**INDEX**

|  |  |
| --- | --- |
| **Table of contents** | **Page No.** |
| a. Abstract | i |
| b. List of Figures | ii |
| 1.Introduction | 01 |
| 2.Literature Survey | 03 |
| 3.Working Principle | 14 |
| 4.Advantages | 20 |
| 5.Disadvantages | 21 |
| 6.Applications | 22 |
| 7.Conclusion | 30 |
| 8.References | 31 |

**ABSTRACT**

In industry reinforcement, learning-based robots or machines are used to perform various tasks. Apart from the fact that these are more efficient than human beings, they can also perform tasks that would be dangerous for people. Industrial robots are very easy to simulate, they can also run without wasting any byproduct, and they provide a great wealth of data. What we see here is called reinforcement learning. It directly connects a robot’s action with an outcome, without the robot having to learn a complex relationship between its activities and results. A great example is the use of AI agents by Deepmind to cool Google Data Centers. This led to a 40% reduction in energy spending. The centers are now fully controlled with the AI system without the need for human intervention. Thus, this Technical Seminar paper focuses on the concept of reinforcement learning in industrial automation, its working, and its applications. During this research, I have learned one of the core concepts that form a major part of our everyday lives.

|  |  |
| --- | --- |
| **List of Figures** | **Page .no** |
| 1.Automation using Robot | 2 |
| 2.First automation device | 4 |
| 3.self-driven machines for grain mills | 5 |
| 4.20th century device | 7 |
| 5.Difference between ml and Rl | 10 |
| 6.Reinforcement algo example | 11 |
| 7.Deeplearning algo example | 17 |
| 8.Action-Environment -MDP | 20 |
| 9.Improving Decision-Making Robots | 23 |
| 10.Safer Robot Collaboration and Productivity with Cobots | 24 |
| 11.Deepmind to cool Google Data Centers | 25 |
| 12.Manufacturing | 26 |
| 13.Product Marketing | 27 |
| 14.Trade Execution | 28 |
| 15.Systemized Logistics | 29 |

**1. INTRODUCTION**

Today’s highly increasing competitiveness over the industry demands high quality and most consistent products with a competitive price. To address this challenge number of industries considering various new product designs and integrated manufacturing techniques in parallel with the use of automated devices.

One of the remarkable and influential movesfor getting the solutions of above mentioned challenge is the **Industrial automation.** In industry reinforcement, learning-based robots are used to perform various tasks.

Automation in the manufacturing industry has evolved from the use of basic hydraulic and pneumatic systems to today’s modern robots. Most industrial operations are automated with the goal of boosting production and reducing the cost of labor. Since its inception, industrial automation has made great advances among activities that were previously carried out manually. A manufacturing organization that uses the latest technologies to fully automate its processes typically sees improved efficiency, production of high-quality products, and reduced labor and production costs.

In 1913, Ford Motor Company introduced a car production assembly line which is considered one of the pioneer types of automation in the manufacturing industry. Before then, a car was built by a team of skilled and unskilled workers. Production automation improved Ford’s production rates and increased its profits. The assembly line and mass car production were the first of their kind globally. It reduced the car assembly time from 12 hours per car to about one and a half hours per car.

During the 1930s, Japan was a leader in developing components that facilitated industrial manufacturing automation. One company developed the first micro-switch, protective relays, and a highly accurate electrical timer. By this time, the rest of the world had started appreciating automation and significant research and development had occurred, like the solid-state proximity switch. Between 1939 and 1945 during the Second World War, automation was highly used in fighter airplanes, landing crafts, warships, and tanks.

Japan surrendered to the US and allied forces in 1945 and an industrial rebuilding program was initiated. The program relied on new and superior technologies as opposed to the old-fashioned manufacturing methods that were being used by the rest of the world. Hence, Japan became the world leader in industrial automation. Automobile companies like Honda, Toyota, and Nissan could produce numerous high quality and reliable cars. They offered standard features that were classified as extras by other car manufacturers as well as competitive prices that triggered the success of Japan’s automobile industry.

The current industrial robots feature high-quality computing capabilities, improved operational degrees of freedom, and vision systems. However, they can only operate in highly structured environments and still require a certain level of human intervention. Moreover, they are quite inflexible and highly-specialized for use in small and medium-sized industries, so industrial automation is usually better suited to large manufacturers and long production runs. Automation in the manufacturing industry relies on computer and software capabilities to automate, integrate and optimize different components of manufacturing systems. As a result, it is also referred to as computer integrated manufacturing.

Thanks to the inception and evolution of industrial automation in manufacturing industries, the world enjoys high-quality products and better energy, resource, and raw materials utilization. Contrary to what most people believe, the manufacturing industry is set to create more jobs with the use of robots, which will continue to drive the operations and benefits of industrial automation.

Since automation is such a fragmented business, all the larger (multi-billion $) companies are mostly a conglomeration of products and services; each product segment generates relatively small volume, but lumped together they form sizable businesses.

Fig.1 – automation using robots.



**2. LITERATURE SURVEY**

Automation describes a wide range of technologies that reduce human intervention in processes. Human intervention is reduced by predetermining decision criteria, subprocess relationships, and related actions — and embodying those predeterminations in machines.

Automation, includes the use of various equipment and control systems such as machinery, processes in factories, boilers, and heat-treating ovens, switching on telephone networks, steering, and stabilization of ships, aircraft, and other applications and vehicles with reduced human intervention.

Automation covers applications ranging from a household thermostat controlling a boiler, to a large industrial control system with tens of thousands of input measurements and output control signals. Automation has also found space in the banking sector. In control complexity, it can range from simple on-off control to multi-variable high-level algorithms.

In the simplest type of an automatic control loop, a controller compares a measured value of a process with a desired set value and processes the resulting error signal to change some input to the process, in such a way that the process stays at its set point despite disturbances. This closed-loop control is an application of negative feedback to a system. The mathematical basis of control theory was begun in the 18th century and advanced rapidly in the 20th.

Automation has been achieved by various means including mechanical, hydraulic, pneumatic, electrical, electronic devices, and computers, usually in combination. Complicated systems, such as modern factories, airplanes, and ships typically use all these combined techniques. The benefit of automation includes labor savings, reducing waste, savings in electricity costs, savings in material costs, and improvements to quality, accuracy, and precision.

The World Bank's World Development Report 2019 shows evidence that the new industries and jobs in the technology sector outweigh the economic effects of workers being displaced by automation.

job losses and downward mobility blamed on Automation has been cited as one of many factors in the resurgence of nationalist, protectionist and populist politics in the US, UK and France, among other countries since the 2010s.

The term *automation*, inspired by the earlier word *automatic* (coming from *automaton*), was not widely used before 1947, when Ford established an automation department. It was during this time that industry was rapidly adopting feedback controllers, which were introduced in the 1930s.

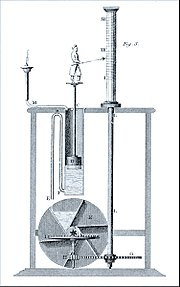


Fig 2. First automation device

It was a preoccupation of the Greeks and Arabs (in the period between about 300 BC and about 1200 AD) to keep accurate track of time. In Ptolemaic Egypt, about 270 BC, Ctesibius described a float regulator for a water clock, a device not unlike the ball and cock in a modern flush toilet. This was the earliest feedback controlled mechanism. The appearance of the mechanical clock in the 14th century made the water clock and its feedback control system obsolete.

The Persian Banū Mūsā brothers, in their *Book of Ingenious Devices* (850 AD), described a number of automatic controls. Two-step level controls for fluids, a form of discontinuous variable structure controls, was developed by the Banu Musa brothers. They also described a feedback controller. The design of feedback control systems up through the Industrial Revolution was by trial-and-error, together with a great deal of engineering intuition. Thus, it was more of an art than a science. It was not until the mid-19th century that the stability of feedback control systems were analyzed using mathematics, the formal language of automatic control theory.

The centrifugal governor was invented by Christiaan Huygens in the seventeenth century, and used to adjust the gap between millstones**.**

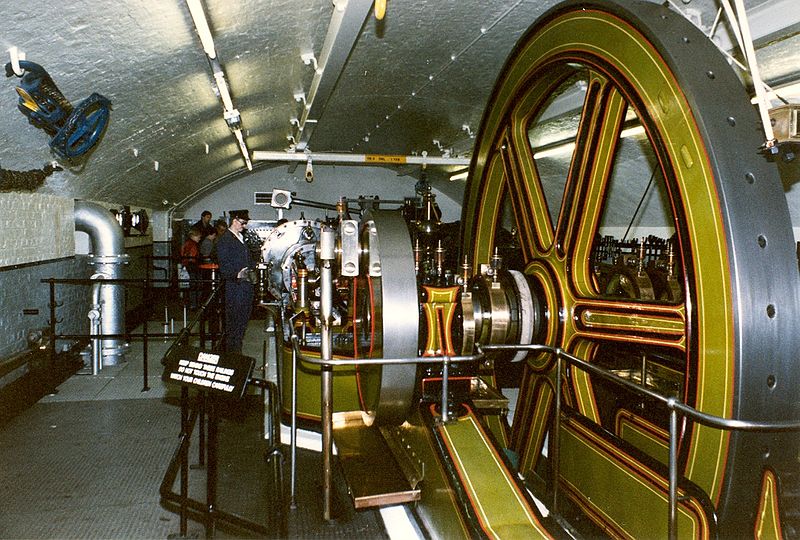


Fig.3 – self-driven machines for grain mills

The introduction of prime movers, or self-driven machines advanced grain mills, furnaces, boilers, and the steam engine created a new requirement for automatic control systems including temperature regulators (invented in 1624; see Cornelius Drebbel), pressure regulators (1681), float regulators (1700) and speed control devices. Another control mechanism was used to tent the sails of windmills. It was patented by Edmund Lee in 1745. Also in 1745, Jacques de Vaucanson invented the first automated loom. Around 1800, Joseph Marie Jacquard created a punch-card system to program looms.

In 1771 Richard Arkwright invented the first fully automated spinning mill driven by water power, known at the time as the water frame. An automatic flour mill was developed by Oliver Evans in 1785, making it the first completely automated industrial process.

A flyball governor is an early example of a feedback control system. An increase in speed would make the counterweights move outward, sliding a linkage that tended to close the valve supplying steam, and so slowing the engine

A centrifugal governor was used by a Mr. Bunce of England in 1784 as part of a model steam crane. The centrifugal governor was adopted by James Watt for use on a steam engine in 1788 after Watt's partner Boulton saw one at a flour mill Boulton & Watt were building. The governor could not actually hold a set speed; the engine would assume a new constant speed in response to load changes. The governor was able to handle smaller variations such as those caused by fluctuating heat load to the boiler. Also, there was a tendency for oscillation whenever there was a speed change. As a consequence, engines equipped with this governor were not suitable for operations requiring constant speed, such as cotton spinning.

Several improvements to the governor, plus improvements to valve cut-off timing on the steam engine, made the engine suitable for most industrial uses before the end of the 19th century. Advances in the steam engine stayed well ahead of science, both thermodynamics and control theory. The governor received relatively little scientific attention until James Clerk Maxwell published a paper that established the beginning of a theoretical basis for understanding control theory.

### 20th century

Relay logic was introduced with factory electrification, which underwent rapid adaption from 1900 through the 1920s. Central electric power stations were also undergoing rapid growth and the operation of new high-pressure boilers, steam turbines and electrical substations created a large demand for instruments and controls. Central control rooms became common in the 1920s, but as late as the early 1930s, most process controls were on-off. Operators typically monitored charts drawn by recorders that plotted data from instruments. To make corrections, operators manually opened or closed valves or turned switches on or off. Control rooms also used color-coded lights to send signals to workers in the plant to manually make certain changes.

The development of the electronic amplifier during the 1920s, which was important for long-distance telephony, required a higher signal-to-noise ratio, which was solved by negative feedback noise cancellation. This and other telephony applications contributed to the control theory. In the 1940s and 1950s, German mathematician Irmgard Flügge-Lotz developed the theory of discontinuous automatic controls, which found military applications during the Second World War to fire control systems and aircraft navigation systems.

Controllers, which were able to make calculated changes in response to deviations from a set point rather than on-off control, began being introduced in the 1930s. Controllers allowed manufacturing to continue showing productivity gains to offset the declining influence of factory electrification.

Factory productivity was greatly increased by electrification in the 1920s. U.S. manufacturing productivity growth fell from 5.2%/yr 1919–29 to 2.76%/yr 1929–41. Alexander Field notes that spending on non-medical instruments increased significantly from 1929 to 1933 and remained strong thereafter.

The First and Second World Wars saw major advancements in the field of mass communication and signal processing. Other key advances in automatic controls include differential equations, stability theory and system theory (1938), frequency domain analysis (1940), ship control (1950), and stochastic analysis (1941).

Starting in 1958, various systems based on solid-state  digital logic modules for hard-wired programmed logic controllers (the predecessors of programmable logic controllers (PLC)) emerged to replace electro-mechanical relay logic in industrial control systems for process control and automation, including early Telefunken/AEG Logistat, Siemens Simatic, Philips/Mullard/Valvo [de] Norbit, BBC Sigmatronic, ACEC Logacec, Akkord [de] Estacord, Krone Mibakron, Bistat, Datapac, Norlog, SSR, or Procontic systems.

In 1959 Texaco's Port Arthur Refinery became the first chemical plant to use digital control. Conversion of factories to digital control began to spread rapidly in the 1970s as the price of computer hardware fell.

Fig. 4



**Reinforcement Learning**

Reinforcement learning is the training of machine learning models to make a sequence of decisions. The agent learns to achieve a goal in an uncertain, potentially complex environment. In reinforcement learning, an artificial intelligence faces a game-like situation. The computer employs trial and error to come up with a solution to the problem. To get the machine to do what the programmer wants, the artificial intelligence gets either rewards or penalties for the actions it performs. Its goal is to maximize the total reward.

Although the designer sets the reward policy–that is, the rules of the game–he gives the model no hints or suggestions for how to solve the game. It’s up to the model to figure out how to perform the task to maximize the reward, starting from totally random trials and finishing with sophisticated tactics and superhuman skills. By leveraging the power of search and many trials, reinforcement learning is currently the most effective way to hint machine’s creativity. In contrast to human beings, artificial intelligence can gather experience from thousands of parallel gameplays if a reinforcement learning algorithm is run on a sufficiently powerful computer infrastructure.

### Examples of reinforcement learning

Applications of reinforcement learning were in the past limited by weak computer infrastructure. However, as Gerard Tesauro’s backgamon AI superplayer developed in 1990’s shows, progress did happen. That early progress is now rapidly changing with powerful new computational technologies opening the way to completely new inspiring applications.

Training the models that control autonomous cars is an excellent example of a potential application of reinforcement learning. In an ideal situation, the computer should get no instructions on driving the car. The programmer would avoid hard-wiring anything connected with the task and allow the machine to learn from its own errors. In a perfect situation, the only hard-wired element would be the reward function.

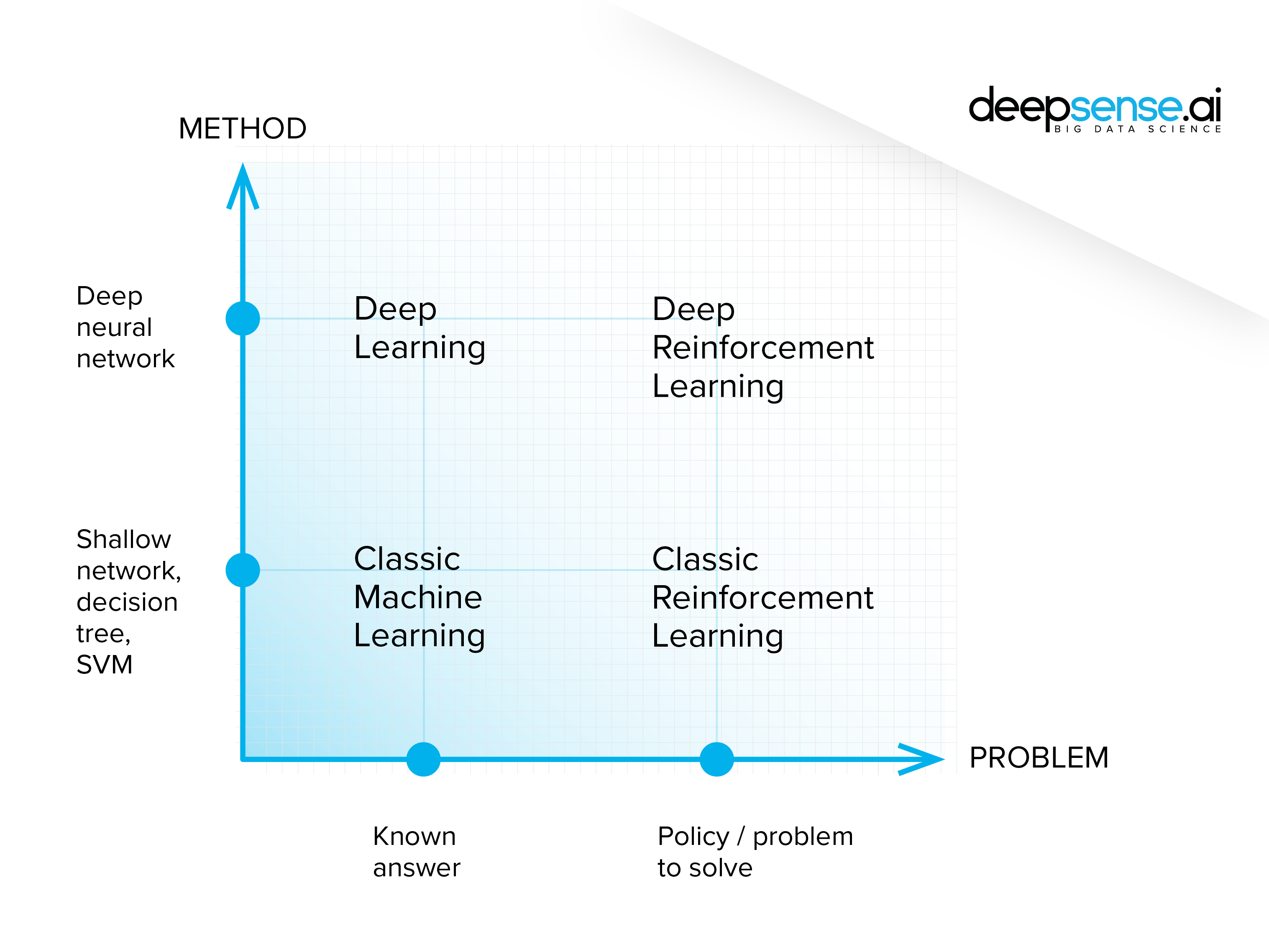
* For example, in usual circumstances we would require an autonomous vehicle to put safety first, minimize ride time, reduce pollution, offer passengers comfort and obey the rules of law. With an autonomous race car, on the other hand, we would emphasize speed much more than the driver’s comfort. The programmer cannot predict everything that could happen on the road. Instead of building lengthy “if-then” instructions, the programmer prepares the reinforcement learning agent to be capable of learning from the system of rewards and penalties. The agent (another name for reinforcement learning algorithms performing the task) gets rewards for reaching specific goals.
* Another example: deepsense.ai took part in the “Learning to run” project, which aimed to train a virtual runner from scratch. The runner is an advanced and precise musculoskeletal model designed by the Stanford Neuromuscular Biomechanics Laboratory. Learning the agent how to run is a first step in building a new generation of prosthetic legs, ones that automatically recognize people’s walking patterns and tweak themselves to make moving easier and more effective. While it is possible and has been done in Stanford’s labs, hard-wiring all the commands and predicting all possible patterns of walking requires a lot of work from highly skilled programmers**.**

**Challenges with reinforcement learning**

The main challenge in reinforcement learning lays in preparing the simulation environment, which is highly dependent on the task to be performed. When the model has to go superhuman in Chess, Go or Atari games, preparing the simulation environment is relatively simple. When it comes to building a model capable of driving an autonomous car, building a realistic simulator is crucial before letting the car ride on the street. The model has to figure out how to brake or avoid a collision in a safe environment, where sacrificing even a thousand cars comes at a minimal cost. Transferring the model out of the training environment and into to the real world is where things get tricky.  
Scaling and tweaking the neural network controlling the agent is another challenge. There is no way to communicate with the network other than through the system of rewards and penalties.This in particular may lead to *catastrophic forgetting*, where acquiring new knowledge causes some of the old to be erased from the network , published during the International Conference on Machine Learning).  
Yet another challenge is reaching a local optimum – that is the agent performs the task as it is, but not in the optimal or required way. A “jumper” jumping like a kangaroo instead of doing the thing that was expected of it-walking-is a great example, and is also one that can be found in our recent blog post.  
Finally, there are agents that will optimize the prize without performing the task it was designed for. An interesting example can be found in the OpenAI video below, where the agent learned to gain rewards, but not to complete the race.

**Distinguish between machine learning, deep learning and reinforcement**

In fact, there should be no clear divide between machine learning, deep learning and reinforcement learning. It is like a parallelogram – rectangle – square relation, where machine learning is the broadest category and the deep reinforcement learning the most narrow one.  
In the same way, reinforcement learning is a specialized application of machine and deep learning techniques, designed to solve problems in a particular way.

**Fig.5**

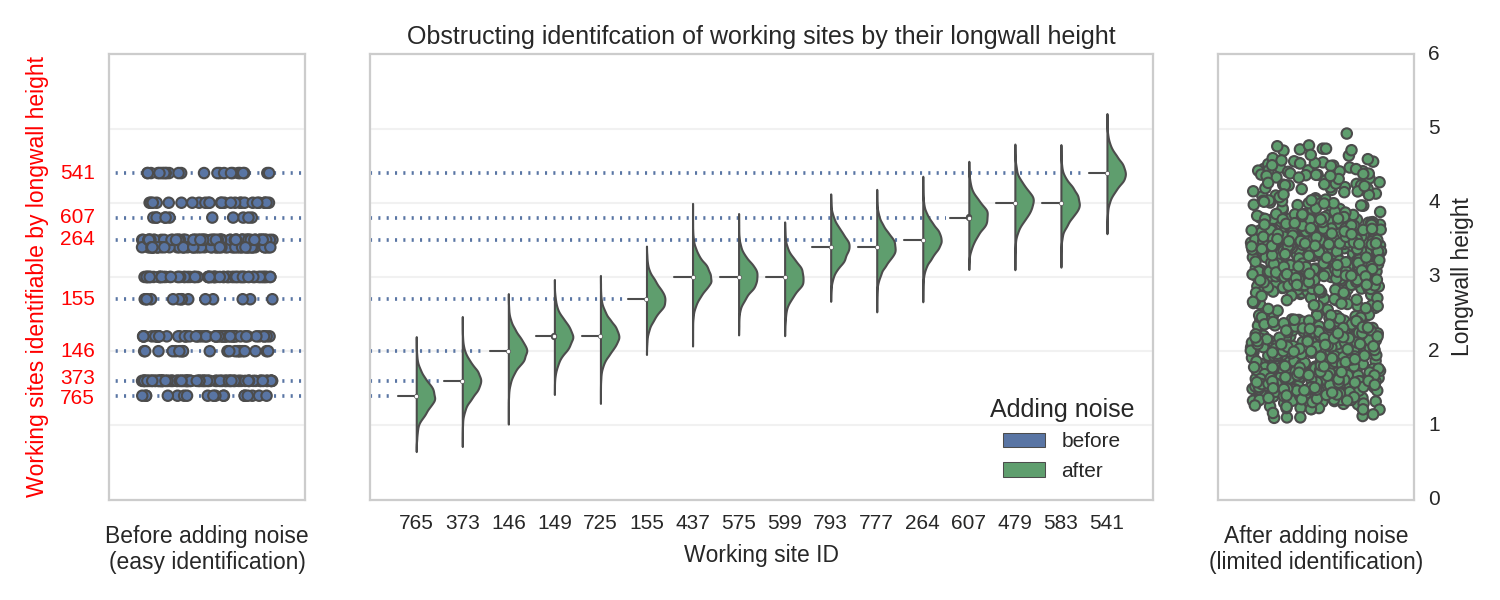
Although the ideas seem to differ, there is no sharp divide between these subtypes. Moreover, they merge within projects, as the models are designed not to stick to a “pure type” but to perform the task in the most effective way possible. So “what precisely distinguishes machine learning, deep learning and reinforcement learning” is actually a tricky question to answer.

* **Machine learning** – is a form of AI in which computers are given the ability to progressively improve the performance of a specific task with data, without being directly programmed ( this is Arthur Lee Samuel’s definition. He coined the term “machine learning”, of which there are two types, supervised and unsupervised machine learning

**Supervised machine learning** happens when a programmer can provide a label for every training input into the machine learning system.

* Example **–** by analyzing the historical data taken from coal mines, deepsense.ai prepared an automated system for predicting dangerous seismic events up to 8 hours before they occur. The records of seismic events were taken from 24 coal mines that had collected data for several months. The model was able to recognize the likelihood of an explosion by analyzing the readings from the previous 24 hours.

Fig.6



*Some of the mines can be exactly identified by their main working height values. To obstruct the identification, we added some Gaussian noise*

From the AI point of view, a single model was performing a single task on a clarified and normalized dataset.

**Unsupervised learning** takes place when the model is provided only with the input data, but no explicit labels. It has to dig through the data and find the hidden structure or relationships within. The designer might not know what the structure is or what the machine learning model is going to find.

* **An example we employed was for churn prediction.** We analyzed customer data and designed an algorithm to group similar customers. However, we didn’t choose the groups ourselves. Later on, we could identify high-risk groups (those with a high churn rate) and our client knew which customers they should approach first.
* **Another example of unsupervised learning** is anomaly detection, where the algorithm has to spot the element that doesn’t fit in with the group. It may be a flawed product, potentially fraudulent transaction or any other event associated with breaking the norm.

**Deep learning** consists of several layers of neural networks, designed to perform more sophisticated tasks. The construction of deep learning models was inspired by the design of the human brain, but simplified. Deep learning models consist of a few neural network layers which are in principle responsible for gradually learning more abstract features about particular data.

Although deep learning solutions are able to provide marvelous results, in terms of scale they are no match for the human brain. Each layer uses the outcome of a previous one as an input and the whole network is trained as a single whole. The core concept of creating an artificial neural network is not new, but only recently has modern hardware provided enough computational power to effectively train such networks by exposing a sufficient number of examples. Extended adoption has brought about frameworks like TensorFlow, Keras and PyTorch, all of which have made building machine learning models much more convenient.

* **Example:** deepsense.ai designed a deep learning-based model for the National Oceanic and Atmospheric Administration (NOAA). It was designed to recognize Right whales from aerial photos taken by researchers. For further information about this endangered species and deepsense.ai’s work with the NOAA, From a technical point of view, recognizing a particular specimen of whales from aerial photos is pure deep learning. The solution consists of a few machine learning models performing separate tasks. The first one was in charge of finding the head of the whale in the photograph while the second normalized the photo by cutting and turning it, which ultimately provided a unified view (a passport photo) of a single whale.

****

**Fig.7**

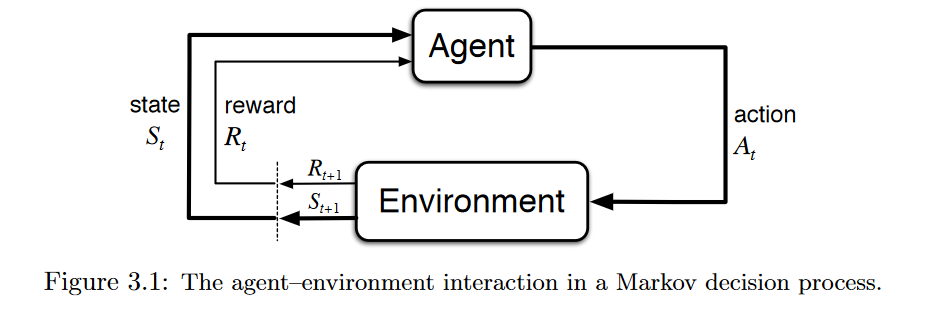
The third model was responsible for recognizing particular whales from photos that had been prepared and processed earlier. A network composed of 5 million neurons located the blowhead bonnet-tip. Over 941,000 neurons looked for the head and more than 3 million neurons were used to classify the particular whale. That’s over 9 million neurons performing the task, which may seem like a lot, but pales in comparison to the more than 100 billion neurons at work in the human brain. We later used a similar deep learning-based solution to diagnose diabetic retinopathy.

**Reinforcement learning**, as stated above employs a system of rewards and penalties to compel the computer to solve a problem by itself. Human involvement is limited to changing the environment and tweaking the system of rewards and penalties. As the computer maximizes the reward, it is prone to seeking unexpected ways of doing it. Human involvement is focused on preventing it from exploiting the system and motivating the machine to perform the task in the way expected. Reinforcement learning is useful when there is no “proper way” to perform a task, yet there are rules the model has to follow to perform its duties correctly. Take the road code, for example.

* **Example:** By tweaking and seeking the optimal policy for deep reinforcement learning, we built an agent that in just 20 minutes reached a superhuman level in playing Atari games. Similar algorithms in principal can be used to build AI for an autonomous car or a prosthetic leg. In fact, one of the best ways to evaluate the reinforcement learning approach is to give the model an Atari video game to play, such as Arkanoid or Space Invaders. According to Google Brain’s Marc G. Bellemare, who introduced Atari video games as a reinforcement learning benchmark, “although challenging, these environments remain simple enough that we can hope to achieve measurable progress as we attempt to solve them”.

**3.WORKING PRINCIPLE**

RL, known as a semi-supervised learning model in machine learning, is a technique to allow an agent to take actions and interact with an environment so as to maximize the total rewards. RL is usually modeled as a [Markov Decision Process](https://en.wikipedia.org/wiki/Markov_decision_process)(MDP).



Imagine a baby is given a TV remote control at your home (environment). In simple terms, the baby (agent) will first observe and construct his/her own representation of the environment (state). Then the curious baby will take certain actions like hitting the remote control (action) and observe how would the TV response (next state). As a non-responding TV is dull, the baby dislike it (receiving a negative reward) and will take less actions that will lead to such a result(updating the policy) and vice versa. The baby will repeat the process until he/she finds a policy (what to do under different circumstances) that he/she is happy with (maximizing the total (discounted) rewards).

If you walk down the street shouting out the names of every object you see — garbage truck! bicyclist! sycamore tree! — most people would not conclude you are smart. But if you go through an obstacle course, and you show them how to navigate a series of challenges to get to the end unscathed, they would.

Most machine learning algorithms are shouting names in the street. They perform perceptive tasks that a person can do in under a second. But another kind of AI — deep reinforcement learning — is strategic. It learns how to take a series of actions in order to reach a goal. That’s powerful and smart — and it’s going to change a lot of industries.

Two industries on the cusp of AI transformations are manufacturing and supply chain. The ways we make and ship stuff are heavily dependent on groups of machines working together, and the efficiency and resiliency of those machines are the foundation of our economy and society. Without them, we can’t buy the basics we need to live and work.

Startups like [Covariant](https://covariant.ai/), [Ocado’s Kindred](https://www.ocadogroup.com/investors/ocado-acquires-kindred-systems-and-haddington-dynamics) and [Bright Machines](https://www.brightmachines.com/) are using machine learning and reinforcement learning to change how machines are controlled in factories and warehouses, solving inordinately difficult challenges such as getting robots to detect and pick up objects of various sizes and shapes out of bins, among others. They are attacking enormous markets: The industrial control and automation market was worth [$152 billion](https://www.marketsandmarkets.com/Market-Reports/factory-industrial-automation-sme-smb-market-541.html) last year, while logistics automation was valued at more than [$50 billion](https://www.globenewswire.com/fr/news-release/2021/02/02/2167982/0/en/The-logistics-automation-market-was-valued-at-USD-52-19-billion-in-2020-and-is-expected-to-be-USD-104-23-billion-in-2026-registering-a-CAGR-of-12-42-during-the-forecast-period-2021.html).

**Deep reinforcement learning consistently produces results that other machine learning and optimization tools are incapable of.**

As a technologist, you need a lot of things to make deep reinforcement learning work. The first piece to think about is how you will get your deep reinforcement learning agent to practice the skills you want it to acquire. There are only two ways — with real data or through simulations. Each approach has its own challenge: Data must be collected and cleaned, while simulations must be built and validated.

Some examples will illustrate what this means. In 2016, GoogleX advertised its robotic “arm farms” — spaces filled with robot arms that were learning to grasp items and teach others how to do the same — which was one early way for a reinforcement learning algorithm to practice its moves in a real environment and measure the success of its actions. That feedback loop is necessary for a goal-oriented algorithm to learn: It must make sequential decisions and see where they lead.

In many situations, it is not feasible to build the physical environment where a reinforcement learning algorithm can learn. Let’s say you want to test different strategies for routing a fleet of thousands of trucks moving goods from many factories to many retail outlets. It would be very expensive to test all possible strategies, and those tests would not just cost money to run, but the failed runs would lead to many unhappy customers.

For many large systems, the only possible way to find the best action path is with simulation. In those situations, you must create a digital model of the physical system you want to understand in order to generate the data reinforcement learning needs. These models are called, alternately, digital twins, simulations and reinforcement-learning environments. They all essentially mean the same thing in manufacturing and supply chain applications.

Recreating any physical system requires domain experts who understand how the system works. This can be a problem for systems as small as a single fulfillment center for the simple reason that the people who built those systems may have left or died, and their successors have learned how to operate but not reconstruct them.

Many simulation software tools offer low-code interfaces that enable domain experts to create digital models of those physical systems. This is important, because domain expertise and software engineering skills often cannot be found in the same person.

Why would you go through all this trouble for a single algorithm? Because deep reinforcement learning consistently produces results that other machine learning and optimization tools are incapable of. [DeepMind](https://deepmind.com/research/case-studies/alphago-the-story-so-far) used it, of course, to beat the world champion of the board game of Go. Reinforcement learning was part of the algorithms that were integral to achieving breakthrough results with chess, protein folding and Atari games. Likewise, [OpenAI](https://openai.com/projects/five/) trained deep reinforcement learning to beat the best human teams at Dota 2.

Just like deep artificial neural networks began to find business applications in the mid-2010s, after Geoffrey Hinton was hired by Google and Yann LeCun by Facebook, so too, deep reinforcement learning will have an increasing impact on industries. It will lead to quantum improvements in robotic automation and system control on the same order as we saw with Go. It will be the best we have, and by a long shot.

The consequence of those gains will be immense increases in efficiency and cost savings in manufacturing products and operating supply chains, leading to decreases in carbon emissions and worksite accidents. And, to be clear, the chokepoints and challenges of the physical world are all around us. Just in the last year, our societies have been hit by multiple supply chain disruptions due to COVID, lockdowns, the Suez Canal debacle and extreme weather events.

Zooming in on COVID, even after the vaccine was developed and approved, many countries have had trouble producing it and distributing it quickly. These are manufacturing and supply chain problems that involve situations we could not prepare for with historical data. They required simulations to predict what would happen, as well as how we could best address crises when they do occur, as Michael Lewis illustrated in his recent book “[The Premonition](https://www.amazon.com/Premonition-Pandemic-Story-Michael-Lewis-ebook/dp/B08V91YY8R).”

It is precisely this combination of constraints and novel challenges that take place in factories and supply chains that reinforcement learning and simulation can help us solve more quickly. And we are sure to face more of them in the future.

**What you need to know before applying RL to your problem**

There are several things needed before RL can be applied:

* Understanding your problem: You do not necessarily need to use RL in your problem and sometimes you just cannot use RL. You may want to check if your problem has some of the following characteristics before deciding to use RL: a) trial-and-error (can be learned to do better by receiving feedback from the environment); b)delayed rewards; c)can be modeled as MDP; d)your problem is a control problem.
* A simulated environment: Lots of iterations are needed before a RL algorithm to work. I am sure that you don’t want to see a RL agent trying different things in a self-driving car on a highway, right? Therefore, a simulated environment that can correctly reflect the real world is needed.
* MDP: You world need to formulate your problem into a MDP. You need to design the state space, action space, reward function and so on. Your agent will do what it is rewarded to do under the constraints. You may not get the results you want if you design the things differently.
* Algorithms: There are different RL algorithms you can choose and questions to ask yourself. You want to directly find out the policy or you want to learn the value function? You want to go model-free or model-based? Do you need to combine other kinds of deep neural network or methods to solve your problems?

**Intuitions from other disciplines**

RL has a very close relationship with psychology, biology and neuroscience. If you think about it, what a RL agent does is just trial-and-error: it learns how good or bad its actions are based on the rewards it receives from the environment. [And this is exactly how human learns to make a decision](http://www.princeton.edu/~yael/ICMLTutorial.pdf). Besides, the exploration and exploitation problem, credit assignment problem, attempts to model the environment are also something we face in our everyday life.

The Economics theory can also shed some light on RL. In particular, the analysis of multi-agent reinforcement learning (MARL) can be understood from the perspectives of game theory, which is a research area developed by John Nash to understand the interactions of agents in a system. In addition to game theory, MARL, Partially Observable Markov Decision Process (POMDP) could also be useful to understand other economic topics like [market structure](https://en.wikipedia.org/wiki/Market_structure) (e.g.monopoly, oligopoly, etc), [externality](https://en.wikipedia.org/wiki/Externality) and [information asymmetry](https://en.wikipedia.org/wiki/Information_asymmetry).

**What could RL possibly achieve in the future**

RL still has lots of problems and cannot be used easily. Yet, as long as more efforts are put in solving the problems, RL would be influential and impactful in the following ways:

* Assisting human: Maybe it is too much to say RL can one day evolve into artificial general intelligence (AGI), but RL surely has the potential to assist and work with human. Just imagine a robot or a virtual assistant working with you and taking your actions into its considerations to take actions in order to achieve a common goal. Wouldn’t it be great?
* Understanding the consequences of different strategies: Life is so amazing because time will not go back and things just happen once. Yet, sometimes we would like to know how things could be different (at least in the short term) if I took a different action? Or would Croatia has a greater chance to win the 2018 World Cup if the coach used another strategy? Of course, to achieve this we would need to model the environment, transition functions and so on perfectly and also analyse the interactions between the agents, which seems to be impossible at the moment.

**Control of Robots**

Learning a control strategy completely from scratch is exactly what this final section is about. There are situations where no analytical model is available to control a process, simply because the process is much too complicated to model. In these cases, we are not worried about optimizing a policy, because that policy does not exist yet. We are actually worried about discovering the policy entirely starting with absolute zero knowledge about the system.

This is clearly a difficult task: it requires a massive amount of data, and is normally only considered for processes that can be simulated. Starting a policy search from scratch means large explorations of the action space, which can be very lengthy and costly on a real machine, not to mention potentially dangerous.

Industrial robots are very easy to simulate, they can also run without wasting any byproduct, and they provide a great wealth of data. Motion trajectory generation is a potential case. But the same could be said for a wind turbine park, where a specific strategy to reduce the mutual disturbances between adjacent turbines could be studied. We could discover a way to align the individual yaw angles in order to maximize the overall park’s output power generation, instead of simply maximizing the individual turbine’s margin without considering the interactions between them. The outcome would be very profitable.

**4.ADVANTAGES**

1. Reinforcement learning can be used to solve very complex problems that cannot be solved by conventional techniques.
2. This technique is preferred to achieve long-term results, which are very difficult to achieve.
3. This learning model is very similar to the learning of human beings. Hence, it is close to achieving perfection.
4. The model can correct the errors that occurred during the training process.
5. Once an error is corrected by the model, the chances of occurring the same error are very less.
6. It can create the perfect model to solve a particular problem.
7. Robots can implement reinforcement learning algorithms to learn how to walk.
8. In the absence of a training dataset, it is bound to learn from its experience.
9. Reinforcement learning models can outperform humans in many tasks. DeepMind’s AlphaGo program, a reinforcement learning model, beat the world champion *Lee Sedol* at the game of *Go* in March 2016.
10. Reinforcement learning is intended to achieve the ideal behavior of a model within a specific context, to maximize its performance.
11. It can be useful when the only way to collect information about the environment is to interact with it.
12. Reinforcement learning algorithms maintain a balance between exploration and exploitation. Exploration is the process of trying different things to see if they are better than what has been tried before. Exploitation is the process of trying the things that have worked best in the past. Other learning algorithms do not perform this balance.

**5.DISADVANTAGES**

1. Reinforcement learning as a framework is wrong in many different ways, but it is precisely this quality that makes it useful.
2. Too much reinforcement learning can lead to an overload of states, which can diminish the results.
3. Reinforcement learning is not preferable to use for solving simple problems.
4. Reinforcement learning needs a lot of data and a lot of computation. It is data-hungry. That is why it works really well in video games because one can play the game again and again and again, so getting lots of data seems feasible.
5. Reinforcement learning assumes the world is Markovian, which it is not. The Markovian model describes a sequence of possible events in which the probability of each event depends only on the state attained in the previous event.
6. The curse of dimensionality limits reinforcement learning heavily for real physical systems. According to Wikipedia, the curse of dimensionality refers to various phenomena that arise when analyzing and organizing data in high-dimensional spaces that do not occur in low-dimensional settings such as the three-dimensional physical space of everyday experience.
7. Another disadvantage is the curse of real-world samples. For example, consider the case of learning by robots. The robot hardware is usually very expensive, suffers from wear and tear, and requires careful maintenance. Repairing a robot system is costs a lot.
8. To solve many problems of reinforcement learning, we can use a combination of reinforcement learning with other techniques rather than leaving it altogether. One popular combination is Reinforcement learning with Deep Learning.

**6.APPLICATIONS**

A variety of problems can be solved using reinforcement learning. Some of them are game-playing, robotics, and many other fields.

As I mentioned earlier, reinforcement learning is the best technology used for game playing. It can even beat world champions.

Reinforcement learning can be used effectively to determine the best move to make in a game, depending on several different factors. It is very handy in games like Chess, Go, etc.

Using reinforcement learning, we can improve and personalize the gaming experience in real-time. It is the algorithm that can solve different games and sometimes achieve super-human performance.

This technology is used for the learning of robots. Robots are trained using the trial and error method with human supervision. Reinforcement learning teaches robots new tasks while retaining prior knowledge.

E-commerce websites like Amazon can use reinforcement learning to solve their problems to generate the maximum revenue by displaying the most relevant ads to interested buyers.

Self-driving cars also implement some reinforcement learning algorithms. Reinforcement learning can also be applied in optimizing chemical reactions.

Industrial automation, trading stock prices forecasting, news recommendations, etc. are some other applications of reinforcement learning.

**1.Improving Decision-Making Robots with Reinforcement Learning**

Reinforcement Learning (LR) is a state-of-the-art machine learning technique that attempts to train ML models for advanced decision-making and strategy-learning. When a model goes through reinforcement learning, it uses trial and error to find a solution to a complex problem. In other words, **the machine gets rewarded or punished for the actions that it takes to achieve a goal, which is a final reward.**

These ML technique has been widely adopted by the gaming industry. In one case, an RL model was used for playing the advanced RPG old video game, Abbey of Crime by itself. The model was capable of taking decisions, learning, and completing the game successfully.

Minecraft, a modern popular video game, organized a competition called MineRL, which encouraged players to program RL models to play the Minecraft game. AWS is also organizing an autonomous Robocar competition, which promises to improve self-driving cars through ML and RL models.

RL helps a machine to learn to perform tasks without knowing much about how to do them at first. When reinforcement learning algorithms run through a massive number of problem-solving quests, the model/machine will attain incredible skills.

RL helps a machine to learn to perform tasks without knowing much about how to do them at first. When reinforcement learning algorithms run through a massive number of problem-solving quests, the model/machine will attain incredible skills.

Aside from games, RL will also shape other industries. For example, **a robot programmed with RL situated in an unknown maze-like environment will observe, navigate, and learn through the process.**The next time the robot goes through the maze, it will be able to make automatic decisions that were previously learned.



**Fig.9**

**2.Safer Robot Collaboration and Productivity with Cobots**

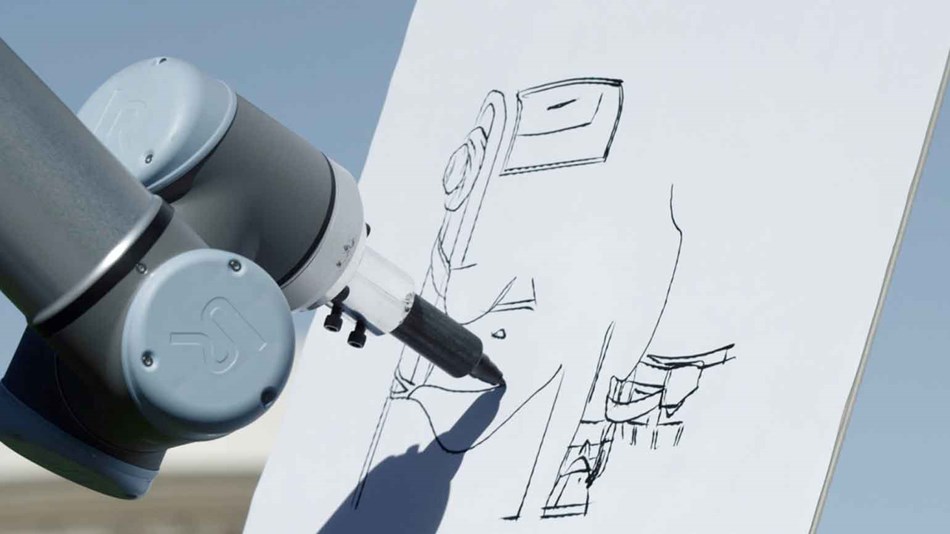
[Amazon](https://www.amazon.com/) took a big leap in the automation industry when they acquired the warehouse robotics [Canvas Technology](https://canvas.technology/) startup, in early 2019. Amazon was able to build new warehouse bots thanks to sophisticated computer vision powered by Canvas Technology.

The new and improved autonomous system is already allowing Amazon warehouse workers to work safely alongside [Collaboration Robots (Cobots)](https://www.lanner-america.com/blog/vision-cobots-reshaping-factory-automation/). The technology is also helping track every movement of every product in the warehouse, which intends to maximize efficiency and productivity.

[Cobots](https://www.lanner-america.com/blog/vision-cobots-reshaping-factory-automation/) play a big role in industries such as manufacturing or laboratories. These robots are intended to work alongside humans, as opposed to the caged big industrial robots, which can be dangerous. Cobots are autonomous systems that can pick and place items, pack them, inject, take analysis, and do a lot more. The power of the Cobots is limited to avoid accidents. They also have monitored stops and can keep track of motion and speed.

**These robots can be configured with so much precision that they can even sketch!**

Fig10



**3.Deepmind to cool Google Data Centers**

Google's data centres contain thousands of servers that power popular services including Google Search, Gmail and YouTube. Ensuring that they run reliably and efficiently is mission-critical. We've designed our AI agents and the underlying control infrastructure from the ground up with safety and reliability in mind, and use eight different mechanisms to ensure the system will behave as intended at all times.

Fig.11



Every five minutes, our cloud-based AI pulls a snapshot of the data center cooling system from thousands of sensors and feeds it into our deep neural networks, which predict how different combinations of potential actions will affect future energy consumption. The AI system then identifies which actions will minimize the energy consumption while satisfying a robust set of safety constraints. Those actions are sent back to the data center, where the actions are verified by the local control system and then implemented.

**4.Manufacturing**

Product differentiation, product assembling, and product distribution are difficult jobs and require high-level qualified labor. Availability and shortage of skilled labor is the biggest concern. Manufacturing companies with heavy production schedules require automated machines that can optimize the production time and production cost. Reinforcement learning can help heavy-duty machines in analyzing the best practices for building qualitative products.

A robot performing tasks with deep RL can read and develop from committed mistakes. And the more it interacts with the environment, the more it can analyze and provide optimal suggestions to advance the production processes automation.

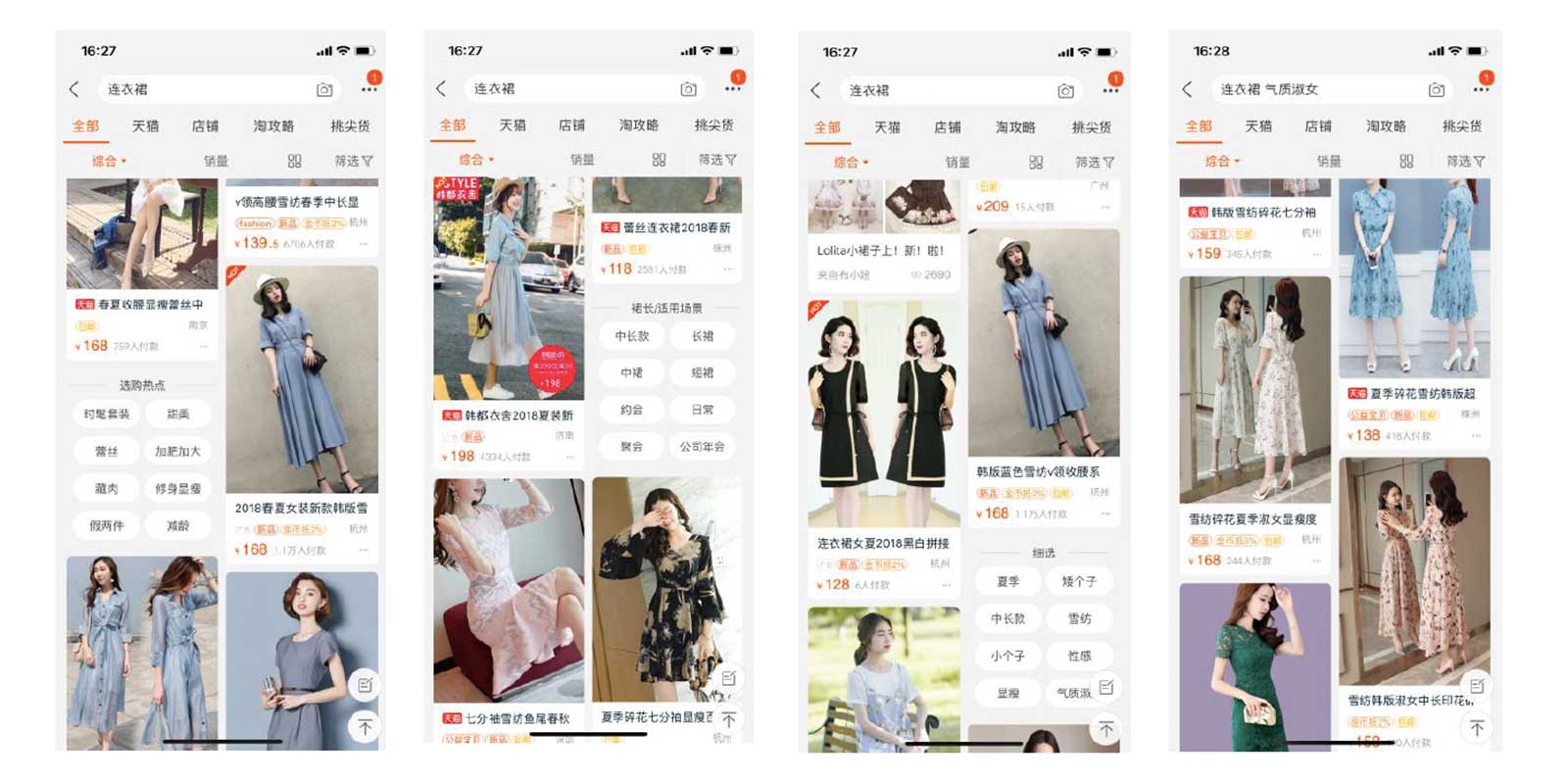
Fig.12



**5.Product Marketing**

Text mining and creating[personalized recommendation engines](https://www.martechadvisor.com/articles/customer-experience-2/recommendation-engines-how-amazon-and-netflix-are-winning-the-personalization-battle/) provide advantages in digital marketing for advertising the right products to the right customers. With advanced analytics, the data insights provide an advantage in customer categorization and targeting. With the application of reinforcement learning, businesses can increase the chance of conversion by providing personalized product recommendations on search tool by analyzing and categorizing real-time data. It helps in reducing advertising costs and constraints in the advertising budget.

70% of the organizations state that 30% of the revenue is spent on product marketing to reach target audience. And in most cases, the ROI on product marketing is negative. A personalized recommendation data engine can be a solution in product marketing. With reinforcement learning, the machine or system can analyze, store, and gain knowledge from the repeated actions of the consumers, which helps in advertising the right product to the target audience at a convenient time to close a lead.

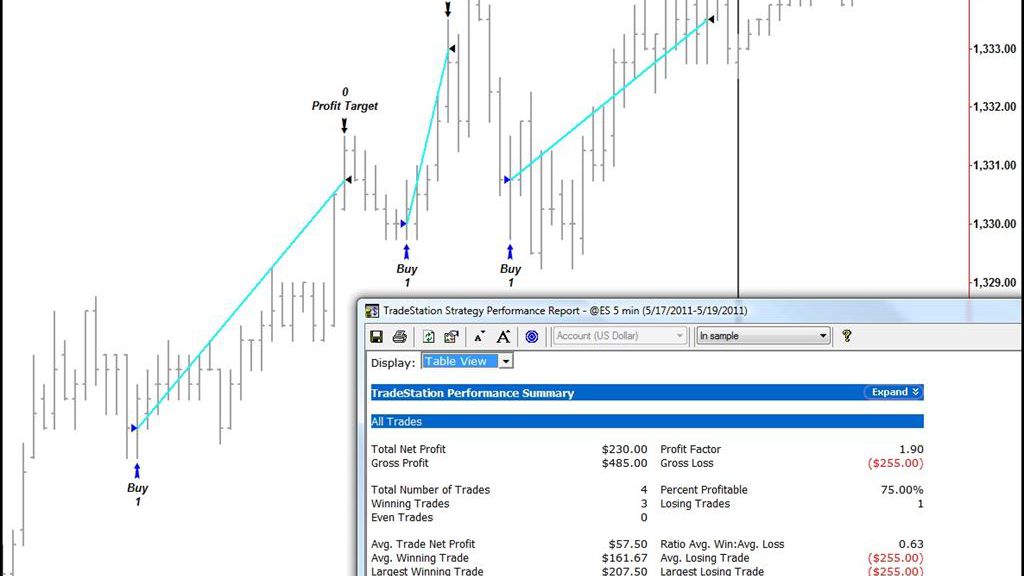


**Fig.13**

**6.Trade Execution**

Optimal liquidation is the biggest concern of traders in the capital and equity markets. Traders can use advanced analytics to forecast the real-time price fluctuations in the market sphere. Integration of deep machine reinforcement learning as a vital tool for understanding market trends and detecting grouped shares with low costs helps in perfect trade execution with minimum risks.

The applications of reinforcement learning helps in building optimal trade execution systems that foresee the flexible time and dynamic behaviour in trade transactions. By understanding market microstructures, the system provides optimal trade execution in both favourable and unfavourable situations.



**Fig.14**

**7.Systemized Logistics**

The execution and perfection in the coordination of inventory policy management systems practiced by various suppliers and manufacturers to reduce the shipment delivery time is a dream of every product manufacturer and supplier. With reinforcement learning the optimized inventory differentiation, stocking, and logistics scheduling is possible. It provides an advantage over time and cost. Retrieving products into warehouse, minimum space utilization, and handling warehouse operations can be performed by machines with deep reinforcement learning. Optimizing [delivery management system](https://www.theseus.fi/bitstream/handle/10024/57102/Vuorinen_Antti.pdf?sequence=1&isAllowed=y) with reinforcement learning provides an advantage in using Q-learning based split delivery system that delivers all the customers with one vehicle by reducing time and cost.

The growth of reinforcement learning as a tool of machine learning has been the last option in industry usage to solve complex problems that are rising every day with changing market dynamics. With increasing intelligent systems, reinforcement learning for industrial applications is unfolding advanced intelligent solutions to tackle complex problems. Industries are significantly realizing the importance of reinforcement learning in their operations which helps them in for being more customer-centric. The future goal of industries using reinforcement learning would be a 100 % return on investment.



Fig.15

**7.CONCLUSION**

Despite training difficulties, reinforcement learning finds its way to be effectively used in real business scenarios. Generally, RL is valuable when searching for optimal solutions in a constantly changing environment is needed.

Reinforcement learning is used for operations automation, machinery and equipment control and maintenance, energy consumption optimization. The finance industry also acknowledged the capabilities of reinforcement learning for powering AI-based training systems. Although trial-and-error training of robots is time-consuming, it allows robots to better evaluate real-world situations, use their skills for completing tasks, or reacting to unexpected consequences appropriately. In addition, RL provides opportunities for eCommerce players in terms of revenue optimization, fraud prevention, and customer experience enhancement via personalization.

For years robotics, advanced analytics, and automation has been a major part of the manufacturing industry. The increasing scale of adoption of AI in manufacturing seems more like an evolution, rather than an industry disruption. [Technology is already here](https://spd.group/ai-development/) and more massive implementation is a matter of time. According to [McKinsey](https://www.mckinsey.com/), by 2025 smart factories will generate $37 trillion.

**8.REFERENCES**

* <https://deepsense.ai/what-is-reinforcement-learning-the-complete-guide/#top>
* <https://towardsdatascience.com/applications-of-reinforcement-learning-in-real-world-1a94955bcd12>
* <https://medium.com/@ffrige/applying-machine-learning-to-industrial-automation-d6b48c7457c0>
* <https://neptune.ai/blog/reinforcement-learning-applications>
* <https://deepmind.com/blog/article/safety-first-ai-autonomous-data-centre-cooling-and-industrial-control>
* <https://pythonistaplanet.com/pros-and-cons-of-reinforcement-learning/>
* <https://spd.group/machine-learning/ai-and-ml-in-manufacturing-industry/>