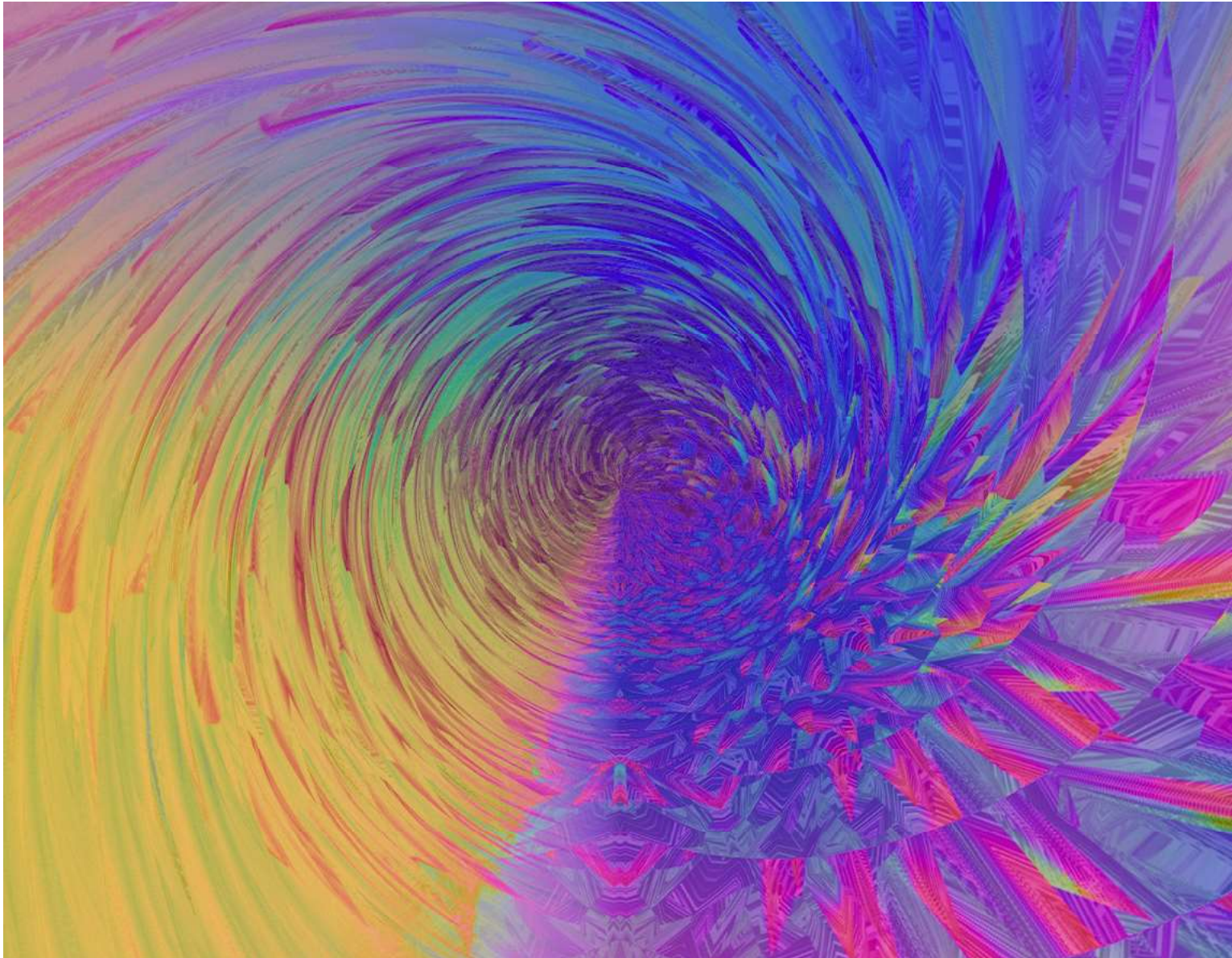


ISSN No: 2582-8312

# *Lattice*

**THE MACHINE  
LEARNING JOURNAL**

**VOLUME-2, ISSUE-4 (October - December 2021)**



AN INTERNATIONAL PEER REVIEWED JOURNAL  
**IN DATA SCIENCE & MACHINE LEARNING**

**ADaSci** | THE ASSOCIATION  
OF DATA SCIENTISTS

[www.adasci.org](http://www.adasci.org)

# Models & Mechanisms for Motivating Machines

autonomous evolution of artificial intelligence

Prithwis Mukerjee  
Praxis Business School, Kolkata  
[prithwis@praxis.ac.in](mailto:prithwis@praxis.ac.in)

**Abstract—** We describe a mechanism that may allow digital machines to display intelligent behaviour without the overt intervention of human programmers. While most studies focus on how systems may be built to demonstrate intelligence, our focus is on how to motivate machines to demonstrate intelligence in new areas when they are not explicitly programmed to do so. Domain generalisation is an attempt in this direction but these attempts seek to enlarge capability in existing domains without moving into new domains of expertise. In this paper, we look at a set of existing software constructs, namely (i) the universally used TCP/IP protocol that connects machines, (ii) Docker containers that create portable programs, (iii) the Ramanujan machine that generates novel conjectures about number theory, (iv) blockchain technology that forms the basis of decentralised autonomous organisations (DAO) and (v) generative adversarial networks (GAN) that use pairs of generator-discriminator neural networks to create original content that is computationally indistinguishable from naturally occurring content. We show how these seemingly disparate dots can be connected to reveal a pattern that delineates the contours of a novel mechanism that may be used to define different levels of motivation for machines to demonstrate intelligent behaviour in new areas in an autonomous manner. We also speculate on how this may lead to a political colour in the behaviour that emerges through this process

**Keywords —** machine learning, motivation, autonomous learning, Ramanujan machine, blockchain, GAN, dockers

## I. INTRODUCTION

Artificial intelligence has made it possible for machines to display competence in many areas that humans and animals excel in, like recognising shapes, driving cars or competing with an adversary. However, none of these systems acquire these capabilities autonomously. They need to be ‘motivated’ to do so by being trained or programmed by humans.

Motivation is a mechanism that allows an artificial agent, which could be hardware or software, to behave in a manner that displays human traits like curiosity and the desire to explore. All such motivation mechanisms are eventually tied to some kind of a reward. In human beings, this reward is often ill-defined and non-material and could be rooted in inherent satisfaction like having fun or completing a challenge. This is intrinsic motivation as opposed to the motivation to perform under external pressure or direction.

The study of intrinsic motivation has its roots in psychology in the works of Berlyne [1] who explored ideas around familiarity and novelty. Festinger [2] viewed the process as being one that would reduce the dissonance between an external and internal view of the world as did Kagan [3] who sought to locate motivation in the desire to reduce the incongruence between experience and cognition.

Moving from human psychology to machine intelligence, Ouedeyer and Kaplan [4] characterised intrinsic motivation as being based on either knowledge, competence or morphology, where for all practical purposes, knowledge devolves into information and information theory. From an information perspective, the simplest possible approach is an objective function as is used in linear programming. However, for complex situations involving a large number of variables and system-states, a practical approach to defining and leveraging motivation is through reinforcement learning [5].

As a key tool in the toolbox of machine learning and artificial intelligence, reinforcement learning does away with the need for labelled or unlabelled data and depends instead on finding a balance between exploration (of uncharted territory) and exploitation (of current knowledge). Because of its generality, it has been used in a wide range of disciplines that include game theory, simulation-based optimisation, multi-agent systems and swarm intelligence. In fact, Q-Learning and the concept of the Deep Q Network that use neural networks to improve the quality of action is perhaps the best way for a machine to learn to solve problems on its own without any human input.

However, one key drawback of reinforcement learning, in fact of all AI and ML techniques is the inability to generalize. These models may excel in one domain and the level of excellence can increase over time but like a fish out of water, these models are of little use in other non-related domains.

What we explore in this paper is the possibility of machines to explore, and acquire competence in, areas that lie beyond what they are currently competent in, and to do this without human intervention. We describe an evolutionary approach with analogies drawn from existing technologies like TCP/IP, Docker, GPT3, the Ramanujan conjecture engine, Blockchain and a class of artificial neural networks called generative adversarial networks.

Software artifacts that display artificial intelligence are increasing in both number and sophistication. There are many definitions of what constitutes intelligence and there are many ways in which software has been programmed to demonstrate the same. Of all the many options, the use of artificial neural networks (ANN), that closely mimic the connectionist approach of animal brains, has been found to be most effective in performing tasks that are both useful and insightful. This includes, for example, recognising faces, driving cars, generating meaningful text passages and playing a wide range of games both against humans and against other programs. It may not be the case that the ANN



will always be the best way to demonstrate this kind of intelligent behaviour, so for the purpose of this study an ANN is neither necessary nor sufficient. All that we need is a digital artifact — a container of data, code, model, APIs or a combination of some of these — that we will refer to as a digital intelligence unit, or DIU. Having access to a DIU equips a digital computing device with the ability to demonstrate intelligence or mimic a specific behaviour of an intelligent biological system.

## II. DIGITAL INTELLIGENCE UNIT

A DIU may be a digital construct but its input and output could be both digital as well as physical. A typical DIU that we deal with today may read in a piece of digital information, like an image or data file as an input and generate digital output, like a name or a class. However, there is no conceptual difficulty in assuming that the DIU is connected to sensors that capture a physical measurement from the environment or that it can cause wheels, drills, arms, actuators or tools to move and do physical work.

For example, a DIU may guide a car to move through traffic or terrain, pick up and assemble physical objects like rocks, machine components or even assemble two objects together. It can also sense and consume energy or where necessary cause the generation or transformation of energy from one state to another. Not all DIUs need to be very sophisticated. There could be very basic DIUs that simply interrogate other devices and exchange information or a DIU that allows a device to share a physical resource like a camera or a disk with another device. Nevertheless, we will treat the DIU as a digital abstraction that is resident on a digital hardware device like a computer.

A set of DIUs that work together may be viewed together as a larger, bigger or more sophisticated DIU. This larger DIU may still reside on one hardware device or its parts may be distributed across multiple hardware devices and identified with something like a Uniform Resource Identifier as is done in web development. However, this set of smaller, compatible DIUs is, conceptually, still another DIU. For example, a *rover* that NASA sends to Mars may consist of a collection of DIUs, each with its own intelligent function but the *rover* itself may be viewed as another DIU.

Continuing with this analogy, we may be tempted to view a biological animal, like a fish or man, as a DIU that is a collection of simpler DIUs. For the sake of argument, and simplicity, we may view, or model, a biological dog as a collection of four DIUs that can scan the environment and identify objects, distinguish between edible and non-edible objects, consume edible objects and generate signals that express facts about the taste of the food. So, four discrete DIUs are collected together to give a bigger DIU called a digital dog.

While this analogy appears tempting, it poses a few challenges.

## III. MOTIVATION

The first challenge is motivation. What motivates a DIU to demonstrate its intelligence? Or what is even more fundamental, what causes a DIU to come into existence?

For a DIU to demonstrate its intelligence, a program must be executed, which means that someone or something must start the program. This is not difficult because most digital platforms (as in computers with operating systems) have mechanisms that could cause certain programs (including DIUs) to start automatically when the system boots and then wait for signals, or interrupts, from the external world. These signals or interrupts could be key-presses or other events like the arrival of mail or the rise of temperature.

What is much more difficult is the process of creating the DIU in the first place. At our current level of understanding, the motivation to create a new DIU, or a new capability, lies in the hands of a human programmer and not within the domain of the digital device. As a human programmer, I can decide that in addition to recognising faces, we need to generate music for which we need an additional DIU. The process of building, or writing the code, for another DIU can be automated — programs to write other programs are not impossible with current technology, for example, the Github CoPilot or GPT3 — but someone must have the motivation to do so.

Current DIUs can be programmed to improve their performance with time. Face recognition programs or self-driving cars can be programmed to become better with use but we still do not have any logical mechanism through which a face recognition program suddenly decides to learn how to drive a car or vice versa. Coming back to our digital dog, its food recognition DIU can become better and better to differentiate good food from bad but it is extremely unlikely that it will acquire an additional DIU that makes it jump over the fence and search for better food outside the house.

Incidentally, a fence jumping DIU is not at all difficult to construct. With current technology it is a trivial exercise to build a robotic system that can jump over a fence. However, what is missing is the motivation to include this DIU in the current digital dog DIU and enhance its capabilities. We need a human programmer to identify this new need and add this additional DIU to the digital dog. Thus, the challenge lies in creating a mechanism, a motivation, that will allow the digital dog DIU to do this on its own.

Let us see how a biological dog does this in the physical world.

## IV. THE COMMUNITY

A biological dog will jump the fence when it sees another biological dog jumping the fence. This ability to jump the fence, or the motivation to do so, is a behaviour, an ability (or intelligence unit), that resides within the community and which is acquired by or triggered in a member by observing other members. Perhaps this is far more so in humans than

in animals who are primarily hardwired. So the existence of a community is an important mechanism that allows an individual member to enlarge its DIU by acquiring the DIU available with some other member. In fact, in the story of human evolution, one of the reasons why humans have been so successful compared to other animals, is because they could form communities, share experiences and learn from each other. Henrich [6]

The first challenge is to devise a mechanism that motivates an individual member of the community to search for and get access to a DIU that is available with others. We refer to this as low level or primary motivation.

Computers that are connected in a network, say a TCP/IP based Ethernet network, can be viewed as a community that can share information with each other. However, they do not share information spontaneously. There needs to be a trigger activated by a human or an external stimulus that causes an exchange of information. Two computers, A and B may be connected by a network and one, say A, may have a DIU to recognise faces and the other, B, may have a DIU to drive cars. But there is no reason or likelihood for A to access the car-driving DIU on B or for B to access the face recognition DIU on A.

First, A would not be aware of the existence of the car-driving DIU on B and even if it were, there would be no reason, or motivation, to access it. To overcome this drawback and to create the mechanism for the primary motivation, we introduce the analogy of a computer virus.

#### V. PRIMARY MOTIVATION : THE VIRUS AND THE DXP

A computer virus is a computer program, but we can view it as another DIU that is capable of performing at least two tasks. First, unlike other programs that sit patiently on their host platform waiting for a signal to do something, a virus program actively seeks out other devices in the network, or the *community*, and actively looks for exploitable exposure points. These exposure points could be TCP/IP ports through which messages could be sent or files and folders that can be written on. Second, it usually makes a copy of itself and places the copy in the second device.

Unlike the DIUs that we are interested in, a computer virus has malicious intentions but the principle under which they operate can be easily adopted by the DIUs. This leads to the idea of a DIU Exchange Protocol (DXP) that is built into the operating system of all digital devices, somewhat similar to the ubiquitous TCP/IP stack. The DXP stack on any host device, is designed to look into other, target devices connected on the network and if allowed to do so by the DXP stack on the target device, to scan it for the existence of new DIUs. This is the primary motivation built into the protocol stack. If a new DIU is found it will be copied back from the target to the host. Obviously, the process works in a symmetrical, peer-to-peer manner. Any machine with DXP can be a host and can pick up DIUs from any other target machine that is running the DXP protocol.

How do we know and recognise a DIU from the many other files, programs, images lying in the target? At the simplest level, a DIU can be identified by something like a file extension. For example, the *http* protocol recognises files with extensions like *htm*, *html* etc but will not interact with *doc* or *ppt* files. But given the complexity of a DIU, a simple file may not be sufficient. So, we may create a DIU as a container — a Docker container is a good analogy — that contains code with APIs, models and perhaps data that may be exchanged across devices. Markers present in the container, for example a correctly formatted XML file, will allow the DXP protocol to recognise them as such and distinct from other artifacts lying in the machine.

The process can be made significantly simpler and more secure if instead of exchanging DIUs from each other in a peer-to-peer manner, DXP protocol on each device publishes its DIU, or saves it, to a central location like a DIU repository. This would be analogous to the Docker hub, CRAN the repository for R packages or even GitHub which has a lot of source code. A better mechanism could be to store the DIUs as smart contracts in an Ethereum, or similar, blockchain. There are two advantages for using a blockchain based approach. First, blockchain *full* clients, that validate transactions and add blocks to the blockchain are designed to operate autonomously without any human intervention and as we show later, this is important in our scheme. Secondly, the DXP protocol that controls the process of adding a new block, with smart contracts, can be configured to include a validation process to ensure that only valid DIUs are added. The validation process will be explored in more detail later.

Once the repository is in place, then any member-device of the DIU user community can pull any DIU that is required or is of interest. The newly pulled DIU can then be assembled with other DIUs already present to create larger and more sophisticated DIU. This would not only mean that the device has evolved by acquiring a new ability but it has done so on the basis of its own primary motivation and not on the commands given by a human operator.

The DIU repository, and the primary motivation built into the DXP protocol, gives us a possible solution to the problem of how our digital dog enhances its ability by acquiring the ability, or DIU, to jump over the fence and find better food. This leads us to two more difficult questions, both of which are tied to the phenomenon of motivation. First, if it is not a human being, then who will create these DIUs and why? Second, why should an existing digital platform that already has a set of DIUs pull one more and add it to the DIUs that it already has.

We shall park the first question for the time being and focus on the second. Which DIU should a platform pull and why? What is the motivation for a device to pull a specific DIU? The basic or primary motivation, namely to scan the DIU hub and pull DIUs at periodic intervals, is baked into the design of the DXP protocol. But the choice of which DIU to pull depends on two factors, namely compatibility and utility.

## VI. DIU COMPATIBILITY

For a DIU to work, it needs certain prerequisites. A DIU to drive a car needs access to a car, that is a device with engine, wheels, radars and many other things. A dog does not have wings and cannot fly in the air but it has legs that allow it to jump. So, it learns how to jump and not how to fly. Similarly, every digital device cannot pull any DIU. Its choice is restricted to a set of DIUs that it is in a position to operate, or for which it already has the prerequisite DIUs.

Prerequisites are usually chained backward. Let us consider that a device attempts to install a face-recognition DIU.

A face recognition DIU needs

- A DIU that already has the ability to access a network camera
- OR
- A DIU that can obtain a camera for the device, that in turn needs
  - A DIU that can execute an eCommerce transaction to purchase a camera and that in turn needs
    - A DIU that can earn money with say crypto mining or performing Amazon Mechanical Turk type assignments
  - AND
  - A DIU that can physically plug a camera, that in turn needs
    - DIU that can operate a robotic arm, etc., that in turn needs
      - {another hierarchy of DIUs}

What if every device were to adopt this chain strategy? That would lead to a situation where every device can do everything which may not be physically possible or even desirable. Can a dog acquire the ability to fly? It may be possible after many generations -- as the evolution of species has shown -- but obviously the physical dog body will die but its genomes will get progressively altered over generations until it can fly. Similarly, the physical platform on which the digital device works may collapse but the software can get transferred from device to device and keep acquiring DIUs until it can do whatever it wants to do. This will take a long time and a lot of resources.

Instead, let us focus on how a device will pull a certain DIU that it wants to. But what is it that the device *wants-to-do*? This is a part of a larger question that will be addressed as the next level of motivation, or secondary motivation. Our current focus is on the question of *Which DIU should a platform pull?* and we said that the answer depends on compatibility and utility. We have addressed the issue of compatibility with primary motivation and we now look at utility and the secondary motivation.

## VII. SECONDARY MOTIVATION : DIU UTILITY

A DIU will be selected for a pull and implementation, if it provides some value to the device. A biological dog will

learn how to jump because it gives it better food and so improves its ability to survive. It will not try to learn how to walk on two legs even if it sees another dog walking on two legs because walking on two legs does not increase its survivability. In the case of biological species, the utility of a particular ability is related to survival and this survival operates at different levels - survival of the individual body, survival of the species or the genome. There is also the possibility or the question of the survival of specific genes in the genome, if we agree to accept Dawkins' principle of The Selfish Gene.

Mapping this issue of biological survival to the world of digital devices is the next challenge and, in a sense, it loops back to the first issue that we identified already, namely motivation. We have already addressed this at one level of primary motivation that partially explains which DIU is to be pulled based on the ability to search for and pull DIUs on the basis of feasibility and compatibility. Now we need a next, or higher level of secondary motivation. Why should a digital platform seek any specific DIU to enlarge its set of DIUs?

In the biological world the only motivation behind the process of acquiring intelligence (or ability to perform certain tasks) is survival. Humans in a certain limited way are governed by Maslow's hierarchy of needs. When it comes to digital devices, we need to determine whether they should be guided, like lower animals, by the need to survive? Or should they be guided by something similar to the human hierarchy of needs? We know that in the case of computer viruses, the motivation is simply to spread to other machines, which is like a survival strategy. For a higher-level digital device, that is one with a complex DIU, the motivation could be something else.

So instead of trying to discover what could be the motivation, we can as humans build our own definition of secondary motivation directly into the algorithm of the DPX protocol. Note that humans are NOT building the DIU, but are designing the DPX protocol. Most optimization problems begin with a motivation, which is generally captured by means of an objective function that we try to minimise or maximise depending on the problem, but there could be others. The Open Shortest Path First is an algorithm that is baked into the heart of the Internet Protocol (IP) and determines the route to be taken by a data packet. Public Key Cryptography is an algorithm that is present in the *https* protocol and ensures data security. Proof of Work is an algorithm that is built into many cryptocurrency protocols to determine which block will be allowed to enter the blockchain.

Similarly, we need a motivation algorithm, the secondary motivation, that is baked into the DPX protocol that determines which DIU is of interest to the device or is useful. The design of this algorithm could be based on certain principles that human society holds dear, like the three principles of Utilitarianism that can be summed up as the *greatest good for the greatest number*. We could also draw upon certain ideas drawn from popular culture like the Three Laws of Robotics created by Isaac Asimov.



Obviously other competing approaches can be explored as well. All that we are saying now is that a motivation function, whatever it may be, needs to be built into the DPX algorithm and this will guide the choice of DIUs that will be allowed to be added to the repository or pulled from it by individual platforms.

With the algorithm of the secondary motivation that decides on which DIU to acquire, in place, we now have another question that we had parked earlier. If it is not a human being, then who will create this pool of DIUs and why? This leads us to a tertiary motivation that operates at the community level.

#### VIII. TERTIARY MOTIVATION : COMMUNITY PARTICIPATION

Mutations that drive biological evolution occur at random. The ones that survive and are passed down through generations survive purely because they make the individuals *fitter* in their respective environments. Thus, evolution works in a brute-force manner, randomly trying out different permutations, and keeping only those mutations that survive the test of natural selection. Similarly, it can be possible to devise mechanisms that will generate newer and newer DIUs and then test them against the principles of secondary motivation. Here we will draw upon three analogies from the world of mathematics and computers and use them to define another level of motivation that can motivate the community as a whole to come up with more and more DIUs.

First let us consider the Ramanujan Machine that was created by Raayoni, et.al [7] to automatically generate new mathematical conjectures using an algorithmic approach. Ramanujan was an Indian mathematician who came up with many unproven conjectures, most of which were validated long after his death. However, these conjectures opened up new vistas in number theory that are still being exploited today. The Ramanujan machine is a network of computers running algorithms dedicated to finding conjectures about fundamental constants in the form of continued fractions. The purpose of the machine is to come up with conjectures (in the form of mathematical formulas) that humans can analyse, and hopefully prove to be true mathematically.[8].

The Ramanujan machine currently generates conjectures from a rather narrow domain of number theory and uses two algorithms, namely MITM and gradient descent. But we can envisage other algorithms that may generate tasks or objectives that are in line with the contours of the secondary motivation algorithm. Then the code for these tasks can be created by a product or process similar to the Open AI's GPT-3. This combination of a secondary motivation task generator and a code creator can then be viewed as a DIU engine that can run autonomously and generate any number of novel DIUs.

The second key piece of our strategy would be a blockchain based decentralised autonomous organisation (DAO). This is a self-sustaining distributed mechanism that creates economic value by encouraging individual machines to validate transactions — in this case DIUs created by the

DIU machine — and rewards successful ones with cryptocurrency tokens. For a DIU to be valid it must meet the conditions of DPX protocol in terms of interoperability and the principles of secondary motivation. Only then it will be accepted as a part of the DIU blockchain and this blockchain will become the DIU hub or repository that we had discussed earlier.

Unlike the Bitcoin or current Ethereum blockchain that is based on an energy intensive Proof of Work protocol, this DIU Blockchain could be based on the principles of Proof of Stake or other energy efficient protocols.

This combination of a DIU generator and blockchain based DIU validator is remarkably similar in principle to the combination of generator-discriminator that is the basis of a class of artificial neural networks called generative adversarial network [GAN] first proposed by Goodfellow [9] A GAN, which is the third piece of our proposed tertiary motivation mechanism, is typically used to generate original artifacts that are nearly indistinguishable from similar artifacts that are found in natural populations. The most common example of this is human faces. Given a training set of human faces, a GAN can generate synthetic images of faces that are not found in the training set, but cannot be distinguished from naturally occurring images. In this case, the training set of DIUs could be the thousands of currently extant DIUs of AI systems that have been developed by humans. In fact, the blockchain could also be seeded by humans as in the first few thousand blocks could contain DIUs built from existing AI systems. However, going forward, the combination of the DIU engine and the blockchain validation process will create a GAN-like mechanism that will create a potentially endless series of DIU.

This mechanism will provide the tertiary motivation to fuel an evolving ecosystem of digital devices with more complex and useful DIUs. As a by-product, the crypto-tokens generated on this DIU Blockchain could be used by digital devices to pay for DIUs that they pull from the DIU hub.

#### IX. CONCLUSION

We have seen that human beings can build very useful software applications that mimic intelligent behaviour. We have also seen that it is possible for these applications to improve their performance based on observing the results of their actions. What has been lacking so far is an autonomous mechanism that has the motivation to create new software applications that mimic a range of intelligent behaviour that goes beyond what was originally coded for.

Biology has addressed this problem by creating new, random behaviours and then checking which of them are best suited for survival. This is called evolution and this biological evolution operates at three levels of survival - of the individual, of the species and of the gene.

We have proposed an analogous mechanism that can be used by digital devices to create a series of applications, the DIUs, that will automatically evolve in terms of complexity based on the three levels of motivation that we have described here.

Most of the software products that are used as analogies in this article already exist. What is missing is an algorithm for the secondary motivation that is stable but flexible and will work for all in a fair and equitable manner. In fact, the logic, or the algorithm that drives the secondary motivation on the DXP protocol is the key to the success of this scheme. This is where we leave technology and could wade into complex socio-political issues. Should this protocol have a socialist or a capitalist bias? This could be the basis of a fork in the DIU blockchain that leads to the emergence of two kinds of machines whose behaviour will reflect this underlying political bias. Even within the same blockchain, there could be DIUs that display either aggressive or conciliatory behaviour -- as is the case with self-driving cars. While the upload, or push, part of the protocol may allow both kinds of DIU to be accepted into the blockchain, the download, or pull part could have options that allow each device to pull DIUs that give it a specific *psychological* profile.

#### ACKNOWLEDGMENT

The first draft of this paper was reviewed by Subhayan Mukerjee, Assistant Professor, National University of

Singapore and his valuable comments have been incorporated in the text.

#### REFERENCES

- [1] Berlyne, D.: Conflict, Arousal and Curiosity. McGraw-Hill, New York (1960)
- [2] Festinger, L.: A theory of cognitive dissonance. Evanston, Row, Peterson (1957)
- [3] Kagan, J.: Motives and development. *Journal of Personality and Social Psychology* 22, 51–66.
- [4] Oudeyer, Pierre-Yves; Kaplan, Frederic (2008). "How can we define intrinsic motivation?". *Proc. of the 8th Conf. on Epigenetic Robotics*. 5. pp. 29–31.
- [5] Kaelbling, Leslie P.; Littman, Michael L.; Moore, Andrew W. (1996). "Reinforcement Learning: A Survey". *Journal of Artificial Intelligence Research*. 4: 237–285. arXiv:cs/9605103. doi:10.1613/jair.301. S2CID 1708582. Archived from the original on 2001-11-20..
- [6] Henrich, J : *The Secret of Our Success*. Princeton University Press, (2017) <https://press.princeton.edu/books/paperback/9780691178431/the-secret-of-our-success>.
- [7] Raayoni, G., Gottlieb, S., Manor, Y. et al. Generating conjectures on fundamental constants with the Ramanujan Machine. *Nature* 590, 67–73 (2021). <https://doi.org/10.1038/s41586-021-03229-4>.
- [8] <https://phys.org/news/2019-07-ramanujan-machine-automatically-conjectures-fundamental.html>
- [9] Goodfellow, Ian; Pouget-Abadie, Jean; Mirza, Mehdi; Xu, Bing; Warde-Farley, David; Ozair, Sherjil; Courville, Aaron; Bengio, Yoshua (2014). *Generative Adversarial Nets* (PDF). *Proceedings of the International Conference on Neural Information Processing Systems (NIPS 2014)*. pp. 2672–2680..