant GANs: Deep Generation of Financial Time Series. arXiv preprint arXiv:1907.06673.
Wiese, Magnus and Bai, Lianjun and Wood, Ben and Buehler, Hans, Deep Hedging: Learning to Simulate Equity Option Markets (October 16, 2019b). Available at SSRN: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3470756

Wood, S. N. (2017). Generalized additive models: an introduction with R. Chapman and Hall/CRC.

Xu, J., & Chen, H. (2005). Criminal network analysis and visualization. Communications of the ACM, 48(6), 100-107.

Xu, K., Ba, J. L., Kiros, R., Cho, K., Courville, A., Salakhutdinov, R., Zemel, R. S., andBengio, Y. (2015). Show, attend and tell: Neural image caption generation with visualattention. In ICML’2015, arXiv:1502.03044.

Yang, W., & Koshiyama, A. S. (2019). Assessing qualitative similarities between financial reporting frameworks using visualization and rules: COREP vs. pillar 3. Intelligent Systems in Accounting, Finance and Management, 26(1), 16-31.

Zhang, Y., & Yang, Q. (2017). A survey on multi-task learning. arXiv preprint arXiv:1707.08114.

Zhang, Z., Zohren, S., & Roberts, S. (2019). Deep Reinforcement Learning for Trading. arXiv preprint arXiv:1911.10107.