INVESTIGATION OF UTILIZATION OF MACHINE LEARNING APPLICATIONS IN MECHANICAL ENGINEERING

Submitted in partial fulfillment of the requirement for the degree of

Bachelor of Science

In

Mechanical Engineering

Submitted by

MUHTADI MUNAWAR ZAHIN

Student ID# 141402

Under the supervision of

Prof. Dr. Md. NURUL ABSAR CHOWDHURY

Dean, FET

Department of Mechanical and Chemical Engineering (MCE)

Islamic University of Technology (IUT)

The Organization of Islamic Cooperation (OIC)

DECLARATION

I hereby declare that thesis entitled “Investigation of Utilization of Machine Learning Applications in Mechanical Engineering” is an authentic report of my study carried out as requirement for the award of degree B.Sc. (Mechanical Engineering) at Islamic University of Technology, Gazipur, Dhaka, under the supervision of PROF. DR. NURUL ABSAR CHOWDHURY, Dean, FET, IUT during January 2018 to October 2018.

The matter embodied in this thesis has not been submitted in part or full to any other institute for award of any degree.

Muhtadi Munawar Zahin

St.ID# 141402

This is to certify that the above statement made by the student concerned is correct to the best of my knowledge and belief.

PROF. DR. NURUL ABSAR CHOWDHURY

Dean, FET

Department of Mechanical and Chemical Engineering

Islamic University of Technology

Signature of Head of Department:

PROF. DR. MD. ZAHID HOSSAIN

Head

Department of Mechanical and Chemical Engineering

Islamic University of Technology

CERTIFICATE OF RESEARCH

The thesis titled "Investigation of Utilization of Machine Learning Applications in Mechanical Engineering" submitted by Muhtadi Munawar Zahin (141402) has been accepted as satisfactory in partial fulfillment of the requirement for the Degree of Bachelor of Science in Mechanical Engineering on November, 2018.

Supervisor

____

Prof. Dr. Md. Nurul Absar Chowdhury

Dean, FET

Department of Mechanical and Chemical Engineering (MCE)

Islamic University of Technology (IUT)

Head of the Department

____

Prof. Dr. Md. Zahid Hossain

Department of Mechanical and Chemical Engineering (MCE)

ACKNOWLEDGEMENT

I would like to thank Allah (SWT) first of all for giving us the patience, potential and blessings to enable the completion of this thesis, and this would not have been possible without His mercy and generosity.

I would like to thank my supervisor, **Prof. Dr. Nurul Absar Chowdhury** for his guidance, enthusiasm and unwavering willingness to spend his time and wisdom for the care of his students that is truly inspiring. I am extremely grateful to have been privy to his sincere advices and earnest support. On that note, I would also like to sincerely thank my respected teacher Assistant Professor Mr. Sayedus Salehin for his suggestions on the research topic and his guidance thereafter, despite being extremely busy at the time.

I also express my gratefulness to the Head of Mechanical and Chemical Engineering Department **Prof. Dr. Md. Zahid Hossain** and all the faculty of the department for providing supports that were truly necessary. I would especially like to thank Assistant Professor Dr. Arafat Ahmed Bhuiyan and his team for providing us access to the case studies which proved integral for the completion of this thesis. I would like give my sincere thanks to my teachers Mr. Abu Shahid Sujon and Mr. Ifat Rabbil Qudrat Ovi for their support and precious guidance.

I would like to thank my parents for the pivotal part they played in allowing me to grow up in a household filled with affection, respect, religious sentiment and grave importance on moral values, all of which allowed me come this far in life.

I am highly indebted to the above-mentioned people and without their encouragement, it would have been impossible to complete this dissertation.

Although I tried my best to complete this thesis flawlessly, I sincerely apologize for any mistakes that may have been made.

Abstract

Machine learning is a subfield of Artificial Intelligence (AI). Its goal is to make sense of a particular group of data and convert them into models that can be utilized by machine and man. The important aspect of Machine Learning is this: given that it is properly executed, a system aided by machine learning will be able to re-define its parameters of consideration and update its mode of action with minimal human interaction with continued accumulation of case data. This aspect of it makes it very suitable for optimization projects, including those related to Mechanical Engineering. In this thesis, an investigation was carried out to assess the prospects of utilizing Machine Learning applications as a comprehensive analytical tool in the research activities by the Islamic University of Technology's Mechanical and Chemical Department (IUT-MCE). It aimed to do so by conducting a project using Machine Learning and assessing the progress of said report to determine the possibilities of inducting the practice of utilizing Machine Learning applications in IUT-MCE and provide a possible route in order to make such an induction possible.

The results of the thesis show that while Machine Learning applications do have a potential to enhance the research capabilities of IUT-MCE, there is a need for extensive investment in the avenue with an expected delay of useful outcome by 5-6 years depending on the specific area of research. Nevertheless, the writers feel confident that utilizing Machine Learning applications will be worth the investment considering the benefits it is likely to bring.

TABLE OF CONTENTS

LIST OF FIGURES

Chapter 1: Introduction and Background Information

Introduction

Mechanical Engineering has been the oldest existing branch of engineering and the branch that covers the most areas of interest from a researcher's point of view. As such, it has had various methods analysis and tools to accommodate the various aspects of these needs [1].

However, there is always room for improvement and nowhere is this truer than in the ever-evolving world of natural research. Therefore every new techniques and applications that can be inducted in the research environment represent an opportunity to discover efficient methods for analysis and possibly a brand new avenue to obtain a perspective that had so far been unavailable.

Machine Learning is a relatively new avenue of computational methodology and it does not have a big history of collaborating with issues related to Mechanical Engineering [2]. However, it is for this reason that Machine Learning is a potent field of investigation which can yield much to the benefit of the future in the research of Mechanical Engineering.

In this paper, we, the writers, aim to work on introducing applications derived from Machine Learning in the realm of analytical research in the fields of Mechanical Engineering. The writers plan to do this by introducing an application of Mechanical Engineering over the course of the thesis, monitor the progress of not only the application itself, but of the circumstances related to it acquirement and operation. The results of the monitoring is to be utilized in recommendations regarding future plans to induct Machine Learning applications as viable research tools under the disposal of The Department of Mechanical and Chemical Engineering of the Islamic University of Technology (IUT-MCE).

Mechanical Engineering

Mechanical engineering is the discipline that applies engineering, physics, engineering mathematics, and materials science principles to design, analyze, manufacture, and maintain mechanical systems. It is one of the oldest and broadest of the engineering disciplines.

The mechanical engineering field requires an understanding of core areas including mechanics, dynamics, thermodynamics, material science, structural analysis and electricity. In addition to the core principles, tools such as Computer Aided Design (CAD) Computational Fluid Dynamics (CFD) and others are used. The avenues of products include, but are not limited to manufacturing planrs, industrial equipment and machinery, heating and cooling system, transport systems, robotics, medical devices and weapons. It is the branch of engineering that involves the design, production, and operation of machinery [3].

For the sake of better understanding aspects of Mechanical Engineering, short notes on its history and core disciplines have been provided in the following sections.

General History

The application of mechanical engineering can be seen in the archives of various ancient and medieval societies. In ancient Greece, the works of Archimedes (287–212 BC) influenced mechanics in the Western tradition and Heron of Alexandria (c. 10–70 AD) created the first steam engine (Aeolipile).[4] In China, Zhang Heng (78–139 AD) improved a water clock and invented a seismometer, and Ma Jun (200–265 AD) invented a chariot with differential gears. The medieval Chinese horologist and engineer Su Song (1020–1101AD) incorporated an escapement mechanism into his astronomical clock tower two centuries before escapement devices were found in medieval European clocks. He also invented the world's first known endless power-transmitting chain drive [5].

During the Islamic Golden Age (7th to 15th century), Muslim inventors made remarkable contributions in the field of mechanical technology. Al-Jazari, who was one of them, wrote his famous Book of Knowledge of Ingenious Mechanical Devices in 1206 and presented many mechanical designs. Al-Jazari is also the first known person to create devices such as the crankshaft and camshaft [6], which now form the basics of many mechanisms.

During the 17th century, important breakthroughs in the foundations of mechanical engineering occurred in England. Sir Isaac Newton formulated Newton's Laws of Motionand developed Calculus, the mathematical basis of physics. Newton was reluctant to publish his works for years, but he was finally persuaded to do so by his colleagues, such as Sir Edmond Halley, much to the benefit of all mankind. Gottfried Wilhelm Leibniz is also credited with creating Calculus during this time period.

During the early 19th century industrial revolution, machine tools were developed in England, Germany, and Scotland. This allowed mechanical engineering to develop as a separate field within engineering. They brought with them manufacturing machines and the engines to power them [7]. The first British professional society of mechanical engineers was formed in 1847 Institution of Mechanical Engineers, thirty years after the civil engineers formed the first such professional society Institution of Civil Engineers [8]. On the European continent, Johann von Zimmermann (1820–1901) founded the first factory for grinding machines in Chemnitz, Germany in 1848.

In the United States, the American Society of Mechanical Engineers (ASME) was formed in 1880, becoming the third such professional engineering society, after the American Society of Civil Engineers (1852) and the American Institute of Mining Engineers(1871) [9]. The first schools in the United States to offer an engineering education were the United States Military Academy in 1817, an institution now known as Norwich University in 1819, and Rensselaer Polytechnic Institute in 1825. Education in mechanical engineering has historically been based on a strong foundation in mathematics and science.

Thermodynamics

Thermodynamics is the science of the relationship between heat, work, temperature, and energy. In broad terms, thermodynamics deals with the transfer of energy from one place to another and from one form to another. The key concept is that heat is a form of energy corresponding to a definite amount of mechanical work [10].

Heat was not formally recognized as a form of energy until about 1798, when Count Rumford (Sir Benjamin Thompson), a British military engineer, noticed that limitless amounts of heat could be generated in the boring of cannon barrels and that the amount of heat generated is proportional to the work done in turning a blunt boring tool. Rumford’s observation of the proportionality between heat generated and work done lies at the foundation of thermodynamics. Another pioneer was the French military engineer Sadi Carnot, who introduced the concept of the heat-engine cycle and the principle of reversibility in 1824. Carnot’s work concerned the limitations on the maximum amount of work that can be obtained from a steam engine operating with a high-temperature heat transfer as its driving force. Later that century, these ideas were developed by Rudolf Clausius, a German mathematician and physicist, into the first and second laws of thermodynamics, respectively.

The most important laws of thermodynamics are:

- The zeroth law of thermodynamics: When two systems are each in thermal equilibrium with a third system, the first two systems are in thermal equilibrium with each other. This property makes it meaningful to use thermometers as the "third system" and to define a temperature scale.
- The first law of thermodynamics, or the law of conservation of energy: The change in a system's internal energy is equal to the difference between heat added to the system from its surroundings and work done by the system on its surroundings.
- The second law of thermodynamics: Heat does not flow spontaneously from a colder region to a hotter region, or, equivalently, heat at a given temperature cannot be converted entirely into work. Consequently, the entropy of a closed system, or heat energy per unit temperature, increases over time toward some maximum value. Thus, all closed systems tend toward an equilibrium state in which entropy is at a maximum and no energy is available to do useful work. This asymmetry between forward and backward processes gives rise to what is known as the "arrow of time."
- The third law of thermodynamics. The entropy of a perfect crystal of an elementin its most stable form tends to zero as the temperature approaches absolute zero. This allows an absolute scale for entropy to be established that, from a statistical point of view, determines the degree of randomness or disorder in a system.

Although thermodynamics developed rapidly during the 19th century in response to the need to optimize the performance of steam engines, the sweeping generality of the laws of thermodynamics makes them applicable to all physical and biological systems. In particular, the laws of thermodynamics give a complete description of all changes in the energy state of any system and its ability to perform useful work on its surroundings.

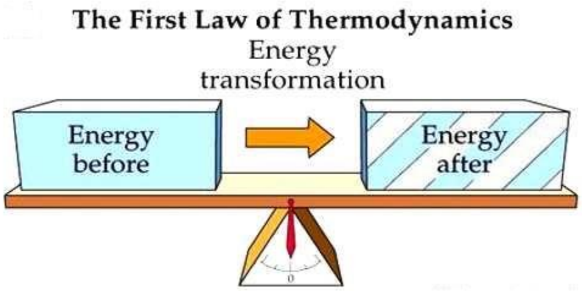

Figure 1 A depiction of the first law of Thermodynamics [11]

Thermodynamics is of two branches, classical thermodynamics that does not consider the conditions of individual atoms/molecules and statistical thermodynamics, which does consider conditions of individual atoms/ molecules

Mechanics

Mechanics describes the motion of macroscopic objects, from projectiles to part of machinery and astronomical objects, such as spacecraft, planets, stars and galaxies [12]. If the present state of an object is known it is possible to predict by the laws of Mechanics how it will move in the future (determinism) and how it has moved in the past (reversibility).

The earliest development of Mechanics is often referred to as Newtonian mechanics. It consists of the physical concepts employed by and the mathematical methods invented by Isaac Newton and Gottfried Wilhelm Leibniz and others in the 17th century to describe the motion of bodies under the influence of a system of forces.

Later, more abstract methods were developed, leading to the reformulations of Mechanics known as Lagrangian mechanics and Hamiltonian mechanics. These advances, made predominantly in the 18th and 19th centuries, extend substantially beyond Newton's work, particularly through their use of analytical mechanics. They are, with some modification, also used in all areas of modern physics.

Mechanics described up to this point can be attributed exclusively to sub-branch of Classical mechanics, which most engineering projects generally utilize in it analysis and calculation. It provides extremely accurate results when studying large objects that are not extremely massive and speeds not approaching the speed of light. When the objects being examined have about the size of an atom diameter, it becomes necessary to introduce the other major sub-field of mechanics: quantum mechanics. To describe velocities that are not small compared to the speed of light, special relativity is needed. In case that objects become extremely massive, General relativity becomes applicable.

Mechanics is an integral part of Mechanical Engineering, where its core principles are extensively used in various parts of the discipline. In fact all of the projects that can be attributed to Mechanical Engineering contain Mechanics in one form or another.

Material Science

Material Science is an interdisciplinary field that deals primarily with the design and discovery of new materials and unveiling previously unknown properties of materials already known, especially in their solid form.

Material Science was originated from analytical observations derived from physics and chemistry that was applied in a rigorous engineering manner and contains clear traces of these disciplines in its works and methodologies to this day.

Rustum Roy described Material Science in his paper [13] as the following:

“Materials science is a syncretic discipline hybridizing metallurgy, ceramics, solid-state physics, and chemistry. It is the first example of a new academic discipline emerging by fusion rather than fission.”

Most fields of studies have a founding father, such as Newton in physics and Lavoisier in chemistry. Materials science on the other hand has no central figure that set in motion materials studies. In the 1940s, wartime collaborations of multiple fields of study to produce technological advances became a structure to the future field of study that would become known as material science and engineering [14]. During the Cold War in the 1950s, US President Science Advisory Committee (PSAC) made materials a priority when it realized that materials were the limiting factor for advances in space and military technology. The Department of defense signed a contract with 5 universities (Harvard, MIT, Brown, Stanford, and Chicago) providing over $13 million for material research. Several institutions departments changed titles from "metallurgy" to "metallurgy and materials science" in 1960's [15].

In the early part of the 20th century, most engineering schools had a department of metallurgy and perhaps of ceramics as well. Much effort was expended on consideration of the austenite-martensite-cementite phases found in the iron-carbon phase diagram that underlies steel production. The fundamental understanding of other materials was not sufficiently advanced for them to be considered as academic subjects. In the post-WWII era, the systematic study of polymers advanced particularly rapidly. Rather than create new polymer science departments in engineering schools, administrators and scientists began to conceive of materials science as a new interdisciplinary field in its own right, one that considered all substances of engineering importance from a unified point of view. Northwestern University instituted the first materials science department in 1955.

Dr. Richard E. Tressler was an international leader in the development of high temperature materials. He pioneered high temperature fiber testing and use, advanced instrumentation and test methodologies for thermos-structural materials, and design and performance verification of ceramics and composites in high temperature aerospace, industrial and energy applications. He was founding director of the Center for Advanced Materials (CAM) which supported many faculty and students from the College of Earth and Mineral Science, the Eberly College of Science, the College of Engineering, the Materials Research Laboratory and the Applied Research Laboratories at Penn State on high temperature materials. His vision for interdisciplinary research played a key role in the creation of the Materials Research Institute. Tressler's contribution to materials science is celebrated with a Penn State lecture named in his honor.

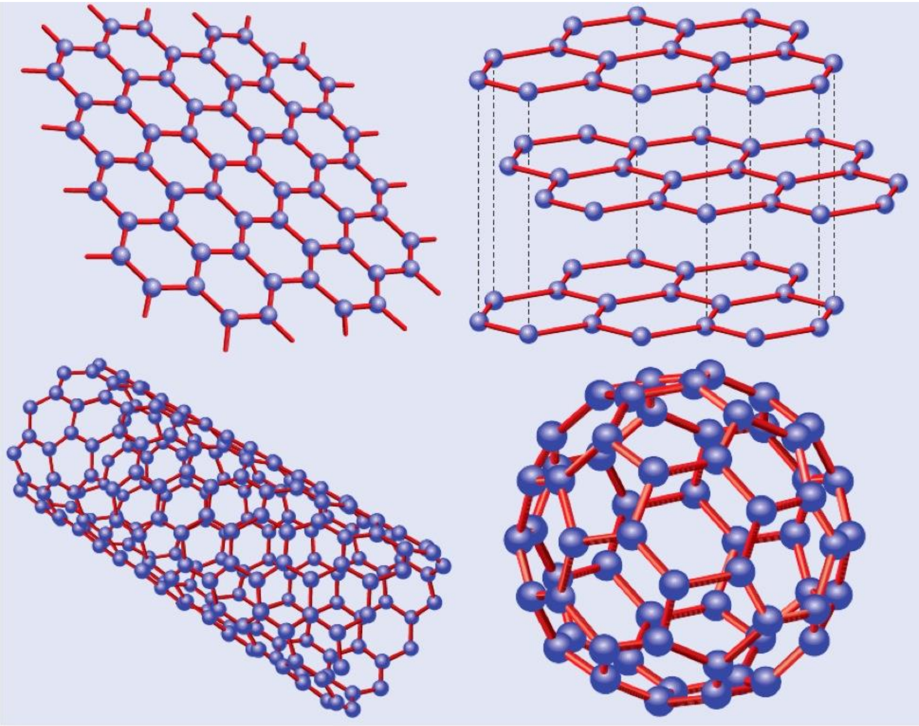

Figure 2 : Various structures of graphene [16]

Material Science has many areas of research at current times that is not limited to the study of metals, for example study of carbon allotropes, especially graphene [17] as shown above. The Open Quantum Materials Database (OQMD) [18], the project chosen to be set as a precedence in this thesis can be considered the result of applying Machine Learning applications in the field Material Science.

Structural Analysis

Structural analysis is the determination of the effects of loads on physical structures and their components. Structures subject to this type of analysis include all that must withstand loads, such as buildings, bridges, vehicles, furniture, attire, soil strata, prostheses and biological tissue. Structural analysis employs the fields of applied mechanics, materials science and applied mathematics to compute a structure's deformations, internal forces, stresses, support reactions, accelerations, and stability. The results of the analysis are used to verify a structure's fitness for use, often precluding physical tests. Structural analysis is thus a key part of the engineering design of structures [19].

The history of structural engineering dates back to at least 2700 BC when the step pyramid for Pharaoh Djoser was built by Imhotep, the first engineer in history known by name. Pyramids were the most common major structures built by ancient civilizations because it is a structural form which is inherently stable and can be almost infinitely scaled. [20]

No record exists of the first calculations of the strength of structural members or the behaviour of structural material, but the profession of structural engineer only really took shape with the industrial revolution and the re-invention of concrete. The physical sciences underlying structural engineering began to be understood in the Renaissance and have been developing ever since. [21]

The foundations of modern structural engineering were laid in the 17th century by Galileo Galilei, Robert Hooke and Isaac Newton with the publication of three great scientific works. In 1638 Galileo published *Dialogues Relating to Two New Science,* outlining the sciences of the strength of materials and the motion of objects (essentially defining gravity as a force giving rise to a constant acceleration). It was the first establishment of a scientific approach to structural engineering, including the first attempts to develop a theory for beams. This is also regarded as the beginning of structural analysis, the mathematical representation and design of building structures. This was followed in 1676 by Robert Hooke's first statement of Hooke's Law, providing a scientific understanding of elasticity of materials and their behavior under load. Eleven years later, in 1687, Sir Isaac Newton published *Philosophiae Naturalis Principia Mathematica*, setting out his Laws of Motion, providing for the first time an understanding of the fundamental laws governing structures [21].

To perform an accurate analysis a mechanical engineer must determine information such as structural loads, geometry, support conditions, and material properties. The results of such an analysis typically include support reactions, stresses and displacements. This information is then compared to criteria that indicate the conditions of failure. Advanced structural analysis may examine dynamic response, stability and non-linear behavior. There are three approaches to the analysis: the mechanics of materials approach (also known as strength of materials), the elasticity theory approach (which is actually a special case of the more general field of continuum mechanics), and the finite element approach. The first two make use of analytical formulations which apply mostly to simple linear elastic models, lead to closed-form solutions, and can often be solved by hand. The by and finite element approach is actually a numerical method for solving differential equations generated by theories of mechanics such as elasticity theory and strength of materials. However, the finite-element method depends heavily on the processing power of computers and is more applicable to structures of arbitrary size and complexity. Regardless of approach, the formulation is based on the same three fundamental relations: equilibrium, constitutive, and compatibility. The solutions are approximate when any of these relations are only approximately satisfied, or only an approximation of reality [20].

In modern times most analysis of structures are done by computational means, as the example below shows:

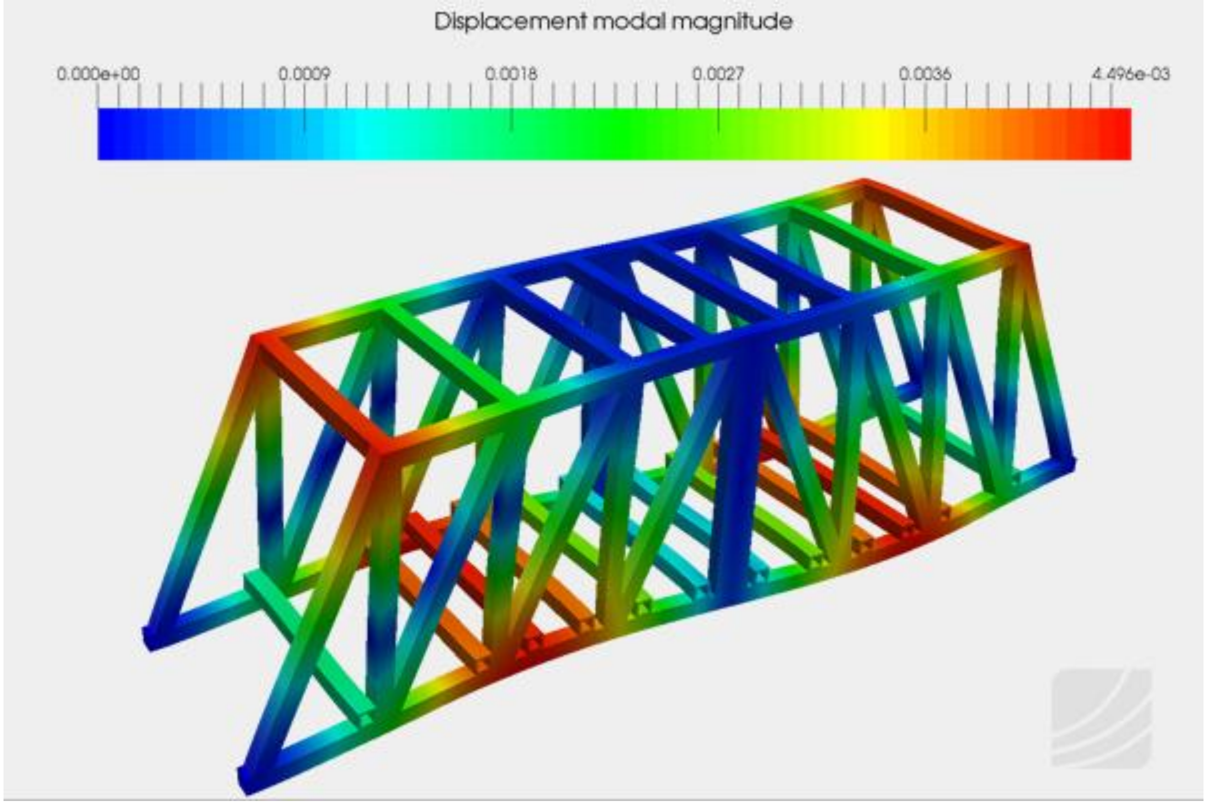

Figure 3: Example of Computational Structural Analysis [22]

Structural analysis is heavily linked with Material sciences. This relationship is a prime example of the trend noted in Mechanical Engineering of the inner disciplines being heavily inter-dependent.

Fluid Mechanics

Fluid mechanics is a branch of physics concerned with the mechanics of fluids (liquids, gases, and plasmas) and the forces on them. Fluid mechanics has a wide range of applications, most notably in mechanical engineering, . Fluid mechanics can be divided into two sub-divisions. It is a branch of continuum mechanics, a subject which models matter without using the information that it is made out of atoms; that is, it models matter from a macroscopic viewpoint rather than from microscopic. Fluid mechanics, especially fluid dynamics, is an active field of research with many problems that are partly or wholly unsolved. Fluid mechanics can be mathematically complex, and can best be solved by numerical methods, typically using computers. A modern discipline, called computational fluid dynamics (CFD) that will be discussed in the next section, is devoted to this approach to solving fluid mechanics problems. Particle image velocimetry, an experimental method for visualizing and analyzing fluid flow, also takes advantage of the highly visual nature of fluid flow [23].

The study of fluid mechanics goes back at least to the days of ancient Greece, when Archimedes investigated fluid statics and buoyancy and formulated his famous law known now as the Archimedes' principle, which was published in his work On Floating Bodies – generally considered to be the first major work on fluid mechanics. Rapid advancement in fluid mechanics began with Leonardo da Vinci (observations and experiments), Evangelista Torricelli (invented the barometer), Isaac Newton (investigated viscosity) and Blaise Pascal (researched hydrostatics, formulated Pascal's law), and was continued by Daniel Bernoulli with the introduction of mathematical fluid dynamics in Hydrodynamica (1739). Inviscid flow was further analyzed by various mathematicians Jean le Rond d'Alembert, Joseph Louis Lagrange, Pierre-Simon Laplace, Siméon Denis Poisson) and viscous flow was explored by a multitude of engineers including Jean Léonard Marie Poiseuille and Gotthilf Hagen. Further mathematical justification was provided by Claude-Louis Navier and George Gabriel Stokes in the Navier–Stokes equations, and boundary layers were investigated (Ludwig Prandtl, Theodore von Kármán), while various scientists such as Osborne Reynolds, Andrey Kolmogorov, and Geoffrey Ingram Taylor advanced the understanding of fluid viscosity and turbulence [24].

The two sub-divisions of Fluid Mechanics mentioned earlier are as follows:

- Fluid Statics: Also called Hydrostatics, it is the branch of fluid mechanics that studies fluids at rest. It embraces the study of the conditions under which fluids are at rest in stable equilibrium; and is contrasted with fluid dynamics, the study of fluids in motion. Hydrostatics offers physical explanations for many phenomena of everyday life, such as why atmospheric pressure changes with altitude, why wood and oil float on water, and why the surface of water is always level and horizontal whatever the shape of its container. Hydrostatics is fundamental to hydraulics, the engineering of equipment for storing, transporting and using fluids. It is also relevant to some aspect of geophysics and astrophysics (for example, in understanding plate tectonics and anomalies in the Earth's gravitational field), to meteorology, to medicine (in the context of blood pressure), and many other fields.

- Fluid Dynamics: It is the branch of fluid mechanics that deals with fluid flow—the science of liquids and gases in motion. Fluid dynamics offers a systematic structure—which underlies these practical disciplines—that embraces empirical and semi-empirical laws derived from flow measurement and used to solve practical problems. The solution to a fluid dynamics problem typically involves calculating various properties of the fluid, such as velocity, pressure, density, and temperature, as functions of space and time. It has several subdisciplines itself, including aerodynamics (the study of air and other gases in motion) and hydrodynamics (the study of liquids in motion). Fluid dynamics has a wide range of applications, including calculating forces and movements on aircraft, determining the mass flow rate of petroleum through pipelines, predicting evolving weather patterns, understanding nebulae in interstellar space and modeling explosions. Some fluid-dynamical principles are used in traffic engineering and crowd dynamics.

Figure 4: Barometer specimen [25]

Fluid mechanics deals with a variety of fluids which can be categorized in many ways. Inviscid and viscous, Newtonian are Non-Newtonian are just two of the popular categorizations. Fluid mechanics is a subject heavily influenced by numerical analysis, which is why computational analysis have become a norm in conducting investigations under fluid mechanics at present day.

Computational Fluid Dynamics (CFD)

Computational fluid dynamics (CFD) is a branch of fluid mechanics that uses numerical analysis and data structures to solve and analyze problems that involve fluid flows. CFD enables one to predict fluid flow, heat and mass transfer, and chemical reactions (explosions) and related phenomena. It is used in almost all industrial sectors: food processing, water treatment, marine engineering, automotive, aerodynamics, and gas turbine design. With the help of CFD software, fluid flow problems are analyzed faster than by testing, in more detail, at an earlier stage in the design cycle, for less money, and with lower risk [26].

The foundation of CFD is built on the Navier-Stokes equations [27], the set of partial differential equations that describe fluid flow. With CFD, the area of interest is subdivided into a large number of cells or control volumes. In each of these cells, the Navier-Stokes partial differential equations can be rewritten as algebraic equations that relate the velocity, temperature, pressure, and other variables, such as species concentrations, to the values in the neighboring cells. These equations are then solved numerically, yielding a complete picture of the flow down to the resolution of the grid. The resulting set of equations can then be solved iteratively, yielding a complete description of the flow throughout the domain [26].

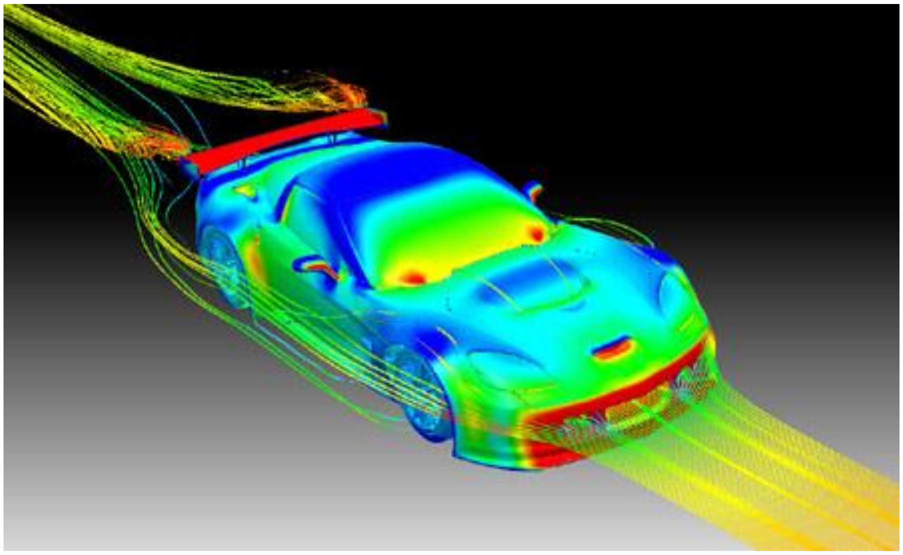

Figure 5: CFD graphic on a GT2 Corvette C.6.R body [28]

These techniques date back to the early 1970s, and the first commercial CFD software became available in the early 1980s. Since then, CFD has come a long way, and geometric flexibility has increased to the point where there are now very few geometries too complex to be represented accurately. Models have been developed for physical phenomena such as turbulence, multiphase flow, chemical reactions, and radiative heat transfer, and the usability of software has increased greatly, with powerful pre- and post-processors.

By solving the fundamental equations governing fluid flow processes, CFD provides information on important flow characteristics such as pressure loss, flow distribution, and mixing rates. CFD analysis complements traditional testing and experimentation, providing added insight and confidence in your designs. This results in better designs, lower risk, and faster time to the marketplace for product or processes.

CFD has become an integral part of the engineering design and analysis environment of many companies because of its ability to predict the performance of new designs or processes before they are manufactured or implemented. The use of CFD software can result in fewer iterations to the final design, shorter lead times, and fewer expensive prototypes to produce. CFD also encourages innovation because it enables cost-effective means for testing novel designs that would otherwise be too expensive and risky to investigate [26].

Solid Mechanics

Solid mechanics is the branch of continuum mechanics that studies the behavior of solid materials, especially their motion and deformation under the action of forces, temperature changes, phase changes, and other external or internal agents. Solid mechanics is fundamental for civil, aerospace, nuclear, biomedical and mechanical engineering, for geology, and for many branches of physics such as materials science [29]. It has specific applications in many other areas, such as understanding the anatomy of living beings, and the design of dental prostheses and surgical implants. One of the most common practical applications of solid mechanics is the Euler-Bernoulli beam equation. Solid mechanics extensively uses tensors to describe stresses, strains, and the relationship between them. The target of Solid mechanics can be ambiguous at times due to manner through which a solid is described under that. For example, a liquid may be modelled as a solid if it can support substantial shear stress over a significant time scale with respect to a natural process [30].

Solid mechanics developed in the outpouring of mathematical and physical studies following the great achievement of Newton in stating the laws of motion, although it has earlier roots. The need to understand and control the fracture of solids seems to have been a first motivation. Leonardo da Vinci sketched in his notebooks a possible test of the tensile strength of a wire. Galileo Galilei had investigated the breaking loads of rods under tension and concluded that the load was independent of length and proportional to the cross section area, this being a first step toward a concept of stress. He also investigated the breaking loads on beams that were suspended horizontally from a wall into which they were built. The English scientist Robert Hooke discovered in 1660, but published only in 1678, that for many materials the displacement under a load was proportional to force, thus establishing the notion of (linear) elasticity but not yet in a way that was expressible in terms of stress and strain. Edme Mariotte in France published similar discoveries in 1680 and, in addition, reached an understanding of how beams like those studied by Galileo resist transverse loadings or more precisely, resist the torques caused by those transverse loadings by developing extensional and compressional deformations, respectively, in material fibres along their upper and lower portions. It was for the Swiss mathematician and mechanician Jakob Bernoulli to observe, in the final paper of his life, in 1705, that the proper way of describing deformation was to give force per unit area, or stress, as a function of the elongation per unit length, or strain, of a material fibre under tension. The notion that there is an internal tension acting across surfaces in a deformed solid was expressed by the German mathematician and physicist Gottfried Wilhelm Leibniz in 1684 and Jakob Bernoulli in 1691. Also, Jakob Bernoulli and Euler introduced the idea that at a given section along the length of a beam there were internal tensions amounting to a net force and a net torque. Euler introduced the idea of compressive normal stress as the pressure in a fluid in 1752. The French engineer and physicist Charles-Augustin Coulomb was apparently the first to relate the theory of a beam as a bent elastic line to stress and strain in an actual beam, in a way never quite achieved by Bernoulli and, although possibly recognized, never published by Euler [30]. It was the great French mathematician Augustin-Louis Cauchy, originally educated as an engineer, who in 1822 formalized the concept of stress in the context of a generalized three-dimensional theory, showed its properties as consisting of a 3 × 3 symmetric array of numbers that transform as a tensor, derived the equations of motion for a continuum in terms of the components of stress, and developed the theory of linear elastic response for isotropic solids.The Austrian physicist Ludwig Boltzmann developed in 1874 the theory of linear viscoelastic stress-strain relations. In their most general form, these involve the notion that a step loading (a suddenly imposed stress that is subsequently maintained constant) causes an immediate strain followed by a time-dependent strain which, for different materials, either may have a finite limit at long time or may increase indefinitely with time [30].

The property of interest in Solid Mechanics is generally deformation: the amount of change in shape an object experiences with respect to its default shape. Analysts in solid mechanics uses models to describe the deformation of an object. Four basic models are as follows [31]:

- Elasticity: When an applied stress is removed, the material returns to its undeformed state. Linearly elastic materials, those that deform proportionally to the applied load, can be described by the linear elasticity equations such as Hooke's law.
- Viscoelasticity: These are materials that behave elastically, but also have damping: when the stress is applied and removed, work has to be done against the damping effects and is converted in heat within the material resulting in a hysteresis loop in the stress–strain curve. This implies that the material response has time-dependence.
- Plasticity: Materials that behave elastically generally do so when the applied stress is less than a yield value. When the stress is greater than the yield stress, the material behaves plastically and does not return to its previous state. That is, deformation that occurs after yield is permanent.
- Thermoelasticity: There is coupling of mechanical with thermal responses. In general, thermoelasticity is concerned with elastic solids under conditions that are neither isothermal nor adiabatic. The simplest theory involves the Fourier's law of heat conduction, as opposed to advanced theories with physically more realistic models.

Solid mechanics are analyzed using mathematical modelling and are thus subjected to Computational analysis as a norm at present time.

Mechatronics

Mechatronics is the synergistic combination of precision mechanical engineering, electronic control and systems thinking in the design of products and manufacturing processes. It relates to the design of systems, devices and products aimed at achieving an optimal balance between basic mechanical structure and its overall control [31]. The word "mechatronics" originated in Japanese-English and was created by Tetsuro Mori, an engineer of Yaskawa Electric Corporation. The word "mechatronics" was registered as trademark by the company in Japan with the registration number of "46-32714" in 1971. However, afterward the company released the right of using the word to public, and the word "mechatronics" spread to the rest of the world. Nowadays, the word is translated in each language and the word is considered as an essential term for industry [32].

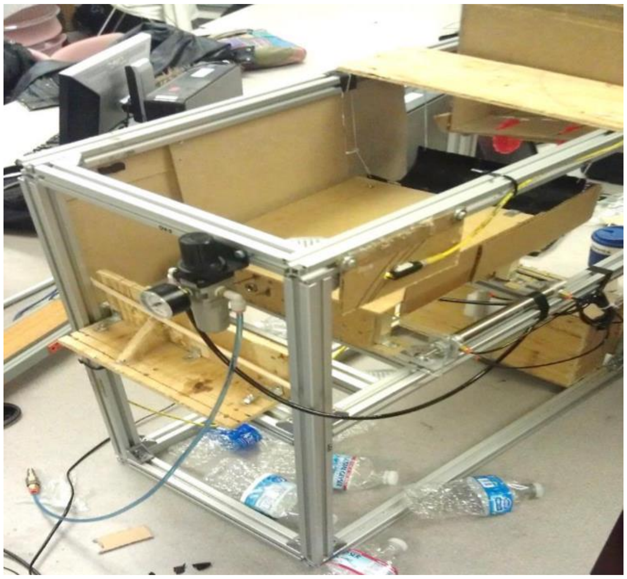

Figure 6: A water bottle crusher made by utilizing mechatronics [33]

The formal history of Mechatronics may be a short one, but the road to its conception is anything but.The industrial revolution enabled design of products and processes to convert and transmit ever-increasing amounts of energy for industrial activities. Engineering designs of this era performed operations of motion transmission, sensing, actuation, and computation using mechanical components e.g., cams, gears, levers, linkages, etc. Watt's flyball governor typifies the engineering designs of this era. Unfortunately, purely mechanical systems suffer from power amplification inability and energy losses due to tolerances, inertia, and friction. The semiconductor revolution leading to the development of integrated circuit (IC) technology created the next key impact on the design of engineering products. On the actuation side, through cost effective, miniaturized, power electronics for efficient power amplification, semiconductor technology provided a practical means of delivering required levels of power to mechanical devices via electrical means. Similarly, on the sensing side, semiconductor technology afforded an ability to condition and encode physical measurements as analog/digital signals. Furthermore, hardwired, onboard, analog/digital electronics provided rudimentary computational abilities to mechanical devices. Rapid developments in electro-mechanical sensing and actuation hardware, which had begun in an earlier era, further fueled the adoption of semiconductor technology in the design and operation of mechanical devices. In the closing decades of 20th century, as the semiconductor and information technology industries experienced explosive growths, the computing hardware became ubiquitous and cheap. A momentous transformation in the design and operation of mechanical devices and systems began to unfold, with information technology emerging as a technology enabler imparting intelligence to numerous products, processes, and machines. Microprocessors began replacing precision mechanical components, e.g., precision-machined camshaft, which in many applications functions as a timing device. In addition, programmability of microprocessors provided a versatile and flexible alternative to the hardwired analog/digital computational hardware. Integrated computer-electrical-mechanical devices now became capable of converting, transmitting, and processing both the physical energy and the virtual energy (information). Thus, in recent years, engineers began developing highly efficient products and processes by judicious selection and integration of sensors, actuators, signal conditioning, power electronics, decision and control algorithms, and computer hardware and software [34].

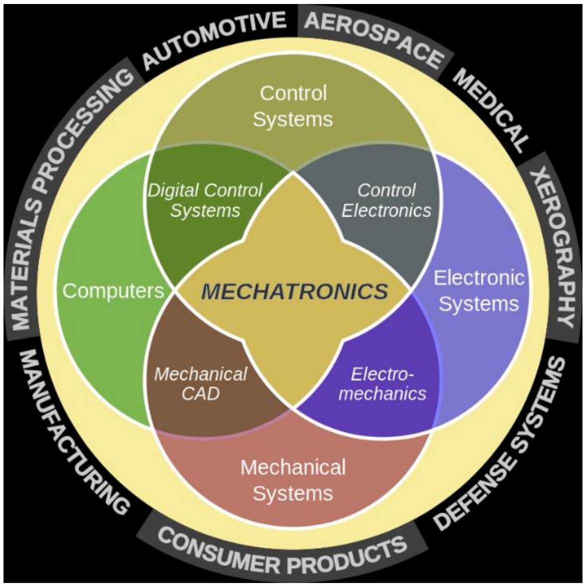

Figure 7 Aerial Euler diagram describing fields that make up Mechatronics [32]

An emerging variant of the field of Mechatronics is Biomechatronics, whose purpose is to integrate mechanical parts with a human being, usually in the form of removable gadgets such as an exoskeleton. This is the "real-life" version of cyberware. Another variant that we can consider is Motion control for Advanced Mechatronics, which presently is recognized as a key technology in mechatronics. The robustness of motion control will be represented as a function of stiffness and a basis for practical realization. Target of motion is parameterized by control stiffness which could be variable according to the task reference. However, the system robustness of motion always requires very high stiffness in the controller. Avionics is also considered a variant of mechatronics as it combines several fields such as electronics and telecom with Aerospace Engineering.

The Internet of things (IoT) [36] is the inter-networking of physical devices, embedded with electronics, software, sensors, actuators, and network connectivity which enable these objects to collect and exchange data. IoT and mechatronics are complementary. Many of the smart components associated with the Internet of Things will be essentially mechatronic. The development of the IoT is forcing mechatronics engineers, designers, practitioners and educators to research the ways in which mechatronic systems and components are perceived, designed and manufactured. This allows them to face up to new issues such as data security, machine ethics and the human-machine interface.

Machine Learning

Machine learning is a subfield of Artificial Intelligence (AI). Its goal is to make sense of a particular group of data and convert them into models that can be utilized by machine and man. The important aspect of Machine Learning is this: given that it is properly executed, a system aided by machine learning will be able to re-define its parameters of consideration and update its mode of action with minimal human interaction with continued accumulation of case data. This aspect of it makes it very suitable for optimization projects, including those related to Mechanical Engineering.

History

Arthur Samuel, an American pioneer in the field of computer gaming and artificial intelligence, coined the term "Machine Learning" in 1959 while at IBM. As a scientific endeavor, machine learning grew out of the quest for artificial intelligence. Already in the early days of AI as an academic discipline, some researchers were interested in having machines learn from data. They attempted to approach the problem with various symbolic methods, as well as what were then termed "neural networks"; these were mostly perceptrons and other models that were later found to be reinventions of the generalized linear models of statistics. Probabilistic reasoning was also employed, especially in automated medical diagnosis.

However, an increasing emphasis on the logical, knowledge-based approach caused a rift between AI and machine learning. Probabilistic systems were plagued by theoretical and practical problems of data acquisition and representation. By 1980, expert systems had come to dominate AI, and statistics was out of favor. Work on symbolic/knowledge-based learning did continue within AI, leading to inductive logic programming, but the more statistical line of research was now outside the field of AI proper, in pattern recognition and information retrieval. Neural networks research had been abandoned by AI and computer science around the same time. This line, too, was continued outside the AI/CS field, as "connectionism", by researchers from other disciplines including Hopfield, Rumelhart and Hinton. Their main success came in the mid-1980s with the reinvention of backpropagation [37]

Machine learning, reorganized as a separate field, started to flourish in the 1990s. The field changed its goal from achieving artificial intelligence to tackling solvable problems of a practical nature. It shifted focus away from the symbolic approaches it had inherited from AI, and toward methods and models borrowed from statistics and probability theory. It also benefited from the increasing availability of digitized information, and the ability to distribute it via the Internet.

Characteristics and Classification

Evolved from the study of pattern recognition and computational learning theory in artificial intelligence, machine learning explores the study and construction of algorithms that can learn from and make predictions on data, such algorithms overcome following strictly static program instructions by making data-driven predictions or decisions or through building a model from sample inputs. Machine learning is employed in a range of computing tasks where designing and programming explicit algorithms with good performance is difficult or infeasible; example applications include email filtering, detection of network intruders or malicious insiders working towards a data breach, optical character recognition (OCR), learning to rank, and computer vision [37].

Machine learning is closely related to (and often overlaps with) computational statistics, which also focuses on prediction-making through the use of computers. It has strong ties to mathematical optimization, which delivers methods, theory and application domains to the field. Machine learning is sometimes conflated with data mining, where the latter subfield focuses more on exploratory data analysis and is known as unsupervised learning. Machine learning can also be unsupervised and be used to learn and establish baseline behavioral profiles for various entities and then used to find meaningful anomalies.

Within the field of data analytics, machine learning is a method used to devise complex models and algorithms that lend themselves to prediction; in commercial use, this is known as predictive analytics. These analytical models allow researchers, data scientists, engineers, and analysts to "produce reliable, repeatable decisions and results" and uncover "hidden insights" through learning from historical relationships and trends in the data.

Machine learning tasks are typically classified into two broad categories, depending on whether there is a learning "signal" or "feedback" available to a learning system:

- Supervised learning: The computer is presented with example inputs and their desired outputs, given by a "teacher", and the goal is to learn a general rule that maps inputs to outputs. As special cases, the input signal can be only partially available, or restricted to special feedback:
    - Semi-supervised learning: the computer is given only an incomplete training signal: a training set with some (often many) of the target outputs missing.
    - Active learning: the computer can only obtain training labels for a limited set of instances (based on a budget), and also has to optimize its choice of objects to acquire labels for. When used interactively, these can be presented to the user for labeling.
    - Reinforcement learning: training data (in form of rewards and punishments) is given only as feedback to the program's actions in a dynamic environment, such as driving a vehicle or playing a game against an opponent.
- Unsupervised learning: No labels are given to the learning algorithm, leaving it on its own to find structure in its input. Unsupervised learning can be a goal in itself (discovering hidden patterns in data) or a means towards an end (feature learning).

Another categorization of machine learning tasks arises when one considers the desired output of a machine-learned system [39]:

- In classification, inputs are divided into two or more classes, and the learner must produce a model that assigns unseen inputs to one or more (multi-label classification) of these classes. This is typically tackled in a supervised manner. Spam filtering is an example of classification, where the inputs are email (or other) messages and the classes are "spam" and "not spam".
- In regression, also a supervised problem, the outputs are continuous rather than discrete.
- In clustering, a set of inputs is to be divided into groups. Unlike in classification, the groups are not known beforehand, making this typically an unsupervised task.
- Density estimation finds the distribution of inputs in some space.
- Dimensionality reduction simplifies inputs by mapping them into a lower-dimensional space. Topic modeling is a related problem, where a program is given a list of human language documents and is tasked with finding out which documents cover similar topics.

Among other categories of machine learning problems, learning to learn, learns its own inductive bias based on previous experience. Developmental learning, elaborated for robot learning, generates its own sequences (also called curriculum) of learning situations to cumulatively acquire repertoires of novel skills through autonomous self-exploration and social interaction with human teachers and using guidance mechanisms such as active learning, maturation, motor synergies, and imitation.

Underlying theory

A core objective of a learner is to generalize from its experience. Generalization in this context is the ability of a learning machine to perform accurately on new, unseen examples/tasks after having experienced a learning data set. The training examples come from some generally unknown probability distribution (considered representative of the space of occurrences) and the learner has to build a general model about this space that enables it to produce sufficiently accurate predictions in new cases [38]

The computational analysis of machine learning algorithms and their performance is a branch of theoretical computer science known as computational learning theory. Because training sets are finite and the future is uncertain, learning theory usually does not yield guarantees of the performance of algorithms. Instead, probabilistic bounds on the performance are quite common. The bias–variance decomposition is one way to quantify generalization error.

For the best performance in the context of generalization, the complexity of the hypothesis should match the complexity of the function underlying the data. If the hypothesis is less complex than the function, then the model has underfit the data. If the complexity of the model is increased in response, then the training error decreases. But if the hypothesis is too complex, then the model is subject to overfitting and generalization will be poorer.

In addition to performance bounds, computational learning theorists study the time complexity and feasibility of learning. In computational learning theory, a computation is considered feasible if it can be done in polynomial time. There are two kinds of time complexity results. Positive results show that a certain class of functions can be learned in polynomial time. Negative results show that certain classes cannot be learned in polynomial time.

The main approaches used in Machine Learning are as follows:

- Decision Tree Learning
- Assosciation rule Learning
- Support Vector machines
- Clustering
- Inductive logic Programming
- Reinforcement Learning
- Representation Learning
- Similarity and Metric Learning
- Sparse Dictionary Learning
- Generic algorithms
- Rule-based Machine Learning
- Deep Learning

Interconnectivity (Relations with other disciplines)

Machine learning is closely related to (and often overlaps with) computational statistics, which also focuses on prediction-making through the use of computers. It has strong ties to mathematical optimization, which delivers methods, theory and application domains to the field. Machine learning is sometimes conflated with data mining, where the latter subfield focuses more on exploratory data analysis and is known as unsupervised learning. Machine learning can also be unsupervised and be used to learn and establish baseline behavioral profiles for various entities and then used to find meaningful anomalies.

Machine learning and data mining often employ the same methods and overlap significantly, but while machine learning focuses on prediction, based on known properties learned from the training data, data mining focuses on the discovery of (previously) unknown properties in the data (this is the analysis step of knowledge discovery in databases). Data mining uses many machine learning methods, but with different goals; on the other hand, machine learning also employs data mining methods as "unsupervised learning" or as a preprocessing step to improve learner accuracy. Much of the confusion between these two research communities (which do often have separate conferences and separate journals, with some exceptions) comes from the basic assumptions they work with: in machine learning, performance is usually evaluated with respect to the ability to reproduce known knowledge, while in knowledge discovery and data mining (KDD) the key task is the discovery of previously unknown knowledge. Evaluated with respect to known knowledge, an uninformed (unsupervised) method will easily be outperformed by other supervised methods, while in a typical KDD task, supervised methods cannot be used due to the unavailability of training data.

Machine learning also has intimate ties to optimization: many learning problems are formulated as minimization of some loss function on a training set of examples. Loss functions express the discrepancy between the predictions of the model being trained and the actual problem instances (for example, in classification, one wants to assign a label to instances, and models are trained to correctly predict the pre-assigned labels of a set of examples). The difference between the two fields arises from the goal of generalization: while optimization algorithms can minimize the loss on a training set, machine learning is concerned with minimizing the loss on unseen samples.

Machine learning and statistics are closely related fields. According to Michael I. Jordan, the ideas of machine learning, from methodological principles to theoretical tools, have had a long pre-history in statistics. He also suggested the term data science as a placeholder to call the overall field. Leo Breiman distinguished two statistical modelling paradigms: data model and algorithmic model, wherein "algorithmic model" means more or less the machine learning algorithms like Random forest. Some statisticians have adopted methods from machine learning, leading to a combined field that they call statistical learning.

Open Quantum Materials Database

The Open Quantum Materials Database (OQMD) is a high-throughput database currently consisting of nearly 300,000 density functional theory (DFT) total energy calculations of compounds from the Inorganic Crystal Structure Database (ICSD) and decorations of commonly occurring crystal structures. To maximize the impact of these data, the entire database is being made available, without restrictions, at www.oqmd.org/download.

Due to its importance in this thesis, a more comprehensive review of the project is given below, courtesy Wolverton et al [18]:

“A foundational concept in materials science and engineering is the processing/structure/properties paradigm: processing determines structure, which in turn defines observed properties. The close ties between structure and properties are historically evident in the development of models to describe materials behavior (e.g., vacancy and dislocation theory). Currently, ab initio predictive models of properties based on structure are commonplace, which has enabled a growing trend in materials design: experimental synthesis in tandem with predictive modeling to facilitate the optimization of materials properties. The promise of such an approach is to dramatically reduce development time for novel materials with innovative design tools and methods. Indeed, accelerating materials design is the primary goal and motivation for the U.S. Materials Genome Initiative. A critical innovation toward accelerating materials design has been accurate and efficient first principles prediction of materials properties with density functional theory (DFT). Employing only quantum mechanical concepts and little experimental input, DFT allows one to predict properties of crystalline solids such as lattice parameters, magnetic moments, formation energies, band structures, etc. Although the fundamental concepts underlying DFT was developed in the 1960s, it took 20 years or so for the practical application of the theory with efficient DFT codes and algorithms. Since then, DFT has been one of the great successes in modeling materials behavior. With ever-increasing computational power at lower costs and improvements in computational algorithms, the CPU time to perform DFT calculations has been steadily declining to the point that performing large-scale calculations on the order of tens or hundreds of thousands of structures is possible in a reasonable amount of time. So-called high-throughput (HT) DFT calculations enable the generation of large databases of DFT-predicted materials properties, which can accelerate materials design through direct searches of materials with desired properties or the development of higher-level models (e.g., data mining). Several efforts are underway to generate largescale HT DFT databases, including the Open JOM, (Vol. 65, No. 11, 2013 DOI: 10.1007/s11837-013-0755-4 2013) The Minerals, Metals & Materials Society (Published online September 28, 2013). Quantum Materials Database (OQMD), the Materials Project, the Computational Materials Repository, and AFLOWLIB. We have developed the OQMD, which is an extensive HT DFT database consisting of DFT predicted crystallographic parameters and formation energies for over 200,000 experimentally observed International Crystal Structure Database (ICSD) and theoretical prototype structures. An important feature of the OQMD is the open nature of our database, meaning we will provide access to the complete database without limitation for the community to use, strongly in line with the Materials Genome Initiative. In the coming months, we will make the OQMD accessible over the web at http://oqmd.org, with a download option available for the database files themselves and the code to use them. Since the development and application of the earliest HT DFT database, HT DFT has proven to be a successful tool for many and varied materials problems.

The Open Quantum Materials Database is a collection of consistently calculated DFT total energies and relaxed crystal structures. Using the Vienna Ab-initio Simulation Package (VASP), DFT calculations have been performed for every unique entry in the ICSD without partial site occupancy and less than 35 atoms in the primitive cell, 32,489 structures as of August 2013. The OQMD serves two primary functions: as a large set of data for known structures from which optimum materials can be searched (such as in the following sections: ‘‘High-Capacity Conversion Anode Screening’’, ‘‘Li2O Battery Screening’’, and ‘‘HF Scavenging LiIon Battery Cathode Coatings’’) and as an accurate description of the chemical potentials and convex hulls of simple and complex systems from which tests of stability can be readily performed. The OQMD is primarily limited by what has been experimentally observed and catalogued in the ICSD (i.e., there may exist novel unexplored systems and compounds which are technologically important). Toward resolving this issue, we also include in the database DFT calculations of many unary, binary, and ternary prototype structures. These include, for example, every possible combination of A3B L12 and X2YZ heusler chemistries for over 80 elements. The inclusion of these prototypes in the OQMD provides an approximation for unexplored convex hulls and possible undiscovered compounds as they sample unexplored compositions and systems. The total number of structures in the OQMD, including both ICSD structures and prototypes, as of August 2013 is over 200,000 and is growing every day. For many materials applications, thermodynamic stability is an important quantity. The long-term stability of c¢ Co3(Al,W) L12 precipitate in Co-based superalloys, hydrogen storage decomposition pathways in metal borohydrides, and spinodal decomposition in IV–VI rock salt thermoelectric semiconductors are several examples where the stability of phases is critical for understanding materials behavior. For a compound to be stable, it must not only be lower in energy than all other compounds at that stoichiometry but also be lower in energy than linear combinations of all other compounds in a given system. Thus, an accurate description of stability requires the calculation of the phase in question (e.g., Co3 (Al,W)) and all other competing phases in the given system (e.g., Co, CoAl, and Co3W33). Because both the composition and the free energy are linear as a function of quantity of different phases in a system, the set of phases that has the minimum total free energy at a given composition can be determined by linear programming. We have employed this approach, grand canonical linear programming (GCLP), to study hydrogen storage reactions, Li-battery anode conversion materials, and general multiphase ground state stability. We have recently revised GCLP to make it more efficient when examining stability in highly multicomponent systems. As an example of the broad application of GCLP with the DFT-calculated formation energies in the OQMD, we used GCLP to determine how many ICSD compounds in the OQMD are thermodynamically stable at 0 K and zero pressure. Under these conditions, 23% of the 32,489 calculated ICSD structures are stable. Before about 1960, most discovered compounds were also thermodynamically stable. After 1960, the number of metastable structures per year grew rapidly, outpacing the fairly constant rate of stable structure discoveries. The surge in thermodynamically metastable yet experimentally observed structures is perhaps due to the advent of complex synthesis techniques for strained and/or high-pressure structures, such as thin-film deposition and the diamond anvil cell. As a large database of DFT calculations, the OQMD can be used to perform a broad comparison of DFT predicted properties to experimental measurements to assess the accuracy of the DFT approach."

OQMD is an extraordinary innovation in the field of Machine Learning but at the same time, an undeniable symbol of its greatest barricade: lack of accessible data. Chris Wolverton began working on OQMD only because he could not gain access to the already available data in then existing database [39].

Previous Works

Machine learning is a relatively new technology and its collaboration with Mechanical Engineering is still uncommon. However, some work has been done on the subject, though on different aspects of Mechanical Engineering. The results therein have been hopeful on the prospect of utilizing Machine Learning applications as a full-fledged tool investigation in the research environment of IUT-MCE.

Details of some of those works are given below:

Rafiee, J. et al [40]

This is a project based on structure analysis. J. Rafiee and team worke on an optimized gear fault identification system using artificial neural networks. A well-designed structure of the system allowed for short training duration and high accuracy. Faults for this project were defined as the conditions of slight-worn, medium-worn and broken-down teeth of a spur gear. It was found that processing of vibration signal became more difficult due to choosing a full-of-oil complex gearbox system was chosen to record raw vibration system. These signals were segmented into signals recorded during one complete revolution of the input shaft using tacometer information and then synchronized using piecewise cubic hermite interpolation to construct sample signals with same length. Standard deviation of wavelet packet co-efficients were considered as feature vector for training purpose of the Artificial Neural Network. To ameliorate the algorithm, GA was exploited to optimize the algorithm so as to determine the best values for “mother wavelet function”, “decomposition level of the signals by means of wavelet analysis”, and “number of neurons in hidden layer” resulted in a high-speed, meticulous two-layer Artificial Neural Network with a small-sized structure. This technique has been eliminated the drawbacks of the type of mother function for fault classification purpose not only in machine condition monitoring, but also in other related areas. The small-sized proposed network has improved the stability and reliability of the system for practical purposes.

Ling. J et al [41]

Despite being used extensively in industry to predict fluid flows, Reynold's Averaged Navier Stokes (RANS) models often produce inaccurate flow predictions and there very limited diagnostics available to assess RANS accuracy for a given flow configuration. In this paper, Machine learning algorithms were used in an attempt to identify regions of high RANS uncertainty. Three different machine learning algorithms were evaluated: support vector machines, Adaboost decision trees, and random forests. The algorithms were trained on a database of canonical flow configurations for which validated direct numerical simulation or large eddy simulation results were available, and were used to classify RANS results on a point-by-point basis as having either high or low uncertainty, based on the breakdown of specific RANS modeling assumptions.

Classifiers were developed for three different basic RANS eddy viscosity model assumptions: the isotropy of the eddy viscosity, the linearity of the Boussinesq hypothesis, and the non-negativity of the eddy viscosity. It was shown that these classifiers were able to generalize to flows substantially different from those on which they were trained. Feature selection techniques, model evaluation, and extrapolation detection were discussed in the context of turbulence modeling applications.

Şencan, A., et al [42]

In this paper a new approach using artificial neural networks (ANN) to determine the thermodynamic properties of two alternative refrigerant/absorbent couples ($LiCl–H_2O$ and $LiBr + LiNO_3 + LiI + LiCl–H_2O$) was presented. These pairs can be used in absorption heat pump systems, and their main advantage is that they do not cause ozone depletion. In order to train the network, limited experimental measurements were used as training and test data. Two feedforward ANNs were trained, one for each pair, using the Levenberg–Marquardt algorithm. The training and validation were performed with good accuracy. The used methodology proved to be much better than linear multiple regression analysis. Using the weights obtained from the trained network, a new formulation is presented for determination of the vapor pressures of the two refrigerant/absorbent couples. The use of this new formulation, which can be employed with any programming language or spreadsheet program for estimation of the vapor pressures of fluid couples, as described in this paper, may make the use of dedicated ANN software unnecessary.

Domanski P.A. et al [43]

Thermodynamic performance limits of the vapor compression cycle is explored in this paper. Evolutionary algorithms were used to explore the performance of hypothetical refrigerants defined by thermodynamic parameters used by the extended corresponding state model for fluid properties. Optimal values of these parameters to reach performance the limit were identified. The study confirmed the fundamental trade-off between the coefficient of performance (COP) and volumetric capacity, and indicated refrigerant critical temperature as the dominant parameter influencing the tradeoff. Thermodynamic performance limits depend on the operating conditions and the cycle design. These limits are represented by Pareto fronts developed for the objective functions COP and volumetric capacity. Predictably, the performance of current refrigerants falls below the Pareto front limits. It was demonstrated demonstrate that for practical cycles the developed methodology and resulting Pareto fronts are more realistic benchmarks for the performance potentials of refrigerants than is COP alone, which is an efficiency only metric.

Chapter 2: Component Discussion

Methods

There are a number of Machine Learning methods available in the world today and that number is ever growing, as is expected of a field that is dynamically evolving at its peak. The methods are generally characterized by their underlying algorithms. The reason “Algorithm” was not used as the subtitle heading is because some of the methods are based on groups of extremely similar, but not identical algorithms. A generic flowchart of which method is chosen for a situation is given below:

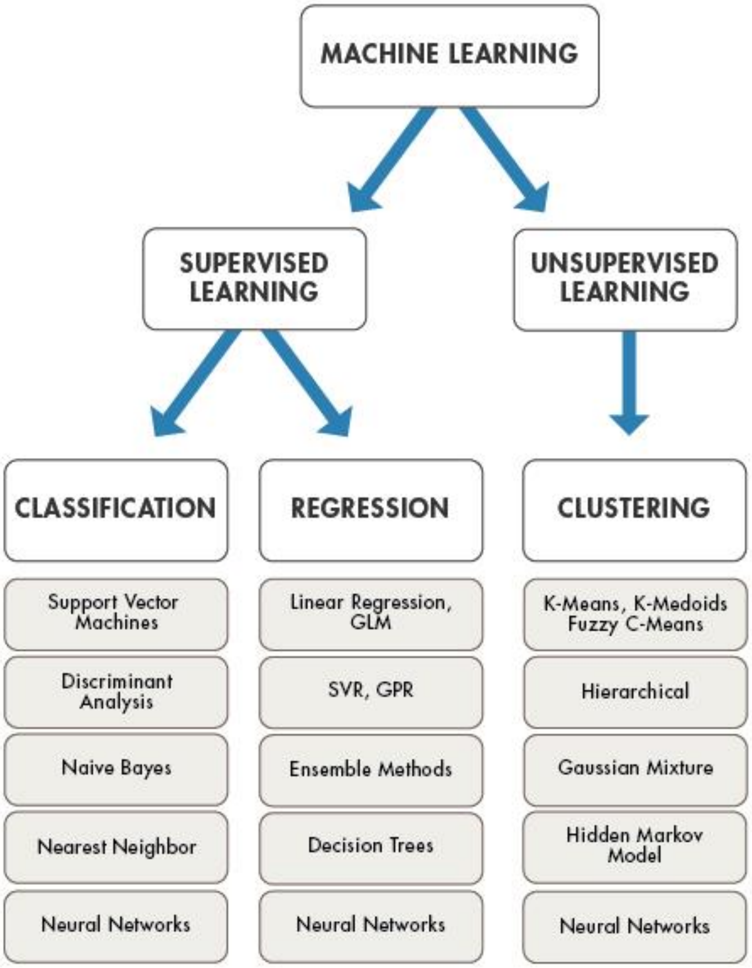

Figure 8: A representation of Machine Learning Algorithms [44]

It should be noted that not all the algorithms shown above will be discussed in details in this dissertation as many of them do not have much to offer in this thesis other than being the variant of the discussed procedure. However, a short overview on algorithm is now due and is given below [45]:

Algorithm, systematic procedure that produces, in a finite number of steps, the answer to a question or the solution of a problem. The name derives from the Latin translation, *Algoritmi de numero Indorum,* of the 9th-century Muslim mathematician al-Khwarizmi's arithmetic treatise *Al-Khwarizmi Concerning the Hindu Art of Reckoning.*

For questions or problems with only a finite set of cases or values an algorithm always exists (at least in principle); it consists of a table of values of the answers. In general, it is not such a trivial procedure to obtain solutions of queries that have an infinite number of cases or values to consider, for example the question, "Is the natural number (1, 2, 3, . . .) a prime?" or the question "What is the greatest common divisor of the natural numbers a and b?" The first of these belongs to a class called decidable; an algorithm that produces a yes or no answer is called a decision procedure. The second question belongs to a class called computable; an algorithm that leads to a specific number answer is called a computation procedure. Algorithms exist for many such infinite classes of questions; Euclid's Elements, published about 300 BC, contained one for finding the greatest common divisor of two natural numbers.

A simplistic approach to understand the general idea of an algorithm could be found by investigating the procedure of long division learned by every elementary school student. The long division is an algorithm which answers the question the question "Upon dividing a natural number x by another natural number y, what are the quotient and the remainder?" Use of this computational procedure leads to the answer to the decidable question "Does y divide x?" (The answer is yes if the remainder is zero). Repeated application of these algorithms eventually produces the answer to the decidable question "Is a prime?" (The answer is no if x is divisible by any smaller natural number besides 1).

Sometimes an algorithm cannot exist for solving an infinite class of problems, particularly when some further restriction is made upon the accepted method. For instance, two problems from Euclid's time requiring the use of only a compass and a straightedge (unmarked ruler)—trisecting an angle and constructing a square with an area equal to a given circle—were pursued for centuries before they were shown to be impossible. At the turn of the 20th century, the influential German mathematician David Hilbert proposed 23 problems for mathematicians to solve in the coming century. The second problem on his list asked for an investigation of the consistency of the axioms of arithmetic. Most mathematicians had little doubt of the eventual attainment of this goal until 1931, when the Austrian-born logician Kurt Gödel demonstrated the surprising result that there must exist arithmetic propositions (or questions) that cannot be proved or disproved. Essentially, any such proposition leads to a determination procedure that never ends (a condition known as the halting problem). In an unsuccessful effort to ascertain at least which propositions are unsolvable, the English mathematician and logician Alan Turing rigorously defined the loosely understood concept of an algorithm. Although Turing ended up proving that there must exist undecidable propositions, his description of the essential features of any general-purpose algorithm machine, or Turing machine, became the foundation of computer science. Today the issues of decidability and computability are central to the design of a computer program—a special type of algorithm.

In this thesis, we shall focus on the description and procedures for the Nelder-Mead method. We shall also study the Density Function Theorem and Green's theorem, due to their importance in OQMD. A cursory introduction has been deemed sufficient for the rest of the methods mentioned in this thesis for the purpose of having the reader familiarize with them.

Nelder-Mead Method

The Nelder–Mead method or downhill simplex method or amoeba method is a commonly applied numerical method used to find the minimum or maximum of an objective function in a multidimensional space. It is one of the best known algorithms for multidimensional unconstrained optimization without derivatives.It is applied to nonlinear optimization problems for which derivatives may not be known. However, the Nelder–Mead technique is a heuristic search method that can converge to non-stationary points [46] on problems that can be solved by alternative methods [47]. The Nelder–Mead technique was proposed by John Nelder and Roger Mead (1965) as a development of the method of Spendley et al [48].

The basic algorithm is quite simple to understand and very easy to use. For these reasons, it is very popular in many fields of science and technology, especially in chemistry and medicine. The method does not require any derivative information, which makes it suitable for problems with non-smooth functions. It is widely used to solve parameter estimation and similar statistical problems, where the function values are uncertain or subject to noise. It can also be used for problems with discontinuous functions, which occur frequently in statistics and experimental mathematics.

One must be aware that the Nelder-Mead method is not one to be directly used as a Machine Learning algorithm, but one to choose ideal parameters and form derived values extensively without prior reference. This aspect of it is what makes the method integral in the proposals to be discussed in Chapter 4.

The Nelder-Mead algorithm is designed to solve the classical unconstrained optimization problem of minimizing a given nonlinear function:

$$f:\mathbb{R}n\rightarrow\mathbb{R} \qquad (1)$$

The method uses only *function values* at some points in R*n* , and does not try to form an approximate gradient at any of these points.

Hence it belongs to the general class of direct search methods. In practice, Nelder-Mead starts off with a randomly-generated simplex. At every iteration, it proceeds to reshape/move this simplex, one vertex at a time, towards an optimal region in the search space. During each step, it basically tries out one or a few modifications to the current simplex, and chooses one that shifts it towards a region of the domain that meets the objective more closely. In an ideal case, the last few iterations of this algorithm would involve the simplex shrinking inwards towards the best point inside it. At the end, the vertex of the simplex that yields that most optimal objective value, is returned. Nelder-Mead method is simplex based.

A simplex S in $\mathbb{R}^n$ is defined as the convex hull of n+1 vertices $x_0,\dots,x_0\in\mathbb{R}^n$. For example, a simplex in $\mathbb{R}^2$ is a triangle, and a simplex in $\mathbb{R}^3$ is a tetrahedron. A simplex-based direct search method begins with a set of n+1 points $x_0,\dots,x_n\in R^n$ that are considered as the vertices of a working simplex S , and the corresponding set of function values at the vertices $f_j:=f(x_j)$, for $j=0,\dots,n$. The initial working simplex S has to be nondegenerate, i.e., the points $x_0,\dots,x_n$ must not lie in the same hyperplane. The method then performs a sequence of transformations of the working simplex S, aimed at decreasing the function values at its vertices. At each step, the transformation is determined by computing one or more test points, together with their function values, and by comparison of these function values with those at the vertices. This process is terminated when the working simplex S becomes sufficiently small in some sense, or when the function values f_j are close enough in some sense (provided f is continuous).The Nelder-Mead algorithm typically requires only one or two function evaluations at each step, while many other direct search methods use n or even more function evaluations. [49]

A more casual and perhaps more easily understood procedure can be found in Sachin Joglekar;s blog at the url: https://codesachin.wordpress.com/2016/01/16/nelder-mead-optimization [50].

An initial simplex is required for the method to initiate operation. There are a variety of this simplex, but the commonly used in practice, most notably in MATLAB (Matrix Laboratory) [51] is as follows:

$$X_{(i+1)} = X_1 + h(X_1, i)*U_1 \quad (2)$$

Where,
$X_{(i+1)}$ is the $(i+1)^{th}$ point of iteration i.
U_i is the i^{th} dimension uit vector.
X_1 is the initial point to be randomly generated or user provided
$h(X_1, i)$ is equal to 0.05 if X_1 is non-zero, otherwise is equal to 0.00025

Decision Tree Learning

Decision tree learning uses a decision tree (as a predictive model) to go from observations about an item (represented in the branches) to conclusions about the item's target value (represented in the leaves). It is one of the predictive modelling approaches used in Machine Learning as well as in statistics and data mining. Tree models where the target variable can take a discrete set of values are called classification trees; in these tree structures, leaves represent class labels and branches represent conjunctions of features that lead to those class labels. Decision trees where the target variable can take continuous values (typically real numbers) are called regression trees. In decision analysis, a decision tree can be used to visually and explicitly represent decisions and decision making. In data mining, a decision tree describes data (but the resulting classification tree can be an input for decision making).

Decision trees are of two main types:

- Classification tree analysis is when the predicted outcome is the class to which the data belongs.
- Regression tree analysis is when the predicted outcome can be considered a real number (e.g. the price of a house, or a patient's length of stay in a hospital).

The term Classification And Regression Tree (CART) analysis is an umbrella term used to refer to both of the above procedures, first introduced by Breiman et al. in 1984.[52] Trees used for regression and trees used for classification have some similarities - but also some differences, such as the procedure used to determine where to split. Some techniques, often called ensemble methods, construct more than one decision tree:

- Boosted trees: Incrementally building an ensemble by training each new instance to emphasize the training instances previously mis-modeled. A typical example is AdaBoost. These can be used for regression-type and classification-type problems.[53]
- Bootstrap aggregated (or bagged) decision trees: an early ensemble method, builds multiple decision trees by repeatedly resampling training data with replacement, and voting the trees for a consensus prediction.[54]
    - A random forest classifier is a specific type of bootstrap aggregating
- Rotation forest: In this which every decision tree is trained by first applying principal component analysis (PCA) on a random subset of the input features.[55]

Decision tree is a supervised Machine Learning method that works best with a large dataset. It essentially categorizes the information it processes based on conditions set by the user. The conditions can then be updated and process can be repeated until a satisfactory categorization has been achieved.

Density Function Theory (DFT)

Density functional theory (DFT) is a computational quantum mechanical modelling method used in physics, chemistry and materials science to investigate the electronic structure (principally the ground state) of many-body systems, in particular atoms, molecules, and the condensed phases. Using this theory, the properties of a many-electron system can be determined by using functionals, i.e. functions of another function, which in this case is the spatially dependent electron density. Hence the name density functional theory comes from the use of functionals of the electron density. DFT is among the most popular and versatile methods available in condensed-matter physics, computational physics, and computational chemistry [56].

DFT is quite simply a method of obtaining an approximate solution to the Schrodinger equation of a many-body system. DFT computational codes are used in practice to investigate the structural, magnetic and electronic properties of molecules, materials and defects. The number of degrees of freedom are limited by the Born-Oppenheimer Appproximation [57].

DFT has been very popular for calculations in solid-state physics since the 1970s. However, DFT was not considered accurate enough for calculations in quantum chemistry until the 1990s, when the approximations used in the theory were greatly refined to better model the exchange and correlation interactions. Computational costs are relatively low when compared to traditional methods, such as exchange only Hartree–Fock theory and its descendants that include electron correlation.

Despite recent improvements, there are still difficulties in using density functional theory to properly describe: intermolecular interactions (of critical importance to understanding chemical reactions), especially van der Waals forces (dispersion); charge transfer excitations; transition states, global potential energy surfaces, dopant interactions and some strongly correlated systems; and in calculations of the band gap and and ferromagnetism in semiconductors [58]. The incomplete treatment of dispersion can adversely affect the accuracy of DFT (at least when used alone and uncorrected) in the treatment of systems which are dominated by dispersion (e.g. interacting noble gas atoms) or where dispersion competes significantly with other effects (e.g. in biomolecules). The development of new DFT methods designed to overcome this problem, by alterations to the functional [59] or by the inclusion of additive terms, is a current research topic.

Although density functional theory has its roots in the Thomas–Fermi model for the electronic structure of materials, DFT was first put on a firm theoretical footing by Walter Kohn and Pierre Hohenberg in the framework of the two Hohenberg–Kohn theorems (H–K). The original H–K theorems held only for non-degenerate ground states in the absence of a magnetic field, although they have since been generalized to encompass these [60].

The first H–K theorem demonstrates that the ground state properties of a many-electron system are uniquely determined by an electron density that depends on only three spatial coordinates. It set down the groundwork for reducing the many-body problem of N electrons with 3N spatial coordinates to three spatial coordinates, through the use of functionals of the electron density. This theorem has since been extended to the time-dependent domain to develop time-dependent density functional theory [61], which can be used to describe excited states.

The second H–K theorem defines an energy functional for the system and proves that the correct ground state electron density minimizes this energy functional.

In work that later won them the Nobel prize in chemistry, The H–K theorem was further developed by Walter Kohn and Lu Jeu Sham to produce Kohn–Sham DFT [62] . Within this framework, the intractable many-body problem of interacting electrons in a static external potential is reduced to a tractable problem of noninteracting electrons moving in an effective potential. The effective potential includes the external potential and the effects of the Coulomb interactions between the electrons, that is the exchange and correlation interactions. Modeling the latter two interactions becomes the difficulty within Kohn–Sham DFT. The simplest approximation is the local-density approximation (LDA), which is based upon exact exchange energy for a uniform electron gas, which can be obtained from the Thomas–Fermi model, and from fits to the correlation energy for a uniform electron gas. Non-interacting systems are relatively easy to solve as the wave-function can be represented as a Slater determinant of orbitals. Further, the kinetic energy functional of such a system is known exactly. The exchange–correlation part of the total energy functional remains unknown and must be approximated.

Another approach, less popular than Kohn–Sham DFT but arguably more closely related to the spirit of the original H–K theorems, is orbital-free density functional theory (OFDFT) [63], in which approximate functionals are also used for the kinetic energy of the non-interacting system.

Green’s Theorem

In mathematics, a Green's theorem function is the impulse reponse of an inhomogenous linear differential equation defined defined on a domain, with specified initial conditions or boundary conditions [64]. Through the superposition principle for linear operator problems, the convolution of a Green's function with an arbitrary function f (x) on that domain is the solution to the inhomogeneous differential equation for f (x). In other words, given a linear ordinary differential equation (ODE), L(solution) = source, one can first solve L(green) = δs, for each s, and realizing that, since the source is a sum of delta functions, the solution is a sum of Green's functions as well, by linearity of L.

Green's functions are named after the British mathematician George Green, who first developed the concept in the 1830s. In the modern study of linear partial differential equations, Green's functions are studied largely from the point of view of fundamental solutions instead. Under many-body theory, the term is also used in physics, specifically in quantum field theory, aerodynamics, aeroacoustics etc. to refer to various types of correlation functions, even those that do not fit the mathematical definition. In quantum field theory, Green's functions take the roles of propagators.

Simply speaking, Green’s theorem is taken to be a more inclusive analysis of the vector fields produced by the microscopic motions of the ions and electrons, giving a more accurate result than DFT, in theory. It was thus mentioned as a possible alternative for the operating algorithm of OQMD by Wolverton et al [18].

Gradient boosting Methods

Gradient boosting is a machine learning technique for regression and classification problems, which produces a prediction model in the form of an ensemble of weak prediction models, typically decision trees. It builds the model in a stage-wise fashion like other boosting methods do, and it generalizes them by allowing optimization of an arbitrary differentiable loss function [65].

The idea of gradient boosting originated in the observation by Leo Breiman that boosting can be interpreted as an optimization algorithm on a suitable cost function. Explicit regression gradient boosting algorithms were subsequently developed by Jerome H. Friedman [66], simultaneously with the more general functional gradient boosting perspective of Llew Mason, Jonathan Baxter, Peter Bartlett and Marcus Frean [67]. The latter two papers introduced the view of boosting algorithms as iterative functional gradient descent algorithms. That is, algorithms that optimize a cost function over function space by iteratively choosing a function (weak hypothesis) that points in the negative gradient direction. This functional gradient view of boosting has led to the development of boosting algorithms in many areas of machine learning and statistics beyond regression and classification.

Clustering

Cluster analysis or Clustering is the task of dividing the population or data points into a number of groups such that data points in the same groups are more similar to other data points in the same group than those in other groups. In simple words, the aim is to segregate groups with similar traits and assign them into clusters [68].

Broadly speaking, clustering can be divided into two subgroups [69] :

- Hard Clustering: In hard clustering, each data point either belongs to a cluster completely or not. For example, in the above example each customer is put into one group out of the 10 groups.
- Soft Clustering: In soft clustering, instead of putting each data point into a separate cluster, a probability or likelihood of that data point to be in those clusters is assigned. For example, from the above scenario each costumer is assigned a probability to be in either of 10 clusters of the retail store.

Since the task of clustering is subjective, the means that can be used for achieving this goal are plenty. Every methodology follows a different set of rules for defining the scope of similarity among data points. In fact, there are more than 100 clustering algorithms known. But few of the algorithms are used popularly and they are [68]:

- Connectivity models: As the name suggests, these models are based on the notion that the data points closer in data space exhibit more similarity to each other than the data points lying farther away. These models can follow two approaches. In the first approach, they start with classifying all data points into separate clusters & then aggregating them as the distance decreases. In the second approach, all data points are classified as a single cluster and then partitioned as the distance increases. Also, the choice of distance function is subjective. These models are very easy to interpret but lacks scalability for handling big datasets. Examples of these models are hierarchical clustering algorithm and its variants.
- Centroid models: These are iterative clustering algorithms in which the notion of similarity is derived by the closeness of a data point to the centroid of the clusters. K-Means clustering algorithm is a popular algorithm that falls into this category. In these models, the no. of clusters required at the end have to be mentioned beforehand, which makes it important to have prior knowledge of the dataset. These models run iteratively to find the local optima.
- Distribution models: These clustering models are based on the notion of how probable is it that all data points in the cluster belong to the same distribution (For example: Normal, Gaussian). These models often suffer from overfitting. A popular example of these models is Expectation-maximization algorithm which uses multivariate normal distributions.
- Density Models: These models search the data space for areas of varied density of data points in the data space. It isolates various different density regions and assign the data points within these regions in the same cluster. Popular examples of density models are DBSCAN and OPTICS.

Clustering is an unsupervised machine learning approach, it can also be used to improve the accuracy of supervised machine learning algorithms as well by clustering the data points into similar groups and using these cluster labels as independent variables in the supervised machine learning algorithm [70].

Inductive Logic Programming

Inductive logic programming (ILP) is a subfield of symbolic artificial intelligence which uses logic programming as a uniform representation for examples, background knowledge and hypotheses. Given an encoding of the known background knowledge and a set of examples represented as a logical database of facts, an ILP system will derive a hypothesised logic program which entails all the positive and none of the negative examples.

Inductive logic programming is particularly useful in bioinformatics and natural language processing. Gordon Plotkin and Ehud Shapiro laid the initial theoretical foundation for inductive machine learning in a logical setting.[71] Shapiro built their first implementation (Model Inference System) in 1981:a Prolog program that inductively inferred logic programs from positive and negative examples. The term Inductive Logic Programming was first introduced in a paper by Stephen Muggleton in 1991.[72]

Reinforcement Learning

Reinforcement learning refers to goal-oriented algorithms, which learn how to attain a complex objective (goal) or maximize along a particular dimension over many steps; for example, maximize the points won in a game over many moves. They can start from a blank slate, and under the right conditions they achieve superhuman performance. Reinforcement learning solves the difficult problem of correlating immediate actions with the delayed returns they produce. Like humans, reinforcement learning algorithms sometimes have to wait a while to see the fruit of their decisions. They operate in a delayed return environment, where it can be difficult to understand which action leads to which outcome over many time steps [73]. The problems of interest in reinforcement learning have also been studied in the theory of optimal control, which is concerned mostly with the existence and characterization of optimal solutions, and algorithms for their exact computation, and less with learning or approximation, particularly in the absence of a mathematical model of the environment.

In machine learning, the environment is typically formulated as a Markov Decision Process (MDP), as many reinforcement learning algorithms for this context utilize dynamic programming techniques [74]. The main difference between the classical dynamic programming methods and reinforcement learning algorithms is that the latter do not assume knowledge of an exact mathematical model of the MDP and they target large MDPs where exact methods become infeasible.[74]. Reinforcement learning differs from standard supervised learning in that pre-determined input/output pairsneed not be presented, and sub-optimal actions need not be explicitly corrected. Instead the focus is on performance in terms of preocess progress, which involves finding a balance between exploration (of uncharted territory) and exploitation (of current knowledge).[75]

Representation Learning

In machine learning, representation learning [76] is a set of techniques that allows a system to automatically discover the representations needed for feature detection or classification from raw data. This replaces manual feature engineering and allows a machine to both learn the features and use them to perform a specific task. Representation learning is motivated by the fact that machine learning tasks such as classification often require input that is mathematically and computationally convenient to process. However, real-world data such as images, video, and sensor data has not yielded to attempts to algorithmically define specific features. An alternative is to discover such features or representations through examination, without relying on explicit algorithms.

Similarity and Metric Learning

Similarity learning is an area of supervised machine learning in artificial intelligence. It is closely related to regression and classification, but the goal is to learn from examples a similarity function that measures how similar or related two objects are. It has applications in ranking, in recommendation systems, visual identity tracking, face verification, and speaker verification [77]. Similarity learning is closely related to distance metric learning. Metric learning is the task of learning a distance function over objects. A metric or distance function has to obey four axioms: non-negativity, Identity of indiscernibles, symmetry and subadditivity/triangle inequality. In practice, metric learning algorithms ignore the condition of identity of indiscernibles and learn a pseudo-metric [78].

Association rule Learning

Association rule learning is a rule-based machine learning method for discovering interesting relations between variables in large databases. It is intended to identify strong rules discovered in databases using some measures of interestingness [79]. Interestingness is defined as the frequency of a particular data with respect to the summation of accumulated data, which is then computed by various factor that is dependent on the time-location or the relative-space location (with respect to other data, especially so if the correlation has precedence) that provides a guideline of sorts to determine how likely is it that that particular data is related to some other data of interest.

One limitation of the standard approach to discovering associations is that by searching massive numbers of possible associations to look for collections of items that appear to be associated, there is a large risk of finding many spurious associations. These are collections of items that co-occur with unexpected frequency in the data, but only do so by chance.

Support Vector Machines

In machine learning, support vector machines [80] are supervised learning models with associated learning algorithms that analyze data used for classification and regression analysis. Given a set of training examples, each marked as belonging to one or the other of two categories, an SVM training algorithm builds a model that assigns new examples to one category or the other, making it a non-probabilistic binary linear classifier. An SVM model is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible. New examples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall. In addition to performing linear classification, SVMs can efficiently perform a non-linear classification using what is called the kernel trick, implicitly mapping their inputs into high-dimensional feature spaces.

When data is unlabeled, supervised learning is not possible, and an unsupervised learning approach is required, which attempts to find natural clustering of the data to groups, and then map new data to these formed groups. The support vector clustering[81] algorithm created by Hava Siegelmann and Vladimir Vapnik, applies the statistics of support vectors, developed in the support vector machines algorithm, to categorize unlabeled data.

Deep Learning

Deep learning is a class of machine learning algorithms that were conceived with aspiration to produce Artificial Intelligence through Machine Learning. Its defining characteristics are [82]:

- Use a cascade of multiple layers of nonlinear processing units for feature extraction and transformation. Each successive layer uses the output from the previous layer as input.
- Learn in supervised (e.g., classification) and/or unsupervised (e.g., pattern analysis) manners.
- Learn multiple levels of representations that correspond to different levels of abstraction; the levels form a hierarchy of concepts.

In deep learning, each level learns to transform its input data into a slightly more abstract and composite representation. In an image recognition application, the raw input may be a matrix of pixels; the first representational layer may abstract the pixels and encode edges; the second layer may compose and encode arrangements of edges; the third layer may encode a nose and eyes; and the fourth layer may recognize that the image contains a face. Importantly, a deep learning process can learn which features to optimally place in which level on its own [83]. For supervised learning tasks, deep learning methods obviate feature engineering, by translating the data into compact intermediate representations akin to principal components, and derive layered structures that remove redundancy in representation. Deep learning algorithms can be applied to unsupervised learning tasks. This is an important benefit because unlabeled data are more abundant than labeled data. Examples of deep structures that can be trained in an unsupervised manner are neural history compressors and deep belief networks [84].

Python

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together. Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourages program modularity and code reuse. The Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms, and can be freely distributed [85].

There are two attributes that make development time in Python faster than in other programming languages [86]:

- Python is an interpreted language, which precludes the need to compile code before executing a program because Python does the compilation in the background. Because Python is a high-level programming language, it abstracts many sophisticated details from the programming code. Python focuses so much on this abstraction that its code can be understood by most novice programmers.

- Python code tends to be shorter than comparable codes. Although Python offers fast development times, it lags slightly in terms of execution time. Compared to fully compiling languages like C and C++, Python programs execute slower. Of course, with the processing speeds of computers these days, the speed differences are usually only observed in benchmarking tests, not in real-world operations. In most cases, Python is already included in Linux distributions and Mac OS X machines.

Python's large standard library, commonly cited as one of its greatest strengths,[87] provides tools suited to many tasks. For Internet-facing applications, many standard formats and protocols such as MIME and HTTP are supported. It includes modules for creating graphical user interfaces, connecting to relational databases, generating pseudorandom numbers, arithmetic with arbitrary precision decimals,[87] manipulating regular expressions, and unit testing.

Since 2003, Python has consistently ranked in the top ten most popular programming languages in the TIOBE Programming Community Index where, as of January 2018, it is the fourth most popular language (behind Java, C, and C++). It was selected Programming Language of the Year in 2007 and 2010 [88].

History

The programming language Python was conceived in the late 1980s,[89] and its implementation was started in December 1989 [90] by Guido van Rossum at CWI in the Netherlands as a successor to ABC capable of exception handling and interfacing with the Amoeba operating system. Van Rossum is Python's principal author, and his continuing central role in deciding the direction of Python is reflected in the title given to him by the Python community, Benevolent Dictator for Life (BDFL). Python was named for the BBC TV show Monty Python's Flying Circus.

Python 2.0 was released on October 16, 2000, with many major new features, including a cycle-detecting garbage collector (in addition to reference counting) for memory management and support for Unicode. However, the most important change was to the development process itself, with a shift to a more transparent and community-backed process [91].

Python 3.0, a major, backwards-incompatible release, was released on December 3, 2008 after a long period of testing. Many of its major features have also been backported to the backwards-compatible Python 2.6 and 2.7.

NumPY

NumPy is a library for the Python programming language, adding support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays. It covers for Python's original deficiency for not being compatible for numerical computing and thus provides a platform that is not overly complicated and cumbersome for such endeavors. NumPy is amongst the fundamental package for scientific computing with Python. It contains among other things [92]:

- A powerful N-dimensional array object.
- Sophisticated (broadcasting) functions.
- Tools for integrating C/C++ and Fortran code.
- Useful linear algebra, Fourier transform, and random number capabilities.

Besides its obvious scientific uses, NumPy can also be used as an efficient multi-dimensional container of generic data. Arbitrary data-types can be defined. This allows NumPy to seamlessly and speedily integrate with a wide variety of databases. NumPy is licensed under the BSD license [93], enabling reuse with few restrictions.

The ancestor of NumPy, Numeric, was originally created by Jim Hugunin with contributions from several other developers. In 2005, Travis Oliphant created NumPy by incorporating features of the competing Numarray into Numeric, with extensive modifications. NumPy is open-source software and has many contributors.

SciPY

SciPy is a free and open-source Python library used for scientific computing and technical computing [94]. SciPy contains modules for optimization, linear algebra, integration, interpolation, special functions, FFT, signal and image processing, ODE solvers and other tasks common in science and engineering.

SciPy builds on the NumPy array object and is part of the NumPy stack which includes tools like Matplotlib, pandas and SymPy, and an expanding set of scientific computing libraries. This NumPy stack has similar users to other applications such as MATLAB, GNU Octave, and Scilab. The NumPy stack is also sometimes referred to as the SciPy stack.[95]

SciPy is also a family of conferences for users and developers of these tools: SciPy (in the United States), EuroSciPy (in Europe) and SciPy.in (in India). Enthought originated the SciPy conference in the United States and continues to sponsor many of the international conferences as well as host the SciPy website.The SciPy library is currently distributed under the BSD license, and its development is sponsored and supported by an open community of developers.

Chapter 3: Methodology

The aim of this thesis was to investigate the efficacy of utilizing Machine Learning in the field of Mechanical Engineering, as well as investigate methods to improve such efficacy. To demonstrate this aspect, an existing project which is related to Mechanical Engineering that utilized Machine Learning in it was searched. The parameters based on which this search was conducted on are:

- Size of the database/amount of distinct information.
- A relation with the subject matters of Mechanical Engineering.
- Suitable classes of data that can be organized and manipulated by programs based on Machine Learning algorithms to conceive tangible discoveries.
- As a secondary parameter, the database being manipulated by an existing algorithm to show results of Machine Learning application.

The project chosen after following the above parameters was the Open Quantum Materials Database (OQMD), an endeavor lead by Northwestern University to quantify atomic data of various elements [18].

The Open Quantum Materials Database (OQMD) is a high-throughput database currently consisting of nearly 300,000 density functional theory (DFT) total energy calculations of compounds from the Inorganic Crystal Structure Database (ICSD) and decorations of commonly occurring crystal structures. To maximize the impact of these data, the entire database is being made available, without restrictions, at www.oqmd.org/download.

DFT is quite simply a method of obtaining an approximate solution to the Schrodinger equation of a many-body system. DFT computational codes are used in practice to investigate the structural, magnetic and electronic properties of molecules, materials and defects. The number of degrees of freedom are limited by the Born-Oppenheimer Approximation.

Python is used for the coding in the database manipulation and translation of the predictive algorithm based on Density Function Theorem. However, the team in charge of the project, headed by Chris Wolverton, has hypnotized a few alternatives for the algorithm, which included the Green’s Theorem, which is a more inclusive analysis of the vector fields produced by the microscopic motions of the ions and electrons, giving a more accurate result, in theory.

In the thesis, the writers attempted to formulate an algorithm based on Green's Theorem to write a program in python under the SciPy module to manipulate the existing database to provide new values for the molecule properties and have them be compared to the values already calculated from the existing algorithms formulated by Mr. Wolverton and his team.

Finally, the experiences gained during the entire process was to be recorded and the procedures and especially the setbacks were to be considered in the perspective of the research settings of IUT-MCE department. With these considered, outlines and proposals for incorporating Machine Learning as an effective tool in the research projects conducted through the department were to be determined, including, but not limited to, steps to form a foundation to support such an avenue.

Chapter 4: Results and Discussions

Results

The writers were invested in being accustomed to python in order to translate the algorithm that was to be derived from Green’s Theorem. The database consisting of the details provided empirically and through calculations based on Density Function Theory was acquired and familiarized with. However, a viable program based on Green’s Theorem to provide a new set of data could not be completed within the given time-frame.

The writers have identified the complexity associated with determining the constant parameters to be selected for the algorithm and the time required to compute such parameters as the main obstruction for the inability to complete this part of the thesis objective at this time. It is to be noted here that the original database based on a relatively simpler theorem, took two years of a post-graduate research team to complete.[39]

Moreover, it is to be noted that the purpose of the attempt to utilize OQMD was to provide a precedence for the possible procedures and to be aware of obstructions faced in the way of utilizing Machine Learning applications in the research environment of IUT-MCE, for which this excursion has fulfilled its part.

The writers have determined that the largest obstacle in the way of procuring Machine Learning applications as a viable research tool for IUT-MCE is the lack of any suitable reserve of information on which the applications could be made. The secondary obstacle would be the lack of specialized algorithm geared towards solving problems related with Mechanical Engineering. This secondary problem can be addressed by utilizing a combination of simpler algorithms in order to generate the given results. The prospect of utilizing Machine Learning can also be considered a means to initiate the culture of perpetual Inter-Department collaborative research on a general topic in Islamic University of Technology.

The writers are confident that despite the lack of a working precedence in the avenue of utilizing Machine Learning present in the present project, the experiences gained during attempt, as well as the proposals that arose solely due to the endeavor undertaken, to be discussed next, will provide a significant contribution to the advent of this research progression in IUT-MCE.

Discussions

Proposals regarding Machine Learning in IUT-MCE

As the results have shown, the writers faced various challenges over the course of the thesis, which have affected adversely on the objective to compare a project as an example for Machine Learning, from the perspective of showing positive results. However, the objective of incorporating Machine Learning in the research projects of IUT-MCE requires a pathway for smooth operation; an objective certain to be made much easier thanks to setbacks the writers faced, should proper feedback be established to the findings.

The writers feel that the proposals to be made should be considered as important findings of the thesis endeavor as their existence would not have come to be, had the effort not been made to try and utilize Machine Learning in the field of Mechanical Engineering not been made to begin with.

The following are a set of proposals the writers wish to make on the course of implementing Machine Learning as an applicable tool in IUT-MCE's arsenal of options regarding research. These proposals will be listed and briefly explained in the order the writers feel they should be acted upon from chronological point of view. The proposals stem from combating two of the most fundamental problems the writers faced while pursuing their endeavor over a proper demonstration of Machine Learning application in the field of Mechanical Engineering:

- Lack of quality databases in terms of volume, categorization, organization and manipulability on which applications of Machine Learning can be used.
- The complexity of formulating and translating algorithms related to Machine Learning directly upon many facets of research topics in Mechanical Engineering, especially those related with empirical simulations using computer programs (ex: CFD)

The writers wish to stress that the proposals are by no means finite in the possibilities they explore, nor devoid of the need to be investigated and modified to more effectively suit the needs of the research environment of IUT-MCE research environment.

The writers would like to take this opportunity to thank once again Dr. Arafat Ahmed Bhuiyan and his team for allowing access to their research projects as theoretical case studies for the implementation of proposals to be mentioned.

Establishment of organized databases

The biggest problem, by far, faced by the writers in the course of this thesis was the lack of suitable databases in digital format regarding many exploits attempted on the field of Mechanical Engineering. In fact, the writers found the database of OQMD to be the only in free circulation on the World Wide Web that met the requirements for a competent dataset. Competency of the dataset was considered based on the following:

- A number of data sufficiently large for gaining meaningful variation.
- Consistency in matters not being observed so as to prevent misidentification of the causes behind observed effects.
- Data being varied just enough to allow for gradual attenuation of the program's calculative parameters

The writers concluded that the lack of such datasets were most likely due to the relative lack of usage of Machine Learning in the field of Mechanical Engineering. This is in part due to the fact that most algorithms available currently cannot be easily incorporated to substitute for more established means of conducting various researches (to be considered in more detail in the upcoming proposals), but another important issue is the volume of data such sets would take up without an appropriate return if conventional means are considered. It is for these reasons that details of unsuccessful (i.e.: not corresponding to the expected results) CFD simulations are kept minimally, mostly with the purpose to serve as a reference for improvement, for future simulations, while most of the details of the simulation are deleted for freeing up the computer/storage device's memory.

In light of the matters mentioned above, the writers conclude that the first step IUT-MCE should take for the prospect of establishing Machine Learning as a viable research tool is the establishment of a database system that will interminably compile maximum information of all researches occurring henceforth under the department. The writers suggest that this database is configured to fulfill the following objectives:

1. Compile all quantitative data regarding the experiments conducted practically or in simulation for all ongoing research projects and store them in a secure environment.
2. Make periodic categorization of collected data based on multiple categories, including but not limited to, conditional similarity, numerical similarity and progression similarity. The writers suggest doing the categorization annually in the months of November or December.
3. Give space for feedback of the preceding researchers on possible utilization of any given set of data, as well as the facility for succeeding researchers to comment on such feedback.

4. Keep iterations of manipulable data within reasonable limits so as to provide references to the progression of manipulation.

A possible type of data to be included in the proposed database would be details of initial conditions, progression details and end results of all simulations made for a particular research utilizing CFD. This includes the details of the simulations that did not contribute directly to the results of that particular research project.

The inclusion of all possible research data is crucial to fulfill the expectations placed on the database: to provide a solid and expansive foundation based on which algorithms can be utilized to produce significant numerical results in multiple avenues.

The effective volume the database is required to reach in order to be able to provide a viable platform for Machine Learning applications (the so-called "initial investment") is dependent on the scope and complexity of the algorithm that will be used as the core of the proposed application. However, for the relatively simpler algorithms that will be mentioned in the following proposals, the database will need 5-6 years' worth of data, based on the number (40 simulation for 5 groups each year for CFD on average) of simulations indicated in the case studies to follow.

Initiation of Machine Learning applications through simple algorithms

The other issue experienced by the writers through course of this thesis was the lack of availability of specific applications of Machine Learning that can effectively replace conventional means at present times. This is due limitations posed by the lack of data needed to define the conditions expansively enough to replace means like ANSYS.

From the perspective of obtaining tangible results, Prof. Yaser S. Abu-Mostafa of California institute of Technology [96] explains that a Machine Learning software would roughly require data amount 10 times the total number of degrees of freedom present in the observed situations. It has been noted that this requirement can be relaxed if sufficient conditions are placed on the algorithm, which are determined based on prior results of similar excursions [39]. For the case of CFD analysis, where the cumulative degrees of freedom present in the meshes numbers to over millions for the average analysis: as such, providing an algorithm that can emulate this from the get-go is a very difficult proposition. Providing conditions to alleviate the situation is also difficult as the number of prior results to determine such conditions require once again, more data generated from some form of logical deduction.

Therefore, it can be concluded that introducing algorithms that can replace the conventional analytic procedures is not viable at present conditions. However, Machine Learning algorithms can optimize the analysis of situations where the procedural uncertainty is mitigated, i.e.: results of analysis that follow the same conditions for analysis or more simply that the flow of intermittent conditions for a given set of simulations/procedures follow roughly the same pattern. This can be done through the usage of algorithms that focus on the correlation of initial inputs and final outputs, for example the Downhill Simplex Method, Powell's Method etc. The overall objective is to first find a correlation between the input and output and then link the correlation to the established mode of the procedure.

Downhill Simplex, or more formally known as Nelder-Mead method is a heuristic approach to find the optimal value for a given objective function in the search space [97]. Heuristic algorithms imbue an attribute of randomness in the analytical aspect of computational calculations. Thus, the same assumed relationship with a given initial condition can give a range of outputs, which can be stored to be compared with other outputs from different relationships. With the increased volume of possible outcomes, it becomes easier to make a correlation between the myriad of factors associated with computational analysis of Mechanical Engineering topics.

Taking our representative scope of consideration as CFD analysis, simply Nelder-Mead alone will not provide a palpable feedback to the queries made on that front. What it can do is to provide a range of possible relationships for a particular simulation whose variables are changed of only one degree at a time as well as provide the significant parameters affecting the simulation results. Doing this to multiple variables gives as an inflow of multiple flexible relations which can then work as foundations for more advanced algorithms suited to arrange the given relationships and/or parameters defined in observable categories.

Utilizing advanced algorithms to correlate simple relations

The end objective of utilizing Machine Learning applications in the research fields related to Mechanical Engineering; the introduction of more efficient means of investigating phenomenon and issues related to the field as well systemized methods to define empirical relationships, will likely not be achieved solely through the usage of relatively simpler algorithms. To achieve results to this end, more complex and expansive algorithms are required to be used.

The relatively simpler algorithms mentioned in the previous proposals are meant to take raw data provided in the observed situations and present them in a relatively organized manner with areas of interest with respect to specific objectives identified for the more complex algorithms to take as input.

The specific algorithm utilized for such a purpose is likely to be different for the fields of investigation they would be used in. It is also very likely that a multitude of variations will need to be made for the same basic form of algorithm to suit for the specific purpose for that particular project. The modifications and implementation of such iterations could a point of collaboration, as will be mentioned in the following proposal.

One possible example draws on from the perspective of CFD analysis discussed in the previous proposal. The flexible range of relations found through simpler algorithm can then be put into categorization on assumed conditions through Decision Tree algorithms or the like. The grouped relations can then be tested to see whether they effectively contribute to the formation of a universal relation for that particular set-up. The findings can be tested against optimization application, of which higher dimension derivatives of conjugate gradient method may be employed, and better categorization conditions may be set forward for more efficient categorization. Repetitive iterations of such procedures in increasing scale may provide an optimal universal relation that can give definite relations for values considered of empirical origins as of yet.

Introduction of Inter-Department collaborative research culture

One aspect of Machine Learning that the writers would like to stress on is that it is ultimately a branch under Computer Science. Development of some of the more intensive algorithms will require the input of those who have been trained in this field with more special attention. At the same time, research environment is interconnected and interdisciplinary at its most effective state.

Keeping the above in mind, the writers feel that for the research environment of IUT-MCE to take qualitative leap to match with global standards and eventually surpass it, it must induce the professional culture of nurturing inter-disciplinary researches and collaborative projects. The writers feel that utilizing Machine Learning Applications as a point of commencement for such an endeavor is a timely takeaway from the experiences on the course of this thesis.

The collaboration on the subject of Machine Learning applications will not only allow the development of the tool for the purpose of utilization in the field of Mechanical Engineering flourish, it will also provide a foundation on which serious investigations can be made on the relationships between aspects of Mechanical Engineering and that of Computer Science and Engineering. The writers have not been able to identify the existence of similar efforts made elsewhere in the academic research global arena, which suggests that should IUT-MCE act upon this proposal, it would very likely have a pioneering role in the particular field in the times ahead to come.

The writers feel that this collaboration should only commence after IUT-MCE has completed setting up a competent database, which will also include values gained from utilizing the relatively simpler algorithms. This way, the projects will be able to be focused on the necessities particular to the field of Mechanical Engineering and will have an adequate source of materials for manipulation and calculation to that end. One of the prime objectives of such a collaboration may be the investigation of the utilization of advanced algorithms and possible modifications of said algorithms for the purpose of optimizing Mechanical Engineering investigative procedures.

Case Studies

Case Study #1: Reduction of Marine Propeller Cavitation

This project has been conducted by Nasimul Karim (MCE 14, St.ID#141440) under the supervision of Dr. Arafat Ahmed Bhuiyan. The description of the project according to the contributor is as follows:

"Cavitation is a common issue for marine propellers. Though there are a couple of existing approaches to mitigate the problem, a new concept has been numerically investigated. The principle goal of this study is to investigate the influence of water injection to the regions, where pressure drops below vapor pressure, to decrease the formation of cavitation, using commercial CFD Code ANSYS. To understand the outcomes, computation has been done on a modified INSEAN E779a propeller. The k-ω SST turbulence model with the curvature correction and the Zwart cavitation model have been applied. The used model has been validated against experimental data and a reasonable agreement has been observed. The influence of the rear part of boats, in the simulation has also been studied."

The simulation parameters for this experiment have been determined to be pressure, surface tension, bubble diameter, viscosity and time. The input variable was limited to the given rpm of the propeller and target rpm had been the amount of cavitation bubbles produced.

Under the premise of utilizing Machine Learning application to optimize the project mentioned above, one improvement to be made could the increase of rang of observed parameters as a basis for more comprehensive analysis to the rear of the marine vehicle. This includes but is not limited to, time taken for a certain threshold for cavitation to occur, distribution of cavitation bubbles across the propeller blade, impact pressure made due to the formation of cavitation bubbles.

There can be an information collection of the optimum value (optimum being the minimum/maximum value for a given objective) by alternatively assigning one parameter as input and one as output for all the available parameters used in the simulation. This will provide access to a large range of possible correlations, which can then be categorized under assumed conditions provided by information verified in previous iterations of the Zwart Cavitation model. The correlations can then be investigated using appropriate algorithms to determine whether any aspect of the phenomenon can be predicted using the logical amalgamation of correlations with necessary modifications grouped under a specific category. Once such amalgamations have been conceived, the process can be repeated until a universal procedure for determination of cavitation influences on propeller properties and general marine vehicle conditions is generated.

Case Study #2: Improvement of thermal-hydraulic performance of Compact Heat Exchangers with multi-corrugated fins and oval tube arrays

This project has been conducted by Rayhanul Islam (MCE 14, St.ID# 141447) and Syed Tasnim Md. Fahim (MCE 14, St.ID#141446) under the supervision of Dr. Arafat Ahmed Bhuiyan. The description of their project is as follows:

“Compact heat exchanger (CHXs) is one of the top choices in residential and industrial applications for its higher thermal efficiency, lower pressure drop, low cost, light weight characteristics. Among various types of compact thermal devices, fin and tube compact heat exchangers are stimulating enormous interest in thermal engineering field. In the present study, airside performance of CHXs having corrugated fin and oval tube heat exchanger are evaluated using a CFD code ANSYS 12.0. Main motivation of this investigation is to find out the pattern of gradual improvement of thermal-hydraulic performance by showing the comparison between plane and corrugated fin geometry. This study has been performed for Reynolds number ranging from 200 to 1000. Here we are analyzed the performance among plane, one, two & three corrugated fin geometry. Also, the variation of tube shape: Changing the circular tube geometry to oval shape has also studied. The significant increase of Nusselt number, Friction factor and Colburn factor has been found for the corrugated compact heat exchangers. Results indicate that incurring a moderate pressure loss the corrugated fin patterns can ensure more thermal efficiency. The pattern of gradual improvement of heat transfer rate and slacken pressure loss with the increase of corrugation in a fin with oval shaped tubes has been experienced.”

The parameters of interest in this project are fin geometry, fin surface area, tube geometry, tube surface area, relative orientation between tube and fins, fluid pressure variations, fluid flow velocity, fluid pressure variation, fluid flow direction relative to the heat exchanger.

This project in particular has the potential to benefit greatly from analysis of design parameters using Machine Learning applications. As stated before, by utilizing the simpler algorithms, many possible relations regarding the variation of rate of heat transfer with respect to change in the parameters given above is available for acquisition, provided the relevant formulaic analysis is used. From there, recurrence of efficient categorization using Decision Tree analysis and subsequent verification of assumed categorization using more complex algorithms such as Conjugate gradient method has a decent possibility of providing an adequately universal analytical solution for the given case.

Case Study #3: Three-dimensional numerical study on airside thermal-hydraulic performance of Rectangular Finned Elliptical Tube Heat

This project is done by Emran Hossain Moin (St.Id#141423) and Kabbir Hossain (St.Id# 141437) under the supervision of Dr. Arafat Ahmed Bhuiyan. The description of the project is as below as stated by the contributors:

"Space heating and cooling is a comfort requirement which results in a significant amount of energy consumption. EATHE (Earth Air Thermal Heat Exchanger) is one of the passive ways of space heating and cooling that can reduce the energy consumption notably. In this paper, the effect of different geometric parameters; cross section of the pipe, inclusion of fins on the pipe, pitch of multiple pipe arrangements on the thermal performance of EATHEs under steady and transient conditions were analyzed. Since, the performance deteriorates over the continuous operation of the EATHEs, their performance under periodic operations were studied. A validated three-dimensional, transient numerical model was used to carry out the analysis. The governing equations, based on the k–$\mathcal{E}$ model and energy equation were used to describe the turbulence and heat transfer phenomena. The results are compared in terms of the outlet temperature of air and heat transfer rate."

The parameters considered in this model are thermal conductivity of soil, thermal conductivity of PVC, Soil surface temperature, Velocity of air, Relative direction of air movement, Size of the tube, specific geometry of tube, number and arrangement of the tubes, surface area of the fins and orientation of the fins, amongst other possible parameters to investigate on.

The general layout of this case study is similar to that of case study #2 and therefore the method suggested for case study #2 will work effectively for this case as well.

Chapter 5: Conclusion and Further Research

Conclusion

In this thesis, an attempt was made to introduce the possibility of utilizing Machine Learning applications in IUT-MCE research environment through creating a precedence of utilizing such a system and providing a guideline based on the experience of the attempt. Although a working example of Machine Learning application could not be provided, the endeavor resulted in the identification of major setbacks to the aspiration made and proposals were made in order to deal with said setbacks. The first of these proposals dealt with the issue of an overwhelming lack of available data required to train Machine Learning applications by suggesting that a database containing primary information regarding various analytical and experimental projects be secured. The second proposal acts along the same line of thought by suggesting acquiring secondary level data (that is, computed) through the usage of relatively simpler Machine Learning algorithms on primary data. The third proposal is to have more advanced algorithms make use of the foundation gained through the first two proposals to gain some meaningful results. The last of the proposals state that since Machine Learning is a field which involves Computer science heavily, IUT-MCE make a collaboration of joint-research on future endeavors using Machine Learning applications, provided that the bulk of the investigative effort remains on Mechanical Engineering field being investigated on.

Further Research

The writers feel that there is a large range of possibilities with regards to future of researching Machine Learning applications in the field of Mechanical Engineering. For now, there are two main areas the writers feel the research focus should be on:

1. Further investigation on the viability and a more detailed guideline on the proposals made in this thesis.
2. Investigation on Machine Learning algorithms that are most suited for chosen research areas under Mechanical Engineering.

References

1. Finger, Susan & R. Dixon, John. (1989). A review of research in mechanical engineering design. Part I: Descriptive, prescriptive, and computer-based models of design processes. *Research in Engineering Design*. *1*. 51-67. 10.1007/BF01580003.
2. Dutton. D.M., Conroy. G.V., (1997), A review of Machine Learning, *The Knowledge Engineering Review, 12, 4,*: 341-367.
3. Engineering "mechanical engineering". The American Heritage Dictionary of the English Language, Fourth Edition.
4. "Heron of Alexandria". Encyclopædia Britannica 2010 - Encyclopædia Britannica OnlineNeedham, Joseph (1986). *Science and Civilization in China: Volume 4*. Taipei: Caves Books, Ltd.
5. Needham, Joseph (1986). *Science and Civilization in China: Volume 4*. Taipei: Caves Books, Ltd.
6. Al-Jazarí. *The Book of Knowledge of Ingenious Mechanical Devices: Kitáb fí ma'rifat al-hiyal al-handasiyya*. Springer, 1973.
7. https://www.britannica.com/technology/engineering
8. R. A. Buchanan. The Economic History Review, New Series, Vol. 38, No. 1 (Feb., 1985): 42–60.
9. http://archive.wikiwix.com/cache/20110223145123/http://anniversary.asme.org/history.shtml
10. https://www.britannica.com/science/thermodynamics
11. https://www.qsstudy.com/physics/significance-first-law-thermodynamics
12. *Knudsen, Jens M.; Hjorth, Poul (2012). Elements of Newtonian Mechanics. Springer Science & Business Media. : 30. ISBN 978-3-642-97599-8.*
13. Rustum Roy (1979) interdisciplinary science on campus, pages 161–96 in *Interdisciplinarity and Higher Education*, J.J. Kockelmans editor, Pennsylvania State University Press
14. Olson, Gregory. "A Materials Science Timeline". *Materials World Modules*.
15. Vincent, Bernedetta. "Materials science and engineering: an artificial discipline about to explode". *History of Recent Materials Science*.
16. https://chemistry.stackexchange.com/questions/69346/a-unit-cell-for-graphene
17. Papageorigiou. D.M., Kinloch I.A.m Young R.J., (2017), Mechanical properties of graphene and graphene-base nanocomposites, *Progress in Materials Science, 90*, : 75-127.
18. Saal, J. E., Kirklin, S., Aykol, M., Meredig, B., and Wolverton, C. "Materials Design and Discovery with High-Throughput Density Functional Theory: The Open Quantum Materials Database (OQMD)", *JOM* **65**: 1501-1509 (2013). doi:10.1007/s11837-013-0755-4
19. https://en.wikipedia.org/wiki/Structural_analysis
20. Victor E. Saouma. "Lecture Notes in Structural Engineering" (PDF). University of Colorado.

21. https://en.wikipedia.org/wiki/History_of_structural_engineering#cite_note-Saouma-1
22. https://www.padeepz.net/ce6602-syllabus-structural-analysis-2-regulation-2013-anna-university/
23. https://en.wikipedia.org/wiki/Fluid_mechanics
24. https://www.scribd.com/document/39741480/History-of-Fluid-mechanics
25. https://www.chmarine.com/acatalog/Barigo-Sole-70mm-Brass-Barometer.html
26. http://www.npd-solutions.com/cfd.html
27. Gorgius. A., (2012), A reliable approach to the solution of Navier-Stokes equations, *Applied Mathematics Letters, 25, 12,*
28. http://www.corvetteblogger.com/images/content/080409_19.jpg
29. Landau. L.D., Lifshitz. E.M., *Course of Theoretical Physics: Theory of Elasticity* Butterworth-Heinemann, ISBN 0-7506-2633-X
30. https://www.britannica.com/science/mechanics-of-solids
31. https://www.journals.elsevier.com/mechatronics
32. https://en.wikipedia.org/wiki/Mechatronics
33. https://medium.com/mistyrobotics/mech-a-what-mechatronics-the-engineering-field-you-didnt-know-existed-1b6edc7b3fa9
34. http://engineering.nyu.edu/mechatronics/Description/evolution.htm
35. Xia F., Yang L.T., Wang L., Vinel A., (2012). Internet of Things. *International Journal of Communications System. 25* : 1101-1102. DOI: 10.1002/dac.2417
36. https://www.forbes.com/sites/bernardmarr/2016/02/19/a-short-history-of-machine-learning-every-manager-should-read/#64e734a615e7
37. https://en.wikipedia.org/wiki/Machine_learning
38. Michalski, R. S., Carbonell, J. G., & Mitchell, T. M. (Eds.). (2013). *Machine learning: An artificial intelligence approach*. Springer Science & Business Media.
39. Morris, A. (2014, August) *World's Largest Materials Database Now Open*, Retrieved from:https://www.mccormick.northwestern.edu/magazine/fall-2014/materials-database.html
40. Rafiee, J., Tse, P. W., Harifi, A., & Sadeghi, M. H. (2009). A novel technique for selecting mother wavelet function using an intelli gent fault diagnosis system. *Expert Systems with Applications*, *36*(3), 4862-4875.
41. Ling, J., & Templeton, J. (2015). Evaluation of machine learning algorithms for prediction of regions of high Reynolds averaged Navier Stokes uncertainty. *Physics of Fluids*, *27*(8), 085103.
42. Şencan, A., & Kalogirou, S. A. (2005). A new approach using artificial neural networks for determination of the thermodynamic properties of fluid couples. *Energy Conversion and Management*, *46*(15-16), 2405-2418.
43. Domanski, P. A., Brown, J. S., Heo, J., Wojtusiak, J., & McLinden, M. O. (2014). A thermodynamic analysis of refrigerants: Performance limits of the vapor compression cycle. *International Journal of Refrigeration*, *38*, 71-79.
44. https://www.mathworks.com/help/stats/machine-learning-in-matlab.html
45. https://www.britannica.com/science/algorithm

46. Powell, M. J. (1973). On search directions for minimization algorithms. *Mathematical programming*, *4*(1), 193-201.

47. Yu, Wen Ci. 1979. "The convergent property of the simplex evolutionary technique". *Scientia Sinica* [*Zhongguo Kexue*]: 69–77.
48. Spendley, W. G. R. F. R., Hext, G. R., & Himsworth, F. R. (1962). Sequential application of simplex designs in optimisation and evolutionary operation. *Technometrics*, *4*(4), 441-461.
49. http://www.scholarpedia.org/article/Nelder-Mead_algorithm
50. https://codesachin.wordpress.com/2016/01/16/nelder-mead-optimization/
51. https://www.mathworks.com/products/matlab.html
52. Zhu, J. (2017). *Improvements on mapping soil liquefaction at a regional scale* (Doctoral dissertation, Tufts University).
53. Friedman, J., Hastie, T., & Tibshirani, R. (2001). *The elements of statistical learning* (Vol. 1, No. 10). New York, NY, USA:: Springer series in statistics.
54. Breiman, L. (1996). Bagging predictors. *Machine learning*, *24*(2), 123-140.
55. Rodriguez, J. J., Kuncheva, L. I., & Alonso, C. J. (2006). Rotation forest: A new classifier ensemble method. *IEEE transactions on pattern analysis and machine intelligence*, *28*(10), 1619-1630.
56. https://en.wikipedia.org/wiki/Density_functional_theory#cite_ref-dftmag_1-0
57. http://newton.ex.ac.uk/research/qsystems/people/coomer/dft_intro.html
58. Assadi, M. H. N., & Hanaor, D. A. (2013). Theoretical study on copper's energetics and magnetism in TiO2 polymorphs. *Journal of Applied Physics*, *113*(23), 233913.
59. Grimme, S. (2006). Semiempirical hybrid density functional with perturbative second-order correlation. *The Journal of chemical physics*, *124*(3), 034108.
60. Vignale, G., & Rasolt, M. (1987). Density-functional theory in strong magnetic fields. *Physical review letters*, *59*(20), 2360.
61. Runge, E., & Gross, E. K. (1984). Density-functional theory for time-dependent systems. *Physical Review Letters*, *52*(12), 997.
62. Gross, E. K., & Dreizler, R. M. (Eds.). (2013). *Density functional theory* (Vol. 337). Springer Science & Business Media.
63. Pearson, M., Smargiassi, E., & Madden, P. A. (1993). Ab initio molecular dynamics with an orbital-free density functional. *Journal of Physics: Condensed Matter*, *5*(19), 3221.
64. Leithold, L. (1996). *The calculus 7* (p. 1394). HarperCollins College Publishing.
65. https://en.wikipedia.org/wiki/Gradient_boosting
66. Friedman, J. H. (2001). Greedy function approximation: a gradient boosting machine. *Annals of statistics*, 1189-1232.
67. Mason, L., Baxter, J., Bartlett, P. L., & Frean, M. R. (2000). Boosting algorithms as gradient descent. In *Advances in neural information processing systems* (pp. 512-518).
68. https://www.analyticsvidhya.com/blog/2016/11/an-introduction-to-clustering-and-different-methods-of-clustering/
69. Wilks, D. S. (2011). Cluster analysis. In *International geophysics* (Vol. 100, pp. 603-616). Academic press.

70. Shipp, M. A., Ross, K. N., Tamayo, P., Weng, A. P., Kutok, J. L., Aguiar, R. C., ... & Ray, T. S. (2002). Diffuse large B-cell lymphoma outcome prediction by gene-expression profiling and supervised machine learning. *Nature medicine*, *8*(1), 68.
71. Shapiro, E. Y. (1981). *Inductive inference of theories from facts*. Yale University, Department of Computer Science.
72. Džeroski, S. (2002). *Data mining tasks and methods: Rule discovery: inductive logic programming approaches* (pp. 348-353). Oxford University Press, Inc.
73. https://skymind.ai/wiki/deep-reinforcement-learning
74. Bertsekas, D. P., & Tsitsiklis, J. N. (1995, December). Neuro-dynamic programming: an overview. In *Proceedings of the 34th IEEE Conference on Decision and Control* (Vol. 1, pp. 560-564). Piscataway, NJ: IEEE Publ..
75. Kaelbling, L. P., Littman, M. L., & Moore, A. W. (1996). Reinforcement learning: A survey. *Journal of artificial intelligence research*, *4*, 237-285.
76. Bengio, Y., Courville, A., & Vincent, P. (2013). Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, *35*(8), 1798-1828.
77. Chopra, S., Hadsell, R., & LeCun, Y. (2005, June). Learning a similarity metric discriminatively, with application to face verification. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*(Vol. 1, pp. 539-546). IEEE.
78. Weinberger, K. Q., Blitzer, J., & Saul, L. K. (2006). Distance metric learning for large margin nearest neighbor classification. In *Advances in neural information processing systems* (pp. 1473-1480).
79. Piatetsky-Shapiro, G. (1991). Discovery, analysis, and presentation of strong rules. *Knowledge discovery in databases* , 229-238.
80. Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine learning*, *20*(3), 273-297.
81. Ben-Hur, A., Horn, D., Siegelmann, H. T., & Vapnik, V. (2001). Support vector clustering. *Journal of machine learning research*, *2*(Dec), 125-137.
82. Deng, L., & Yu, D. (2014). Deep learning: methods and applications. *Foundations and Trends® in Signal Processing*, *7*(3–4), 197-387.
83. Bengio, Y., Courville, A., & Vincent, P. (2013). Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, *35*(8), 1798-1828.
84. Bengio, Y., Courville, A., & Vincent, P. (2013). Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, *35*(8), 1798-1828.
85. https://www.python.org/doc/essays/blurb/
86. https://www.techopedia.com/definition/3533/python
87. Python, J. (2007). Python programming language. In *USENIX Annual Technical Conference*.
88. Chemisky, Y., Chatazigeorgiou, G., & Meraghni, F. Simmit: An open source project for Interactive Learning Experience in Mechanics of Materials.

89. https://www.artima.com/intv/pythonP.html
90. http://python-history.blogspot.com/2009/01/brief-timeline-of-python.html
91. http://www.amk.ca/python/2.0
92. http://www.numpy.org/
93. http://www.numpy.org/license.html#license
94. https://en.wikipedia.org/wiki/SciPy
95. https://www.scipy.org/about.html
96. Abu-Mostafa, Y.S. (2014, February), Learning from Data, Retrieved from: https://www.edx.org/course/learning-data-caltechx-cs1156x-0
97. VanderPlas, J. (2015, July 06), The Model Complexity Myth, Retrieved from: https://jakevdp.github.io/blog/2015/07/06/model-complexity-myth/

1.