**Abstract**

The use of machine learning (ML) to replicate aspects of legal decision making is already well advanced, with various ‘Legal Tech’ applications being used to model litigation risk, and data analytics informing decisions on issues with relevance to law which include probation, predictive policing and credit evaluation. The next step, already being trialled in a number of jurisdictions, will be the use of ML to replicate core functions of legal systems, including adjudication.

We consider the likely consequences of this step using a systemic-evolutionary model of law. From this point of view, many aspects of legal reasoning have algorithmic features which could lend themselves to automation. However, an evolutionary perspective also points to features of legal reasoning which are not consistent with ML: these include the reflexivity of legal knowledge and the incompleteness of legal rules at the point where they encounter the ‘chaotic’ and unstructured data generated by other social sub-systems. We illustrate this point with an example taken from labour law concerning the classification of work relationships.

We argue that the goal of a ‘legal singularity’ which has been advanced by advocates of the use of ML in law is based on a conception of a functionally complete legal system which, while a mirage, has the potential to divert resources to ultimately fruitless uses, while compromising the autonomy of the legal system and undermining its core modes of operation. Finding the institutional means to maintain law’s system-boundary with technology is an urgent task but one whose success cannot be guaranteed, as there is no principle of societal organisation which guarantees the perpetuation of the rule of law, and the democratic-liberal order it maintains, in the face of current technological changes.

* University of Cambridge, respectively: cpm49@cam.ac.uk; s.deakin@cbr.cam.ac.uk.
‘[Mathematics] did not, as they supposed, correspond to an objective structure of reality; it was a method and not a body of truths; with its help we could plot regularities—the occurrence of phenomena in the external world—but not discover why they occurred as they did, or to what end.’

—Isaiah Berlin, ‘Counter Enlightenment’ in Dictionary of the History of Ideas

1. Introduction

One consequence of recent advances in Machine Learning—a family of statistical techniques enabling an algorithm to ‘learn’ over time through the iterative adjustment of mathematical parameters and optimise performance at a task—is renewed interest in applying computation to more aspects of the law and legal processes. Concurrent breakthroughs in Natural Language Processing—a theory-driven subfield of computer science and artificial intelligence (AI) exploring the use of computers to automatically analyse and represent human language—have contributed to the emergence of the so-called ‘Legal Technology’ (LegalTech) industry and development of various tools for use in legal practice and administration. Included within this are those leveraging Big Data and related techniques to forecast the outcome of legal cases—with some demonstrating predictive capabilities greater than human experts. A number of algorithmic decision-making (ADM) systems using ML to simulate aspects of human
reasoning are also used in both public and private-sector contexts. From medicine to finance and immigration to criminal justice, ADM systems have proliferated at a remarkable pace—albeit with sometimes lamentable results.

Because law has language at its core, researchers have long been exploring how to bring AI research to bear on the legal domain, and the cognitive domain of judges and lawyers. Earlier logic-based approaches to AI were used to develop systems for searching legal databases as early as the 1960s-70s, and the advent of Legal Expert Systems (LES) contributed to a swell of optimism for using AI to compliment, extend, and potentially replace the work of human lawyers and judges in the 1970-80s. Thanks to the success of connectionist models and availability of data, recent years have seen a major renewal of interest in the area, with a number of start-ups competing to apply advanced computational techniques to entire areas of law.

While new use cases in law are being identified with each subsequent breakthrough and performance leap, little attention has been paid to how we might assess the fundamental limits of computation in relation to legal reasoning and various decision-making

---


processes. With mounting issues around bias, accountability, and transparency of ADM systems—and rule of law concerns more generally—we pose a question about limits: how do we determine where AI-leveraging systems should be used, where they should be prohibited, and why?

We think these are urgent questions for legal scholars. Proponents frame the totalisation of ADM as inevitable once various technical issues are resolved and more data and methods of better statistical inference become available. But this current AI hype cycle has allowed promises of what’s to come overshadow virtually any democratic deliberation over whether they should be invested in, built, and deployed in the first place. Even when such issues have been tabled—as they were with the EU Commission’s High Level Expert Group on AI—lobbying has been successful in forestalling discussion of regulatory ‘red lines’—contexts in which AI should be strictly prohibited on legal, moral, or humanitarian grounds. Nonetheless, there are compelling cases for prohibiting AI—broadly conceived—in autonomous weapons, facial recognition—and as the French courts have ruled—the use of state legal data for judicial analytics, an important—and potentially lucrative—LegalTech domain.

In light of the disruptive impact of technology on political processes and social discourse in recent years, a lack of meaningful deliberation becomes all the more concerning in light of predictions about a forthcoming ‘legal singularity’. This is a hypothetical point where

---


computational intelligence and decision-making capabilities exceed those of human lawyers, judges and other decision-makers.¹⁷

Lest we forget, a primary justification for ADM systems is the belief they will offer quicker, cheaper, and more reliable decisions untainted by human shortcomings. As physicist Max Tegmark, reasons:

> Since the legal process can be abstractly viewed as computation, inputting information about evidence and laws and outputting a decision, some scholars dream of fully automating it with robojudges: AI systems that tirelessly apply the same high legal standards to every judgment without succumbing to human errors such as bias, fatigue or lack of the latest knowledge.¹⁸

Practical and juridicial reasoning are thus weak links to be severed and reforged with strategic reasoning expressed and imputed computationally. However, the wholesale replacement of juridical reasoning with computation risks undermining one of the principal institutions of a democratic-liberal order. We suggest that it must be treated with all due caution and scepticism. Particularly when one considers that a primary function of the legal system is guarding individuals against the potentially dehumanising dangers of science and technology, and threat of totalitarianism.¹⁹ As Hannah Arendt reminds, ‘the first essential step on the road to total domination is to kill the juridical person.’²⁰

To avoid such an outcome scrutiny must be directed to identifying and establishing limits for the use of AI and other data-driven approaches in replicating core aspects of legal administration and the role of human lawyers and judges. The success of these measures ultimately depends on the law’s capacity to maintain its autonomy in the face of all-encompassing technological change, an outcome which is far from guaranteed. The hypothesis of this paper is that there are limits to the computability of legal reasoning and, hence, the use of AI to replicate the core processes of the legal system. The method we employ is to consider the extent to which there are resemblances between ML and legal decision making as systems involving information retention, adaptive learning, and error correction. Our argument is that while there are certain structural resemblances, there are also critical differences which set limits to the project of replacing juridical reasoning with ADM and ML more generally.

1.1 Methodology

If there are limits to using ADM systems in legal contexts, the next step is identifying and defining them. While there are material and practical limits to computation and data


¹⁸ M Tegmark, Life 3.0: Being Human in the Age of Artificial Intelligence (Allen Lane 2017) 105.


storage,\(^{21}\) and theoretical limits to the computability of problems,\(^{22}\) this does not mean that legal problems are necessarily non-computable. Some might be, but not necessarily all. We thus explore the extent to which the legal system—a system in the sense implied by the theory of social systems\(^{23}\)—is amenable to computation and automation, and how the replacement of juridical reasoning with strategic and computational reasoning might impact the autonomy of the legal system, erode the rule of law, and diminish state authority in structuring and mediating legal relations. We adopt a systemic-evolutionary understanding of law to identify unifying principles that help explain the legal system’s mode of operation with respect to other social sub-systems, including the economy, politics and technology itself, and which help to clarify the role of juridical reasoning in facilitating legal evolution.

Our suggestion is that the hypothetical ‘legal singularity’—which presumes the elimination of all legal uncertainty—conflates simulation and the probabilistic capabilities of ML and Big Data for the process of legal judgement. As Hildebrandt observes, ‘[w]hereas machines may become very good in such simulation, judgement itself is predicated on the contestability of any specific interpretation of legal certainty in the light of the integrity of the legal system—which goes beyond a quasi-mathematical consistency.’\(^{24}\) Following Supiot’s identification of the ‘anthropological function’ of law as a ‘technique [for the] humanization of technology’\(^{25}\) we contend that the replacement of juridical reasoning with computation would ultimately result in an embedding of society in law and the subordination of the ‘rule of law’ to a new ‘rule of technology’.

### 2. Machine Learning (ML)

The recent resurgence in AI research, investment, and applications is primarily driven by the promise posed by a family of computational techniques collectively known as machine learning (ML). Generally, ML ‘involves developing algorithms through statistical analysis of large datasets of historical examples.’\(^{26}\) Through the iterative adjustment of mathematical parameters, data retention, and error correction techniques ML algorithms are said to automatically update (or ‘learn’) through repeated exposure to data and optimise

---


\(^{25}\) A Supiot, Homo Juridicus, 117.

\(^{26}\) D Spiegelhalter, The Art of Statistics: Learning from Data (Pelican, 2019) 144.
performance at various classification, prediction, and decision-making tasks. ML bears more than a ‘family resemblance’ to the practice of ‘data mining’—the ‘study of collecting, processing, analyzing, and making inferences from data.’—as both techniques involve processing large datasets to identify correlations (alternatively referred to as ‘relationships’ or ‘patterns’) between variables.

‘Learning’ in this case refers to the properties of an algorithm—a series of mathematical instructions for transforming an informational input into an output—that allow for the iterative adjustment and dynamic optimisation of parameters upon repeated exposure to example data when directed towards specific tasks (i.e. classifying data or identifying images) without being explicitly programmed. Through a process referred to as ‘training’, an algorithm is repeatedly exposed to data so that over time its performance is optimised at an objective function; a mathematical formula defining the algorithms ‘goal’. ML algorithms are said to ‘learn’ from previous calculations by retaining information and using error correction techniques—such as backpropagation—to produce increasingly stable and repeatable computational models.

Training an algorithm is ultimately a process of trial and error and involves ‘tuning’ various mathematical parameters. Ultimately, it is both an art and a science. To assess performance at an objective function it must be re-trained, re-tuned, and re-assessed multiple times to correct for errors using test data it did not encounter in the course of training. This allows performance to be assessed viz a viz ‘unfamiliar’ data—that is: data not ‘seen’ by the algorithm in the course of training. Performance can be assessed using a variety of strategies. Most often these involve testing it by sectioning or partitioning a dataset; running each iteration of an algorithm on a section of the data and then validating the ‘accuracy’ (as defined by its objective function) of outputs against iterations that were not exposed to a particular section of data. However, these methods are not as reliable as using separate test data.

2.1 Supervised and Unsupervised Learning

27 CC Aggarwal, Data Mining: The Textbook (Springer 2015) 1.


29 S Becker and RS Zemel, ‘Unsupervised Learning with Global Objective Functions’ in A Arbib (ed) The Handbook of Brain Theory and Neural Networks (MIT Press 1997) 997 (‘The main problem in unsupervised learning research is to formulate a performance measure or cost function for the learning, to generate this internal supervisory signal. The cost function is also known as an objective function, since it sets the objective for the learning process.’)

ML algorithms are broadly classifiable as ‘supervised’ or ‘unsupervised’.31 Supervised algorithms are provided with a defined outcome variable representing what should be predicted on the basis of input data. This involves defining a problem by correctly labelling the outcome variable (i.e. this image is dog, that number is 6). Outcome variables can be variously defined, but those defined with binaries (i.e. True/False, Cat/Dog) are referred to ‘classification’ algorithms. In ‘multi-label classification’ an algorithm can predict indicators with more than two classes (i.e. Vehicle/Cyclist/Sign/Tree; Red/Yellow/Green). Algorithms applied to ‘regression’ problems are used to predict a continuous quantitative result in the form of a specific numerical value. Finally, and related to both classification and regression algorithms, are algorithms that help predict ordinal outcomes; involving multiple classes of information that despite not adhering to a continuous number line, have some innate ordering (i.e. First/Second/Third; High/Average/Low).

In supervised ML systems, a programmer defines the desired output. Through training, a program processes input data through its statistical model to produce a an output and dynamically adjusts the weightings applied to each variable (or ‘node’) within the statistical model so that it is ‘tuned’ to produce a desired output. This process of incremental adjustment and re-adjustment can be repeated over thousands or millions of iterations until a model outputs a desired value or one within a defined range. A statistical model is considered ‘trained’ when the relative weighting of neuronal inputs has been adjusted to produce a desired output and it can reliably perform a designated task (i.e. identifying symptoms of breast cancer). However, because a model is constructed and trained on data provided by the designer, the choices they make—such as the composition of the model, section of training data, weighting of inputs—significantly determines how a system functions, how inputs are transformed into outputs, and by extension, the subsequent decision-making process relying on those outputs. This is to say: a supervised algorithm is only as good as the people constructing, tuning, and training it, and the data it has available.

In unsupervised learning an algorithm uses a vector of variables—such as the physical symptoms or characteristics of breast cancer—and a ‘correct’ label for this vector (i.e. a positive diagnosis of breast cancer). This is referred to as a ‘ground truth’. Whereas the goal in supervised learning is to accurately predict a ground truth from specific input variables when only input variables exist, unsupervised learning is not ‘supervised’ by the ground truth. Instead, unsupervised ML systems try to identify a ground truth from a data set or cluster of data points based on their proximity or similarity. For example, an unsupervised ML system might be used to identify a ‘cluster’ of provisions in an employment contract to determine the rights and status of a worker. The clustering of data points can be done as an end in itself or as intermediary step to refine a dataset for a supervised learning approach, where the clusters identified by an unsupervised algorithm could be used as classes to be predicted through supervised learning. While unsupervised algorithms are not without their use in the legal domain, supervised algorithms are of the greatest saliency as they are

driving the most legally consequential decisions in contexts such as risk assessment, immigration, predictive policing, credit scoring, and so forth.

2.2 Artificial Neural Networks (ANNs)

A primary tool of ML are so-called artificial neural networks (ANNs). ANNs are computational models that use a simplified understanding of how the human brain learns through the use of essentially statistical models. These models generally consist of: 1) an input layer or ‘neurons’ that receive information, 2) a hidden layer consisting of equations to transform inputs, and 3) ‘synapses’ that link neurons together by transferring the outputs of one neuron to form the inputs of another. Neurons are arranged in ‘layers’ that effectively divide the model into different stages of calculation through which data passes, with synapses typically weighted so their values are modified by a multiplier. Models can involve several layers each consisting of multiple neurons, with neurons in each layer linked to the neurons in the previous and subsequent layers by synapses.

---


Inputs (which may consist of significant quantities of data) are provided to the first layer of neurons, which calculate these inputs and then pass them on as outputs to the next layer through synapses. The neurons in the subsequent layer in turn perform calculations and produce an output. Each layer repeats this process until the final output of the model is produced. Depending on the nature and complexity of the task, a programmer will choose the number of layers, the number of neurons in each layer, and the equation used to calculate the weighting and behaviour of data between the neurons.

2.3 Deep Learning (DL)

DL is a subset of ML, and is the technique largely responsible for some of the most high-profile breakthroughs in modern AI research.\footnote{Y LeCun, Y Bengio and G Hinton, ‘Deep Learning’ (2015) 521 Nature 436.; D Silver, T Hubert, J Schrittwieser, et al., ‘AlphaZero: Shedding new light on the grand games of chess, shogi and Go’ <https://deepmind.com/blog/alphazero-shedding-new-light-grand-games-chess-shogi-and-go/>]. DL approaches essentially involve large ANNs where ‘depth’ is determined by the number of hidden layers and neurons within them. Generally, DL uses both supervised and unsupervised approaches to predict an output from a set of input variables. This is done by assigning a numerical weight to each connection between neurons determining the importance of an input value in the overall model. For instance, a DL algorithm designed to predict the price of airline tickets would assign a higher weight to a factor such as the departure date/time given their importance as variables in determining cost. Through training, the use of an error correction algorithm called backpropogation allows a process called gradient descent to dynamically adjust the mapping of connections between neurons so that any given input is correctly mapped to the corresponding output.\footnote{I Goodfellow, Y Bengio, and A Courville, Deep Learning, 19; cf. DE Rumelhart, G Hinton and JW Williams, ‘Learning representations by back-propagating errors.} Training a DL system is an intensive process due to the vast amounts of data required to produce valid models, but the development of the backpropogation algorithm has markedly improved training efficiency.

Most DL approaches also employ a technique known as convolution.\footnote{Y LeCun, ‘Generalization and network design strategies’ (1989) Technical Report CRG-TR-89-4 <https://yann.lecun.com/exdb/publis/pdf/lecun-89.pdf>.} This allows neuronal connections within a network to be constrained so that they can capture a property referred
to as translational invariance.\textsuperscript{41} In tasks involving image recognition, this allows a system to recognise a specific object and maintain recognition when its appearance varies in some way in subsequent images. For instance, a traffic light in one image can be presumed to be the same object in a subsequent one, without direct experience of it. This is, of course, a critical issue for computer vision systems, particularly those embedded in autonomous vehicles where recognising phenomena in the real-world is paramount for safety.\textsuperscript{42} Due to their strengths in local connectivity, weight sharing, and pooling, CNNs are also particularly helpful in speech recognition tasks where invariance is highly desired.\textsuperscript{43}

**Figure 2:** Examples of Translation Invariance

### 2.4 Deep Learning and Natural Language Processing

Another promising domain for DL is Natural Language Processing (NLP). From a scientific perspective, NLP models the cognitive dimensions of how natural language is understood and produced by humans. From an engineering perspective, NLP entails developing practical applications for facilitating interactions between computers and human languages. Bates observes:

---


NLP has evolved to incorporate aspects of many other disciplines (such as artificial intelligence, computer science, and lexicography). Yet it continues to be the Holy Grail of those who try to make computers deal intelligently with one of the most complex characteristics of human beings: language.\(^4^4\)

Common NLP applications include speech recognition, lexical analysis, machine translation, information retrieval and question answering, among others.\(^4^5\) Natural language can be understood as a system for conveying meaning or semantics and is by nature a symbolic or discrete system.\(^4^6\) The observable aspect of language, text, is a physical signal that exists in purely symbolic form. Text has a counterpart in the ‘speech’ signal, which is reducible to the continuous correspondence of symbolic text. Together they share the same linguistic hierarchy of natural language.

NLP has been a facet of AI research since the 1950s, when machine translation was the primary focus of researchers but ‘abandoned when it was discovered that, although it was easy to get computers to map one word string to another, the problem of translating between one natural language and another was much too complex to be expressible as such a mapping’.\(^4^7\) The most familiar example of this is the ‘Turing Test’ which used natural language exchanges between a human and a computer designed to generate human-like responses.\(^4^8\) Deng and Liu indicate that NLP research has gone through three distinct waves: 1) rationalism; 2) empiricism; and 3) deep learning.

### 2.4.1 Rationalism and NLP

The first of these, rationalism, dates from the 1960s-1980s. During this time research was primarily oriented around having machines ‘understand’ and respond to questions. To accomplish this, researchers preferred a ‘knowledge-based’ approach that involved devising techniques to capture and encode human expert knowledge into a queryable database. The enterprise was guided the idea that knowledge of language in the human mind was pre-determined by generic inheritance.\(^4^9\) Rationalist approaches were largely derived from the work of Noam Chomsky on the innateness of language and grammatical structure and

---


\(^4^5\) T Young, D Hazarika, S Poria, and E Cambria, ‘Recent Trends in Deep Learning Based Natural Language Processing’, 8-10.


\(^4^8\) A Turing, ‘Computing machinery and intelligence’ (1950) 14 Mind.

\(^4^9\) L Deng and Y Liu, ‘A Joint Introduction to Natural Language Processing and to Deep Learning’ in L Deng and Y Liu (eds) *Deep Learning in Natural Language Processing*. 
his rejection of N-grams.\textsuperscript{50} Believing that core aspects of language were biologically encoded into the brain at birth through genetic inheritance, rationalist approaches to NLP attempted to design hard-coded rules for capturing human knowledge and incorporating reasoning mechanisms into NLP systems using various techniques borrowed from the emerging field AI research.\textsuperscript{51}

This first wave of NLP systems had particular strengths in transparency and interpretability but were limited in their capacity to perform logical reasoning tasks. Following the approach of so-called Expert Systems, NLP systems such as ELIZA or MARGIE relied upon hard coded rules and codifying human expertise.\textsuperscript{52} This meant they they were only useful for narrow applications and could not cope with uncertainty enough to be useful in practical applications.\textsuperscript{53} Instead, most NLP applications during this time relied upon symbolic rules and templates using various grammatical and ontological constructs. When this approached worked, it worked quite well. However, successes were few and far between and there were major difficulties to their use in practical contexts.\textsuperscript{54}

\subsection*{2.4.2 Empiricism and NLP}

The second wave, referred to as empiricism, relies upon data sets and early ML and statistical techniques to make sense of it. Armstrong-Warwick observes that empirical NLP methods offer several advantages over rationalist approaches, such as 1) acquisition—the ability to identify and encode relevant domain knowledge, 2) coverage—incorporating all the phenomena of a domain or application, 3) robustness—combining (often ‘noisy’) empirical data and factors not explicitly accounted for in the underlying NLP model, and 4) extensibility—the ability to extend an application from one domain or task to another.\textsuperscript{55}

While prevalent between the 1920s and 1960s, the ICT revolution made large tracts of machine-readable data available, driven by a steady increase in computing power, and better, faster, and cheaper components—driven by the relentless technological selection enshrined by Moore’s Law.\textsuperscript{56} The exponential growth of digital information—which by one


\textsuperscript{52} H Shah, K Warwick, J Vallerdú and D Wu, ‘Can machines talk? Comparison of Eliza and modern dialogue systems’ (2016) 58 Computers in Human Behavior 278,


\textsuperscript{54} E Brill and RJ Mooney, ‘Empirical Natural Language Processing’, 13-18.

\textsuperscript{55} E Brill and RJ Mooney, ‘Empirical Natural Language Processing’, 16.

estimate will total 175 zettabytes worldwide by 2025—has meant that data-hungry empirical approaches have dominated NLP research since the 1990s. In contrast to rationalist approaches, which assumed language was a genetic inheritance, empirical approaches presumed that the human brain begins with only rudimentary capacity for association, pattern recognition and generalisation. To learn the complex structure of natural language the mind was thought to require a stream of rich sensory data. Early empirical NLP approaches used generative models, such as the Hidden Markov Models (HMM), to create stochastic models describing a sequence of probable events where the probability of an event depends on the state attained by a previous event. Among many other uses, these HMM models remain remarkably useful for determining the structure of natural languages from large data sets, and developing various probabilistic language models.

Generally, ML based approaches to NLP perform much better than earlier knowledge-based ones. Virtually every successful NLP application, including speech recognition, handwriting recognition, and machine translation are attributable to empiricist approaches. Whereas recent developments have yielded major performance increases in translation quality, NLP is not yet nearly as pervasive in real-world deployment as many hope it can be.

---


59 cf. EA Feinberg and A Shwartz (eds) Handbook of Markov Decision Processes: Methods and Applications (Springer 2002) In the 1948 A Mathematical Theory of Communication, Claude Shannon for all intents founded information theory and by doing so revolutionised the telecommunications industry and laid the groundwork for the Information Age. Therein Shannon proposed using a Markov chain to create a statistical model of the sequences of letters in a piece of English text. They were remarkably successful. Markov chains are now widely used in speech recognition, handwriting recognition, information retrieval, data compression, and spam filtering among many other uses. Markov models and decision processes also numerous scientific applications including: the genemark algorithm for gene prediction, the Metropolis algorithm for measuring thermodynamical properties, and Google’s PageRank algorithm for Web search.


2.4.3 Deep Learning and NLP

While NLP applications such as speech recognition, language understanding and machine translations developed out of this second wave—and generally performed much better than those in the first—they fell far short of human-level performance. The ML models used were insufficient for dealing with large sets of a training data and algorithmic design, method and implementation were lacking. This, however, changed dramatically a few years ago with the third wave of NLP under the deep learning paradigm. As discussed above, ML requires human programmers to define various features, as such, feature engineering is a bottleneck that requires significant human expertise. Moreover, comparatively ‘shallow’ ANNs lack the capacity to produce decomposable abstractions and ontologies that allow for the automatic disentangling of complex language structures.

The sophistication of current DL systems enables the learning of concepts at higher-levels of abstraction by building them out of lower-level representations. This has led to the development of so-called Deep belief networks—a form of ANN that can generate connections between layers without needing to do so between neurons in the same layer—and Convolutional Neural Networks (CNNs) which replicate the organisation of the animal visual cortex to produce vast increases in image classification and recognition. However, the overarching promise of DL is the ability to discover intricate patterns and correlation in high-dimensional data. This has enabled the development of applications useful in real-world tasks, perhaps most notably speech recognition, image recognition and machine translation.

2.5 Limitations of Machine and Deep Learning

Successive studies have begun to probe the inherent limitations and discriminatory effects of deploying ADM to various social contexts. From concerns about the opacity and explainability of ML algorithms, to flawed selection of data, to technical and legal strategies to ensure the trustworthiness, accountability, and compliance of automated systems, a clear picture has emerged as to the dangers of relying on what Wigner termed the ‘unreasonable

---

effectiveness of mathematics in the natural sciences’. But in addition to immediate concerns looms much bigger and more entangled questions about the epistemic and practical viability of using AI and Big Data to replicate core aspects and processes of the legal system, if not the cognitive domain of lawyers or judges altogether.

If we are to assess claims of a forthcoming ‘legal singularity’ we must first ask basic questions: what is AI good at?, what is AI bad at?, and to what does it mean to claim that ML (in current and aspirational forms) can replicate core processes of the legal system, if not human reasoning and authority altogether? Contrary to the assumptions of empiricists, Kuhn argued that reality is not directly accessible to human observers as ‘facts’ that can be recorded and mathematically formalised. While we may access reality as limited by our senses, we cannot do so without the interference of ‘meaning making’. What we can sense, through a camera lens, for instance, already has meaning, and meaning is not a function of sensory perception. Sensory perception is not possible without biological cognition that allows an observer to observe.

Data science, on the other hand, does not simply record facts about the world. It transforms data in ways that can be apprehended by the human senses. The same applies to AI and ML applications using vast data sets. Efforts to formalise legal knowledge into mathematical axioms and transform juridical reasoning into something that can be modelled echo the Neo-platonism of the early scientific era and revive the Leibnizian assumption that there exists a hidden mathematical order underlying the structure of reality and human cognition. As a subsequent section of this paper explores, it is now presumed that mathematical formalisation is not just possible, but that strategic reasoning expressed via computation should be considered ontologically superior to inherently faulty practical reasoning expressed through natural language categories.

Understanding the assumptions of the new normative order posed by AI and Big Data will help legal scholars better understand not just the consequences, but their role in expediting or mitigating the ‘legal singularity’. However, we must first consider some of the inherent limitations of AI and ML, particularly as they relate to the task of formalising legal reasoning and imputed legal consequences from algorithmic processes.

### 2.5.1 Finite Data

Data is the ‘lifeblood’ of AI research and a major bottleneck for the development of new applications. Despite an ongoing lack of labelled data, compelling and effective use cases for ML continue to be found in a variety of real-world contexts. The use cases for DL, on the

---


other hand, have been slow coming, leading some to question its long term viability.\textsuperscript{68} There are several reasons for this, but the primary reason is the data intensive nature of DL and sheer scale of data required to train and produce valid models.\textsuperscript{69} It is for this very reason that the entities prioritising AI research are engaging in elaborate multiyear games to overcome the data acquisition problem and compete for the data they need.\textsuperscript{70} Because DL systems must generalise solutions beyond their training data (i.e. pronouncing a new word or identifying an unseen image) data availability limits algorithmic performance such that it cannot guarantee high-quality solutions.

Generalisation takes place by either interpolation of known (i.e. correctly identified) examples or extrapolation which requires an algorithm generalising beyond the examples in training data. For a DL network to accurately generalise, it must, at least in most cases, have access to a vast library of data with test data similar to that of its training data so that new solutions can be interpolated from previous ones. In the paper credited with re-igniting interest in the viability and practicality of DL applications, the landmark performance results were achieved using a nine layer Convolutional Neural Network (CNN) with 60 million parameters and 650 000 nodes trained on approximately a million images from approximately 1000 categories. While the brute force approach to image recognition yielded impressive results, it did so in the large but nonetheless finite context of the ImageNet database.\textsuperscript{71} Moreover, DL works particularly well in stable domains where training exemplars can be mapped onto finite or limited categories, but struggles in tasks requiring open-ended inference, examined in greater detail below.

2.5.2 Data Intensiveness

Humans are able to learn abstract relationships with only a few trials. For instance, if a drotch was defined as a brother between the ages of 15 and 30, perhaps using an example, a human could easily infer whether they had any drotches or if anyone they knew did. Using an explicit definition a human does not require potentially millions of training examples to generalise and abstract out what a drotch is. Rather, the capacity to infer abstract relationships between algebraic variables (male/age range) through explicit and implicit heuristics is a tacit and innate quality of biological intelligence.


\textsuperscript{69} A Woodie, ‘Deep Learning is Great, But Use Cases Remain Narrow’ (Datanami, 3 October 2018) <https://www.datanami.com/2018/10/03/deep-learning-is-great-but-use-cases-remain-narrow>.


Psychological researchers observe that the capacity for inference and abstraction is seen in 7-month old toddlers who can learn language rules from a limited number of labelled examples in under two minutes. At present, DL approaches cannot learn abstract relationships through explicit verbal definitions, and instead work best (if at all) when trained using millions or even billions of training examples; best evidenced by Deep Mind’s success with video games and Go. As successive researchers demonstrate, humans are far more efficient at learning complex rules and generalising abstraction relationships than DL systems. This has not been lost on DLs staunchest proponents, leading some to question whether CNNs dependence on large labelled might and difficulty generalising novel viewpoints might ‘lead to their demise.’

2.5.3 Transfer Learning

While DL has yielded impressive results in computationally intensive domains, the word ‘deep’ refers to a technical property (i.e. the number of hidden or convolutional layers) and does not imply conceptual depth or sophistication. For instance, a DL network thousands of layers deep cannot induce understanding or generalise abstract and subjective concepts such as ‘fairness’, ‘justice’ or ‘employee’. Even more mundane concepts can elude understanding. In his analysis of Deep Mind’s success using reinforcement learning methods to conquer Atari video games, Marcus critiques the suggestion that the program has ‘learned’ anything, but rather ‘doesn’t really understand what a tunnel, or what a wall is; it has just learned specific contingencies for particular scenarios. The systems represents ‘knowledge’ through converting raw informational data into inputs that allow a game to be ‘played’ but there is no player as such.

Transfer tests—where a deep reinforcement learning system is confronted with scenarios that differ in minor ways from those it was trained on—show that DL solutions are often extremely superficial. Marcus argues that it is misleading to suggest that reinforcement learning enables a program to induce a semantic understanding of the narrow computational environment: ‘It’s not that the Atari system genuinely learned a concept of

---

wall that was robust but rather the system superficially approximated breaking through walls within a narrow set of highly trained circumstances.\textsuperscript{77}

The superficiality of DLs ‘learning’ is particularly evident in natural language contexts. For instance, research by Jia and Liang trained artificial neural networks (ANNs) on the Stanford Question Answering Database (SQuAD) were the goal was to highlight words in a passage corresponding to a given question. In one example, the system was correctly trained to identify John Elway as the winning quarterback of Super Bowl XXXIII based on a short summary of the game. However, the inclusion of red herring sentences—such as a fictional one about Google AI researcher Jeff Dean winning another ‘bowl’ game—led to a major drop off in accuracy from 75% to 36%.\textsuperscript{78}

\subsection*{2.5.4 Non-Hierarchical Language}

Most language-based DL models represent sentences as strings of words. This contrasts with the hierarchical view of language proposed by Chomsky where language is ordered into four classes by its complexity and larger linguistic structures are recursively constructed out of smaller components.\textsuperscript{79} Earlier research cast doubts that even the most sophisticated neural networks could systematically represent and extend the recursive structure of language to new and unfamiliar sentences.\textsuperscript{80} More recently, however, Lake and Baroni conclude that neural networks are ‘still not systematic after all these years’ and that they could ‘generalize well when the differences between training and test…are small [but] when generalization requires systematic compositional skills, [they] fail spectacularly.’\textsuperscript{81}

While similar difficulties are likely to be encountered in other domains, the inability to deal with complex hierarchical language structures presents specific challenges within the domain of law, where systems will inevitably encounter novel fact patterns requiring generalising abstract relationships from agonistic accounts of underlying ‘facts’. As Marcus observes:

\begin{quote}
The core problem, at least at present, is that deep learning learns correlations between sets of features that are themselves ‘flat’ or non-hierarchical, as if in a simple,
\end{quote}

\begin{thebibliography}{99}
\end{thebibliography}
unstructured list, with every feature on equal footing. Hierarchical structure (e.g., syntactic trees that distinguish between main clauses and embedded clauses in a sentence) are not inherently or directly represented in such systems, and as a result deep learning systems are forced to use a variety of proxies that are ultimately inadequate, such as the sequential position of a word presented in a sequences.  

Whereas people learn and acquire many different types of tacit and innate knowledge from diverse experiences over many years, in most cases becoming better learners over time, DL systems are comparatively narrow and able to learn only a single function or data model from statistical analysis of a single data set. Although progress has been made in representing words in the form of vectors and complete sentences in ways that are compatible with DL techniques, they remain limited in the ability to reliably represent and generalise rich semantic structure, such as that found in legal judgements.

A promising solution is, however, posed by the never-ending language learning (NELL) paradigm developed by researchers at Carnegie Mellon University. Since 2010, Carnegie Mellon researchers have programmed NELL to run around the clock to identify fundamental semantic relations between hundreds of data categories such as cities, corporations, emotions and sports team. This has involved NELL processing millions of web pages for connections between what it is has already ‘learned’ and what it finds through an exhaustive search process. The goal of the Carnegie Mellon team is to have NELL draw inferential connections and discern a hierarchical linguistic structure to deduce subsequent connections so that it can answer natural language questions with no human intervention.

2.5.5 Open-Ended Inference

Consider the sentences: ‘Adam promised Eve to stop’ and ‘Adam promised to stop Eve’. Without the ability to draw semantic inferences about who is stopping who, the meaning of these sentences can dramatically diverge. While ML systems have proved adept at machine reading tasks such as SQuAd—where the answers are specified in the text being read—they have been far less successful at tasks requiring inference beyond those made explicit in a text. Common problems encountered include the combining of multiple sentences into a single semantic string (multi-hop inference) or combining explicit sentences with information not included in a particular text selection. In contrast, when humans read a text

---


they can automatically draw a variety of novel and implicit inferences, such as the ability to
derive a fictional characters intentions from indirect dialogue.87

3. Law as Algorithm: Exploring the Limits of ‘Computable Law’

The next step in our analysis is to consider how far legal reasoning can be understood in
terms equivalent to those developed for use in the domain of ML. In so far as legal
reasoning can be broken down into a series of algorithmic processes similar to those
employed in ML, the case for applying ML to law is to that extent enhanced. Conversely,
identifying features of legal reasoning which are not captured by an algorithmic logic can
help us understand the limits to ML in legal contexts. Understanding those limits will also
enable us to more clearly identify potential social harms resulting from the over-extensive
application of ML in the legal sphere.

Our analysis proceeds as follows. We first of all consider the degree to which legal
reasoning contains elements of information retention and error correction which are the
approximate equivalent of those used in ML. Then we look more closely at how legal
reasoning involves processes of data partitioning and weighting which are analogous to
those used in ML. We identify similarities between legal reasoning and ML, but also some
critical differences. In particular, we question the applicability to the legal context of the
idea of ‘optimisation’. The ‘learning’ involved in legal reasoning, we suggest, a process of
non-linear adjustment between the legal system and its external context which is not
adequately captured by the optimisation function of ML.

3.1 Classification, Information Retention and Error Correction in Legal
Reasoning: The Case of Employment Status

As we have seen,88 ‘learning’ in the context of ML refers to the iterative adjustment and
dynamic optimisation of an algorithm, understood a process for correctly mapping inputs
into outputs in response to repeated exposure to data. Success is measured in terms of the
algorithm’s ability to predict or identify a specified outcome or range (a ‘ground rule’) from
a given vector of variables. In order to ‘learn’, in the sense of becoming more accurate in its
predictions, the algorithm must be able to retain information from previous iterations (or
‘states’), while correcting errors. The statistical techniques used to tune and refine an
algorithm’s performance in this way include the partitioning of data into conceptual
categories and the assignment of differential weights to those categories in order to
establish their relative importance in the process of identification and prediction.89

88 Above, 2.1-2.3.
89 For a more thorough overview of the training and tuning process cf. D Lehr and P Ohm, ‘Playing
3.1.1 Classifying Employment Relationships

To see how the legal system might be thought of as containing analogous mechanisms of information retention and error correction, we will use as an illustration the way in which courts and tribunals perform the task of classifying work relationships. This is a foundational issue in a number of areas of law, including employment law, tax law and tort law, and is one found in a functionally similar form (albeit with some important cross-national differences as well as variations across different areas of law) in virtually all legal systems today.\footnote{The literature on this question is huge. Here we draw principally on the historical analysis of the evolution of labour classifications in English law set out in S Deakin and F Wilkinson, The Law of the Labour Market: Industrialization, Employment and Legal Evolution (Oxford University Press 2005).}

The equivalent of the outcome or ‘ground rule’ here is the classification of an individual supplier of labour as either an ‘employee’, indicating one particular set of legal rights and obligations, or an ‘independent contractor’ or ‘self-employed’ person, indicating another. The terms used in various legal systems differ, but the basic classifications, and their normative functions, are remarkably consistent across national regimes.\footnote{On the degree of functional continuity across systems, notwithstanding differences in terminology, cf. S Deakin, ‘The comparative evolution of the employment relationship’ in G Davidov and B Langille (eds) Boundaries and Frontiers of Labour Law (Hart 2006).} ‘Employee’ status signifies that the individual supplier of labour is under a duty to obey orders of the counter-party, here signified as the ‘employer’, in return to receiving certain protections against work-related and labour market risks. An ‘independent contractor’ supplying their labour to a ‘client’ is in a different type of relationship: in essence, they have more autonomy over how they work but acquire fewer legal rights against their contractual counter-party. The outcome here is binary and so, at least in principle, is capable of being captured in mathematical form using a pairwise code \((1, 0)\).\footnote{B Alarie, A Niblett and A Yoon, ‘Using machine learning to predict outcomes in tax law’.} In this sense it is like very many legal determinations, which can be expressed as binary alternatives, such as (liable, not liable), and so on.

Classification systems, which appear more complex, can be reduced to binary outcomes by narrowing the focus of the decision-making process, or in other words: making it more fact-intensive. Thus UK labour law currently knows not just one, but three basic categories of work relation: the employee, the self-employed or independent contractor and, somewhere in between them, an intermediate category known as a ‘limb (b) worker’ which has some, but not all, of the characteristics of each of the other two.\footnote{Employment Rights Act 1996, s. 230(3)(b). For further detail cf. S Deakin and G Morris Labour Law (6th rev ed. Hart 2012) ch 3; J Prassl, ‘Pimlico Plumbers, Uber Drivers, Cycle Couriers, and Court Translators: Who is a Worker?’ (2017) Oxford Legal Studies Research Paper No. 25/2017.}

On closer inspection the existence of this third category does not detract from the essentially binary nature of the classification process. It just means that the court goes down
one further layer of analysis in the sense of conducting an additional factual inquiry. After answering the question of whether the individual is an ‘employee’ or ‘self-employed’ in favour of the latter category, it will then undertake a further review of the facts to see whether the individual is a ‘limb (b) worker’, in which case they will acquire (among other things) the right to the minimum wage, or not, in which case they will not be accorded this legal right, among others. In effect, the set (limb (b) worker) is a set located entirely within the wider set (self-employed).

This example illustrates a further feature of legal categories, which is that they are arranged hierarchically. ‘Higher level’ categories, meaning those defined at a higher level of abstraction or generality, contain within them ‘lower-level’ or more fact-specific classifications. The process of applying a concept to a given fact situation can be understood as one of moving from the general to the specific, or the abstract to the factual. The process continues until the scope for conceptual classification is exhausted and the court is left with the unstructured empirical data of the facts presented to it.

The point at which the law reaches the limits of its discursive capabilities demarcates its boundary with other systems, such as the economy, politics, or technology. Beyond the boundary of juridical analysis, as far as the legal system is concerned, data are unstructured or ‘chaotic’. With respect to such variables, law is unavoidably incomplete. There is no unique ‘right answer’ to cases of classification which involve novel or untested points. When a new type of social or economic relationship comes before the courts for classification (as in the case, currently, of ‘gig’ or ‘platform’ work), rather than a clear solution appearing immediately, what tends to happen instead is that different courts propose a number of alternative solutions which are then tested against each other, and selected in or out as the case may be, through external pressures arising from litigation and lobbying, as well as from academic legal commentary. Eventually, data which were initially unstructured or ‘chaotic’


96 We draw here on Niklas Luhmann’s concept of coevolving social subsystems, cf. N Luhmann, Theory of Society: Volumes I and II.


are translated into the conceptual language of the law and thereby absorbed into its processes.\footnote{N Luhmann, Law as a Social System (Oxford University Press 2004) 250.} Conceptual stability is restored, if only provisionally, through a ruling of a higher appellate court, or a legislative clarification or modification of the relevant rule. Law remains bounded with respect to its environment, even if the ‘shape’ of this boundary is altered as a result of the selective pressures exerted by other systems.\footnote{On law’s boundary, cf. A Morrison, ‘The law is a fractal: the attempt to anticipate everything’ (2013) 44 Loyola University Chicago Law Journal 649; D Post and M Eisen, ‘How long is the coastline of the law? Thoughts on the fractal nature of legal systems’ (2000) 29 Journal of Legal Studies 545.}

We might next ask: in what sense does the allocation of legal status to particular types of work relationship involve a process of learning functionally similar to that involved in ML as described above? The first point of similarity to note is that with classification problems, courts are applying tests which are essentially algorithmic in nature, since they consist of an ordered series of decisions which will result in a particular output for a given set of inputs. Thus according to the ‘control’ test, for example, which is widely used in labour law systems in some form or another to demarcate employee status, the individual will be (or least is more likely to be) an employee if they are under the command or subordination of another as to how they should do their work.\footnote{Yewens v Noakes (1880) 6 QBD 530; cf. S Deakin and F Wilkinson, The Law of the Labour Market, 91.} The control test is a verbal algorithm used to reconcile ‘data’ drawn from the facts presented to the court with an ‘outcome’ which is a legal determination of the parties’ rights.

\section*{3.1.2 Information Retention in Legal Reasoning}

The next point of similarity is that the use of abstract categories involves a form of information retention.\footnote{S Deakin, ‘Juridical Ontology’.} Legal concepts condense and hence retain information in two senses. The first is a function of concepts being hierarchically ordered in the sense we have just identified, and that implied by the Chomsky Hierarchy of formal grammars. Since higher order concepts can be decomposed into lower-order ones which become more fact-specific at each descending level of definition, a general category such as ‘employee’ becomes a shorthand form of expression for information embedded in categories operating at lower levels.\footnote{S Deakin and GS Morris, Labour Law (6th ed, Hart 2012) 131-257.} Thus the control test groups together a range of more precisely defined indicators of employee status such as having a supervisor or boss, being paid a wage, being subject to disciplinary control, paying tax at source, and so on.\footnote{S Deakin and GS Morris, Labour Law (6th ed, Hart 2012) 131-257.} At the very final (lowest) level of analysis, when conceptual reasoning is exhausted, only the facts of individual disputes remain: this is the systemic boundary just referred to. The capacity of the law to order
external ‘factual’ data into juridical categories of varying degrees of specificity allows for its internal processing as legal material. As this conceptual ordering takes place, information of a detailed kind can be stored within linguistic categories of varying degrees of abstraction.\textsuperscript{106}

The second sense in which legal concepts embed and retain information concerns their inter-temporal effect.\textsuperscript{107} The processing of external information does not have to be repeated every single time a new case falls to be decided. The information ‘learned’ by the system is retained there through the persistence of conceptual forms across time.\textsuperscript{108} The legal system’s meta-norm of precedent—’like cases should be decided alike’—constrains a later court to adapt its reasoning to the linguistic categories used in earlier iterations. While in one sense constraining, the norm of precedent is also facilitative, as it ensures that information on these past iterations is retained for future use: information retention over time.

### 3.1.3 Error Correction in Legal Reasoning

Error correction can also be observed in how courts and judges deal with cases. We can see an error correction occur in the capacity of legal classifications to adjust to new data inputs over time. The doctrine of precedent notwithstanding, the categories used to determine employee status are dynamic, not static. They are continuously being adjusted in the light of new fact situations coming before the courts for resolution. Thus the ‘control’ test no longer (since the 1940s) places a primary emphasis on ‘personal’ subordination as the essence of employee status (as it did, for example, in the 1880s) but stresses instead the individual’s incorporation in, and subjection to, procedures and processes of a more bureaucratic and depersonalised nature, reflecting a change in the way that managerial functions are performed in practice within firms and organisations.\textsuperscript{109}

The importance attributed to ‘control’ itself as a guide to employee status is not fixed: it can be eclipsed by other tests which are seen as more useful or appropriate for their time, such as ‘integration’ at the point when vertically integrated firms and public service delivery organisations were the norm, ‘economic reality’ when the capacity of the post-1945 welfare state to deliver protection against labour market risks was at its height, and ‘mutuality of obligation’ at the point where the labour market was becoming more ‘flexible’ as a result of

\textsuperscript{106} N Luhmann, \textit{Law as a Social System}, 257.


\textsuperscript{108} N Luhmann, \textit{Law as a Social System}, 255.

a combination of political and technological changes. Even the foundational concept of the ‘employee’ is not as stable as it looks: it is in a line of descent from earlier juridical categories such as ‘artificer’, ‘servant’ and ‘workman’ which served somewhat different classificatory purposes from the contemporary ‘employee’, purposes which reflected the technological and political conditions of previous phases of industrialisation.

The updating of legal categories is an incremental process which occurs through the experimental testing of concepts against new fact situations as they come before the courts for decision. Incremental as the process is, its effects, over a sufficient period of time, can be radically transformative. Existing linguistic categories can be remoulded, given entirely new meanings, or even abandoned in favour of entirely new typologies. The process can be understood in an evolutionary sense as a dynamic adjustment of the legal system to its economic, political and, especially relevant for this discussion, technological context. It is a version of the variation-selection-retention algorithm which explains the survival and persistence of approximately functional features of a system’s operation in a context where that system is subject to selective pressures from its environment. In the context of legal evolution, the element of variation is supplied by experimentation in the stock of judicial decisions as different courts arrive at diverse outcomes when faced with novel fact situations; selection, by pressures to challenge or alter existing rules through litigation and legislation; and retention, by the meta-rule or doctrine of precedent which ensures continuity at the point of change by requiring courts to justify innovations as extrapolations from, or adjustments of, existing modes of reasoning.

3.2 Data Partitioning and Weighting in Legal Reasoning

We saw above that one of the techniques used by ML to achieve error correction is to partition input data into categories which are connected to one another through ‘mapping equations’. These equations have the effect of treating the output of one category as the input to another. The categories are arranged in hierarchical layers so that the model is divided into different stages of calculation. Error correction is achieved by assigning different numerical weights to the different inputs and outputs to achieve a desired result. In DL models ‘backpropagation’ allows for the more efficient dynamical mapping of connections. The distinctive aspect of DL models is that solutions can emerge through trial and error as different weightings are tested. Thus in the case of the algorithm for pricing


112 N Luhmann, Law as a Social System. 255.

113 2.2 above.
airline tickets cited above,\textsuperscript{114} the precise weight to be attributed to departure date as against other input variables cannot be known in advance, but can be identified through repeated iterations of the mapping process, as different weights are tried out and ‘errors’ progressively minimised.

Data partitioning and weighting have equivalents in legal reasoning. Data partitioning is a feature of the tendency, which we have already observed (3.1), to bundle fact situations into conceptual categories expressed as abstractions at different levels of generality. Thus ‘employee’ (at a higher level), ‘control’, ‘integration’, ‘economic reality’ and ‘mutuality of obligation’ (at an intermediate level) and ‘mode of payment’, ‘ownership of equipment’, ‘paying tax at source’, ‘given orders’, ‘receives regular work’, and so on (at a lower level), are categories to which data from cases (detailed factual descriptions of working relationships drawn from the parties' pleadings and submissions) are assigned. ‘Mapping’ occurs when lower level indicators are assigned to higher level categories, so that, for example, ‘given orders’ is assigned to the ‘control’ concept while ‘receives regular work’ is assigned to the ‘mutuality of obligation’ concept.

The process also works in the other direction: higher level concepts inform the selection and identification of lower level ones.\textsuperscript{115} Thus once an intermediate-level concept such as ‘mutuality of obligation’ becomes established as a relevant test of employee status (as it did under conditions of an increasingly flexible labour market in the course of the 1980s), lower-level categories which were previously of minimal importance in the law (such as ‘receives regular work’) achieve a heightened relevance (or ‘weight’) within legal reasoning.

Legal reasoning also involves ‘weighting’ in the sense of an experimental and iterative adjustment of the relative importance of inputs in deciding cases. In the context of the rise of the ‘mutuality of obligation’ test in the 1980s, the additional importance accorded to the ‘given regular work’ indicator led to the eclipse of other, previously well-established factors for determining employee status, in particular ‘given orders’. While it would be artificial to regard these different factors being assigned precise numerical weights in the way that occurs in ML, it is not artificial, we suggest, to see their relative importance in judicial decision making being adjusted over time as the courts dealt with a growing number of cases involving ‘precarious’ or insecure work, which tested the established boundaries of the employee concept.\textsuperscript{116}

\begin{itemize}
\item \textsuperscript{114} 2.3 above.
\item \textsuperscript{116} S Deakin and F Wilkinson, The Law of the Labour Market, 309.
\end{itemize}
More generally, it would seem that ‘weighting’ (assigning degrees of importance to indicators) is a near-universal feature of legal reasoning, just as ‘data partitioning’ (allocating facts to concepts) is. Complex fact situations of the kind which do not admit of an easy resolution are the most likely to be litigated. These are precisely the cases in which courts must make adjustments of the kind implied by the ‘weighting’ analogy. For the English courts of the 1980s to decide that a casual worker, despite being subject to a duty to obey orders while working and economically dependent on a particular user of labour, was not an employee because of the irregular and discontinuous nature of the hiring, was to accord a decisive weight to one variable among practically dozens which courts had relied on in a case law stretching back over decades.

This example is also instructive because it requires us to focus on why this (or indeed, any) single variable should have acquired the weight that it did, at that time. From one point of view, we might conclude that the variable acquired a high ‘weighting’ in an incremental and emergent way, as it was initially proposed in a number of first-instance decisions, taken up and adopted by intermediate courts, and finally endorsed by the highest appellate court and thereby stabilised. The ‘error correction’ process—which we might liken to a metaheuristic inspired by natural selection—benefited from the operation of the doctrine of precedent in a dual sense: not just the influence of preceding decisions, but the stabilising effect of the hierarchy of courts, helped to elevate the concept of mutuality of obligation to its pre-eminent position as the first point of reference for courts when deciding employee status cases.

Yet the example also poses difficulties for the idea that ML can replicate legal reasoning, since it is far from obvious that the mutuality case law was ‘correcting’ an ‘error’ in the system. On the contrary, the test has proved controversial to the point of being regarded by many commentators as ‘error’ in itself, warping the application of employment law. In what sense, then, is it appropriate to think of the end result of legal ‘learning’ as a process of ‘optimisation’, and not an ongoing process of ‘training’ and ‘tuning’ without an explicitly defined ‘objective function’?


3.3 Optimisation and Reflexivity in Legal Learning

As we have seen, in unsupervised learning ML systems identify an output or ‘ground truth’ from an initial clustering of data points on the basis of their similarity or complementarity. The clustering of data points can be tested and refined through an unsupervised DL analysis and then adopted as an output in a supervised learning approach. This is already being done in a number of contexts with a bearing on the administration of civil and criminal justice, including probation, immigration, police resourcing, and credit scoring. Although its use in legal adjudication as such is currently limited to a small number of isolated trials, it would be surprising if it were not more widely taken up at the level of judicial decision making in the relatively near future.

An important factor which accounts for the success of DL approaches in matching processes to outcomes to date is that they have generally been applied to contexts in which the ultimate output variable is invariant to the process being used to identify or predict it. In other words, a medical condition such as diabetes or cancer is an invariant reality which exists regardless of the techniques used, whether through ML or otherwise, to diagnose it. The ‘success’ of the algorithm or model used to predict a medical condition can be tested against that invariant reality: the model will be more or less effective in ensuring improved survival or recovery rates on the basis of the diagnoses it makes. In that sense, there is an objectively ascertainable measure of ‘success’ against which the model can be benchmarked.

In the context of legal adjudication, there is no equivalently invariant measure of a model’s success. This is because the output variable – in the case we have been considering, the classification of the individual as either an employee or self-employed – is not invariant with respect to the processes used to define it.

Legal knowledge is reflexive. In other words, the law’s epistemic categories alter the social forms to which they relate, as well as being altered by them. The category ‘employee’ is a legal construct which would not exist if the law did not create it. It is not a reality which exists independently of the law. A concept of this kind operates in complex, two-way relationship to a given social referent, in part reflecting it (just as the notion of ‘employee’ has changed over time as forms of working have also changed), but also shaping it (since the way in the law defines employment has tangible consequence for, an effects on, the way in which labour relations are constituted in the economy).

---

122 S Deakin, ‘Juridical ontology’.
It follows that no legal determination, even of the most straightforward kind in which a stable rule is applied to uncontroversial facts, is a simple description of reality. It is a normative act intended to change that reality.\(^{124}\) When ‘work’, understood in its most elemental material form as a physical act of labour, becomes ‘employment’ as a result of a legal classification, what follows is the assignment to that material form of normative effects: legal rights, liabilities, powers, immunities, and so on. Thus the legal classification of work relations, while it undoubtedly has a technical (juridical) dimension, is also, necessarily, political in nature, in the sense of being concerned with the articulation of values. It defines and instantiates a particular conception of justice in the ordering of work.\(^{125}\)

This suggests the need for some care when specifying the outcome variable or ground truth which legal reasoning achieves. What exactly is being optimised? If our argument to this point is correct, it at least insufficient, and arguably misleading, to suggest workers might be classified more ‘accurately’ through ML than through human decision making. There is no technically ‘optimal’ solution to the question of how many ‘employees’ and ‘independent contractors’ there are in a given industry or economy. The relative proportions of the two groups is ultimately a normative question which turns on competing conceptions of the public good in the regulation of labour relations.

The application of ML to legal adjudication at the very least obscures the political issues at stake in the process of juridical classification. But it also undermines the effectiveness of legal reasoning as a means of resolving political issues. Legal reasoning involves more than the algorithmic application of rules to facts.\(^{126}\) Because of the unavoidable incompleteness of rules in the face of social complexity,\(^{127}\) legal reasoning is best thought of an exercise in experimentation. Approximate solutions to recurring issues of dispute are proposed through adjudication and then subjected to scrutiny through a number of mechanisms of selection which include litigation, lobbying and commentary. It is a dynamic process which, while drawing on experience from the past (information retained in the system from earlier iterations), is forward-looking in the sense of adjusting existing rules to novel contexts and projecting their normative effect into future anticipated states of the world.\(^{128}\) ML, however effective it may be in replicating the effects of known iterations of a particular classification problem, can only work on

---


\(^{125}\) A Supiot, L'Esprit de Philadelphie. La justice sociale face au marché total (Seuil 2011).

\(^{126}\) A Supiot, Governance by Numbers, 25-30.

\(^{127}\) Pistor and Xu, ‘Incomplete law’.

\(^{128}\) H Kelsen, ‘Pure theory of law’.
existing data. By comparison to legal reasoning, it is unavoidably backward-looking. ML’s effectiveness is diminished in direct relation to the novelty of the cases it must process and, relatedly, to the pace of social change in the particular context to which it is applied.

Where ML has so far been used to ‘predict’ the outcome of cases, this has been done by using data from past decisions with a view to ‘training’ an algorithm to replicate the outcomes in those decisions on similar facts. An algorithm of this kind can only be used as basis for adjudication if it is assumed that the facts of future cases will be unchanged from those in the past. Yet this is almost certainly not going to be the case in many social contexts, including the one we are considering: it is generally agreed that the current rise of ‘gig’ or ‘platform’ work is posing challenges to the existing definitional structure of labour law which are just as foundational as those posed by the rise of precarious or ‘flexible’ forms of work four decades ago. Novel fact situations arising from platform work are coming before labour courts all the time. New technologies will challenge existing conceptual categories in ways that will undoubtedly require adaptation of those concepts, and may render many of them otiose. Under these circumstances, using ML in place of legal reasoning as a basis for adjudication would lock in existing solutions, leading to the ossification of the law.

3.4 ML overreach

It is not our argument to claim that ML cannot, by its nature, be applied to many areas of legal decision making. On the contrary, because much of legal reasoning is algorithmic, there is huge scope for ML applications in the legal context. We can expect further advances in ML which will overcome some of its current limits. As the costs of ML come down, and its predictive capacities improve, we can anticipate growing pressure for it to be used to replicate or substitute for adjudication. This will involve a step beyond its current use, in various ‘Legal Tech’ applications, to estimate litigation risks. Employing ML to predict how a particular judge might decide a future case on the basis of their past performance may raise ethical and legal issues of a kind which could justify restrictions on its use, as has already happened in at least one jurisdiction, but it does not pose a direct threat to the autonomy of the judicial process. ML adjudication, by contrast, does pose such a threat, as it would lead to the de-norming of law, as well as to its ossification.

For some supporters of an enhanced role for ML in the legal sphere, this scenario, far from being a matter for concern, should be actively welcomed. Their argument is that biases and inefficiencies in the policy making process would justify the

---

extension of ML beyond adjudication. While in the short run (the ‘next few decades’) more data and improved machine learning are likely to be ‘complement to human judgment rather than substitutes’, over time the goal of a ‘functionally complete law’\textsuperscript{130} will come into view. Governments will use machine learning to ‘optimise the content’ of the law in the light of ‘prevailing politically endorsed social values’. ML will facilitate the implementation of ‘transfer systems that achieve the distributive justice trade-offs that democratic political processes endorse’. As ML is used to identify an ‘efficiency frontier’ at which different values are traded off against each other, the law will become not only more complete but also more generally stable. In this state of a ‘legal singularity’, ML will generate a ‘stable and predictable legal system whose oscillations will be continuous and yet relatively insignificant’\textsuperscript{131}

We would suggest an alternative prediction. This is one in which the mirage of ‘complete law’ leads to the removal of political contestation from the legal system. In its place will be a version of the ‘rule of technology’ according which solutions based on computation are accorded an elevated status. Challenging outcomes based on supposedly neutral modes of decision making will prove increasingly problematic. At the same time, the techniques used to generate these outcomes are likely to prove more remote and opaque as time goes on. The tendency, already evident in the development of Legal Tech, for algorithms to be concealed from view behind a veil of commercial confidentiality and official secrecy, will only intensify.

4. Conclusion

This paper has explored the hypothesis that there are limits to the computability of legal reasoning and hence to the use of machine learning to replicate the core processes of the legal system. It has considered the extent to which there are resemblances between machine learning and legal decision making as systems involving information retention, adaptive learning, and error correction. Our argument has been that while there are certain resemblances, there are also critical differences which set limits to the project of replacing legal reasoning with machine learning.

Machine learning uses mathematical functions to perform various analysis and classification tasks. In ‘deep learning’ approaches, data is organised into concepts which are expressed hierarchically, with more complex concepts being built out of simpler ones. The classification of data into concepts is achieved through the use of weights whose values are adjusted over time as the system receives new inputs.


Through its recursive operations, a system adjusts its internal mode of operation or ‘learns’, so that, in principle, errors are corrected and gradually purged.

The legal system also makes use of concepts to store and retain information. Legal concepts are ordered hierarchically, with higher-order categories informing the content of sub-categories operating at lower levels of abstraction. Legal concepts possess the capacity for self-adjustment in response to external signals. Error correction at a systemic level is achieved through a number of mechanisms including claimant-led litigation, appellate review of lower court judgments, and statutory reversals of unworkable or dysfunctional rules.

At a structural level, then, the resemblances between machine learning and legal reasoning are more than superficial. Some of the objections to the use of machine learning in a legal context are perhaps less fundamental than they might first seem. For example, the implication of the hierarchical ordering of legal concepts is that even a very widely framed general clause such as ‘reasonableness’ or ‘good faith’ can be understood as operating symbiotically with lower-level concepts and, ultimately, with fact-specific instances of individual disputes. The application of a legal rule to a set of social facts is, in this sense, an algorithmic process depending upon the interaction between concepts and rules which are expressed at different levels of generality, not unlike, in principle, the neural layering and assigning of relative weights to new informational inputs that characterise the artificial neural networks used in deep learning.

However, for machine learning to replicate legal reasoning it requires the translation of the linguistic categories used by the law into mathematical functions. This is not straightforward, even with advances in natural language processing which are making it possible to convert text into computer code with ever more sophistication. As the debate over ‘smart’ and ‘semantic’ contracts makes clear, there is an element of flexibility and contestability in the natural language used to express juridical forms which cannot be completely captured by mathematical algorithms.

We suggest that the natural language categories used in juridical reasoning, precisely because of their imprecision and defeasibility, are superior in a number of respects to mathematical functions. Not only are they better at storing and representing complex information about the social world; they are more adaptable in light of new information. Machine learning, for all its advances, remains backwards-looking and prone to error through lock-in effects. It also has relatively few options for error correction by comparison with the range of techniques available in legal decision making. While progress is being made to ensure greater algorithmic transparency and explainability, the intricate layering and opacity of
artificial neural networks remain, at least for now, more opaque than those of the law.

Underlying the project to apply machine learning to law is the goal of a perfectly complete legal system. This implies that the content and application of rules can be fully specified ex ante no matter how varied and changeable the social circumstances to which they are applied. In this world of a ‘legal singularity’ the law operates in a perpetual state of equilibrium between facts and norms. Advanced as an answer to the incompleteness and contingency of the law, the project for the legal singularity is also a proposal for the elimination of juridical reasoning as a basis for dispute resolution and the allocation of powers, rights and responsibilities. As such it risks undermining one of the principal institutions of a democratic-liberal order.

To avoid such an outcome, thought will need to be given to identifying and establishing limits for the use of machine learning and other data-driven approaches in core aspects of the legal system. The success of these measures will ultimately depend on the law’s capacity to maintain the autonomy of its operations in the face of all-encompassing technological change, an outcome which is far from guaranteed.