NORTH ATLANTIC TREATY ORGANIZATION

SCIENCE AND TECHNOLOGY ORGANIZATION

AC/323(IST-163)TP/1080

www.sto.nato.int

STO TECHNICAL REPORT **TR-IST-163**

Deep Machine Learning for Cyber Defence

(Apprentissage automatique profond pour la cyberdéfense)

Report of STO Research Task IST-163 (IWA).

Published October 2022

NORTH ATLANTIC TREATY ORGANIZATION

SCIENCE AND TECHNOLOGY ORGANIZATION

AC/323(IST-163)TP/1080

www.sto.nato.int

STO TECHNICAL REPORT **TR-IST-163**

Deep Machine Learning for Cyber Defence

(Apprentissage automatique profond pour la cyberdéfense)

Report of STO Research Task IST-163 (IWA).

The NATO Science and Technology Organization

Science & Technology (S&T) in the NATO context is defined as the selective and rigorous generation and application of state-of-the-art, validated knowledge for defence and security purposes. S&T activities embrace scientific research, technology development, transition, application and field-testing, experimentation and a range of related scientific activities that include systems engineering, operational research and analysis, synthesis, integration and validation of knowledge derived through the scientific method.

In NATO, S&T is addressed using different business models, namely a collaborative business model where NATO provides a forum where NATO Nations and partner Nations elect to use their national resources to define, conduct and promote cooperative research and information exchange, and secondly an in-house delivery business model where S&T activities are conducted in a NATO dedicated executive body, having its own personnel, capabilities and infrastructure.

The mission of the NATO Science & Technology Organization (STO) is to help position the Nations' and NATO's S&T investments as a strategic enabler of the knowledge and technology advantage for the defence and security posture of NATO Nations and partner Nations, by conducting and promoting S&T activities that augment and leverage the capabilities and programmes of the Alliance, of the NATO Nations and the partner Nations, in support of NATO's objectives, and contributing to NATO's ability to enable and influence security and defence related capability development and threat mitigation in NATO Nations and partner Nations, in accordance with NATO policies.

The total spectrum of this collaborative effort is addressed by six Technical Panels who manage a wide range of scientific research activities, a Group specialising in modelling and simulation, plus a Committee dedicated to supporting the information management needs of the organization.

- AVT Applied Vehicle Technology Panel
- HFM Human Factors and Medicine Panel
- IST Information Systems Technology Panel
- NMSG NATO Modelling and Simulation Group
- SAS System Analysis and Studies Panel
- SCI Systems Concepts and Integration Panel
- SET Sensors and Electronics Technology Panel

These Panels and Group are the power-house of the collaborative model and are made up of national representatives as well as recognised world-class scientists, engineers and information specialists. In addition to providing critical technical oversight, they also provide a communication link to military users and other NATO bodies.

The scientific and technological work is carried out by Technical Teams, created under one or more of these eight bodies, for specific research activities which have a defined duration. These research activities can take a variety of forms, including Task Groups, Workshops, Symposia, Specialists' Meetings, Lecture Series and Technical Courses.

Published October 2022

ISBN 978-92-837-2397-4

Table of Contents

List of Figures

Acknowledgements

The Chair and Vice-Chair of this Research Task Group (RTG) activity would like to acknowledge the important contribution of their fellow team members in the creation of this technical report on Deep Machine Learning for Cyber Defense. It was possible only because of your dedication, perseverance and hard work.

The entire team of this NATO RTG Information Systems Technology (IST) would like to thank the NATO Collaboration Support Office, the IST Panel, and each of their respective organisations for supporting this work. Additionally, the team wants to thank the reviewers of this document for their valuable time and input provided.

IST-163 Membership List

CHAIR

Dr. Frederica FREE NELSON*
Army Research Laboratory (ARL)
UNITED STATES
Email: frederica.f.nelson.civ@army.mil

VICE-CHAIR

Mr. Raphael ERNST*
Fraunhofer FKIE
GERMANY
Email: raphael.ernst@fkie.fraunhofer.de

MEMBERS

Dr. Lieutenant Commander Bernt AKESSON*
Finnish Defence Research Agency
FINLAND
Email: bernt.akesson@mil.fi

Prof. David ASPINALL*
University of Edinburgh
UNITED KINGDOM
Email: david.aspinall@ed.ac.uk

Mr. Markus ASPRUSTEN*
Norwegian Defence Research Establishment (FFI)
NORWAY
Email: markus.asprusten@ffi.no

Mr. Bulent BASKUS
SSB
TURKEY
Email: bbaskus@ssb.gov.tr

Dr. Tracy BRAUN*
US Army Research Laboratory
UNITED STATES
Email: tracy.d.braun.civ@army.mil

Capt. Thibault DEBATTY
Royal Military Academy
BELGIUM
Email: thibault.debatty@rma.ac.be

Maj. Dr. Jerzy DOLOWSKI
Military University of Technology
POLAND
Email: jerzy.dolowski@wat.edu.pl

Lt Col Matthias FRANK
KdoCIRU
GERMANY
Email: matthias1frank@bundeswehr.org

Dr. Gudmund GROV
Norwegian Defence Research Establishment (FFI)
NORWAY
Email: Gudmund.Grov@ffi.no

Mrs. Yeliz HENDEN*
Roketsan
TURKEY
Email: yeliz.topcu@roketsan.com.tr

Mr. Espen KJELLSTADLI*
Norwegian Defence Research Establishment (FFI)
NORWAY
Email: Espen-Hammer.Kjellstadli@ffi.no

Mr. Pascal LLORENS
THALES System & Software Architectures and Cybersecurity
FRANCE
Email: pascal.llorens@thalesgroup.com

*** Contributing Author**

Mr. Marek MALOWIDZKI
Military Communication Institute
POLAND
Email: m.malowidzki@wil.waw.pl

Mr. Joey MATHEWS*
US Naval Research Laboratory
UNITED STATES
Email: joseph.mathews@nrl.navy.mil

Prof. Dr. Ir. Wim MEES
Ecole Royale Militaire
BELGIUM
Email: Wim.Mees@rma.ac.be

M.S. Col. Fuat OZCAKMAK*
Turkish Air Force
TURKEY
Email: fozcakmak@hvkk.tsk.tr

Dr. Ferhat OZGUR CATAK
Tubitak Bilgem
TURKEY
Email: ozgur.catak@tubitak.gov.tr

Col.Dr. Zbigniew PIOTROWSKI
Military University of Technology
POLAND
Email: zbigniew.piotrowski@wat.edu.pl

Mr. François SAUSSET
THALES
Email: francois.sausset@thalesgroup.com

Burkay SUCU
Havelsan
TURKEY
Email: bsucu@havelsan.com.tr

CDR (ret.) Topi TUUKKANEN
Finnish Defence Research Agency
FINLAND
Email: topi.tuukkanen@mil.fi

Mr. Cornelis VERKOELEN
TNO
NETHERLANDS
Email: cor.verkoelen@tno.nl

Dr. Emre YÜCE
Havelsan
TURKEY
Email: eyuce@havelsan.com.tr

ADDITIONAL CONTRIBUTORS

Dr. Christian CALLEGARI*
National Laboratory – CNIT
ITALY
Email: christian.callegari@cnit.it

PANEL/GROUP MENTOR

Prof. Dr. Nazife BAYKAL
Middle East Technical University (ODTU)
TURKEY
Email: baykal@metu.edu.tr

*** Contributing Author**

Deep Machine Learning for Cyber Defence

(STO-TR-IST-163)

Executive Summary

Cyber threats grow increasingly pervasive. Recent high-profile intrusions illustrate how surreptitious cyberspace effects can challenge the 21st century's strategic international security landscape. The growing reliance upon digital technology in every economic sector and aspect of human life strongly suggest this trend will continue. NATO Allies are responding with increasingly robust security and defence of the cyber landscape, especially as it intersects with military systems, platforms, and missions.

The demand for increased resilience and robustness has accelerated the exploration and adoption of Artificial Intelligence technology, i.e., techniques that enable computers to mimic human intelligence, for cyber defence. Deep Machine Learning (DML) is one such state-of-the-art technique which demonstrates considerable potential in cybersecurity as well as many other application domains. Deep Machine Learning can enhance cyber resilience with defences that evolve with threats over time and reduce the overall burden of manual data analysis by human experts. Deep Machine Learning facilitates faster responses, most especially with ample and sufficient training. Some possible consideration includes adversarial examples within training and model development in building or generating data models.

This technical report takes initial steps in consolidating NATO-wide knowledge in the field of cyber defence applications of DML. It further identifies gaps between current solutions and military needs and structures the pursuit of promising cyber defence applications of DML for the military domain accordingly. The research group, with the embodiment of the technical report at a core, examines the National Institute of Standards and Technology security guidelines from the perspective of Malware Detection, Event Management, Information Management, Vulnerability Management, Software Assurance, Asset Management, Licence Management, Network Management, and Configuration Management.

The report examines the intricate utility of DML, practical implementations as well as open challenges. The Research Task Group comprises experts across the fields of data science, machine learning, cyber defence, modelling & simulation, and systems engineering. Researchers and practitioners consider aggregation of data, characterization of data, the need to share data, and the sharing of data models, or the generators thereof. These factors, including how data will be processed, trained, accessed, and related techniques such as transfer, or federated learning are also considered.

Apprentissage automatique profond pour la cyberdéfense

(STO-TR-IST-163)

Synthèse

Les cybermenaces sont de plus en plus omniprésentes. De récentes intrusions très médiatisées illustrent de quelle façon le cyberespace peut subrepticement bouleverser le paysage international stratégique de la sûreté. La confiance croissante accordée à la technologie numérique dans tous les secteurs économiques et aspects de la vie humaine suggère fortement que cette tendance perdurera. Les Alliés de l'OTAN y répondent par une sûreté et une défense robustes du paysage cybernétique, en particulier lorsque celui-ci recoupe les systèmes, plateformes et missions militaires.

La demande de renforcement de la résilience et de la robustesse a accéléré l'exploration et l'adoption des technologies d'intelligence artificielle, autrement dit, des techniques qui permettent aux ordinateurs d'imiter l'intelligence humaine, pour la cyberdéfense. L'apprentissage automatique profond (DML) est l'une de ces techniques de pointe qui montre un potentiel considérable en cybersécurité, comme dans beaucoup d'autres domaines d'application. L'apprentissage automatique profond peut favoriser la cyber-résilience en faisant évoluer la défense avec les menaces et en réduisant la charge générale d'analyse manuelle des données par des spécialistes humains. L'apprentissage automatique profond accélère la réponse, plus particulièrement avec un entraînement suffisant et de grande ampleur. Il peut être envisagé de fournir des exemples d'adversaire pendant l'entraînement et le développement des modèles, pour établir ou produire des modèles de données.

Le présent rapport technique constitue un premier pas vers la consolidation des connaissances de l'OTAN en matière d'applications du DML à la cyberdéfense. Ce rapport identifie l'écart existant entre les solutions actuelles et les besoins militaires et structure en conséquence la recherche d'applications prometteuses du DML à la cyberdéfense dans le domaine militaire. Le groupe de recherche, dans l'optique du rapport technique, a examiné les lignes directrices du National Institute of Standards and Technology en matière de sûreté, sur le plan de la détection des logiciels malveillants, la sûreté des logiciels et la gestion des événements, des informations, des vulnérabilités, des actifs, des licences, du réseau et de la configuration.

Le rapport étudie l'utilité complexe du DML, les applications pratiques de celui-ci et les défis lancés. Le groupe de recherche se compose d'experts en science des données, apprentissage automatique, cyberdéfense, modélisation et simulation et ingénierie des systèmes. Les chercheurs et praticiens étudient l'agrégation et la caractérisation des données, le besoin de partager les données et le partage des modèles de données ou de leurs générateurs. Ces facteurs, notamment le mode de traitement des données, l'entraînement du système, l'accès aux données et les techniques liées telles que l'apprentissage par transfert ou l'apprentissage fédéré, sont également étudiés.

Chapter 1 – IST-163 BACKGROUND

Cyber threats grow increasingly advanced, and adversaries are more strategic and can manifest threats from anywhere in the world. Adversaries of today have resources and time and can invoke damaging attacks with ease, given availability of time and resources.

The availability of and abundance of data in different formats also help to create a level of flexibility for the adversaries that would not have existed without the influx of data [1]. Because of the easy access to tools and technologies by adversaries, availability of all forms of big data, cyber-attacks are at an all-time high and NATO countries have to enhance their strategic positions by mitigating tools and techniques to alleviate cyber threats against military systems, platforms, and mission [2].

Mitigating technologies will include the latest and greatest technologies to create resilience, detect and respond to attacks in time and recover before any damage or compromise to platforms occurs.

The world is becoming more digitised [3] and the military is no exception. With the onset of advanced tools and digitalisation of technologies, researchers must prepare by examining defensive techniques to prevent disruption and degradation to military systems and platforms.

The RTG plans to explore applications of Deep Machine Learning (DML) to implement and enhance the military strategic cyber position and create a defence that not only addresses threats of today, but threats that might manifest in the future made possible with resources such as increased processing power, advanced tools, and data manipulation techniques.

The main goal of the proposed "IST 163 – Deep Machine Learning for Cyber Defence" activity is to consolidate the NATO-wide knowledge in the field of DML and cyber defence, identify the gaps between civilian solutions and military needs, and collaborate with other NATO countries to use data processing, share data and models, and pursue the transfer of the most promising technologies and applications to the military domain while adhering to standards to ensure data is well tailored to the chosen technologies.

The RTG works to discover DML techniques across NATO nations, uncover how data is processed and fitted for the neural networks and identify gaps within these techniques across the various nations in a comparison of best of breed solutions that could potentially be adopted by other nations that may not have the potential or not be as technologically advance.

The research creates an opportunity for Nations to holistically take a look at capabilities and gaps of DML for cyber defence and study means to enhance cyber defence with state of the art DML methods.

When creating data for DML, researchers from various backgrounds will work together to support use cases that reflect best case scenarios of data utility and models as well as work to ensure data is best fitted for the research. Given the dynamics of proposed data from multiple backgrounds, the curation and sanitising of the data to fit the models will create an opportunity to see the various aspects of what the different types of data does to the DML model.

Special attention will be paid to the alignment of terminology with related activities within other NATO initiatives. As such, it will address a multidisciplinary audience from the fields of artificial intelligence, machine learning, modelling and simulation, and systems engineering.

The working group's efforts will focus on machine learning encompassing the deep learning aspect.

1.1 REFERENCES

[1] Kumari, M. (2019). Application of Machine Learning and Deep Learning in Cybercrime Prevention – A Study. Int. J. Trend Res. Dev, 1-4.

[2] Paruchuri, P., Pearce, J.P., Tambe, M., Ordonez, F., and Kraus, S. (2007). An Efficient Heuristic Approach for Security Against Multiple Adversaries. In Proceedings of the 6th International Joint Conference on Autonomous Agents and Multiagent Systems, 1-8.

[3] Rege, M. (2018). Machine Learning for Cyber Defense and Attack. Data Analytics 2018, 83.

Chapter 2 – MILITARY RELEVANCE

Cyber defence impacts all areas of military operations, including communications, operations, and logistics. As threats evolve in sophistication and adversaries become more innovative, traditional signature-based approaches to detecting threats are easily evaded. Existing defences do not keep pace with the scale at which new vulnerabilities, exploits, and attack vectors emerge. There exists a clear need to develop automated and data-driven defences with models that are appropriate for military systems and coalition operating environments.

Cyber defence techniques which reduce the burden of data analysis and scale to diverse and federated environments are and will continue to be of considerable importance to military operations. One promising area within this category is the application of Machine Learning (ML), the study and development of approaches to pattern recognition without pre-programmed instructions on how to interpret data. Theobold [1] clearly articulates the utility of machine learning:

> *For decades, machines operated on responding to direct user commands. In other words, computers were designed to perform set tasks in response to pre-programmed commands. Now, computers don't strictly need to receive an input command to perform a task but rather input data. Specifically, the machine creates a predictive model based on previous experiences captured in the data. From the input data the machine is then able to formulate a decision on how, where, and when to perform a certain action.* [1]

For two decades in the first half of the 20th century, the armed forces of the United States were the single most important driver of digital computer development [2]. As the commercial computer industry began to take shape, the armed forces and defence industry served as its major marketplace. During its growth, humans programmed all software and served as the principal drivers of computational and algorithmic advances. Object oriented programming made software reusable and increased its scale. The Internet later democratised software. This landscape is now poised to fundamentally shift again with the advent of Deep Machine Learning (DML), a subset of ML. DML technology enables computers to "write" their own software by training models which describe relationships between inputs and outputs. This breakthrough is already accelerating advances in every industry. Studies suggest deep learning will increase the global equity market over the next two decades by nearly 50% [3].

Cyber defence is no exception to this trend. The growing adoption of digital technologies among social and military applications in the latter half of the 20th century, and routine data breaches which characterise the first decades of the 21st century, illustrate the importance of a resilient cyberspace. Artificial Intelligence (AI) applications, to include ML and DML for cyber defence, have already garnered considerable exposure among defence research forums [4], [5], [6], [7], [8], [9], [10], [11]. These applications hold considerable military promise, particularly is it relates to vulnerability discovery, threat identification, situational awareness, and resilient systems.

2.1 THE NATO PERSPECTIVE

Cyber defence is integral to NATO's core mission of cooperative security [12]. In 2002, Allied leaders first publicly acknowledged the need to strengthen capabilities to defend against cyber-attacks [13]. Shortly thereafter in 2003, they established The NATO Computer Incident Response Capability (NCIRC), a team of "first responders" to prevent, detect, and respond to cyber incidents. Since then, the cyber landscape has only grown in importance and focus. In 2008, NATO established the Cooperative Cyber Defence Centre of

Excellence, currently comprising 25 sponsoring nations, and with the mission to enhance capability, cooperation, and information sharing among NATO Allies and partners [14]. In 2014, Allied leaders declared that a cyber-attack could lead to the invocation of the collective defence clause of NATO's founding treaty. In 2016, Allies recognised cyberspace as a domain of military operations. Allied leaders further pledged to enhance the resilience of their national networks and infrastructure as a matter of priority and affirmed that international law applies in cyberspace [15]. Though NATO's main focus is to protect the communications and information systems owned and operated by the Alliance, it provides for streamlined cyber defence governance, assistance to Allied countries in response to cyber-attacks, and integrating cyber defence into operational planning, including civil emergency planning. NATO clearly recognises that its Allies and partners benefit from a predictable and secure cyberspace.

Cyber threats to the security of NATO are growing more frequent, complex, destructive, and coercive. The Alliance must be prepared to defend its networks and operations against the growing sophistication of the cyber threats it faces. Accordingly, Allied doctrine notes cyber defence as one of six key factors that will affect the balance of future military power [16]. NATO policy further frames the pursuit of cyber defence with six key objectives [17]:

- Integrate cyber defence considerations into NATO structures and planning processes in order to perform NATO's core tasks of collective defence and crisis management.
- Focus on prevention, resilience, and defence of critical cyber assets to NATO and its Allies.
- Develop robust cyber defence capabilities and centralise protection of NATO's own networks.
- Develop minimum requirements for cyber defence of national networks critical to NATO's core tasks.
- Provide assistance to achieve a minimum level of cyber defence and reduce vulnerabilities of national critical infrastructures.
- Engage with partners, international organisations, the private sector, and academia.

Recent studies expound on how these objectives are being addressed [18]. Though its members are responsible for protecting their own segments of cyberspace, NATO plays a key role in facilitating interaction, maintaining situational awareness, and moving assets from one ally or tactical situation to another as a crisis or conflict develops. It further champions a high degree of interoperability among multinational forces, to include federating the collection, decision making, and execution elements of allied operations in cyberspace [19]. In 2013, the NATO Defence Planning Process began assigning a number of collective minimum capabilities to its allies to ensure a common baseline to include national Cyber Emergency Response Teams (CERT), encryption, education, training, and information sharing. In cyberspace as well as other domains, NATO plays an integral role in establishing international norms and codes of conduct that promote clarity regarding unacceptable behaviour, condemnation, sanctions, and indictments.

2.2 THE UNITED STATES NATIONAL PERSPECTIVE

United States National Cyber Strategy [20] asserts responsibility for defending US interests from cyber-attack and deterring any adversary that seeks to harm national interests. It further acknowledges the development of capabilities for cyberspace operations by which to achieve this objective. US military doctrine defines cyber operations as an array of actions to prevent unauthorised access, defeat specific threats, and deny adversarial effects [21]. Two key functions stand out in the context of this report:

- **Cyberspace Security (Cybersecurity)**, refer to actions taken within protected cyberspace to prevent unauthorised access to, exploitation of, or damage to computers, electronic communications systems, and other information technology, including platform information technology, as well as the information contained therein, to ensure its availability, integrity, authentication, confidentiality, and nonrepudiation.
- **Cyberspace Defence (Cyber Defence)**, on the other hand, refers to actions taken within protected cyberspace to defeat specific threats that have breached or are threatening to breach cyberspace security measures and include actions to detect, characterise, counter, and mitigate threats, including malware or the unauthorised activities of users, and to restore the system to a secure configuration.

Despite the distinction, both cybersecurity and cyber defence necessitate extensive continuous monitoring of systems and security controls. Joint military doctrine further acknowledges challenges to integrating capabilities that include:

- Nation-State threats with access to resources, personnel, or time unavailable to other actors. Some nations may employ cyberspace capabilities to attack or conduct espionage against the US and its allies. These actors include traditional adversaries; enemies; and perhaps even traditional allies and may be outsourced to third parties, including front companies, patriotic hackers, or other surrogates, to achieve their objectives.
- Non-State threats include organisations unbound by national borders, including legitimate Non-Governmental Organisations (NGOs), criminal organisations, and violent extremist organisations. Non-state threats use cyberspace to raise funds, communicate with target audiences and each other, recruit, plan operations, undermine confidence in governments, conduct espionage, and conduct direct terrorist actions within cyberspace. They may also be used as surrogates by nation-states to conduct attacks or espionage through cyberspace.
- Individual or Small Group threats are enabled by affordable and accessible malware and offensive capabilities. Encompassing a wide array of groups or individuals, these small-scale threats can be co-opted by more sophisticated threats, such as criminal organisations or nation-states, often without their knowledge, to execute operations against targets while concealing the identity of the threat/sponsor and also creating plausible deniability.
- Accidents and Natural Hazards can disrupt the physical infrastructure of cyberspace. Examples include operator errors, industrial accidents, and natural disasters. Recovery from these events can be complicated by the requirement for significant external coordination and reliance on temporary back-up measures.
- Anonymity and Attribution. To initiate an appropriate defensive response, attribution of threats in cyberspace is crucial for any actions external to the defended cyberspace beyond authorised self-defence.
- Geography. The cumulative effects of a defensive response may extend beyond the initial threat. Because of transregional considerations, some defensive actions are coordinated, integrated, and synchronised using centralised execution from a location remote from the supported commander.
- Technology Challenges. Using a cyberspace capability that relies on exploitation of technical vulnerabilities in the target may reveal its functionality and compromise the capability's effectiveness for future missions. This means that once discovered, these capabilities will be widely available to adversaries, in some cases before security measures can be updated to account for the new threat.
- Private Industry and Public Infrastructure. Many of DOD's critical functions and operations rely on contracted commercial assets, including Internet Service Providers (ISPs) and global supply chains, over which DOD and its forces have no direct authority.

- Globalisation. The combination of DOD's global operations with its reliance on cyberspace and associated technologies means DOD often procures mission-essential information technology products and services from foreign vendors.
- Mitigations. DOD partners with the Defense Industrial Base (DIB) to increase the security of information about DOD programs residing on or transiting DIB unclassified networks.

The 2018 National Defense Strategy [22] expressed serious concerns to the US military in every domain – air, land, sea, space, and cyberspace. It further acknowledged the current international security landscape is affected by rapid technological advancements and the changing nature of war. To address this challenge, the US Department of Defense established modernisation priorities that include AI/ML, Autonomy, and Cyber. Cyber is a unique operational domain with significant challenges and potential leap-ahead capabilities for military operations requiring enhanced command, control and situational awareness, and autonomous operations.

The 2019 Federal Cybersecurity Research and Development Strategic Plan [23] articulates the need to augment cybersecurity Research and Development (R&D) with Artificial Intelligence (AI) models, algorithms, and the human-AI interactions in other domains. Incorporating AI techniques into cyber autonomous and semi-autonomous systems will help human analysts operate at faster speed and scale in automated monitoring, analysis, and responses to adversarial attacks. Applications of this include the deployment of Intelligent Autonomous Agents to detect, respond, and recover from adversarial attacks in the increasingly complex cyber battlespace. Expected outcomes include predicting unprecedented security vulnerabilities in firmware, software, and hardware; continuous learning and modelling from attack scenarios based on a learned history of interactions and expected behaviour; defence against attacks that target the AI systems itself using communication patterns, application logic, or authorisation frameworks; and semi/fully autonomous systems decreasing the role of the Human-In-The-Loop (HITL) among cyber operations.

In 2020, the US National Security Commission on Artificial Intelligence [24] highlighted the potential impact of AI techniques to the economy, national security, and human welfare. It noted that America's military rivals are integrating AI concepts and platforms to challenge the United States' decades-long technological advantage. AI deepens the threat posed by cyber-attacks and disinformation campaigns that our adversaries can use to infiltrate society, steal data, and interfere with democracy. It clearly asserts the US Government should leverage AI-enabled cyber defences to protect against AI-enabled cyber-attacks, though they alone can't defend digital infrastructure that is inherently vulnerable.

2.3 A BRIEF HISTORY OF CYBER WARFARE INCIDENTS

According to the NATO Cooperative Cyber Defence Centre of Excellence, at least 83 nations have drafted a national strategy for cybersecurity [25]. Further, all 30 NATO member countries have published one or more governance documents reflecting the strategic importance of defending the cyber landscape. This steadfast posture stems from increasingly common and impactful cyber-attacks occurring over the past two decades. In this section we examine a short history of high-profile breaches that impacted NATO allies, cultivated the current climate, and emphasise the need for better cyber protection, deterrence, detection, and reaction techniques.

In 2003, a series of coordinated attacks originating in China compromised US computer systems. The attacks, designated as "Titan Rain" by the US Government, lasted three years and resulted in the theft of unclassified information from Government agencies, national laboratories, and US defence contractors. The public allegations and denials that followed, stemming from the difficulty of accurately detecting and attributing cyber-attacks, characterised emerging international distrust in cyberspace.

In 2007, Estonia fell victim to a politically motivated cyber-attack campaign lasting twenty-two days. Distributed denial of service attacks led to temporary degradation and loss of service on many commercial and government servers. The majority of attacks targeted non-critical services, i.e., public websites and e-mail. However, a small portion concentrated on more vital targets, such as online banking and the Domain Name System (DNS). The attacks triggered a number of military organisations to reconsider the importance of network security to modern military doctrine and led to the establishment of the NATO Cooperative Cyber Defence Centre of Excellence (CCDCOE) that operates out of Tallinn, Estonia.

In 2008, a series of cyber-attacks disabled websites of Georgian organisations. The attacks were initiated three weeks before a shooting war began in what is regarded as a coordinated cyberspace attack synchronised with major combat actions.

In 2015, Russian computer hackers targeted systems belonging to the US Democratic National Committee. The attack led to a data breach that was determined to be an act of espionage. In addition to highlighting the need to bolster cyber resilience, the response to this event highlighted the need for action to counter disinformation and propaganda operations.

In 2017, the WannaCry ransomware infected more than 200,000 computers in 150 countries. The indiscriminate attack, enabled by ransomware which exploited a vulnerability in the Microsoft Windows operating system, locked the data and demanded payment in bitcoin. The malware was stopped after the lucky discovery of a kill switch, but not before it caused factories to shut operations and hospitals to divert patients.

In 2018, Norwegian military and allied officials confirmed that Russia had disrupted NATO's Trident Juncture exercise in Europe's High North region by persistently jamming GPS signals during the exercise [26]. China has claimed 'the ability to use space-based systems and to deny them to adversaries as central to modern warfare' [27]. The dependence of military operations on space-based assets has grown over the last few decades and space-based assets are increasingly desirable targets for cyber-attack. Both Russia and China prioritise electronic warfare, cyber-attacks, and superiority within the electromagnetic battlespace as part of a strategy to achieve victory in future missions. Available doctrine from these nations highlights a key focus on preventing adversarial satellite-based communication systems from impacting their operational effectiveness. Satellites depend on cyber technology including software, hardware, and other digital components. Space systems are fundamental to provisioning data and services among operations conducted in the air, land, maritime, and even cyber domains. A threat to a satellite's control system or bandwidth poses a direct challenge to national assets and objectives and promote the demand for mitigation measures by which to achieve resilience among these systems.

In 2020, hackers from Armenia and Azerbaijan targeted websites during the Nagarno-Karabekh War. Misinformation and videos of older events have been shared as new and different events related to the war. New social media account creation that posts about Armenia and Azerbaijan spiked, with many from authentic users, but many inauthentic accounts have also been detected. This incident highlights the emergence of social cybersecurity as an emerging area of research [28].

In 2020, a major cyber-attack penetrated thousands of organisations globally by compromising the software supply chain of a popular network monitoring tool, Solarwinds. The extent of breaches which followed were reported to be among the worst cyber-espionage incidents ever suffered by the US, due to the sensitivity and high profile of the targets and the long duration in which the hackers had access. Within days of its discovery, at least 200 organisations around the world had been reported to be affected by the attack.

2.4 BROAD TRENDS IN CYBERSPACE

A growing number of trends characterise the evolution of cyberspace. Cyber technologies play an increasing role among every aspect of our lives. This trend extends to military conflict. Increasing dependence on cyber technologies will introduce new vulnerabilities and erode the boundaries which characterise traditional cyber defences. An increasing intersection among cyberspace and other domains, including critical infrastructures, military weapons systems, and integrated biological, physical, and quantum systems will grow in importance as underlying technological components and interfaces mature. In this section, we identify technological and non-technological trends that will impact the evolution of cyberspace and the underlying utility of ML in its defence applications.

2.4.1 Technological Trends

Hardware, software, and protocols, grow increasingly programmable and complex. Increased programmability affords fast development and delivery windows, but further introduces new vulnerabilities with each new codebase. Increased complexity lends itself to unused code paths, i.e., software bloat, which sustains undesirable attack paths. The increasing presence of third party and open-source hardware and software enable rapid prototyping but are vulnerable to opaque supply chains and loss of provenance.

Applications of autonomy and accelerated decision loops characterise the direction and speed of cyber conflict. Humans will rely in machine intelligence to the confluence of big data, increased computing power, and novel computational algorithms. Increasing cyber speed requires a greater reliance on prevention of compromise, resilience, and optimal man-machine teaming with human experts. At the same time, cyberspace grows increasingly untrustworthy, and emerging security architectures prescribe the need to protect assets and information based on their importance to mission context [29].

Cyberspace application domains grow increasingly diverse. As edge devices remain powered on and accessible, and as applications of low size, weight, and power device connections grow, the pervasive connectivity will increase military dependence on cyberspace. As with cyber-physical systems (i.e., Internet of Things), emerging biological, physical, and quantum applications will necessitate new interfaces with cyberspace. These interfaces will create new opportunities and challenges for cyber defence, such as instrumentation and sensing, side channel attacks, and formal verification.

Machine Learning (ML) will continue to grow its multi-faceted relationship with cyberspace technologies and cyber defence applications. On one hand, ML can augment nearly all cyber technologies and their applications (i.e., design, development, and testing of microelectronics, networking, computing architectures, etc.). On the other hand, advances in cyber technologies (e.g., tensor processing units, quantum computers) can enhance ML capabilities. Given the underlying challenge of pattern recognition in large quantities of data, ML could significantly improve the capabilities and resilience of cyberspace.

2.4.2 Non-Technological Trends

The number of Internet users encompasses more than half the world's population [30]. Though signs point to near term slowdowns in growth caused by declining smartphone shipments and the global pandemic in 2020, innovation continues to drive product improvements. The rapid rise of gathered digital data is key to the success of those fastest growing companies, often through data mining and context-rich enhancements that help personalise products and services. This has led to concerns over misuse of data, user privacy, and problematic content poised to drive market changes or regulation. As digital systems become increasingly sophisticated,

data-rich, and mission critical, so too has the opportunity and will for exploitation. Increasingly, the cybersecurity implications of emerging technology are folded into international diplomacy and national defence considerations. Recent examples include the Vulnerability Equities Process [31], Paris Call for Trust and Security in Cyberspace [32] and the Algorithmic Bill of Rights [33].

Strategic global demand signals, including climate change and resource shortages, could generate new territorial ambitions and alliances, causing the political landscape to dramatically shift. For instance, power generated from space-based solar technology may be beamed to the terrestrial surface, necessitating new critical infrastructure and global points of presence in cyberspace. Similarly, population shifts bought about by natural resource shortages may alter the political and national security landscapes. These changes will introduce new critical infrastructure with dependencies on cyberspace.

Military operations have become critically dependent on cyberspace. This dependency is a vulnerability that can be exploited for asymmetric advantage [34]. The loss, degradation, corruption, unauthorised access, or exploitation of digital terrain provide a significant advantage to an adversary and represents a threat to military objectives. Near peer actors will continue attempting to disrupt cyberspace or counter offensive cyber operations. The democratisation and proliferation of offensive cyber capabilities will further offer specific advantages to non-near peer competitors. Increasingly, a country's capability and influence may be measured by its ability to weaponize consumer electronics, particularly as those same commercially-developed systems will underpin military applications. Consequently, cyber-attacks will grow in scope, frequency, and impact.

At the same time, globalisation will drive increased scrutiny of standards and accountability for military operations. Political and public demands for accountability will be challenged by the increasingly opaque nature of warfare. For instance, deterrence operations carried out in the physical domain necessitate a carefully planned narrative and messaging that is aligned with 24-hour news cycles. Offensive cyber operations, however, are poised to achieve far more surreptitious effects not readily observable or attributable. Cyberwarfare tools have transformed cyberspace into a grey zone battlefield where conflict falls below the threshold of outright war but above that of peacetime.

Warfighting will increasingly integrate cyber with traditional domains (e.g., land, sea, air, space). Warfighting doctrine, international treaties, and general laws will reactively evolve with the balance of power, available technology, and regional conflicts. Democratisation of offensive cyber tools will counter traditional advantages of warfighting in the kinetic domain. Unprecedented connectivity and increasing nationalism will drive the continued use of cyberspace for asymmetric advantage. Disinformation and influence campaigns fuelled by worldwide societal turmoil will likely spill into cyberspace. The increased desire to minimise external influence, enforce data privacy, and govern digital content may drive Internet Balkanization.

This is already evidenced in Russia's declaration to close off its national segment from the global Internet and become 'digitally sovereign' while pursuing a decisive military advantage in cyberspace. Included in this goal, is the creation of information security standards for artificial intelligence systems. Such new technological applications will likely influence the way Russia chooses to achieve its objectives. For instance, Kukkola et al. [35] assert that AI might present Russia with an opportunity to define its digital borders in a flexible manner, reflecting prevailing opinions and loyalty, rather than geographic location. Russian leadership has further asserted that the nation leading AI will be the "ruler of the world," indicating the advances to be transformational, with an impact not fully understood.

2.5 REFERENCES

[1] Theobald, O. (2017). Machine Learning for Absolute Beginners: A Plain English Introduction. 157, Scatterplot Press.

[2] Edwards, P.N. (1997). Why Build Computers? The Military Role in Computer Research. The Closed World: Computers and the Politics of Discourse in Cold War America, 54-74, Cambridge: MIT.

[3] ARK Invest. (Jan 2021). Big Ideas 2021. https://ark-invest.com/big-ideas-2021/

[4] Apruzzese, G., Colajanni, M., Ferretti, L., Guido, A., and Marchetti, M. (2018). On the Effectiveness of Machine and Deep Learning for Cyber Security. 10th International Conference on Cyber Conflict (CyCon). IEEE, 2018.

[5] Väisänen, T., Trinberg, L., and Pissanidis, N. (2016). I Accidentally Malware-What Should I Do... Is This Dangerous? Overcoming Inevitable Risks of Electronic Communication. NATO Cooperative Cyber Defence Centre of Excellence (CCDCOE).

[6] Vaarandi, R., Blumbergs, B., and Kont, M. (2018). An Unsupervised Framework for Detecting Anomalous Messages from Syslog Log Files. NOMS 2018 – 2018 IEEE/IFIP Network Operations and Management Symposium, IEEE.

[7] Hartmann, K., and Steup, C. (2020). Hacking the AI-the Next Generation of Hijacked Systems. 12th International Conference on Cyber Conflict (CyCon), vol. 1300, IEEE.

[8] Lavrenovs, A., Heinäaro, K., Orye, E., NATO CCDCOE, (2020). Towards Cyber Sensing: Venturing Beyond Traditional Security Events. ECCWS 2020 20th European Conference on Cyber Warfare and Security. Academic Conferences and Publishing Limited.

[9] Sharma, A. (2010). Cyber Wars: A Paradigm Shift from Means to Ends. Strategic Analysis 34(1): 62-73.

[10] Apruzzese, G., Colajanni, M., Ferretti, L., Guido, A., and Marchetti, M., (2019). Addressing Adversarial Attacks Against Security Systems Based on Machine Learning. 11th International Conference on Cyber Conflict (CyCon), vol. 900, IEEE.

[11] Hartmann, K., and Giles, K. (2020). The Next Generation of Cyber-Enabled Information Warfare. 12th International Conference on Cyber Conflict (CyCon), vol. 1300, IEEE.

[12] NATO (2021). Cyber Defence. NATO. https://www.nato.int/cps/en/natohq/topics_78170.htm

[13] Brent, L. (2019). NATO's Role in Cyberspace. NATO Review, 2019.

[14] NATO CCDCOE (16 Oct 2018). Expertise and Cooperation Make Our Cyber Space Safer. e-Estonia. Retrieved 29 Aug 2019.

[15] Schmitt, M.N., and Vihul, L.(2014). The Nature of International Law Cyber Norms. Tallinn Papers 5.

[16] NATO Standard (2017). Allied Joint Doctrine AJP-01. (2017). https://www.coemed.org/files/stanags/01_AJP/AJP-01_EDE_V1_E_2437.pdf

[17] NATO (2011). C-M 2011 0042, NATO Policy on Cyber Defence and Cyber Defence Action Plan, 7 June 2011.

[18] Shea, J. (2018). Cyberspace as a Domain of Operations: What Is NATO's Vision and Strategy? Marine Corps University Press Quantico United States.

[19] NATO Standard (2020). Allied Joint Doctrine AJP-3.20 Allied Joint Doctrine for Cyberspace Operations.

[20] Carter, A. (2015). The Department of Defense Cyber Strategy. The US Department of Defense, Washington, DC.

[21] Joint Chiefs of Staff (2018). Cyberspace Operations. Joint Publication 3-12, Joint Chiefs of Staff, Washington, DC.

[22] Mattis, J. (2018). Summary of the 2018 National Defense Strategy of the United States of America. Department of Defense, Washington, United States.

[23] Shannon, G., Bogner, K., Epstein, J., Fraser, T., King, S., Martin, W.B. et al. (2016). Federal Cybersecurity Research and Development Strategic Plan: Ensuring Prosperity and National Security. Technical Report, National Science and Technology Council, Washington, DC.

[24] Schmidt, E., Work, B., Catz, S., Chien, S., Darby, C., Ford, K. et al. (2021). National Security Commission on Artificial Intelligence (AI). National Security Commission on Artificial Intelligence.

[25] NATO Cyber Defence Centre of Excellence (2021). Strategy and Governance. https://ccdcoe.org/library/strategy-and-governance/

[26] Jančárková, T., Lindström, L., Signoretti, M., Tolga, I., and Visky, G. (Eds.) (2020). 12th International Conference on Cyber Conflict 20/20 Vision: the Next Decade.

[27] Office of the Secretary of Defense (2018). Annual Report to Congress: Military and Security Developments Involving the People's Republic of China, US Department of Defense,

[28] Beskow, D.M., and Carley, K.M. (2019). Social Cybersecurity: An Emerging National Security Requirement. Carnegie Mellon University, Pittsburgh, United States.

[29] National Security Agency (Feb 2021). Embracing a Zero Trust Security Model (U/OO/115131-21). https://media.defense.gov/2021/Feb/25/2002588479/-1/-1/0/CSI_EMBRACING_ZT_SECURITY_MODEL_UOO115131-21.PDF

[30] Meeker, M. and Wu, L. (2018). Internet Trends Report, Kleiner Perkins.

[31] United States Government. Vulnerabilities Equities Policy and Process for the United States Government. White House Report (2017).

[32] Macron, E. (2018). Paris Call For Trust and Security in Cyberspace, Paris Call International.

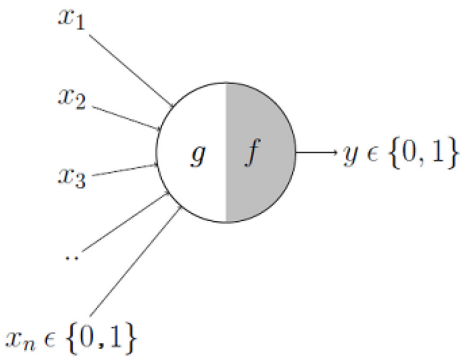
[33] Hosanagar, K. (2020). A Human's Guide to Machine Intelligence: How Algorithms Are Shaping Our Lives and How We Can Stay in Control. Penguin Books.

[34] Reveron, D.S. (2015). China and Cybersecurity: Espionage, Strategy, and Politics in the Digital Domain, 42, Oxford University Press.

[35] Kukkola, J., Ristolainen, M., and Nikkarila, J.P. (2019). Game Player: Facing the Structural Transformation of Cyberspace. Riihimäki, Finnish Defence Research Agency.

Chapter 3 – INTRODUCTION TO DEEP MACHINE LEARNING

The history of deep learning starts with the computational model for neural networks of the human brain created by McCulloch and Pitts in 1943 [1]. Theoretical neurophysiology rests on "all-or-none" law, which proposes the neuron like a switch, i.e., if the excitation exceeds a threshold, the neuron lets the impulse propagate; otherwise, it does not. McCulloch and Pitts suggested that this law ensures that all the neural events and the relations among them can be modelled using propositional logic. McCulloch-Pitts' neuron, also known as **Threshold Logic Unit** (TLU), and models the neuron based on "all-or-none" law (Figure 3-1).



$$g(x_1, x_2, x_3, ..., x_n) = g(\mathbf{x}) = \sum_{i=1}^{n} x_i$$

$$y = f(g(\mathbf{x})) = 1 \quad if \quad g(\mathbf{x}) \geq \theta$$
$$= 0 \quad if \quad g(\mathbf{x}) < \theta$$

Figure 3-1: Threshold Logic Unit.

McCulloch and Pitts showed that any logical proposition can be encoded by an appropriate artificial neural network. This study is very enlightening in means of Artificial Intelligence which starts as an attempt to answer the question "Can computers think?", because it proposes that the brain, which is also a neural network, can encode a computer program.

McCulloch-Pitts' neuron model is a simplified model. It only accepts binary inputs and generates binary output. Threshold values are fixed. The input connections have weights, but it is not about their importance on the result, but about their being of excitatory or inhibitory input.

In the late 1940s, Hebb proposed a theory based on the ability of neural networks to change structurally and physiologically [2]. **Hebbian learning theory** is often summarised as "*Cells that fire together wire together*", which means that if a neuron A repeatedly activates a neuron B, then their connection will be stronger and neuron A will have more effect on the activation of neuron B. One can say that the weight (or importance) of the connection between neuron A and neuron B will be updated incrementally. This also means the connection stores the information and learns that the signal coming from neuron A is important for its activation.

In 1958, Rosenblatt came up with the advanced version of McCulloch-Pitts neuron, which is called **Perceptron** [3] (Figure 3-2). He improved the McCulloch-Pitts neuron to adapt Hebbian learning mechanism. This time, values are not binary inputs, but real-valued. The inputs have importance values (**weights**), and these values also are real-valued. There is also a bias term, which is the threshold value equivalent of McCulloch-Pitts neuron. The output is determined by not only input values, but **weighted sum of input values**. The function which determines the activation initiated by a given input vector, which is also called **activation function**, is the unit step function.

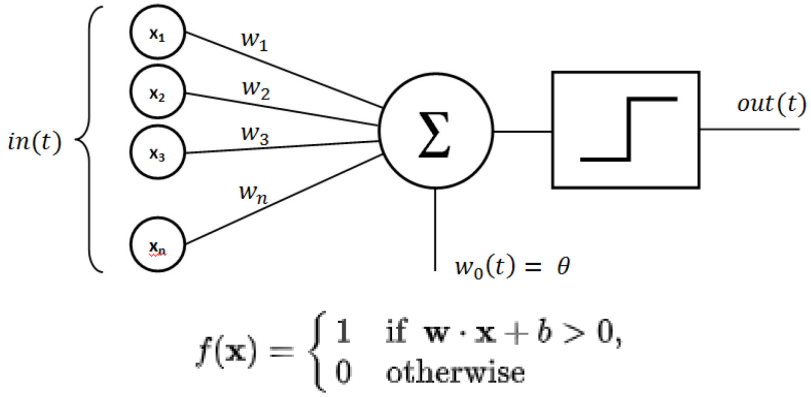


Figure 3-2: Perceptron.

The computationally modelled neurons can be considered as artificial neurons because they are simplified versions of real neurons. The network of artificial neurons is called **Artificial Neural Network**. The first functional ANN is developed by Ivakhnenko and Lapa in 1965 [4]. This was a learning algorithm using **multi-layer perceptron** instead of Rosenblatt's single-layer perceptron. This can be considered as the first demonstration of Deep Learning; because the adjective "deep" in Deep Learning indicates that the learning occurs via multiple layers in the network, and all the methods of Deep Learning is based on the Perceptron structure.

This chapter presents several popular architectures of Deep Learning as means of their approach, state-of-the-art, utilities, practical implementations, and open challenges. Note, that the list of practical implementations is not a finite list, and only serve the purpose of informing about a handful applications the architecture has been known to be applied to.

3.1 CLASSIC NEURAL NETWORKS

3.1.1 Multi-Layer Perceptron (MLP)

3.1.1.1 History

The Multi-layer perceptron is the oldest Deep Learning algorithm. The first known Multi-layer Perceptron was developed by Ivakhnenko and Lapa in 1965 [4] (Figure 3-3). Ivakhnenko's multi-layer perceptron had three layers: an input layer, a hidden layer, and an output layer. Each node is a neuron (perceptron), except for the input nodes. Each node in the preceding layer is connected to every node in the descendent layer. The activation function of perceptrons should be **non-linear**, because if all of them has a linear activation function, then the network can be reduced to single-layer perceptron. The activation function of Ivakhnenko's perceptrons was polynomial. In addition, multi-layer perceptron is said to be **feed-forward**, which means it has no cycles or loops, and the information moves in only one direction from the input nodes, through the hidden nodes and to the output nodes. Training was done with **Group Method of Data Handling**. The method is to iteratively create layers of neurons using combinatorial search to find the best activation function that predicts the output most accurately. In 1971, Ivakhnenko designed an ANN with eight layers, and showed that there can be more than one hidden layer in multi-layer perceptrons [5].

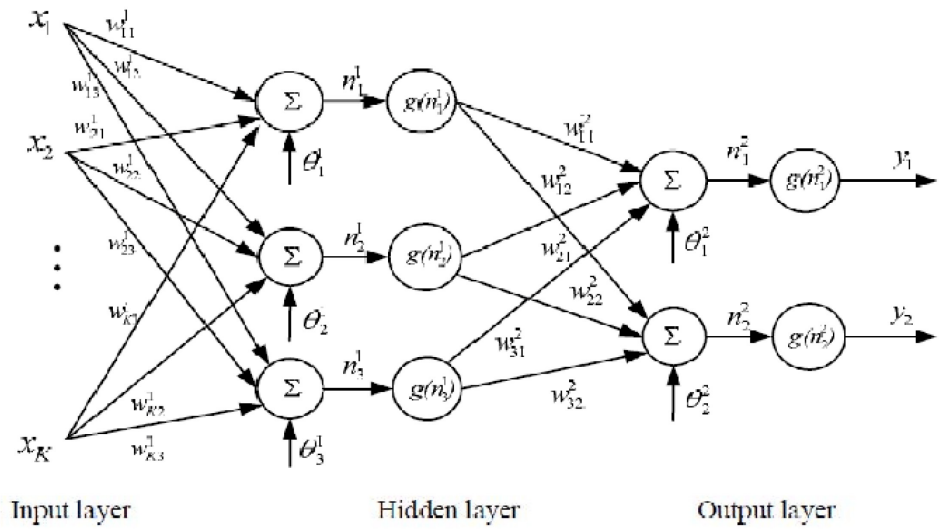


Figure 3-3: The Architecture of Ivakhnenko's Multi-Layer Perceptron.

3.1.1.2 Approach

The Multi-Layer Perceptron (MLP) is a supervised learning[1] algorithm. All the types of multi-layer perceptrons are based on Ivakhnenko's multi-layer perceptron. Therefore, they all have the same properties of Ivakhnenko's. In addition, the number of hidden layers is not limited, and it can be any number. The MLP architecture approximates the non-linear function that maps the features to corresponding classes (classification) or real-number values (regression) with the help of the hidden layer nodes with non-linear activation functions. Some commonly used activation functions are given in the Figure 3-4:

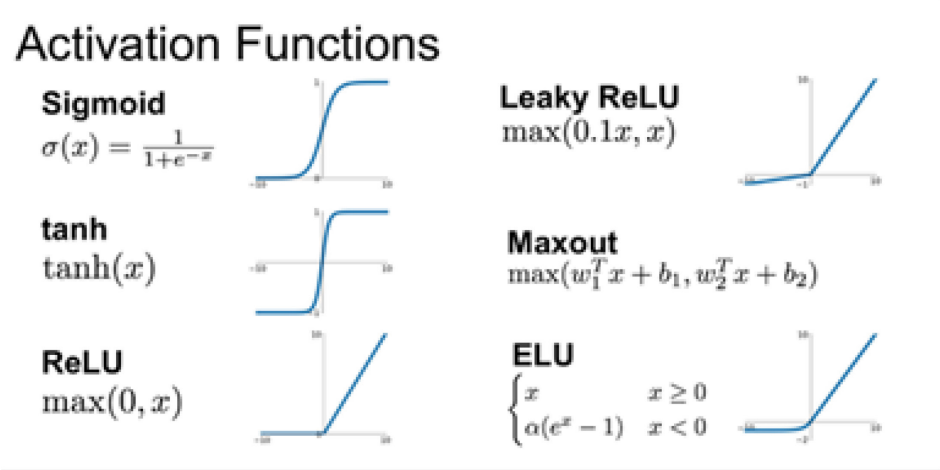


Figure 3-4: Most Commonly Used Activation Functions.

[1] **Supervised learning** means that the algorithm learns a function that maps the i-dimension input to o-dimension output.

Several learning methods have been proposed for artificial neural networks, but the best practice for training was and still is Backpropagation. The method was first proposed in 1970 [6]. In 1982, the method is used specifically for the training of neural networks [7]. Backpropagation method is about finding the difference between actual output and expected output and updating the weights to minimise that difference by iterating backward from the last layer. The method performs through the error function, which defines the error between the actual output and expected output. Gradient descent algorithm updates the weights and biases each iteration according to a learning rate, using the gradient of the error function with respect to the weights and biases. Learning rate indicates how quickly the learning process converge to the right model.

3.1.1.3 State-of-the-Art

In 1991, research on artificial neural networks was interrupted by the vanishing gradient problem, which can make the learning of deep neural networks extremely slow and almost impractical. It is claimed that the learning will be more problematic as the number of inputs fed into the model increases, even learning may completely stop at one point [8].

Moreover, in 2011, it was shown that the vanishing gradient problem can be solved by using **Rectified Linear Unit (ReLU)** as activation function [9]. It is also proved that the networks which uses ReLU activation function, which is called **Deep Rectifier Nets**, will be able to achieve their best without requiring any unsupervised pre-training.

3.1.1.4 Utility

- MLP is capable to learn non-linear models very well.
- MLP is capable to learn models online.[2]
- Training[3] can be parallelised.
- MLPs are noise tolerant. Even noisy or incomplete inputs can be classified correctly.
- MLPs are fault tolerant. Even if some of the neurons or interconnections fail, it keeps working correctly because of its distributed nature. In addition, damage can easily be reducted by re-learning.
- An arbitrary number of input features can be specified.
- An arbitrary number of output values can be specified.

3.1.1.5 Practical Implementations

- Time Series Forecasting [10], [11], [12], [13], [14].
- Facial Expression Recognition [15], [16].
- Healthcare Data Classification [17], [18].

[2] **Online learning** is learning to update the model from each data samples sequentially. On the other hand, **Offline (Batch) learning** needs all the training data to construct the model. There are two terms frequently used for offline learning: batch and epoch. **Batch size** defines the number of samples processed before applying backpropagation. One **Epoch** means that entire dataset passed through the model once. One must specify the batch size and number of epochs for a learning algorithm.

[3] The term **training** represents the process of feeding all the data samples to the architecture to make it learn the mapping function.

- Natural Language Processing [19], [20].
- Speech Synthesis [21].
- Speech Recognition [22], [23].
- Cyber Security [24], [25].

3.1.1.6 Open Challenges

- Learning process is computationally expensive and slow.
- MLP requires tuning several hyperparameters such as the number of hidden layers, the number of neurons in the hidden layers, initial weights, and type of activation function.
- It is hard to find the correct number of hidden layers and the number of neurons in the hidden layers. A small network provides limited learning capabilities, while a large one will overfit the training data and lose the capability of generalisation.

3.1.2 Autoencoders (Auto-Associative Neural Network)

3.1.2.1 History

Auto-association Neural Network is an unsupervised[4] learning technique (Figure 3-5). It uses the inputs as the outputs as well and tries to learn the identity function. First auto-associative neural network was composed of 3 hidden layers: mapping layer, bottleneck layer, de-mapping layer. The nodes of mapping layer and de-mapping layer had sigmoidal activation functions. The nodes of bottleneck layer and output layer had sigmoidal or linear activation functions. The bottleneck layer was there to enforce an internal encoding and compression of inputs. It is claimed that the tasks such as noise filtering, missing sensor replacement and gross error detection and correction can be done [26].

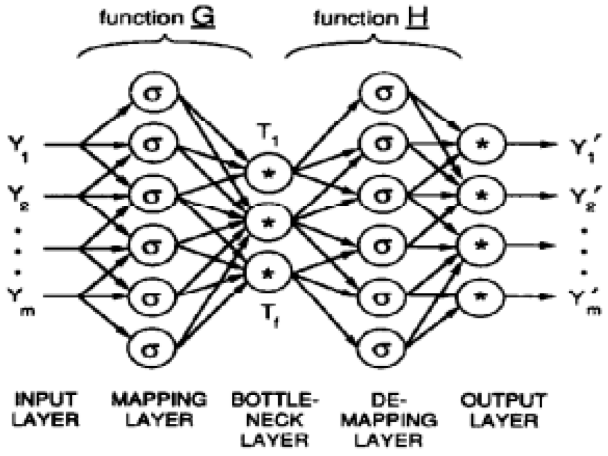


Figure 3-5: Kramer's Auto-Associative Neural Network.

[4] **Unsupervised learning** means that there is no output associated with inputs, and the aim is to detect patterns and correlations on the data distribution.

3.1.2.2 Approach

In auto-associative neural networks, output is identical to input, i.e., trying to reconstruct input at output layer. Therefore, there exist same number of nodes in input layer and output layer. There is a hidden layer (code) which represents the input, not perfectly and exactly but by extracting useful properties of data. The architecture can be broken into two parts according to that hidden layer: encoder and decoder. The Encoder part compresses the information to that hidden layer, and decoder part reconstructs the information from the compressed representation. The activation function commonly used for autoencoders are sigmoid and ReLU. Backpropagation is used for training.

In this technique, the most important thing is to construct the code which extracts the useful information about the data. To achieve this, it is seen that the dimension of the code layer should be less than the dimension of input. The autoencoders which have this property called **undercomplete autoencoders**. This forces the autoencoder learn the most salient features about the data. But there is a pitfall that there is no one-dimension code which represents all the data although it is possible theoretically, the model will probably memorise the data in this situation. The problem is that autoencoders cannot perform the task of learning useful features if they are given too much capacity, and they are prone to memorisation instead of generalisation. The problem also occurs when the dimension of the code is equal to the input dimension or more than the input dimension, which is the case of **overcomplete autoencoders** (Figure 3-6).

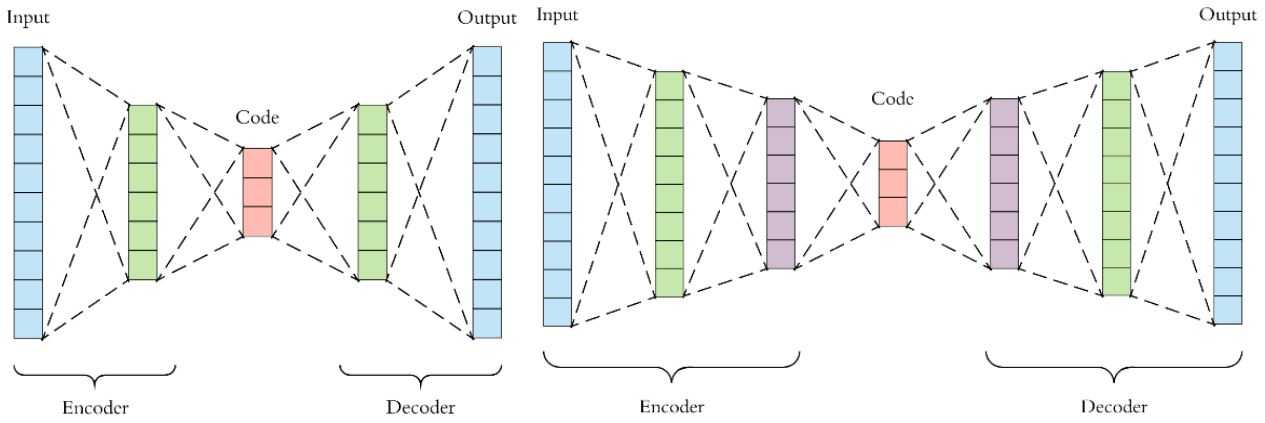


Figure 3-6: Shallow (Left) and Deep (Right) Undercomplete Autoencoders.

Autoencoders are generally trained with only a single-layer encoder and a single-layer decoder. But deep autoencoders -the encoders and decoders are composed of more than one layer- have less computational cost for training, and they can learn with less amount of training data [27]. In addition, it is showed that deep autoencoders produce much better compression than corresponding shallow or linear auto-encoders [28]. For training deep autoencoders, several methods are proposed. The most commonly used method is to pre-train by treating each two consecutive layers as Restricted Boltzmann Machine, then to use a backpropagation to fine-tune the results [28].

3.1.2.3 State-of-the-Art

- To overcome the memorisation problem, regularisation is applied. This is an alternative method without requiring a reduction in the dimension of the code or getting encoders and decoders shallow. Regularisation is choosing the loss function in a way that encourage generalisation in the model. The most known example

of regularisation is sparse autoencoders. In sparse autoencoders, the loss function penalises the activations in the code layer. There are also denoising autoencoders, which first corrupts the input by adding noise and then tries to find identity function with a loss function that penalises the dissimilarity between the real input and the reconstruction of the input. The other regularised type of autoencoders are contractive autoencoders. This type uses a loss function which penalises the dissimilarity between the compressions of similar inputs.

- Variational Autoencoders (VAE): Variational autoencoders share same architecture with traditional autoencoders. However, the mathematics underlying them are different. Variational autoencoders are generative models, which learns how the data is generated and underlying causal relations, whereas traditional autoencoders are discriminative models, which learns copying given observation and preserves only the most relevant aspects of the data.

3.1.2.4 Utility

Autoencoders can be considered as non-linear version of Principal Component Analysis (PCA), which is commonly used for dimensionality reduction and feature extraction in Machine Learning tasks. They can learn more powerful generalisations with minimum loss of information, due to their non-linearity.

3.1.2.5 Practical Implementations

- Dimensionality Reduction [28], [29], [30], [31].
- Information Retrieval [32], [33], [34], [35].
- Anomaly Detection [36], [37], [38].
- Image Processing [39], [40], [41], [42].

3.1.2.6 Open Challenges

- Autoencoders need a large number of noiseless data.
- Autoencoders are domain-specific and need relevant data. For example, if an AE will be trained for car pictures, the data should be composed of only car-related pictures.
- Autoencoders tend to misunderstand what is relevant to problem space and lose important information about the data. VAE is proposed to better catch causal relationships between features.

3.1.3 Generative Adversarial Networks (GAN)

3.1.3.1 History

In 2014, Goodfellow et al. proposed a framework called **Generative Adversarial Nets**, which learns to generate new data with the same distribution as the training set [43]. This framework consists of two models which are simultaneously trained and have influence on each other. One is called generative model G, and it captures the data distribution and generates candidates as if it is sampled from the real distribution. The other one is called discriminative model D, which estimates the probability that a sample came from the training data rather than G. The training objective of G is producing data which fools D into thinking the sample is not synthesised and is part of true data distribution. The authors demonstrated the method by using multi-layer perceptrons for two of the models.

3.1.3.2 Approach

The Generative Adversarial Networks (GAN) approach assembles generative modelling and discriminative modelling (Figure 3-7). Generative modelling is an unsupervised learning method. It aims to discover and learn patterns in input data in a way that the model can be used to generate new examples just like it has been drawn from the original training set, e.g., autoencoders and DBNs are examples of generative modelling. Discriminative modelling is referred to as classification that maps each input to a class label. The architecture of GANs has not changed since Goodfellow proposed it. Two sub-models work together to learn the model in an adversarial way. The generator part generates new examples while discriminator part classifies the generated examples as real or fake. Training occurs in a way that generator tries to generate more realistic examples to fool the discriminator successfully. Both the generator and the discriminator are neural networks. The output of the generator is fed into the discriminator model. Through backpropagation, the generator and discriminator networks are trained one after another, the generator constant is kept during the discriminator training phase and vice versa, until the discriminator has a 50% accuracy. 50% accuracy means that the discriminator tosses a coin to determine whether generated example is real or fake.

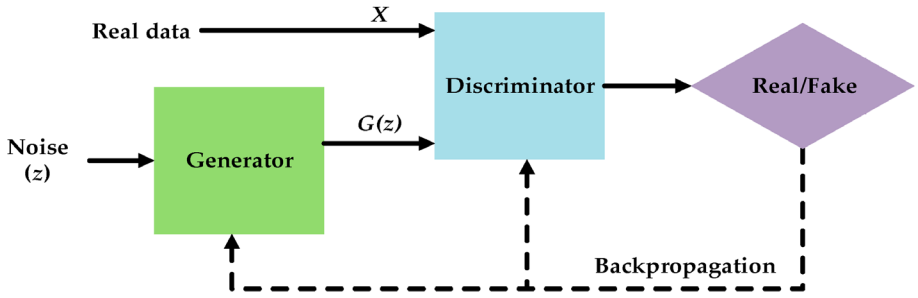


Figure 3-7: Architecture of GAN.

GANs can be extended with the use of conditions. The generative model and discriminator model can be trained in a way that both models are provided with some additional inputs which are conditional on real input. For example, if the models are also provided with the class labels of input, **conditional GANs** are used to generate targeted examples of a given class. It is first demonstrated with specific handwritten digits using MNIST dataset and specific object photographs using CIFAR-10 dataset [44].

3.1.3.3 Utility

GANs are generative models, which learns the causal relationships between features. GANs are generally used to generate multimedia data such as image, music, voice, prose.

3.1.3.4 State-of-the-Art

- StyleGAN [45].
- CycleGAN [46].
- DCGAN [47].
- seGAN [48].
- BiGAN [49].

3.1.3.5 Practical Implementations

- Image-to-Image Translation [50].
- Text-to-Image Synthesis [51].
- Super-Resolution [52].
- Text-to-Speech [53].

3.1.3.6 Open Challenges

- The convergence of GAN is hard to identify. If the GAN continues training after the accuracy of discriminator is 50%, which means the discriminator starts to make random decisions about the performance of generator, the generator starts to train on meaningless response. This problem can affect the quality.
- If the discriminator is too good, the generator may suffer from vanishing gradient problem. Techniques such as Wasserstein loss [54] and modified minimax loss [43] are proposed to overcome this problem.
- There is a phenomenon called mode collapse. When the generator produces a persuasive output to the discriminator, it may learn to produce only that output. The discriminator cannot realise the problem, does not converge, and keep accepting same outputs again and again. Techniques such as Wasserstein loss [54] and unrolled GANs [55] are proposed to overcome this problem.

3.2 CONVOLUTIONAL NEURAL NETWORKS (CNN OR CONVNET)

3.2.1 History

In 1969, Minksy and Rapert showed that complicated functions like XOR cannot be solved by single-layer perceptrons, which means that it is not possible to find the appropriate weights for such problems, and they said that this problem can be solved only if some learning theorem is proposed for multi-layer perceptrons [56]. Thereupon, neural networks research stagnated until 1979. In 1979, Fukushima built a new neural network architecture, which is seen as an inspiration for convolutional neural networks [57] (Figure 3-8). It was specialised to visual pattern recognition through learning, and base on two basic visual cell types in the brain: simple cells and complex cells. This design is again a multilayered and hierarchical ANN, but it was different in such aspects that the input is not a vector, but a two-dimensional array, and there are different types of layers with different tasks. **S-cells** (equivalent to simple cells) are feature-extracting cells. The weights of their input connections are variable and modified through learning. **C-cells** (equivalent to complex cells) tolerates the positional errors of the features. Their input connections come from a layer of S-cells, and the weights of their input connections are fixed and invariable. Features in the input are integrated gradually, e.g., first layers learn how to extract the features like edges besides the last layers learn how to interpret the shape and classify. Learning was done by a method called Selective Response, which repeatedly presents a set of stimulus patterns to the network, chooses S-cells which are more representative, and reinforces their input connections which non-zero signals come through.

In 1989, LeCun et al. proposed the CNN architecture used today, and applied backpropagation method to train a convolutional neural network to recognise handwritten digits, and it was successful (Figure 3-9). The architecture was reminiscent to Fukushima's architecture, but it allowed for supervised and automatic learning unlike the unsupervised and hand-crafted learning of Neocognitron [58].

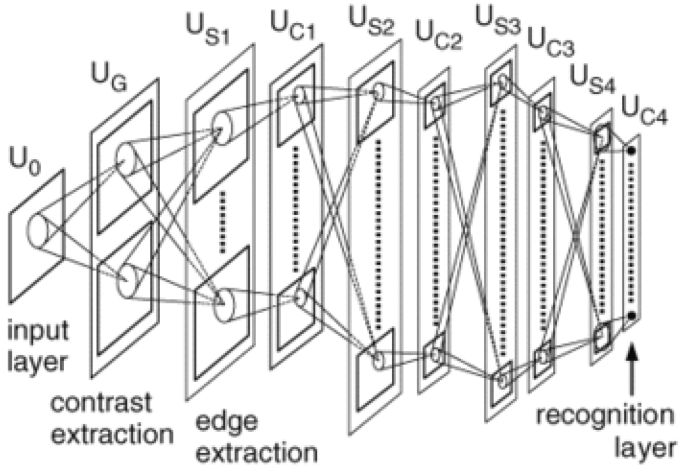


Figure 3-8: Architecture of Fukushima's Neocognitron.

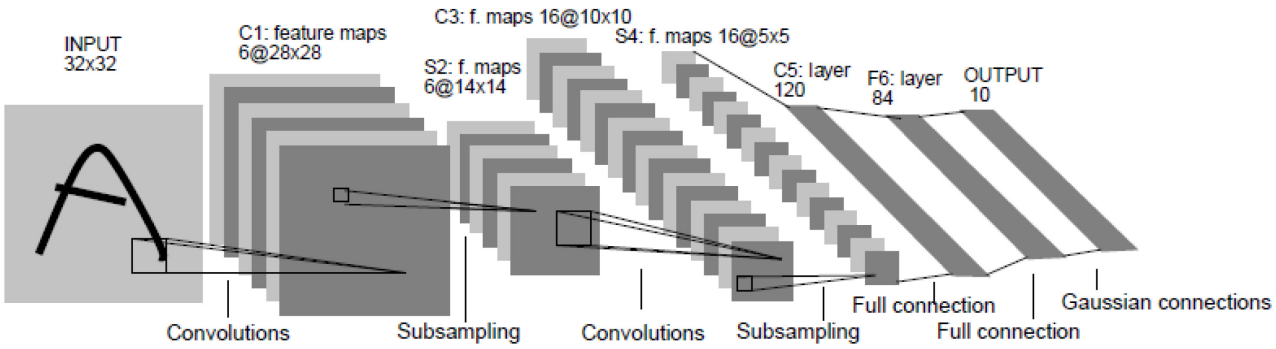


Figure 3-9: Architecture of LeCun's Convolutional Neural Network.

3.2.2 Approach

The convolutional neural networks take an input 2D image and classifies it. An input 2D image is equivalent to a (**width**) x (**height**) x (**number of colour channels**) matrix whose entries are raw pixel values.

CNNs have two major components: feature extraction and classification (Figure 3-10).

Feature extraction is employed with two types of layers: **convolutional layer** and **pooling layer**.

- Convolutional layer uses a mathematical operation called **convolution**, which combines two functions to produce a third function. The convolution operation is carried out via a **filter/kernel** matrix. The size (width-height) of a filter is predetermined. However, the depth of the filter is same with the depth of the image. Each filter looks for different features of the input. Therefore, there can be more than one filter. The filter moves from left to the right until it goes through the complete width, then goes down to the left beginning, and repeats the process until it traverses the entire input. There is a hyperparameter called **stride**, which is the step size of the convolution filter moves each time. Matrix multiplication of filter and input forms the **feature map**. When one does not want that the feature map shrinks too much

or want that the size of the input and output is the same, zero-padding can be applied. Zero-padding surrounds the matrix with zero-values to increase the size of width and height. To summarise, hyperparameters for convolutional layer are number of filters, size of the filters (same size will be applied for all the filters), stride, and the amount of **zero-padding**. ReLU is the most commonly used activation function for convolutional layers. Convolutional layers are responsible for reducing the input size to ease the process, but without losing important features. Learning occurs by changing the entries of filters (weights).

- Pooling layer reduces the size of the convolved feature. The aim is to reduce the number of parameters and computational power required, and to control overfitting. **Max-pooling** is the popular type of pooling. Its size n and stride s are hyperparameters. Max-pooling takes nxn pieces beginning from the left and returns the maximum value of that nxn piece. Then, it moves to the right with a stride s until the end of complete width, and then goes down and repeats the same process. The convolutional layers and pooling layers are one after another.
- After feature extraction part, it is time to classify the image. **Fully Connected Layers** (FCL) are responsible for the classification, and they are simply feed-forward neural networks, i.e., Multi-layer Perceptron. The input of FCL is the flattened output of the last pooling/convolutional layer. **Flattening** means converting 3D data to 1D data. ReLU activation function is used except for last hidden layer. Last hidden layer before output layer has neurons with **softmax** activation function, and the number of neurons of softmax layer should be the same with the output layer. Softmax assigns probabilities to each class. For example, an image might belong to a cat with the probability of 0.95 and might belong to a horse with the probability of 0.05.

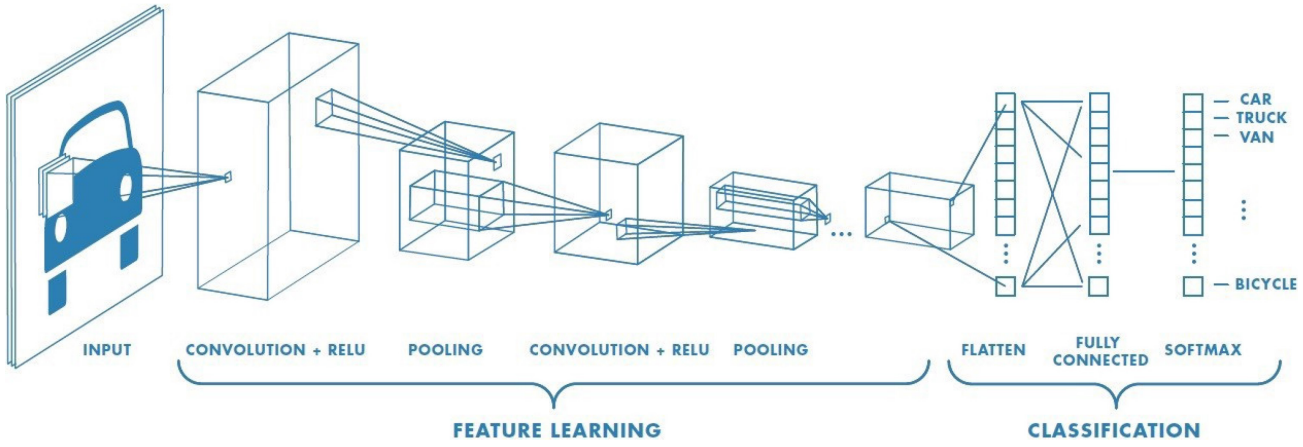


Figure 3-10: CNN Architecture.

3.2.3 State-of-the-Art

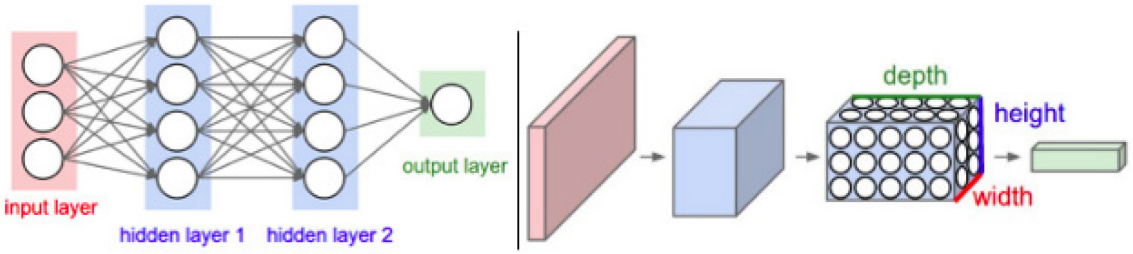
There are various architectures of convolutional networks:

- LeNet [58].
- AlexNet [59].
- ZFNet [60].

- GoogLeNet [61].
- VGGNet [62].
- ResNet [63].
- Temporal CNN [64].

3.2.4 Utility

Regular neural networks do not scale well for images. For example, if we have an image of 200 x 200 x 3 (200 width, 200 height, 3 colour channels), the first hidden layer of a regular neural network would have 120.000 weights. This many weights lead to enormous computational power required and overfitting. Convolutional Neural Networks are the networks specialised to images. Neurons are not in a fully connected manner, but they are connected to a small region of the preceding layer. Moreover, unlike the neurons of regular neural networks, the neurons of CNNs are arranged in 3 dimensions to cope with the 3D nature of image data (Figure 3-11).



Left: A regular 3-layer Neural Network. Right: A ConvNet arranges its neurons in three dimensions (width, height, depth), as visualized in one of the layers. Every layer of a ConvNet transforms the 3D input volume to a 3D output volume of neuron activations. In this example, the red input layer holds the image, so its width and height would be the dimensions of the image, and the depth would be 3 (Red, Green, Blue channels).

Figure 3-11: Comparison of Regular Neural Networks and CNNs.

3.2.5 Practical Implementations

- Image classification [59], [65], [66], [67], [68], [69], [70], [71], [72], [73], [74].
- Visual object detection [75], [76], [77], [78], [79], [80], [81].
- Anomaly Detection [82].
- Time Series Forecasting [83], [84], [85], [86].

3.2.6 Open Challenges

Some researchers question the necessity of pooling layers and claim that pooling layers has wiped away valuable information and the relation between the part and the whole [87]. CNNs are too sensitive to translation, rotation, or scale changes without pooling layers. Laptev et al. proposed TI pooling and showed that it handles rotations and scale changes [88]. Jaderberg et al. also proposed a spatial transformer module which learns translation, scale, rotation, and warping invariance [89].

Because of their computational costs, it is hard to employ them to embedded systems such as mobile device or FPGAs. CNNs should be improved in a way that compromises with limited memory and battery constraints of embedded systems. There are several hardware developments and improved algorithms, but it is still an open challenge [90], [91].

The classification accuracy of CNNs is not robust when they are faced with intentionally perturbated images. While humans are able to classify the altered image correctly, CNNs classify the image as from another class. There are several promising findings [92], [93], [94].

Multi-labelled images and semantic content description is another open challenge for CNNs. To overcome this challenge, the most recent trend is to use Deep CNN-RNN hybrid architectures [95], [96].

Another interesting area of research is how CNNs can be modified to support sequential modelling, through architectural modifications that enable memory. Early research indicate that such architectures exhibit longer memory, than recurrent neural networks (RNN) with the same capacity [97]. This is especially relevant to understand context in e.g., video classification.

3.3 RECURRENT NEURAL NETWORKS (RNN)

3.3.1 History

Recurrent Neural Network is a term for feed-forward neural networks with memory. They are recurrent because the output of the current input depends on the preceding input. In other words, the output of an input will affect the output of following input. The network considers the current input and the last output calculated when making a decision.

The history of Recurrent Neural Networks started with **Hopfield Networks**, which were proposed by Hopfield to model human memory in 1982 [98] (Figure 3-12). The difference from classic neural networks studied until that time was that he used a single-layer and recurrent neural network to learn and recognise patterns. The difference between single-layer perceptron and Hopfield network was that the connections between neurons of Hopfield Networks are symmetrical and directed, and there are no self-connections. This was a break-through because he showed that single-layer neural networks can also learn, but with the basic modification of symmetric connections.

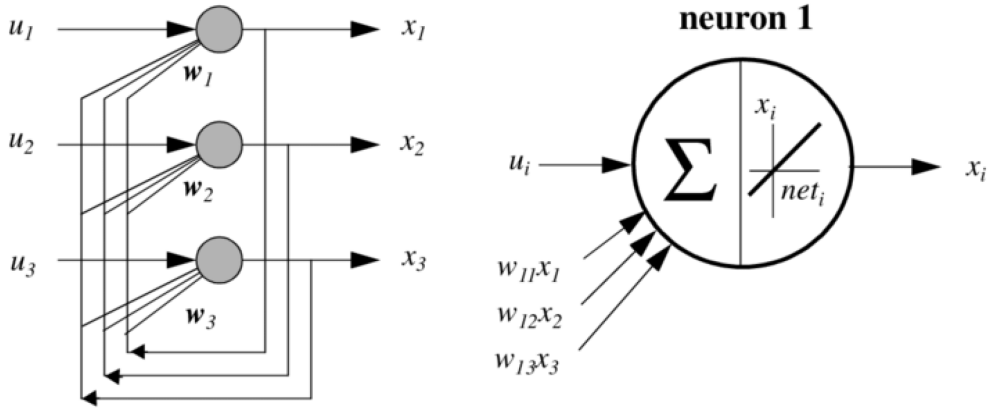


Figure 3-12: Architecture of Hopfield Networks.

In 1985, **Boltzmann Machine** was created (Figure 3-13). The Boltzmann Machine is a stochastic recurrent neural network [99]. It is also known as stochastic Hopfield network with hidden units. There are **visible units** (V), which receive information from the environment, which can be considered as input layer. Only the content of these nodes can be measured and changed. In addition, there are hidden units (H), which represents extracted features from the information, and changes through new inputs to keep the information about past inputs. There is no output layer. All the nodes are connected to each other, and the connections between nodes are also symmetrical but not directed. It makes this model different from others so that there are connections between the nodes of input layer. Furthermore, this is an **unsupervised** model, because the weights of the system are adjusted according to only input data. Until this time, all the models were **supervised**, which means the weights are adjusted according to minimise the difference between actual output and expected output from a given input.

In 1986, the Boltzmann Machine was modified by removing intra-layer connections, and then called **Restricted Boltzmann Machine** [100]. This alteration has led more efficient training algorithms to be applied.

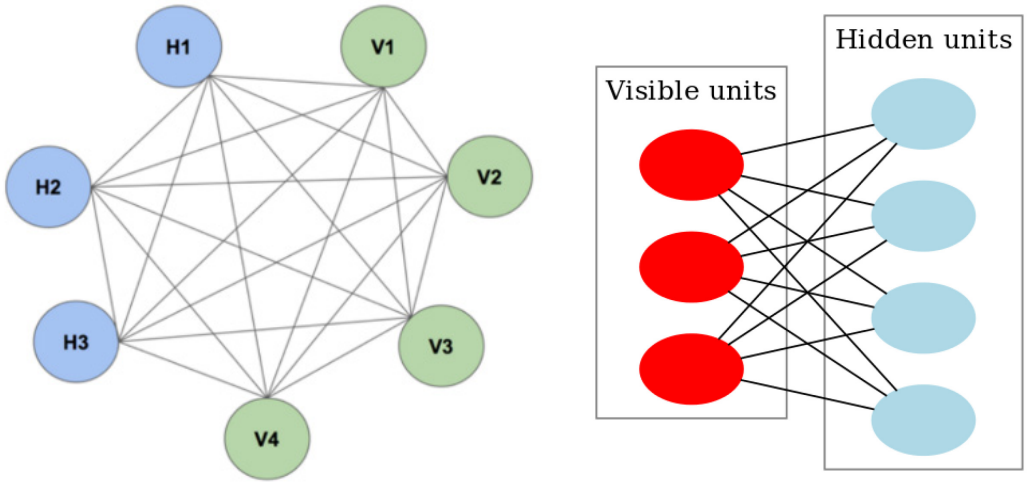


Figure 3-13: Boltzmann Machine (Left) and Restricted Boltzmann Machine (Right).

All the architectures stated above form the foundation of state-of-the-art RNN-architectures.

3.3.2 Traditional/Standard/Vanilla RNN

3.3.2.1 Approach

Recurrent Neural Networks are Multi-Layer Perceptrons with the hidden layers which have incoming activations from both input layer and hidden layer one time step back. In this way, the sequential relations are preserved. An RNN with single hidden layer can also presented as a feed-forward neural network with n (input size) hidden layers, and the hidden layers get both the input data and the output of the preceding hidden layer (Figure 3-14). The most common activation functions are hyperbolic tangent (tanh), sigmoid, and ReLU.

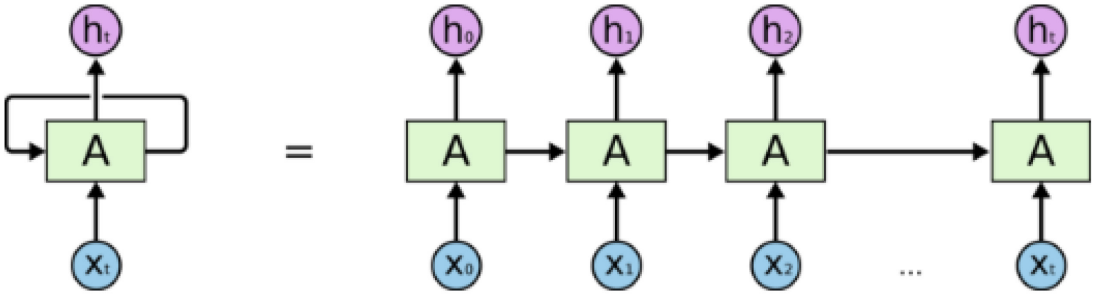


Figure 3-14: Architecture of a Single-Layer RNN.

3.3.2.2 State-of-the-Art

The **Bidirectional RNN** (BRNN) is proposed for getting information from past and future at the same time [101]. It is useful not for predicting the next one, but for predicting current observation using future and past ones. There is an extra hidden layer, which passes the information in a backward direction. The input is fed to forward layer starting from the first token and fed to backward layer starting from the last token.

Deep RNN (DRNN) is the RNN with multiple recurrent hidden layers. Deep RNNs facilitates high-level representation of low-level data beside the sequential prediction. For an example of stock markets, DRNN can predict the future values, besides it can evaluate that the trend is bullish or bearish [102].

3.3.2.3 Utility

RNNs are generally used for sequentially related data. In addition, it can handle large length of data, and the size of the model does not increase with the size of data.

3.3.2.4 Practical Implementations

- Machine translation [103].
- Image caption [104].
- Time series prediction [105].

3.3.2.5 Open Challenges

In prior networks the vanishing gradient problem, has been solved through the use of ReLU activation functions. This solution can however introduce exploding gradients, a phenomenon occurring when the gradient is back-propagated through the network, and it grows exponentially from layer to layer. While the issue can be minimised by scaling weights, the issue is prevalent for deep neural networks e.g., which RNN-architectures often entails. To reduce the problem with both the vanishing and exploding gradient, techniques such as Gated Recurrent Unit (GRU) or Long Short-Term Memory (LSTM) are proposed. These techniques will be addressed in the upcoming sections.

3.3.3 Long Short-Term Memory (LSTM)

3.3.3.1 History

One of the first solutions to the vanishing gradient problem for recurrent neural networks was multi-level hierarchy of networks in 1992 [106]. An attempt was made to reduce the number of considered inputs by finding causal dependencies between the inputs and learning to attend unexpected inputs instead of focusing on every input. This is because an input consistent with the model will not change anything on the model and it will be not efficient to evaluate that input.

In 1997, Long Short-Term Memory (LSTM) was proposed for recurrent neural networks. LSTM concerns learning long-term dependencies by remembering long-term information and further developing the idea of multi-level hierarchy [107]. It was a three-layer network consisting of input, hidden and output layer. In the hidden layer, there are additional memory cells and corresponding gate units. The memory cells facilitate information storage, and the gates protects the memory cell from perturbation by irrelevant inputs, and likewise, protects the other units from perturbation by irrelevant memory contents. With this technique, they guaranteed that learning is succeeded with minimal 1000 inputs even in case of noisy, incompressible input sequences. Then, in 1999, the "forget gate" was introduced, which enabled the cell to reset its state, and puts LSTM into final form [108].

3.3.3.2 Approach

As stated above, when one unrolls RNN, it may be seen that there are consecutive blocks which are actually MLPs. In LSTM, the general architecture is same but the blocks inside are designed differently to overcome the vanishing gradient problem (Figure 3-15). Each block consists of a memory cell and three gates that regularise read, write, and reset operations for the memory cell. The state of memory cell is the key component for the operation of LSTM. Forget gate decides which information from previous time step is not important and should be deleted from the cell state. Input gate determines the significance of current information. Then, tanh layer updates the cell state scaled by importance values given by input gate. Output gate determines which parts of cell state are relevant to current information. Then, output is produced as filtered version of cell state.

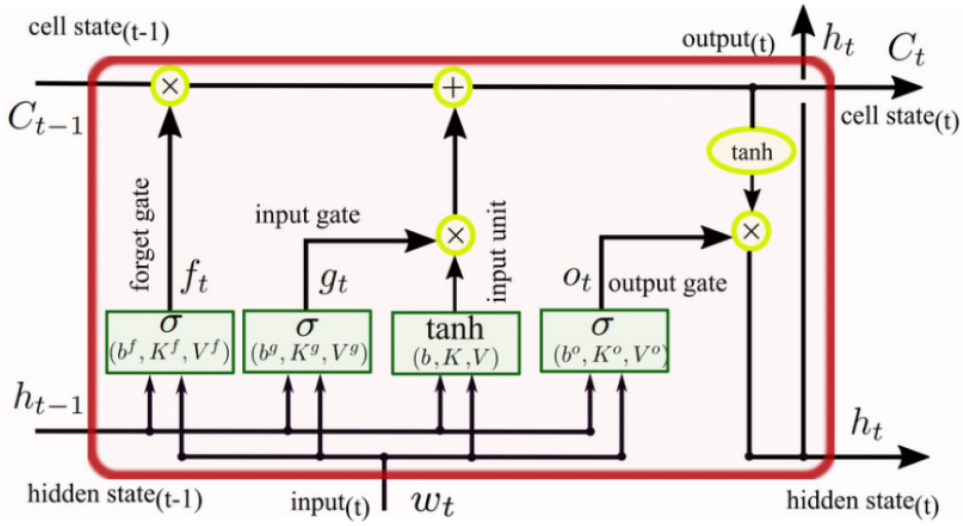


Figure 3-15: Architecture of LSTM Blocks.

3.3.3.3 State-of-the-Art

- **Gated Recurrent Unit (GRU):** GRU is a variation of LSTM, proposed by Cho et al. It merges hidden state and cell state. Only the hidden state will be updated. It combines the forget gate and input gate into a single gate called "update gate". This update gate controls how much information from previous hidden state will be used when updating new hidden state. It adds a "reset gate", which decides to ignore previous hidden state and continue with current input. GRU aims to have all the facilities of LSTM with a simplified architecture [109].

- **Grid LSTM:** The 2D grid LSTM extends the architecture by proposing that the LSTM memory cells. It suggests that the gates should extend to the vertical depth dimension as well as the horizontal temporal direction. It is not limited to 2-dimension and can be extended to n-dimension. In this way, the aim is to handle multi-dimensional data and both deep and sequential computation [110].

- **Bidirectional LSTM:** Bidirectional LSTMs are the BRNNs with LSTM blocks. In this way, the aim is to capture long-term dependencies and using the utilities of bidirectional RNNs [111].

- **Seq-to-seq LSTM/RNN Encoder-Decoder:** Encoder LSTM reads an input sequence to obtain a vector-representation of input and decoder LSTM extracts the output sequence from the vector. It aims to map a fixed-length input to a fixed-length output, but the lengths of input and output does not have to be the same. It has been the architecture lying under Google Translate [112].

3.3.3.4 Utility

Although classic RNNs are theoretically capable of handling long-term dependencies, they are practically not because of vanishing and exploding gradient problem. LSTMs are a specialised version of RNN to deal with this problem. It can remember information for long periods of time.

3.3.3.5 Practical Implementations

- Machine Translation [112].
- Stock Price Prediction [113].
- Speech Recognition [114].
- Video Captioning [115].

3.3.3.6 Open Challenges

Some critics of LSTM have suggested that it has components whose purpose is not immediately apparent. It is not clear that LSTM is an optimal architecture and there may be a better approach. Jozefowicz et al. tested more than ten thousand RNN-architectures and share the results to show that some architectures worked better for some cases [116].

3.4 TRANSFORMER NETWORKS

3.4.1 History

Transformer networks were introduced in 2017 [117] and made strides by highlighting that attention mechanisms, techniques that mimics cognitive attention, alone can match the performance of RNN-architectures such as LSTMs and GRUs. It has since then, increasingly become the architecture of choice for natural language processing (NLP) problems [118]. The innovations brought forth by the Transformer network, can be boiled down to three aspects: positional encodings, attention, and self-attention. All of these aspects enable the Transformer network to perform computations in parallel, and thus result in a more time efficient model. Beyond these innovations, it shares properties with prior networks, e.g., the vanishing gradient is solved by utilising skip connections akin to CNN-based ResNet, in which the gradient can bypass a number of activations functions when traversing the network.

3.4.2 Approach

The old way of understanding input order is by processing the input sequentially, this also made them hard to parallelise. A workaround for this constraint, is to add a number specifying the location of a given input in its sequence. In other words, the input order is encoded into the data structure rather than dealt with through the neural network. We refer to this location as a positional embedding, and the original authors computed the location with sine and cosine functions and sum it with the original input.

This information is further used in its attention mechanisms, in which the model allows the evaluation of every single input in a given sequence, when deciding how to solve a particular task. In other words, every intermediate state in a sequence is utilised and may contain important information that otherwise loose importance over time. In comparison, RNN-architectures tend to only evaluate the final state. The information in these intermediate states can be modelled as a heat map, showing which states the network is attending when it produces a given output. Knowing what to attend, is knowledge obtained from the training data. This is rather straightforward in translation tasks, in which you have an exact answer to which inputs that are important to attend to. However, in many tasks we do not have this exact setup. In such scenarios, self-attention is useful to build a model that understand e.g., the underlying meaning and patterns in the dataset.

3.4.3 State-of-the-Art

We've identified two directions among the state-of-the-art:

- **Generalisation.** There exists a large number of models tailored to particular tasks, but few have attempted to generate a generic model which can be reused. Google and OpenAI have made progress to produce such models, which can be fine-tuned to particular tasks. This effort effectively reduces the data resources needed to obtain a model with reasonable performance, due to extensive pre-training on a large text corpus. This effort has later advanced into making these models more hardware efficient, to make them accessible to devices with less compute resources. Google has launched the "Long-Range Arena" a benchmark to evaluate attempts at efficient Transformers.[5]
- **Variable input sequences.** Most networks dealing with sequential modelling specify a max constraint on the number of inputs allowed in a given sequence. In scenarios in which the sequence is below this constraint, the input is padded with pre-defined values defined to be ignored by the network.

[5] https://github.com/google-research/long-range-arena.

For the opposite, some choose to truncate the sequence. Neither of these solutions are optimal, and research is done to create a Transformer network that will allow variable input sequences. XLNet is a Transformer network which allow such inputs.

3.4.4 Utility

Transformers benefits from its ability to scale to enormous amounts of data, but the performance gain compared to LSTM-inspired models are not significant on a number of benchmarks.[6] Moreover, the networks often require a large number of parameters, which struggle to fit onto a number of devices' memory. Thus, Transformer are preferable if you have access to high capacity hardware, seek shorter time spent on training, and is able to benefit from transfer-learning on a publicly available model.

3.4.5 Practical Implementations

- Machine Translation [119].
- Text auto-completion [120].
- Multivariate forecasting [121].
- Source code analysis [122].

3.4.6 Open Challenges

Creating a Transformer network that can fit onto edge devices and other low-resource devices is still an open challenge, recent research have attempted to tackle it by improving components of the architecture, or by generating multiple models and search for the optimal given certain hardware constraints [123], [124].

3.5 DEEP BELIEF NETWORKS (DBN)

3.5.1 History

The concept of "Belief Network", also called "Bayesian Network", was introduced in 1985 [125]. It is a Probabilistic Graphical Model that represents causal relationships between random variables through a Directed Acyclic Graph. In 200, Hinton et al. proposed a learning method for feed-forward neural networks using Belief Networks [126]. They claimed that this is a fast and greedy learning algorithm even in deep networks with millions of parameters and many hidden layers. They designed a hybrid model which consists of two types of networks. The top two hidden layers of the architecture have undirected connections and forms an associative memory. The remaining hidden layers form a directed acyclic graph (belief network). They also showed the equivalence between infinite belief networks and Restricted Boltzmann Machine. Therefore, the architecture turned into stacked RBMs. Bottom-up directed connections carry inferred representation of previous layer. Top-down directed connections are generative, and they are used to map the associative memory to the data distribution. They stated that this method is unsupervised but can be also applied to labelled data.

[6] https://paperswithcode.com/sota/sentiment-analysis-on-sst-2-binary?tag_filter=8%2C4.

3.5.2 Approach

The architecture can be considered as stacked Restricted Boltzmann Machines (Figure 3-16). The top two layers have undirected connections, whereas all the other layers have directional connections. Learning starts from the bottom (input) layer and move up. It is layer-by-layer basis, which means that layers are trained one at a time. After training of one RBM is finished, then training of another RBM starts with the outputs of preceding RBM. In other words, hidden layer of one RBM is visible layer of subsequent RBM. RBMs are trained in an unsupervised manner. It stops when all RBMs are trained. DBNs are generative models. Alternatively, an auto-associative memory can be trained as last module, which is fed with the output of last RBM, and the labels associated with input. Then, the model can be used for recognition tasks.

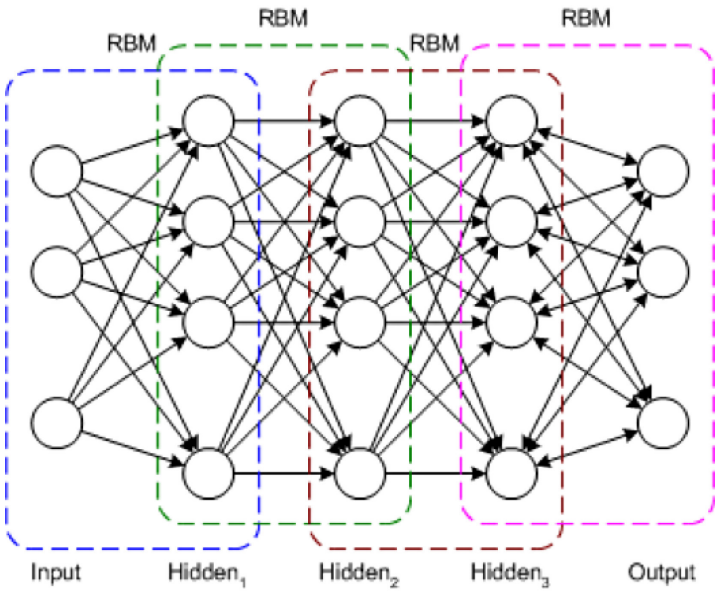


Figure 3-16: Architecture of Deep Belief Network.

3.5.3 State-of-the-Art

The architecture of DBNs has not changed from the proposal until now. Deep Boltzmann Machines are proposed as a variation of DBNs. They allow undirected connections all the layers. Because they carry on the inferences and training in both directions, they are better to reveal the input representation [127].

3.5.4 Utility

The learning process of DBNs is less computationally expensive. The computational cost grows linearly with the number of layers, instead of exponentially as with feed-forward networks. DBNs are also less vulnerable to vanishing gradient problem.

DBNs are generally used for pre-training or dimensionality reduction rather than direct application development. The weights can be used to initialise a deep feed-forward network, which can be fine-tuned with supervised learning using backpropagation then. Moreover, output of the last RBM can be used as the representation of input with reduced dimensionality, and it can be connected to a classifier network [28].

3.5.5 Practical Implementations

- Image Recognition [128], [129].
- Video Recognition [130].
- Motion-capture data [131].

3.5.6 Open Challenges

As the network architecture gets more complex, it has greater ability to solve complex problems. However, some problems will arise as the number of hidden layers get higher. Probably, the training will be harder, the training error accumulates much more, and the correctness of the model degrades. There is no theoretical support to find the right architecture. Therefore, the depth of network and the number of hidden layers need to be set by trial-and-error [132].

3.6 DEEP REINFORCEMENT LEARNING NETWORKS

Beginning around 2013, DeepMind have tried to combine reinforcement learning[7] and deep learning (**Deep Reinforcement Learning**) for the game-playing tasks which requires decision-making using 2D dimensional data as input [133], [134]. In 2016, they reached a milestone to prove the success of deep reinforcement leaning by demonstrating a computer program called AlphaGo to play Go, and this program became the first computer program which defeated a human professional Go player [135]. It used Q-Learning, which is the most known reinforcement learning algorithm. Deep reinforcement learning aims at learning the relationship between state-action pairs and corresponding reward value, predicting the reward from given state-action pair, and choosing the best decision to make (Figure 3-17).

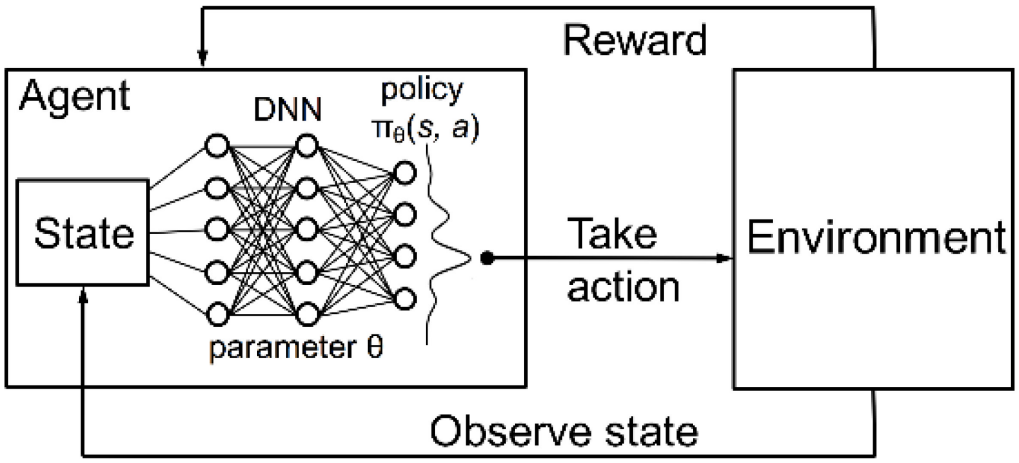


Figure 3-17: The Agent-Environment Interaction Model for Deep Reinforcement Learning.

[7] Reinforcement learning is a technique which the model (agent) learns by evaluating the reaction of the environment after the interaction. The state of the agent changes when the agent takes an action in the environment. Agent also takes feedback from the environment about the result of its action according to how correctly it behaved. This feedback is called "reward". The learning occurs by evaluating current state and reward, and accordingly taking an action to maximize the reward.

3.6.1 Deep Q-Networks (DQN)

3.6.1.1 Approach

In Q-Learning, a Q-value is the expected reward when an action a is taken when the state is s. If agent wants to exploit, it performs the action with highest Q-value for a given state. It is always guaranteed to act with highest Q-values. If agent wants to explore, it chooses an action randomly. It is important for discovering new states which may result in higher Q-values, other than the states which will be evaluated during the exploitation. Agent does not have to always explore or always exploit, and the choice between exploration and exploitation can be made according to a given probability, i.e., Epsilon-Greedy Policy.

There is a table called a Q-Table, which maps state-action pairs to their Q-values. The Q-Table is updated using Bellman Equation after every interaction with the environment (Figure 3-18).

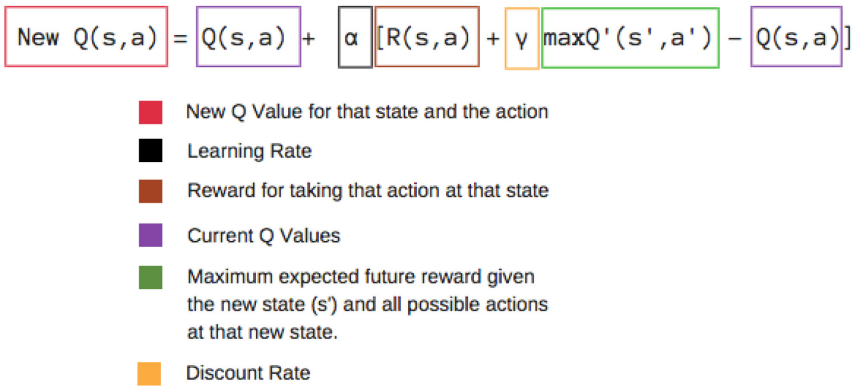


Figure 3-18: Bellman Equation.

Deep Q-Learning is proposed by Mnih et al., for utilising the advances of deep learning to estimate Q-values when the input is high dimensional (Figure 3-19). For a 2D example, the gameplay image is fed into a convolutional neural network, and the network outputs a vector of Q-values for every possible action. An action is chosen according to epsilon-greedy policy (randomly or the action with highest Q-value). Then, action is executed, reward and new state is observed.

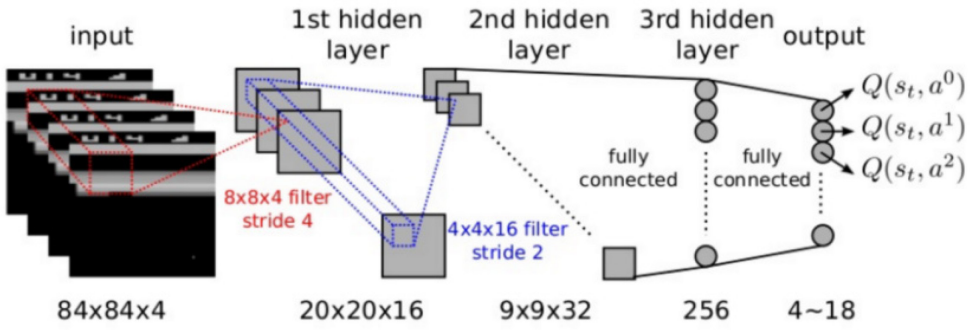


Figure 3-19: Architecture of Deep Q-Networks.

The agent's experiences, i.e., the tuples of state, action, reward, new state, are stored at each time step in a replay memory. Then, a batch of randomly chosen experiences from this replay memory is used for training. This technique is called Experience Replay. In this way, significant weight oscillations and divergence are avoided.

Training is done according to the error between the target value and expected reward calculated by the Q-network (Q-value). If the experience taken by replay memory corresponds to the last observed state, the target value is directly the reward of that experience. Otherwise, the target value is supported with the maximum of the Q-values for new state besides the reward value. At this point, the learning process uses two neural networks: main network and target network. The target network calculates the predicted Q-values for new state. Then, using the outputs of target network, target value is calculated and used for training of main network. The main/online network is updated at each time step, whereas target network is updated with the last version of main network after a number of steps. This leads to more stability in the learning process [134].

3.6.1.2 State-of-the-Art

- **Double DQN:** In DQN, the Q-values for actions of new state is calculated by target network, and the maximum of them is used for training process of main network. Double DQN again uses the notions of main network and target network; but this time, it proposes that the Q-values for actions of new state should be calculated by main network itself, then the target value will be calculated with the support of the target network which calculates Q-value for the action which had maximum Q-value among all the actions of new state. In this way, it aims to reduce overestimation, which is the estimation of Q-values higher than the value it should be [136].
- **Duelling Deep Q-Networks (DDQN):** To understand DDQN, the notions of value function and advantage function should be introduced. Value function calculates the value of being in a particular state independent of the possible actions. Advantage function calculates the value of taking a certain action compared to the others in a particular state. Q-value can be thought of the combination of value function and advantage function. DDQN proposes two separate networks for calculating these two functions sharing same convolutional feature learning module, and then combines them to produce Q-values. By this way, it leads to better policy evaluation in the presence of many similar-valued actions [137].
- **Deep Recurrent Q-Network (DRQN):** It aims to integrate information across frames to detect relevant information. The fully connected layer of DQN is replaced with a recurrent LSTM layer [138].

3.6.1.3 Utility

DQN is found to exhibit learning control policies for high dimensional sensory input. With the help of convolutional layers, high dimensional input is turned into features. Then, the features are used as the state of the agent.

3.6.1.4 Practical Implementations

- Game-playing [133], [135].
- Robot navigation [139].
- Physics based simulation [140].
- Traffic light control [141].

3.6.1.5 Open Challenges

If the action space is discrete and low-dimensional, DQN works well. However, DQN cannot be straightforwardly applied to continuous (real-valued) and high dimensional domains like physics simulations. Lillicrap et al. proposed a technique called DDPG to solve this problem. They stated that their approach finds solutions a factor of 20 fewer steps than DQN requires for good solutions on Atari video games. However, it requires many training episodes to find solutions [142].

3.6.2 Asynchronous Advantage Actor-Critic (A3C)

3.6.2.1 Approach

A3C is based on a different approach called Actor-Critic instead of Q-Learning. In Actor-Critic, value function and policy function are used (Figure 3-20). Value function specifies the value how good a state is in terms of expected future rewards. Policy function is the probabilistic distribution of the action space. In other words, policy function specifies the probability of choosing an action when in a particular state. Actor determines which action to be executed using policy function, whereas Critic criticizes the last state that the agent caused using value function. Actor chooses the action with the highest probability value. After an agent selects an action *a* in state *s*, and the reward and new state is received, Critic compares the new state with the last state to tell the Actor how good the chosen action *a* was. If the value of new state is more than the value of last state, it means that action a was a good choice, and the probability of choosing the action *a* in state *s* should be strengthened. Otherwise, it should be weakened [143].

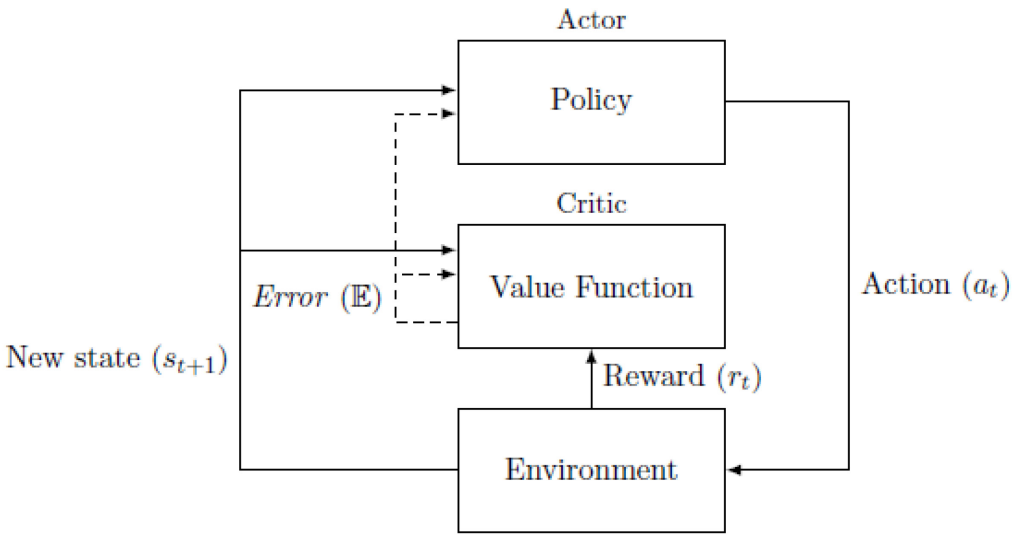


Figure 3-20: Actor-Critic Interaction Model.

In A3C architecture, there are two separate networks for estimating actor (policy) and critic (value) (Figure 3-21). These two networks are trained using advantage function. Advantage function, as used in DDQN, calculates the advantage of taking an action compared to the others in a particular state. But this time, advantage function is not related to Q-values, but it is coming from the comparison of the values of states before and after the action is taken.

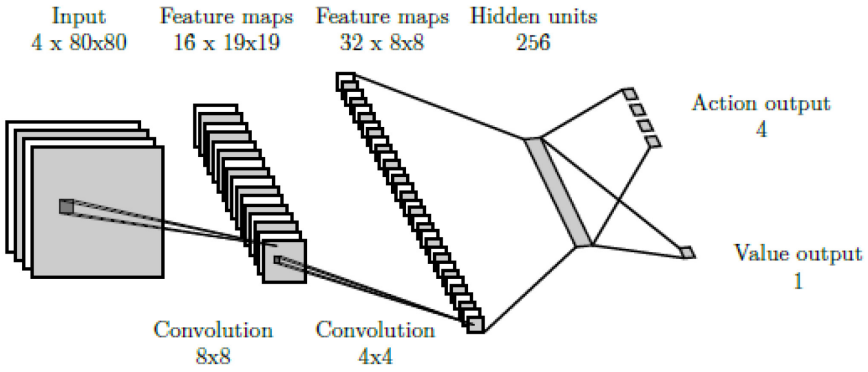


Figure 3-21: A3C Network Architecture.

In the A3C paradigm, multiple agents are asynchronously executed in parallel on multiple instances of the environment, unlike DQN, where a single agent interacts with a single environment (Figure 3-22). Each agent interacts with its own environment and has experiences independent of the others'. There is one global network, and all the worker threads copy the network parameters of the global network to themselves at the beginning. Then, each one starts to interact with its own copy of the environment and collects experience until arriving terminal state or executing predetermined size of steps. Each worker stores their own experiences. Then, each worker updates global network accordingly as it calculates the errors for each experience [144].

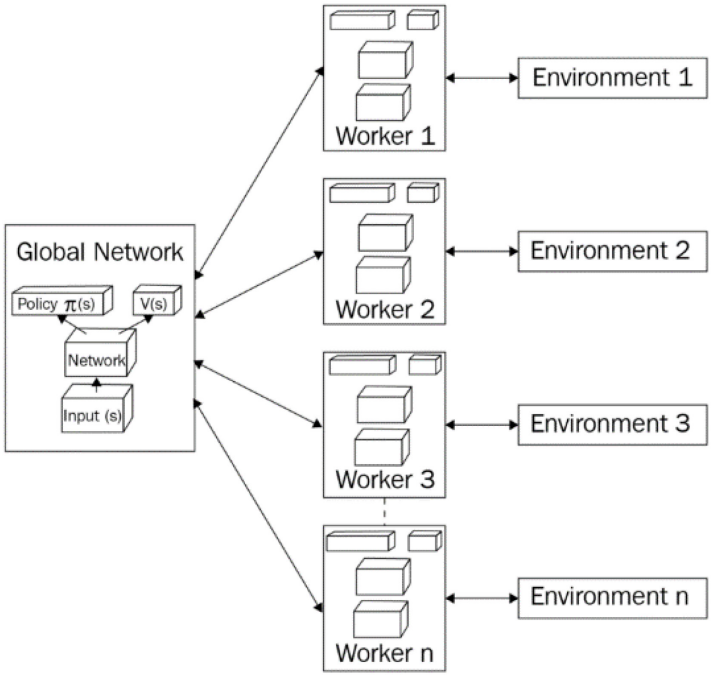


Figure 3-22: Diagram of A3C High-Level Architecture.

3.6.2.2 State-of-the-Art

- **A3C-LSTM:** It is the recurrent version of A3C, so that additional LSTM cells are added after the fully connected networks to extend its capabilities and performance.

3.6.2.3 Utility

It is faster, simpler, more robust, and able to achieve much better scores. In addition, it can deal with continuous domains as well as discrete action spaces.

3.6.2.4 Practical Implementations

- Animal cognition [145].
- Resource allocation management [146].
- Robot navigation [147].

3.6.2.5 Open Challenges

In A3C, policy function is trained by using gradient descent. It especially suffers from the problem that learning rate highly affects training. Small learning rate may cause vanishing gradient problem, whereas large learning rate may cause exploding gradient problem. Some techniques such as Trust Region Policy Optimisation (TRPO) [148], Proximal Policy Optimisation (PPO) [149], and Actor-Critic with Experience Replay (ACER) [150] have been proposed and it is reported that they outperform the A3C especially for continuous domains.

3.7 REFERENCES

[1] McCulloch, W., and Pitts, W. (1943). A Logical Calculus of Ideas Immanent in Nervous Activity. The Bulletin of Mathematical Biophysics, 5(4), 115-133. doi: 10.1007/BF02478259.

[2] Hebb, D.O. (1949). The Organization of Behavior. New York: Wiley & Sons.

[3] Rosenblatt, F. (1958). The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain. Psychological Review, 65(6), 386-408. doi: 10.1037/h0042519.

[4] Ivakhnenko, A.G., and Lapa, V.G. (1965). Cybernetic Predicting Devices. New York: CCM Information Corp.

[5] Ivakhnenko, A.G. (1971). Polynomial Theory of Complex Systems. SMC, 1(4), 364-378.

[6] Linnainmaa, S. (1970). The Representation of the Cumulative Rounding Error of an Algorithm as a Taylor Expansion of the Local Rounding Errors. Masters Thesis, University of Helsinki, Helsinki.

[7] Werbos, P.J. (1982). Applications of Advances in Nonlinear Sensitivity Analysis. In R.F. Drenick, and F. Kozin (Eds.), System Modeling and Optimization, 762-770. Springer. doi: 10.1007/BFb0006203.

[8] Hochreiter, S. (1991). Untersuchungen zu dynamischen neuronalen Netzen. Diploma Thesis, Technical University of Munich, Institute of Computer Science.

[9] Glorot, X., Bordes, A., and Bengio, Y. (2010). Deep Sparse Rectifier Neural Networks. Journal of Machine Learning Research, 15.

[10] Weigend, A., and Gershenfeld, N. (1994). Time Series Prediction: Forecasting the Future and Understanding the Past. Addison-Wesley.

[11] Basu, S., and Mukherjee, A. (1999). Time Series Models for Internet Traffic. Proceedings of 24th Conference on Local Computer Networks, 164-171.

[12] Box, G., and Jenkins, G. (1976). Time Series Analysis: Forecasting and Control. San Francisco: Holden-Day.

[13] Abu Bakar, N.M., and Mohd Tahir, I. (2009). Applying Multiple Linear Regression and Neural Network to Predict Bank Performance. International Business Research, 2(4). doi: 10.5539/ibr.v2n4p176.

[14] Alegado, R., and Tumibay, G. (2020). Statistical and Machine Learning Methods for Vaccine Demand Forecasting: A Comparative Analysis. Journal of Computer and Communications, 8, 37-49. doi: 10.4236/jcc.2020.810005.

[15] Zhang, Z., Lyons, M., Schuster, M., and Akamatsu, S. (1998). Comparison between Geometry-based and Gabor-Wavelets-Based Facial Expression Recognition Using Multi-layer Perceptron. Proceedings of 3rd IEEE International Conference on Automatic Face and Gesture Recognition. doi: 10.1109/ AFGR.1998.670990.

[16] Danisman, T., Bilasco, I.M., Martinet, J., and Djeraba, C. (2013). Intelligent Pixels of Interest Selection with Application to Facial Expression Recognition Using Multilayer Perceptron. Signal Processing, 93, 1547-1556. doi: 10.1016/j.sigpro.2012.08.007.

[17] Chittem, L.K., and Reddy, P.V. (2019). An Efficient Deep Neural Network Multilayer Perceptron Based Classifier in Healthcare System. 3rd International Conference on Computing and Communication Technologies (ICCCT'19). doi: 10.1109/ICCCT2.2019.8824913.

[18] Moreira, M., Rodrigues, J.J., Kumar, N., Niu, J., and Sangaiah, A.K. (2018). Multilayer Perceptron Application for Diabetes Mellitus Prediction in Pregnancy Care. International Conference on Frontier Computing, 200-209. doi: 10.1007/978-981-10-7398-4_22.

[19] Ciayandi, A., Mawardi, V., and Hendryli, J. (2020). Retrieval Based Chatbot on Tarumanagara University with Multilayer Perceptron. IOP Conference Series: Materials Science and Engineering.

[20] Jotheeswaran, J., and Seerangan, K. (2015). Decision Tree Based Feature Selection and Multilayer Perceptron for Sentiment Analysis. ARPN Journal of Engineering and Applied Sciences, 5883-5894.

[21] Sejnowski, T., and Rosenberg, C. (1987). Parallel Networks that Learn to Pronounce English Text. Complex Systems, 1(1), 145-168.

[22] Yadav, M., Verma, V.K., Yadav, C.S., and Verma, J.K. (2020). MLPGI: Multilayer Perceptron-Based Gender Identification over Voice Samples in Supervised Machine Learning. In Applications of Machine Learning, 353-364, Springer, Singapore.

[23] An, Q., Bai, K., Zhang, M., Yi, Y. and Liu, Y. (2020). Deep Neural Network Based Speech Recognition Systems Under Noise Perturbations. 21st International Symposium on Quality Electronic Design (ISQED), 377-382. doi: 10.1109/ISQED48828.2020.9136978.

[24] Al-Ahmadi, S., and Lasloum, T. (2020). PDMLP: Phishing Detection using Multilayer Perceptron. International Journal of Network Security & Its Applications, 12(3). doi: 10.5121/ijnsa.2020.12304.

[25] Crow, D., Graham, S., Borghetti, B., and Sweeney, P. (2020) Engaging Empirical Dynamic Modeling to Detect Intrusions in Cyber-Physical Systems. In J. Staggs and S. Shenoi (Eds.) Critical Infrastructure Protection XIV. ICCIP 2020. IFIP Advances in Information and Communication Technology, vol 596. Springer, Cham. doi: 10.1007/978-3-030-62840-6_6.

[26] Kramer, M.A. (1992). Autoassociative Neural Networks. Computers and Chemical Engineering. doi: 10.1016/0098-1354(92)80051-A

[27] Goodfellow, I., Bengio, Y., and Courville, A. (2016). Deep Learning. MIT Press. ISBN 978-0262035613.

[28] Hinton, G.E., and Salakhutdinov, R.R. (2006). Reducing the Dimensionality of Data with Neural Networks. Science 313(5786), 504-507. doi: 10.1126/science.1127647.

[29] Wang, W., Huang, Y., Wang, Y., and Wang, L. (2014). Generalized Autoencoder: A Neural Network Framework for Dimensionality Reduction. IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops. doi: 10.1109/CVPRW.2014.79.

[30] Thomas, S.A., Race, A.M., Steven, R.T., Gilmore, I.S., and Bunch, J. (2016). Dimensionality Reduction of Mass Spectrometry Imaging Data Using Autoencoders. 2016 IEEE Symposium Series on Computational Intelligence, SSCI 2016. doi: 10.1109/SSCI.2016.7849863.

[31] Zabalza, J., Ren, J., Zheng, J., Zhao, H., Qing, C., Yang, Z., Du, P., and Marshall, S. (2016). Novel Segmented Stacked Autoencoder for Effective Dimensionality Reduction and Feature Extraction in Hyperspectral Imaging. Neurocomputing. doi: 10.1016/j.neucom.2015.11.044.

[32] Salakhutdinov, R., and Hinton, G. (2009). Semantic Hashing. International Journal of Approximate Reasoning, 50(7), 969-978. doi: 10.1016/j.ijar.2008.11.006.

[33] Torralba, A., Fergus, R., and Freeman, W.T. (2008). 80 Million Tiny Images: A Large Data Set for Nonparametric Object and Scene Recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence. doi: 10.1109/TPAMI.2008.128.

[34] Hinton, G., Krizhevsky, A., and Wang, S. (2011). Transforming Auto-Encoders. International Conference on Artificial Neural Networks, 44-51.

[35] Liou, C. Y., Cheng, W. C., Liou, J. W., and Liou, D. R. (2014). Autoencoder for Words. Neurocomputing. doi: 10.1016/j.neucom.2013.09.055.

[36] Sakurada, M., and Yairi, T. (2014). Anomaly Detection Using Autoencoders with Nonlinear Dimensionality Reduction. ACM International Conference Proceeding Series. doi: 10.1145/2689746.2689747.

[37] An, J., and Cho, S. (2015). Variational Autoencoder Based Anomaly Detection Using Reconstruction Probability. Special Lecture on IE.

[38] Zhou, C., and Paffenroth, R.C. (2017). Anomaly Detection with Robust Deep Autoencoders. Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. doi: 10.1145/3097983.3098052.

[39] Theis, L., Shi, W., Cunningham, A., and Huszár, F. (2017). Lossy Image Compression with Compressive Autoencoders. 5th International Conference on Learning Representations, ICLR 2017 – Conference Track Proceedings.

[40] Cho, K.H. (2013). Boltzmann Machines and Denoising Autoencoders for Image Denoising. 1st International Conference on Learning Representations, ICLR 2013 – Workshop Track Proceedings.

[41] Xu, J., Xiang, L., Liu, Q., Gilmore, H., Wu, J., Tang, J., and Madabhushi, A. (2016). Stacked Sparse Autoencoder (SSAE) for Nuclei Detection on Breast Cancer Histopathology Images. IEEE Transactions on Medical Imaging. doi: 10.1109/TMI.2015.2458702.

[42] Zeng, K., Yu, J., Wang, R., Li, C., and Tao, D. (2015). Coupled Deep Autoencoder for Single Image Super-Resolution. IEEE Transactions on Cybernetics. doi: 10.1109/TCYB.2015.2501373.

[43] Goodfellow, I.J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative Adversarial Networks. arXiv preprint arXiv:1406.2661.

[44] Mirza, M., and Osindero, S. (2014). Conditional Generative Adversarial Nets. arXiv preprint arXiv:1411.1784.

[45] Karras, T., Laine, S., and Aila, T. (2019). A Style-Based Generator Architecture for Generative Adversarial Networks. Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition. doi: 10.1109/CVPR.2019.00453.

[46] Zhu, J.Y., Park, T., Isola, P., and Efros, A.A. (2017). Unpaired Image-to-Image Translation Using Cycle-Consistent Adversarial Networks. Proceedings of the IEEE International Conference on Computer Vision. doi: 10.1109/ICCV.2017.244.

[47] Radford, A., Metz, L., and Chintala, S. (2016). Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks. 4th International Conference on Learning Representations, ICLR 2016 – Conference Track Proceedings.

[48] Pascual, S., Bonafonte, A., and Serra, J. (2017). SEGAN: Speech Enhancement Generative Adversarial Network. Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH. doi: 10.21437/Interspeech.2017-1428.

[49] Donahue, J., Darrell, T., and Krähenbühl, P. (2017). Adversarial Feature Learning. 5th International Conference on Learning Representations, ICLR 2017 – Conference Track Proceedings.

[50] Isola, P., Zhu, J.Y., Zhou, T., and Efros, A.A. (2017). Image-to-Image Translation with Conditional Adversarial Networks. Proceedings of 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017. doi: 10.1109/CVPR.2017.632.

[51] Zhang, H., Xu, T., Li, H., Zhang, S., Wang, X., Huang, X., and Metaxas, D. (2017). StackGAN: Text to Photo-Realistic Image Synthesis with Stacked Generative Adversarial Networks. Proceedings of the IEEE International Conference on Computer Vision. doi: 10.1109/ICCV.2017.629.

[52] Ledig, C., Theis, L., Huszár, F., Caballero, J., Cunningham, A., Acosta, A., Aitken, A., Tejani, A., Totz, J., Wang, Z., and Shi, W. (2017). Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network. Proceedings of 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017. doi: 10.1109/CVPR.2017.19.

[53] Yang, S., Xie, L., Chen, X., Lou, X., Zhu, X., Huang, D., and Li, H. (2018). Statistical Parametric Speech Synthesis Using Generative Adversarial Networks Under a Multi-Task Learning Framework. 2017 IEEE Automatic Speech Recognition and Understanding Workshop, ASRU 2017 – Proceedings. doi: 10.1109/ASRU.2017.8269003.

[54] Arjovsky, M., Chintala, S., and Bottou, L. (2017). Wasserstein Generative Adversarial Networks. 34th International Conference on Machine Learning, ICML 2017.

[55] Metz, L., Sohl-Dickstein, J., Poole, B., and Pfau, D. (2017). Unrolled Generative Adversarial Networks. 5th International Conference on Learning Representations, ICLR 2017 – Conference Track Proceedings.

[56] Minsky, M., and Papert, S. (1969). Perceptrons. M.I.T. Press

[57] Fukushima, K. (1979). Neural Network Model for a Mechanism of Pattern Recognition Unaffected by Shift in Position-Neocognitron. IEICE Technical Report, A, 62(10), 658-665.

[58] LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-Based Learning Applied to Document Recognition. Proceedings of the IEEE, 86(11). doi: 10.1109/5.726791.

[59] Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). ImageNet Classification with Deep Convolutional Neural Networks. Advances in Neural Information Processing Systems. doi: 10.1061/(ASCE)GT.1943-5606.0001284.

[60] Zeiler M.D., and Fergus R. (2014) Visualizing and Understanding Convolutional Networks. In D. Fleet, T. Pajdla, B. Schiele and T. Tuytelaars (Eds.) Computer Vision – ECCV 2014. Springer, Cham. doi: 10.1007/978-3-319-10590-1_53.

[61] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A. (2015). Going Deeper with Convolutions. Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition. doi: 10.1109/CVPR.2015.7298594.

[62] Simonyan, K., and Zisserman, A. (2015). Very Deep Convolutional Networks for Large-Scale Image Recognition. 3rd International Conference on Learning Representations, ICLR 2015 – Conference Track Proceedings.

[63] He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep Residual Learning for Image Recognition. Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition. doi: 10.1109/CVPR.2016.90.

[64] Bai, S., Kolter, J.Z., and Koltun, V. (2018). An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling. arXiv preprint arXiv:1803.01271.

[65] Lowe, D.G. (2004). Distinctive Image Features from Scale-Invariant Keypoints. International Journal of Computer Vision. doi: 10.1023/B:VISI.0000029664.99615.94.

[66] Oliva, A., and Torralba, A. (2006). Building the Gist of a Scene: The Role of Global Image Features in Recognition. In Progress in Brain Research 155, 23-26. doi: 10.1016/S0079-6123(06)55002-2

[67] Ahonen, T., Hadid, A., and Pietikäinen, M. (May 2004). Face Recognition with Local Binary Patterns. In European Conference on Computer Vision, 469-481. Springer, Berlin, Heidelberg.

[68] Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A.C., and Fei-Fei, L. (2015). ImageNet Large Scale Visual Recognition Challenge. International Journal of Computer Vision, 115, 211-252. doi: 10.1007/s11263-015-0816-y.

[69] Lin, T.Y., Roychowdhury, A., and Maji, S. (2015). Bilinear CNN Models for Fine-Grained Visual Recognition. Proceedings of the IEEE International Conference on Computer Vision. doi: 10.1109/ICCV. 2015.170.

[70] Branson, S., Van Horn, G., Belongie, S., and Perona, P. (2014). Bird Species Categorization Using Pose Normalized Deep Convolutional Nets. BMVC 2014 – Proceedings of the British Machine Vision Conference 2014.

[71] Kagaya, H., Aizawa, K., and Ogawa, M. (2014). Food Detection and Recognition Using Convolutional Neural Network. MM 2014 – Proceedings of the 2014 ACM Conference on Multimedia. doi: 10.1145/2647868.2654970.

[72] Yang, L., Luo, P., Loy, C.C., and Tang, X. (2015). A Large-Scale Car Dataset for Fine-Grained Categorization and Verification. Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition. doi: 10.1109/CVPR.2015.7299023.

[73] Cui, Y., Zhou, F., Lin, Y., and Belongie, S. (2016). Fine-Grained Categorization and Dataset Bootstrapping Using Deep Metric Learning with Humans in the Loop. Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition. doi: 10.1109/CVPR.2016.130.

[74] Zhou, F., and Lin, Y. (2016). Fine-Grained Image Classification by Exploring Bipartite-Graph Labels. Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition. doi: 10.1109/CVPR.2016.127.

[75] Gundogdu E., Solmaz B., Yücesoy V., and Koç A. (2017) MARVEL: A Large-Scale Image Dataset for Maritime Vessels. In S.H. Lai, V. Lepetit, K. Nishino and Y. Sato (Eds.) Computer Vision – ACCV 2016. Springer, Cham. doi: 10.1007/978-3-319-54193-8_11

[76] Solmaz, B., Gundogdu, E., Yucesoy, V., Koç, A., and Alatan, A.A. (2018). Fine-Grained Recognition of Maritime Vessels and Land Vehicles by Deep Feature Embedding. IET Computer Vision. doi: 10.1049/iet-cvi.2018.5187.

[77] Solmaz, B., Gundogdu, E., Yucesoy, V., and Koc, A. (2017). Generic and Attribute-Specific Deep Representations for Maritime Vessels. IPSJ Transactions on Computer Vision and Applications, 9(22). doi: 10.1186/s41074-017-0033-4.

[78] Dai, J., Qi, H., Xiong, Y., Li, Y., Zhang, G., Hu, H., and Wei, Y. (2017). Deformable Convolutional Networks. Proceedings of the IEEE International Conference on Computer Vision. doi: 10.1109/ICCV.2017.89.

[79] Redmon, J., and Farhadi, A. (2018). Yolov3: An Incremental Improvement. arXiv preprint arXiv:1804.02767.

[80] Girshick, R., Donahue, J., Darrell, T., and Malik, J. (2014). Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition. doi: 10.1109/CVPR.2014.81.

[81] Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.Y., and Berg, A.C. (2016). SSD: Single Shot Multibox Detector. In B. Leibe, J. Matas, N. Sebe, and M. Welling (Eds.) Computer Vision – ECCV 2016. doi: 10.1007/978-3-319-46448-0_2.

[82] Ren, H., Xu, B., Wang, Y., Yi, C., Huang, C., Kou, X., Xing, T., Yang, M., Tong, J., and Zhang, Q. (2019). Time-Series Anomaly Detection Service at Microsoft. Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. doi: 10.1145/3292500.3330680.

[83] Bai, S., Kolter, J.Z., and Koltun, V. (2018). An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling. arXiv preprint arXiv:1803.01271.

[84] Tsantekidis, A., Passalis, N., Tefas, A., Kanniainen, J., Gabbouj, M., and Iosifidis, A. (2017). Forecasting Stock Prices from the Limit Order Book Using Convolutional Neural Networks. Proceedings of 2017 IEEE 19th Conference on Business Informatics, CBI 2017. doi: 10.1109/CBI.2017.23.

[85] Borovykh, A., Bohte, S., and Oosterlee, C.W. (2017). Conditional Time Series Forecasting with Convolutional Neural Networks. arXiv preprint arXiv:1703.04691.

[86] Mittelman, R. (2015). Time-Series Modeling with Undecimated Fully Convolutional Neural Networks. arXiv preprint arXiv:1508.00317.

[87] Springenberg, J.T., Dosovitskiy, A., Brox, T., and Riedmiller, M. (2015). Striving for Simplicity: The All Convolutional Net. 3rd International Conference on Learning Representations, ICLR 2015 – Workshop Track Proceedings.

[88] Laptev, D., Savinov, N., Buhmann, J.M., and Pollefeys, M. (2016). TI-POOLING: Transformation-Invariant Pooling for Feature Learning in Convolutional Neural Networks. Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition. doi: 10.1109/CVPR.2016.38.

[89] Jaderberg, M., Simonyan, K., Zisserman, A., and Kavukcuoglu, K. (2015). Spatial Transformer Networks. Advances in Neural Information Processing Systems 28 (NIPS 2015).

[90] Dettmers, T. (2016). 8-bit Approximations for Parallelism in Deep Learning. 4th International Conference on Learning Representations, ICLR 2016 – Conference Track Proceedings.

[91] Kim, Y.D., Park, E., Yoo, S., Choi, T., Yang, L., and Shin, D. (2016). Compression of Deep Convolutional Neural Networks for Fast and Low Power Mobile Applications. 4th International Conference on Learning Representations, ICLR 2016 – Conference Track Proceedings.

[92] Zhao, Q., and Griffin, L.D. (2016). Suppressing the Unusual: Towards Robust CNNS Using Symmetric Activation Functions. arXiv preprint arXiv:1603.05145.

[93] Maharaj, A.V. (2015). Improving the Adversarial Robustness of ConvNets by Reduction of Input Dimensionality. Stanford, CA: Department of Physics, Stanford University.

[94] Bastani, O., Ioannou, Y., Lampropoulos, L., Vytiniotis, D., Nori, A.V., and Criminisi, A. (2016). Measuring Neural Net Robustness with Constraints. 30th Conference on Neural Information Processing Systems (NIPS 2016), Barcelona, Spain.

[95] Wang, J., Yang, Y., Mao, J., Huang, Z., Huang, C., and Xu, W. (2016). CNN-RNN: A Unified Framework for Multi-Label Image Classification. Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition. doi: 10.1109/CVPR.2016.251.

[96] Vinyals, O., Toshev, A., Bengio, S., and Erhan, D. (2015). Show and Tell: A Neural Image Caption Generator. Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition. doi: 10.1109/CVPR.2015.7298935.

[97] Bai, S., Kolter, J. Z., and Koltun, V. (2018). An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling. arXiv preprint arXiv:1803.01271.

[98] Hopfield, J.J. (1982). Neural Networks and Physical Systems with Emergent Collective Computational Abilities. Proceedings of the National Academy of Sciences of the United States of America. doi: 10.1073/pnas.79.8.2554.

[99] Ackley, D.H., Hinton, G.E., and Sejnowski, T.J. (1985). A Learning Algorithm for Boltzmann Machines. Cognitive Science 9, 147-169. doi: 10.1016/S0364-0213(85)80012-4.

[100] Smolensky, P. (1986). Information Processing in Dynamical systems: Foundations of Harmony Theory. Parallel Distributed Processing: Volume 1: Foundations. Colorado Univ at Boulder Dept of Computer Science.

[101] Schuster, M., and Paliwal, K.K. (1997). Bidirectional Recurrent Neural Networks. IEEE Transactions on Signal Processing. doi: 10.1109/78.650093.

[102] Pascanu, R., Gulcehre, C., Cho, K., and Bengio, Y. (2014). How to Construct Deep Recurrent Neural Networks. 2nd International Conference on Learning Representations, ICLR 2014 – Conference Track Proceedings.

[103] Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y. (2014). Learning Phrase Representations Using RNN Encoder-Decoder for Statistical Machine Translation. EMNLP 2014 – 2014 Conference on Empirical Methods in Natural Language Processing. doi: 10.3115/v1/d14-1179.

[104] Karpathy, A., and Li, F.F. (2015). Deep Visual-Semantic Alignments for Generating Image Descriptions. Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition. doi: 10.1109/CVPR.2015.7298932.

[105] Prasad, S.C., and Prasad, P. (2014). Deep Recurrent Neural Networks for Time Series Prediction. arXiv preprint arXiv:1407.5949.

[106] Schmidhuber, J. (1992). Learning Complex, Extended Sequences Using the Principle of History Compression. Neural Computation 4(2), 234-232. doi: 10.1162/neco.1992.4.2.234.

[107] Hochreiter, S., and Schmidhuber, J. (1997). Long Short-Term Memory. Neural Computation 9(8), 1735-1780. doi: 10.1162/neco.1997.9.8.1735.

[108] Gers, F.A., Schmidhuber, J., and Cummins, F. (1999). Learning to Forget: Continual Prediction with LSTM. IEE Conference Publication. doi: 10.1049/cp:19991218.

[109] Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y. (2014). Learning Phrase Representations Using RNN Encoder-Decoder for Statistical Machine Translation. EMNLP 2014 – 2014 Conference on Empirical Methods in Natural Language Processing. doi: 10.3115/v1/d14-1179.

[110] Kalchbrenner, N., Danihelka, I., and Graves, A. (2016). Grid Long Short-Term Memory. 4th International Conference on Learning Representations, ICLR 2016.

[111] Graves, A., and Schmidhuber, J. (2005). Framewise Phoneme Classification with Bidirectional LSTM Networks. Proceedings of the International Joint Conference on Neural Networks. doi: 10.1109/IJCNN.2005.1556215.

[112] Sutskever, I., Vinyals, O., and Le, Q.V. (2014). Sequence to Sequence Learning with Neural Networks. Advances in Neural Information Processing Systems 27 (NIPS 2014).

[113] Selvin, S., Vinayakumar, R., Gopalakrishnan, E.A., Menon, V.K., and Soman, K.P. (2017). Stock Price Prediction Using LSTM, RNN and CNN-Sliding Window Model. 2017 International Conference on Advances in Computing, Communications, and Informatics, ICACCI 2017. doi: 10.1109/ICACCI.2017.8126078.

[114] Prabhavalkar, R., Rao, K., Sainath, T.N., Li, B., Johnson, L., and Jaitly, N. (2017). A Comparison of Sequence-to-Sequence Models for Speech Recognition. Proceedings of the Annual Conference of the International Speech Communication Association, Interspeech. doi: 10.21437/Interspeech.2017-233.

[115] Venugopalan, S., Rohrbach, M., Donahue, J., Mooney, R., Darrell, T., and Saenko, K. (2015). Sequence to Sequence – Video to Text. Proceedings of the IEEE International Conference on Computer Vision. doi: 10.1109/ICCV.2015.515.

[116] Jozefowicz, R., Zaremba, W., and Sutskever, I. (2015). An Empirical Exploration of Recurrent Network Architectures. 32nd International Conference on Machine Learning, ICML 2015.

[117] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., .et al. (2017). Attention is All You Need. In Advances in Neural Information Processing Systems, 5998-6008.

[118] Wolf, T., Chaumond, J., Debut, L., Sanh, V., Delangue, C., Moi, A. et al. (Oct 2020). Transformers: State-of-the-art natural language processing. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, 38-45.

[119] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N. et al. (2017). Attention Is All You Need. In Advances in Neural Information Processing Systems, 5998-6008. Advances in Neural Information Processing Systems, 30.

[120] HuggingFace, Write With Transformer. Accessed on 28 Dec 2021. arXiv preprint arXiv:1910.03771.

[121] Grigsby, J., Wang, Z., and Qi, Y. (2021). Long-Range Transformers for Dynamic Spatiotemporal Forecasting. arXiv preprint arXiv:2109.12218.

[122] Rahali, A., and Akhloufi, M.A. (2021). MalBERT: Using Transformers for Cybersecurity and Malicious Software Detection. arXiv preprint arXiv:2103.03806.

[123] Wang, H., Wu, Z., Liu, Z., Cai, H., Zhu, L., Gan, C., and Han, S. (2020). Hat: Hardware-Aware Transformers for Efficient Natural Language Processing. arXiv preprint arXiv:2005.14187.

[124] Kitaev, N., Kaiser, Ł., and Levskaya, A. (2020). Reformer: The Efficient Transformer. arXiv preprint arXiv:2001.04451.

[125] Pearl, J. (1985). Bayesian Networks A Model of Self-Activated Memory for Evidential Reasoning. In Proceedings of the 7th Conference of the Cognitive Science Society.

[126] Hinton, G.E., Osindero, S., and Teh, Y.W. (2006). A Fast Learning Algorithm for Deep Belief Nets. Neural Computation 18, 1527-1554. doi: 10.1162/neco.2006.18.7.1527.

[127] Salakhutdinov, R., and Hinton, G. (2009). Deep Boltzmann Machines. Journal of Machine Learning Research 5(2):448-455.

[128] Ranzato, M., Huang, F.J., Boureau, Y.L., and LeCun, Y. (2007). Unsupervised Learning of Invariant Feature Hierarchies with Applications to Object Recognition. Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition. doi: 10.1109/CVPR.2007.383157.

[129] Bengio, Y., Lamblin, P., Popovici, D., and Larochelle, H. (2007). Greedy Layer-Wise Training of Deep Networks. Advances in Neural Information Processing Systems 19. doi: 10.7551/mitpress/7503.003.0024.

[130] Sutskever, I., and Hinton, G. (2007). Learning Multilevel Distributed Representations for High-Dimensional Sequences. Journal of Machine Learning Research 2.

[131] Taylor, G.W., Hinton, G.E., and Roweis, S. (2007). Modeling Human Motion Using Binary Latent Variables. Advances in Neural Information Processing Systems 19 (NIPS 2006). doi: 10.7551/mitpress/7503.003.0173.

[132] Guangyuan, P., Wei, C., and Junfei, Q. (2015). Depth Determination Method of DBN Network. Journal of Control and Decision, 30(2), 256-260.

[133] Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., and Riedmiller, M. (2013). Playing Atari with Deep Reinforcement Learning. arXiv preprint arXiv:1312.5602.

[134] Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A.A., Veness, J., Bellemare, M.G., Graves, A., Riedmiller, M., Fidjeland, A.K., Ostrovski, G., Petersen, S., Beattie, C., Sadik, A., Antonoglou, I., King, H., Kumaran, D., Wierstra, D., Legg, S., and Hassabis, D. (2015). Human-Level Control through Deep Reinforcement Learning. Nature 518, 529-533. doi: 10.1038/nature14236.

[135] Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., Van Den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., Dieleman, S., Grewe, D., Nham, J., Kalchbrenner, N., Sutskever, I., Lillicrap, T., Leach, M., Kavukcuoglu, K., Graepel, T., and Hassabis, D. (2016). Mastering the Game of Go with Deep Neural Networks and Tree Search. Nature 529, 484-489. doi: 10.1038/nature16961.

[136] Van Hasselt, H., Guez, A., and Silver, D. (2016). Deep Reinforcement Learning with Double Q-Learning. 30th AAAI Conference on Artificial Intelligence, AAAI 2016.

[137] Wang, Z., Schaul, T., Hessel, M., Van Hasselt, H., Lanctot, M., and De Frcitas, N. (2016). Dueling Network Architectures for Deep Reinforcement Learning. 33rd International Conference on Machine Learning, ICML 2016.

[138] Hausknecht, M., and Stone, P. (2015). Deep Recurrent Q-Learning for Partially Observable MDPs. AAAI Fall Symposium – Technical Report.

[139] Zhang, M., McCarthy, Z., Finn, C., Levine, S., and Abbeel, P. (2016). Learning Deep Neural Network Policies with Continuous Memory States. IEEE International Conference on Robotics and Automation. doi: 10.1109/ICRA.2016.7487174.

[140] Wolf, P., Hubschneider, C., Weber, M., Bauer, A., Hartl, J., Durr, F., and Zollner, J.M. (2017). Learning How to Drive in a Real World Simulation With Deep Q-Networks. IEEE Intelligent Vehicles Symposium, Proceedings. doi: 10.1109/IVS.2017.7995727.

[141] Arel, I., Liu, C., Urbanik, T., and Kohls, A.G. (2010). Reinforcement Learning-Based Multi-Agent System for Network Traffic Signal Control. IET Intelligent Transport Systems. doi: 10.1049/iet-its.2009.0070.

[142] Lillicrap, T.P., Hunt, J.J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., Silver, D., and Wierstra, D. (2016). Continuous Control with Deep Reinforcement Learning. 4th International Conference on Learning Representations, ICLR 2016.

[143] Barto, A.G., Sutton, R.S., and Anderson, C.W. (1983). Neuronlike Adaptive Elements that Can Solve Difficult Learning Control Problems. IEEE Transactions on Systems, Man and Cybernetics. doi: 10.1109/TSMC.1983.6313077.

[144] Mnih, V., Badia, A.P., Mirza, L., Graves, A., Harley, T., Lillicrap, T.P., Silver, D., and Kavukcuoglu, K. (2016). Asynchronous Methods for Deep Reinforcement Learning. 33rd International Conference on Machine Learning, ICML 2016.

[145] Beyret, B., Hernández-Orallo, J., Cheke, L., Halina, M., Shanahan, M., and Crosby, M. (2019). The Animal-AI Environment: Training and Testing Animal-Like Artificial Cognition. arXiv preprint arXiv:1909.07483.

[146] Chen, M., Wang, T., Ota, K., Dong, M., Zhao, M., and Liu, A. (2020). Intelligent Resource Allocation Management for Vehicles Network: An A3C Learning Approach. Computer Communications 151(C), 485-494. doi: 10.1016/j.comcom.2019.12.054.

[147] Sasaki, Y., Matsuo, S., Kanezaki, A., and Takemura, H. (2019). A3C Based Motion Learning for an Autonomous Mobile Robot in Crowds. IEEE International Conference on Systems, Man and Cybernetics. doi: 10.1109/SMC.2019.8914201.

[148] Schulman, J., Levine, S., Moritz, P., Jordan, M., and Abbeel, P. (2015). Trust Region Policy Optimization. 32nd International Conference on Machine Learning, ICML 2015.

[149] Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. (2017). Proximal Policy Optimization Algorithms. arXiv preprint arXiv: 1707.06347.

[150] Wang, Z., Mnih, V., Bapst, V., Munos, R., Heess, N., Kavukcuoglu, K., and De Freitas, N. (2017). Sample Efficient Actor-Critic with Experience Replay. 5th International Conference on Learning Representations, ICLR 2017.

Chapter 4 – DEEP MACHINE LEARNING IN CYBER DEFENCE

Traditional approaches to cybersecurity and cyber defence are reliant on manual data analysis to support risk management activities and decisions. Though certain aspects of these activities can be automated, automations often fall short due to their simplicity and limited understanding of the problem domain. In this chapter, we will investigate literature for DML applications which aid information security continuous monitoring for a set of security automation domains defined by the US National Institute of Standards [1]. We do this to create a structured understanding of the status quo in state-of-the-art research, practical implementations, open challenges, and future vision. With these insights we point out a set of challenges for DML applications across the cybersecurity domain and conclude our findings.

4.1 MALWARE DETECTION

Malware is any malicious software that is intentionally designed to infiltrate, modify, or damage a computer system without the owner's knowledge or consent. Malware assumes many forms of digital content that include executable code, scripts, and active objects embedded inside of interactive files. Common types of malware and their characteristics are enumerated below:

- Exploits take advantage of a vulnerability.
- Adware hijacks browsing for economic gain.
- Spyware steals sensitive information.
- Ransomware encrypts files for blackmail.
- Trojans masquerade as benign software.
- Rootkits provide persistent, covert privileged access.
- Viruses replicate themselves on other computer programs.
- Worms duplicate themselves on other computers.
- Bots remotely execute commands.
- Backdoors provide illicit access.
- Cryptojackers mine cryptocurrency.
- Droppers download and install further unwanted software.
- Scareware tricks users into installing unnecessary software.

The struggle between security analysts and malware developers is a continuous battle. The earliest documented virus appeared during the 1970s. Today, the complexity of malware changes quickly, leveraging ever-increasing innovation. Recent studies underscore the role malware plays in facilitating cybersecurity breaches, note trends towards malware with targeted payloads motivated by economic gain, and provide evidence asserting the proliferation of Internet connected devices will catalyse the malware trade [2], [3].

Malware detection refers to the process of identifying the presence of malware on an endpoint device, and distinguishing whether a specific program exhibits malicious or benign characteristics. Traditional signature-based approaches to identifying and characterising malware grow increasingly disadvantaged given trivial alterations

allow malware to evade common detection methods [4], [5]. Signature-based methods are essentially regular expression-based pattern matches garnered from empirical knowledge of observed malware. Unique strings of bytes extracted from known malware samples build a database of signatures, usually provided by a subscription service from an endpoint protection vendor. When an anti-malware program receives a file to test, it compares the byte content of the file with those signatures in its database. This approach is effective and computationally efficient (i.e., low Type 1 error) provided the malware does not employ evasive measures. However, as the quantity of signatures and adoption of tricky evasions increase, pattern matching becomes computationally expensive and increasingly ineffective. Heuristic approaches somewhat address this challenge with rules, but simultaneously increase false positive rates. The fragile nature of signature and heuristic-based approaches is a long-recognised problem that has catalysed research into alternative and complementary techniques.

These complementary techniques are generally arduous processes requiring an exhaustive combination of software reverse engineering, source code debugging, runtime execution analysis, and network and memory forensics. Static analysis techniques identify surface characteristics such as cryptographic hash, size, type, header, embedded content, and the presence of software packers. Static analysis tools include source code and bytecode analysers, digital signature verification tools, and configuration checkers. Dynamic analysis techniques identify runtime characteristics such as alterations to the file system, operating system, process listings, mutexes, and network touchpoints. Dynamic techniques demand a corpus of specialty tools including unpackers, debuggers, disassemblers, decoders, fuzzers, and sandboxes by which to safely execute, instrument, and observe the behaviour of suspicious files. Many military organisations with robust information security programs adopt a hybrid approach in which unknown files of suspicious provenance are triaged and scrutinised by an array of techniques and tools [6].

Despite the comprehensive approach, many tools have limitations, and no single technique can confidently assure the provenance and hygiene of software. For example, the presence of a software packer and other tricks obfuscating file content hamper static analysis methods. Similarly, dynamic analysis through sandboxes is expensive to implement, often lack forensic traceability, and are easily subverted through virtual kill switches which instrument the execution environment. ML applications of malware discovery date back two decades. Early approaches relied on feature vectors such as ASCII strings, instructions, n-grams, header fields, entropy, and imports of dynamically linked libraries, all extracted from executable files. These approaches yielded mixed results. Though providing indications of great success and remarkable accuracy, they ultimately lacked scalability and failed to keep pace with evolving threats, necessitating the continued employment of traditional, precise signatures. The adversarial nature of malware creation and discovery ensures that adversaries will adopt new techniques once they become aware of the traits used to identify their code. Therefore, these techniques prove limiting due to a lack of obvious or natural features suggesting malicious intent.

4.1.1 Current Research

Machine Learning has been widely explored in an attempt to improve malware detection by overcoming extant challenges [7]. In the US, the last NIST standard was by Souppaya, et al. [8]. For a good review, and listing of recent research challenges, see the work done by Gilbert et al. [9] and McCallam et al. [10]. Finally, Apruzzese et al. [11] assess the security of ML solutions and identifies their main limitation which prevent immediate adoption, noting that all approaches are vulnerable to adversarial attacks and require continuous re-training and careful parameter tuning that cannot be automated. When the same ML classifier is applied to identify different threats, the detection performance is unacceptably low.

Traditional ML approaches usually rely on manually designed features derived from expert knowledge of the domain. These solutions provide an abstract view of malware that a machine learning classifier uses to make a decision. Feature engineering and feature extraction are key, time-consuming processes of the ML workflow. Following advances in other fields (e.g., Ref. [12]), malware detection is now utilising DML architectures. These solutions replace the feature engineering process with a fully trainable system beginning from raw input to the final output of recognised objects [8].

Research suggests that the application of DML, combined with unique representations and the application of unique classifiers for different threats, may overcome traditional ML challenges in malware detection. Resultingly, tools such as Dshell [12] afford a framework for analysing network traffic and creating custom decoders to parse and filter network traffic. This and similar tools enable discovery of custom protocols, encryption, and communications that can be fed into a DML pipeline. More recent research focuses on novel representations of digital content by which to train models.

4.1.1.1 Stylometry

Code stylometry is the attribution of source code or binaries via stylistic authorship identification, similar to handwriting analysis or authorship attribution of prose. In the cyber domain, attribution of code is an extremely challenging yet essential task that can identify the origin of malware. Recent research demonstrates more accurate and timely authorship attribution and characterisation and devises de-anonymisation countermeasures [13]. The approach exploits neural network learning of structural code and disassembly features, i.e., Abstract Syntax Trees, which are difficult to obfuscate. The approach can be extended to verify and validate firmware on military platforms. Stylometry attributes authorship to known authors based on a differentiation from the baseline of the firmware being installed by detecting the difference from manufacturing to installation.

4.1.1.2 Domain Transforms

ML and DML techniques have enjoyed great success in the signal processing domain [14]. Emerging research has demonstrated how representing cyber data can afford a similar degree of success [15]. Domain transforms involve representing binary data (i.e., programs) in a form more amenable to exploitation by DML. In one novel approach, binary objects are cast as visual images, as illustrated in Figure 4-1. This enables the application of computer vision and imagery-based similarity descriptors. The technique provides a key advantage by exploiting visual similarity and dis-similarity among variants [16], [17].

Like traditional static anti-malware analysis tools, this approach takes as input a binary executable file, scans it, compares the scan results to a knowledge base, and informs the user of the result. However, the scan and knowledge base are fundamentally unique and, most importantly, independent, and orthogonal from traditional methods. It would be considerably difficult for an attacker to modify existing malware in a way that defeats both traditional signatures and its visual structure. In this application, visual features derived from the structure of a file lends themselves to detection of polymorphic code. After evaluating the technique against 1.2 million packed and unpacked malware samples, its creators found the tool classified 50% of the incoming samples with more than 99% precision and further claim an ability to detect up to 70 samples per day before any antivirus vendor could detect them. This technique attempts to convert malware detection into a signal or image processing problem, wherein DML has enjoyed great success.

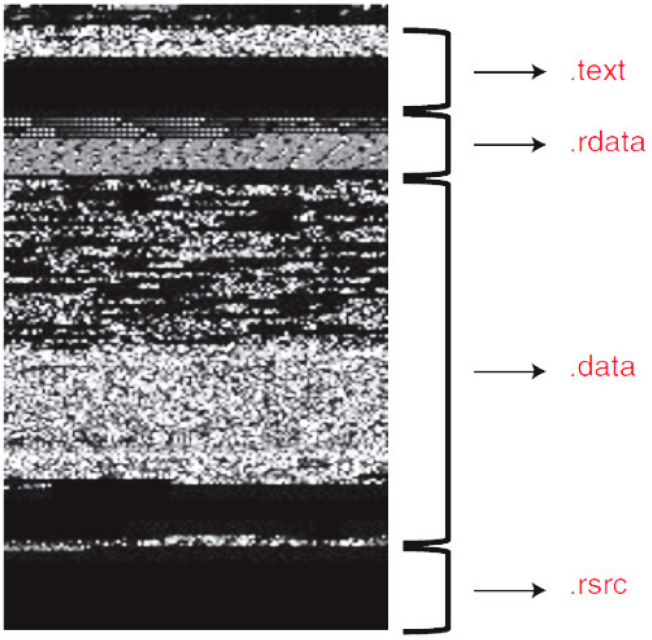


Figure 4-1: Image Casting Binary Content.

4.1.2 Practical Implementations

Industry trends point to a growing sector of companies offering AI-based cybersecurity solutions, including those which implement some form of DML for malware detection. Independent assessments value the global market for malware analysis technology between $3B and $11B USD annually and cite more than two dozen technology vendors offering software tools for identifying malware. Many of these incorporate some form of DML-based capabilities integrated within an endpoint security monitoring product [18]. Often, these models are limited to portable executable file formats.

The products commonly implement a variety of supervised learning models including random forests, gradient boosted trees, neural networks, and logistic regression models for static features that capture information about the structure and content of the file. In related applications, word embedding applies natural language processing techniques to disassembled binary code. Alternatively, convolutional neural network or recurrent neural networks have been trained to learn long sequences of raw bytes in executable files. In one documented instance, DML models have been shown to detect malware with upwards of 99% accuracy [19]. This ostensibly eases reliance upon signatures for every new instance of malware.

4.1.3 Open Challenges

Custom malware designed to penetrate military systems is unlikely to be released into the wild and thus unlikely to be identified by commercial antivirus vendors. Adversaries have further adapted to the latest advances in artificial intelligence and computing to craft attacks that better evade detection [20]. This presents many difficulties for malware analysis efforts: rapid releases of malware variants, polymorphic and metamorphic malware, obfuscation to avoid detection, timing-based evasions, and use of stolen certificates.

Further, ML algorithms that apply directly to executable files often suffer from overfitting. Malware can be readily transformed to obfuscate malicious intent while remaining interpretable by a CPU using techniques such as code reordering, register renaming, operator substitution, and opaque constants. Therefore, traditional approaches lack scalability, as well as any meaningful capability of explaining "why" a program is malicious, other than it shares characteristics seen in other programs labelled as malware.

Malware targeting military systems is frequently detected by sensors that identify anomalous network traffic or during law enforcement investigations where systems are forensically examined. Traditional approaches by which endpoints and networks are instrumented do not scale for an emergent class of fileless, in-memory malware. Lastly, training data for malware further from challenges found among other domains. Curating a comprehensive corpus of both good and bad software is challenging [21]. Compiled datasets remain susceptible to poisoning attacks and tampering. It is also worth noting that DML applications are inherently software applications and can potentially add complexity and attack surface to the digital environment.

In addition, training data sets often suffer from the class imbalance problem. There will be an artificially high number of malicious signatures in order to speed up training time, minimise data storage, and minimise processing time. But this is often very different than real-world data. Adversarial machine learning is also a challenge. Adversarial machine learning is a technique used to attempt to fool machine learning detection by automatically crafting adversarial examples. For example, benign data points might be inserted to stay below a known threshold.

Malware is forced to evolve in order to survive and operate. That is, malicious software has to constantly change to avoid detection by anti-malware engines. As a result, over time, concept drift can render some detection techniques less effective. In other words, over time malicious software changes, the detection training data becomes less relevant, and the detection technique is deprecated.

4.1.4 Future Work

Researchers in this area face several challenges. The problem of concept drift, the challenges of adversarial learning, and improved methods for attribution, mentioned above, are all active areas of research.

Because of its rapid, widespread adaptation to various diverse fields, ML is often treated as a black box, leading to difficulty interpreting results. Advances in explainability and interpretability are necessary to effectively triage suspicious software. Advances in generative adversarial networks will be of considerable importance to overcoming limitations imposed by training data. Training deep neural networks requires vast amounts of data from all classes which can be quite challenging in the field of proprietary software. Finally, novel representations of software, to include additional domain transforms and abstract representations (e.g., abstract syntax trees, program control graphs) may serve as more effective features by which to train models.

Future directions of research include the following. Classifiers that rely on more than one type of feature or modality of data to detect malware. Combining different features could produce more robust detection techniques. Mimicking biological immune systems, to use analogues to human immune systems to detect foreign objects. Lastly, automated data set generation and labelling. Having realistic data sets that could be automatically generated, labelled, and are safe to share is on ongoing challenge.

4.2 EVENT MANAGEMENT

Event management covers tools and technologies to monitor and, if necessary, respond to observed occurrences in a network or system. These occurrences can be referred to as "alarms" or "alerts" if they indicate malicious or questionable activity. They are generally captured in logs that record events within an organisation's perimeter. There is a magnitude of tools that can be considered part of this domain, but we consider two in particular: *Security Information and Event Management* (SIEM) systems and *Intrusion Detect Systems* (IDS). The former strives to enable analysis by aggregating logs from multiple security controls. The latter is deployed on strategic locations and analyse logs on the local system or network.

4.2.1 Security Information and Event Management

To enable log aggregation and analysis, each individual log source needs a blueprint that define how the accumulated data is to be structured for storage. These blueprints are referred to as *schemas* and constitute an essential component in SIEM systems. The properties of these schemas, differ depending on the applied schema creation strategy such as *Schema-on-write* and *Schema-on-read* [22], [23]. These strategies set different SIEM systems apart, as it lays the groundwork for data analytics. In *Schema-on-write* the logs are expected to be processed beforehand to extract valuable information for storage in objects called *fields*. This allows information to be queried in a quick and responsive manner but enforces a static interpretation of the logs. In comparison *Schema-on-read* store logs in a format akin to the original log and expect field extraction to be done for every attempt to query data. This allows a dynamic interpretation of the logs but requires more computational resources available upfront for individual data queries. Despite these differences, a SIEM's choice of schema creation strategy does not particularly constrain its analysis capabilities.

The level of analysis is first and foremost constrained by the utilities that are exposed through each SIEM's respective *Domain-Specific Language* (DSL). The DSL makes a repository of utilities available that can be applied for different business purposes to enable a more streamlined workflow by a security analyst. Because of this, the extent these repositories support any particular task varies depending on the available utilities. For detection tasks, this is traditionally done with signatures translated to each SIEM's DSL as a data query. A framework that facilitates this is Sigma[1] which defines a generic signature format that can be translated to Splunk, Elasticsearch, and IBM QRadar SIEM compatible queries.[2] Although prior work has by and large been invested in signature-based detections, current research attempt to enable application of machine learning for detection purposes.

4.2.2 Intrusion Detection System

There are two main axes that define an Intrusion Detection System (IDS), its deployment method and its detection method. The system is deployed either on the host-level or the network-level. There are advantages and disadvantages with both deployment methods, namely that host-based IDS (HIDS) require more computational resources and network-based IDS (NIDS) can only access data on the network-level [24].

The detection methods can be divided into signature-based detection and anomaly-based detection. Signature-based detection is useful for detecting known attacks but does not necessarily work as well with novel or mutated attacks. Anomaly-based detection can be used for detecting abnormal activity by training on activity known to be benign. This can be used to detect novel attacks but is also known for challenges with respect to false positives rates. It is also possible to train models directly with data from known attacks in a supervised manner.

[1] https://github.com/Neo23x0/sigma.

[2] https://uncoder.io/.

4.2.3 Current Research

Both of the mentioned tools support deep learning applications to a certain extent, but most attempts to current date have bypassed the SIEM or IDS systems' analysis capability to enable such applications. For instance, by using established tools to perform initial preprocessing (or tools designed specifically to extract features from raw data captures [25], [26]) and exporting resulting data to a third-party deep learning platforms [27], [28]. Recent efforts do however attempt to improve upon this by including deep learning frameworks as an extension to existing tools [29], [30], [31]. The more promising attempt to integrate such functionality are brought by Splunk's Deep Learning Toolkit (DLTK)[3] released in 2019, which includes access to frameworks such as Keras and PyTorch for deep learning applications. We have also identified attempts to integrate machine learning functionality in ThreatEye4, which is a proprietary NIDS.

4.2.4 Practical Implementations

Because these developments are so recent, there is a lack of literature that address *deep learning applications* with existing analysis capabilities in SIEMs and IDS systems. Numerous attempts that make use of *traditional machine learning* do exist, however. Shah et al. [32] propose a machine learning plugin for the Snort NIDS system, which runs in parallel with its signature rule set to reduce the false positive alarms. It has support for algorithms such as SVM, Fuzzy Logic, Decision Tree, hybrids of SVM and FuzzyLogic, and certain optimised SVMs [33]. Moreover, Splunk have ML applications that detect unknown and hidden threats with behavioural analysis.[4,5] In other words, there are great opportunities to explore and research deep learning integration and thus applications in both SIEM and IDS systems.

4.2.5 Open Challenges

There are certain challenges that the current offerings do not address. For instance, with SIEMs, even though these platforms have accumulated a great quantity of information, the underlying data structures prohibit data labels to be appended due to their immutable characteristics [34], [35]. This implies that data either needs to be reingested with labels or enriched with labels in a post-processing stage of a data query to make supervised deep learning applicable. This is unless routines to accumulate labelled data, including curation and maintenance, are already established. Moreover, there is a need to arrange ongoing efforts on how to re-train, evaluate, and deploy deep learning models. Scaling these deep learning models to multiple tasks can introduce considerable overhead. Such overhead could be partially reduced through incremental deep learning, relaxed requirements for evaluation, and by allowing ad-hoc deployments. However, software that allow such functionality is in its infancy. Nonetheless, newer iterations of current deep learning extensions plan to improve upon this overhead through streamlining the management of models and related deployment logic [31].

For IDS systems, many of the same challenges apply in regard to scalable analytics. But one of the main challenges in using machine learning with such systems, is still developing adequate methods for applying machine learning in an operational setting. Consequently, few use them as more than a data source [24] for detecting malicious network traffic with machine learning. Additional challenges include more encrypted data, training algorithms in a live environment and detection management.

Moreover, both systems also need to be aware of more attack vectors, e.g., by the introduction of malicious traffic in the training data or fooling the method by adversarial methods.

[3] https://github.com/splunk/deep-learning-toolkit

[4] https://www.counterflow.ai/threateye

[5] https://splunkbase.splunk.com/app/3617

4.2.6 Future Work

We suspect research of deep learning applications within event management will become increasingly popular once tools commonly deployed by Security Operations Centres (SOC) teams integrate deep learning frameworks or algorithms. All existing security controls need to integrate and coexist with each other and introduce a great level of challenges with respect to complexity and maintenance that must be overcome. We also predict that differentiating malicious and benign activity in data sources will prove challenging, with malicious actors being increasingly interested in compromising the underlying data.

4.3 INFORMATION MANAGEMENT

The classification of data is a standard requirement in the military domain. Traditionally paper documents have been marked with labels like “unclassified” or “secret” and users had to follow strict regulations to ensure the required confidentiality. One property of this paper-based system is a direct connection between the document and its classification as it is part of the document. The meta information on the document classification cannot be separated from the document itself. This cannot be implemented in the same way in the digital environment because it is very often easily possible to separate the classified data from its metadata and therefore its classification. Some systems try to guarantee such an inseparable connection. However, they are limited to edge cases. In practice, data is stored in a myriad of systems, transferred, altered, converted, and uses uncountable formats. Some examples are:

- Text documents stored in office formats like PDF, Office Open XML, or plain text;
- Images stored in simple formats like BMP (bitmap image file format) or JPEG; and
- Audio data stored as WAVE or MP3.

Some of these formats offer protected metadata others are plain formats without anything but the information.

This section focusses on a general approach often referred to as Data Loss Prevention/Data Leak Prevention (DLP) which can handle arbitrary data. Such DLP systems analyse if a user action applied to data (e.g., sending a document via email or printing it) is allowed by a given rule set. Metadata, like classification can ease the process but are (in theory) not required. We can formalise such a DLP as a decision task where we want to decide if a given action a might be applied to a document d following the rules r. We could model this as the function $f(a,d,r) \rightarrow \{allowed,not\_allowed\}$.It is important to note, that we have two options for a default rule. In a whitelist approach, we limit the actions to data to allowed rules. Everything else is forbidden. Blacklist approaches reverse the idea. Everything is allowed unless explicitly forbidden. Both approaches are common in cyber security.

We can distinguish two major system designs. End point solutions work like AntiVirus (AV). They monitor activities on the given device. End point solutions can access the data in an unencrypted form on access (also known as “data in use”) or search proactively for data on the system (also known as “data at rest”) such that the major challenge is the classification of the given data and applying the policy, e.g., stopping classified files from being printed or transferred via insecure channels or to untrusted destinations. Network solutions monitor data exchanges also referred to as “data in motion.” Therefore, they cannot enforce rules on a given host, but restrict the information exchange. A common problem the network solutions face is that more and more network traffic is encrypted by an end-to-end encryption and therefore not readable by a monitoring system. A third class in between the given two are cloud-based solutions where the DLP is enforced on data stored in a cloud-based system. Cloud-based solutions seems to be very special, but they are similar to end point solutions as they can

operate on "local" data in their cloud and to network solutions as they can monitor the traffic flows. However, end points might store encrypted data in the cloud such that cloud systems might suffer from less access to unencrypted data.

DLP systems face the following challenges:

1) **Data acquisition:** The DLP must access the data itself to analyse if an action is allowed. This becomes more and more complicated for network-based solutions.
2) **Analysing data:** DLP systems must "understand" and classify the content. This means, they must support a wide range of different file types.
3) **Representing rules:** The rules are necessary to decide if an action may be applied to given data. The representation of the rules is straight forward for some rules like "do not allow transfer of documents marked as classified." However, "fuzzy" rules are much harder. For example, "do not allow transfer of pictures of military locations" as there is no clear definition when a picture contain a military location.

DML can be applied to all challenges but analysing the data is the most obvious one and will be discussed briefly in "Current Research."

4.3.1 Current Research

Analysing the data and its classification is the most important functionality in DLP and a lot of research in the field is centred around this topic. DML excels in interpretation of complex data like texts, images, or videos. So, DML can give a DLP machine readable access to the information encoded in such data. Thus, theoretically Deep Neural Networks that are being used for NLP or image or text classification, object detection etc. should be applicable to DLP. Nayak et al. [36] claim in their survey of DLP approaches that DL might be used for sensitivity classification with increased accuracy. Trieu et al. [37] use a DL-based NLP approach to classify sensitive textual data with good results. Cheng et al. [38] highlight the potential of behavioural analysis based on DL for DLP. Ong et al. [39] present a system for real-time sensitivity classification of files via DL. Tarawneh et al. [40] utilise DL for the classification of scanned invoices.

4.3.2 Practical Implementations

Most commercial security vendors offer some form of DLP solutions. However, the documentation about the used techniques is sparse such that it is unclear if and what DML/ML is used. According to marketing material, manuals, and survey papers, current DLP systems are very likely a mixture of at least the following data classification techniques:

- **Labelled Data:** Data can be labelled with classification information like in the traditional paper-based world. Unless the system in use can guarantee the label integrity, labels should be used mainly in blacklist systems, where a classification label prohibits particular usage.
- **Exact Matching:** Contents are not analysed but hashes are used to identify known data. This can be file hashes, database fingerprints etc. This approach is well known and has been used to identify malware in the AV sector for decades. It is very fast, is computational cheap on current hardware, and has a low false positive[6] and zero false negative rate. However, attackers can easily bypass such systems by altering the data.

[6] False positives are caused by hash collisions.

- **Partial Document Matching:** Looks for partial matches of documents. Although it seems similar to exact matches, it differs significantly as there is no easy way to represent partial matches. Similarity hashes are a promising approach but require special algorithms for different data types. E.g., two images look very similar when they are encoded as PNG or JPEG for human observers, but their byte representation is very different.
- **Regular Expression (RegEx):** RegEx can be used to find data in particular formats. This approach is common in commercial system to comply with regulations, e.g., from the payment industry. Regex is a powerful first-level-filter technique as RegExs are computationally cheap but might have a high false positive rate. An example rule might filter all 16-digit numbers which are likely credit card numbers. However, this regex matches common phone numbers as well. Therefore, such rules should be enriched with further features (e.g., credit card checksums).
- **"Simple" Machine Learning and Statistics:** ML and statistical methods have been used as classifiers in particular situation like email classification. They require a large volume of data to initialise the scanner and are prone to misclassification (False Positive and False Negative).
- **"Advanced" Machine Learning:** As DML/ML is very effective for certain tasks like object identification in visual data, text recognition (Optical Character Recognition (OCR)), speech recognition, or NLP, it is very likely that DML/ML is part of commercial solutions. In contrast to the previously mentioned techniques, they are not filters but can be used to pre-process the data for further analysis. E.g., an image classifier could detect particular objects in an image which might be an indicator if a given image is shareable or not. E.g., an image of a fighter jet is more likely a classified information than a cat picture.
- **Combinations:** In practice, most systems combine different techniques. E.g., exact matching as a first-level-filter, followed by a regex and labelled data. This reduces the load for computationally expensive tasks like partial document matching.

4.3.3 Open Challenges

In this section, we briefly discuss open challenges based on the identified challenges from the introduction of this chapter.

Current IT security measures strengthen data confidentiality on different layers. Data acquisition on the network layer will be a significant challenge as soon as end-to-end-encryption is widely adopted as intended by large parts of the IT security community. Similar problems arise for local endpoint solutions if strong data separation will be enforced on the OS level which is partly in place, e.g., in iOS and Android. It is still an open question if we will see similar trends on more tradition PC-style systems, but the trends of containerisation and virtualisation point in that direction. Furthermore, recent OSs implement more sandboxing features and secure enclaves or trusted execution environments in hardware are available on modern CPUs (e.g., Intel SGX, ARM TrustZone). DLP system could counter this trend by a deep integration into the OS kernel which is common in AV solutions. Parts of the IT security community are very concerned about this trend as such solutions might introduce new security weaknesses.[7] Especially closed source systems are in danger, e.g., Windows, as the vendors have a limited knowledge about the inner processes of the OS kernel and potential side effects of their

[7] There is a long list of security weaknesses in AV products risking system integrity and user privacy. e.g., https://success.trendmicro.com/solution/000151730-SECURITY-BULLETIN-Trend-Micro-OfficeScan-Arbitrary-File-Upload-with-Directory-Traversal-Vulnerability, which was the starting point for a cyber-attack on Mitsubishi Electric https://www.zdnet.com/article/trend-micro-antivirus-zero-day-used-in-mitsubishi-electric-hack/ or https://support.kaspersky.com/general/vulnerability.aspx?el=12430#110719, which compromised user privacy by introducing code into all opened websites.

products. Thus, significant research is necessary if DLPs as described above will be possible in the future. Some of these topics are under research for related systems like Intrusion Detection Systems (IDS).

The data analysis is necessary for any DLP solution as described above and must be addressed on at least two levels. First, the analysis of the raw data. E.g., does a given picture contain sensitive information? Questions like this are one of the hot research topics in DLP-DML research and well covered. Nevertheless, there will be a very large gap from research to commercial solutions to an application in military systems. A practical challenge for researchers as well as users in the future is the availability of training data for DML-based systems. They typically depend on large quantities of training data to "configure" the DML system to the environment. The second level to address is a very technical question. The same information can be stored in different ways. For example, pictures can be encoded as simple bitmaps (pixels with colour information), compressed formats like PNG or JPEG, or as vector graphics. Thus, it is not sufficient to analyse the raw displayed pixel data on a screen. Further information like included metadata might be relevant as well. Furthermore, the different formats alter the image (e.g., compression artefacts). Therefore, specialised tools are required, or the user must be limited to a supported format set. Although some research questions are open and could improve the situation, this problem seems to be an engineering question.

The representation of rules in the system seems straight forward but is in practice a challenge if they are formulated in an open or fuzzy way E.g., it might be forbidden to share pictures of certain areas, the acoustic signature of a submarine, or texts discussing the political situation in a country based on intelligence information. The system needs the information to identify such rule violations but there is no obvious way to represent the information for complex data like images. DML can help if the information extracted by the DML system are rich enough. E.g., if a DML can extract geo coordinates for a given picture, the required rule to forbid pictures of a certain area is very simple as it is a simple georeferenced bounding box.

4.3.4 Future Work

As discussed in the Open Challenges section, DLPs face challenges on different key components. The mentioned challenges related to IT security trends and ML are relevant for a lot of applications and will be researched not exclusively for DLP. Advances in these fields can be adopted to DLP needs. The rule representation challenge seems almost unique for DLPs and as such need's special attention in the future.

We describe current systems in Section 4.3.2. With advances in ML, the building block of "Advanced Machine Learning" can be improved. Such improved classifiers are necessary to implement advanced DLP system allowing open or fuzzy rules.

4.4 VULNERABILITY MANAGEMENT

The Committee on National Security Systems (CNSS) Glossary No. 4009 defines a vulnerability as a weakness in an information system, system security procedures, internal controls, or implementation that could be exploited or triggered by a threat source [41]. A software vulnerability is a security flaw, glitch or weakness found in software code that could be exploited by an attacker [42].

Vulnerability management is the cyclical practice of identifying, classifying, remediating, and mitigating vulnerabilities [43]. The National Institute of Standards and Technology (NIST) defines vulnerability management capability as an Information Security Continuous Monitoring (ISCM) capability that identifies vulnerabilities on devices that are likely to be used by attackers to compromise a device and use it as a platform

from which to extend compromise to the network [44]. The purpose of vulnerability management is to ensure that software and firmware vulnerabilities are identified and patched to prevent attackers from compromising a system or device which in turn may be used to compromise other systems or devices.

4.4.1 Vulnerability Management

The vulnerability management process comprises the following phases [45], [46], [47]:

- **Creating baseline:** Critical assets are identified and prioritised.
- **Vulnerability assessment:** Identify known vulnerabilities in the organisation infrastructure.
- **Risk assessment:** Summarise the risk level identified for each asset.
- **Remediation:** Prioritise and fix vulnerabilities in order according to risk. Three possible types of remediations are installing a patch, adjusting configuration settings, and uninstalling a software application.
- **Verification:** Verify that vulnerabilities have been remediated.
- **Monitoring:** Continuously monitor for vulnerabilities, remediations and threats.

A central part of the vulnerability assessment phase is scanning the assets for vulnerabilities. This is done using tools known as vulnerability scanners, which can identify vulnerabilities, measure exposure, identify out-of-date software versions, validate compliance with an organisational security policy and generate alerts and reports about identified vulnerabilities [48] The scanners scan against vulnerability databases, such as the National Vulnerability Database (NVD).

The identified vulnerabilities are typically given a score, which is then used when prioritising patching. The score may be based on several factors, such as the severity of the vulnerability itself context-based factors, which include the value and priority of the affected asset.

The Common Vulnerability Scoring System (CVSS) is a published standard for communicating the characteristics and severity of software vulnerabilities. CVSS allows prioritisation of remedial actions and calculating the severity of vulnerabilities discovered in one's systems. Vulnerabilities are classified based on severity level (low, medium, or high) and exploit range (local or remote). NVD provides CVSS scores for almost all known vulnerabilities.

The vulnerability management process can and needs to be automated to a large degree, in order to handle to vast amount of data in a consistent and timely manner. A number of software solutions are available for this purpose.

4.4.2 Current Research

4.4.2.1 Vulnerability Discovery

Typically, vulnerability scanners only scan for known vulnerabilities. Previously unknown vulnerabilities can be discovered, e.g., from source code or by fuzz testing. A fuzz tester provides invalid, unexpected, or random data to software, to determine whether problems occur (e.g., crashes or failed built-in assertions) [49].

Zeng et al. [50] review studies where deep machine learning has been employed to detect vulnerabilities from source code. The authors conclude that the application of deep machine learning techniques is not yet mature.

Heidbrink et al. [51] investigate the use of multimodal deep learning for detecting software flaws using both source and binary representations of a software as inputs. They demonstrate that multimodal learning can improve flaw detection performance over baseline deep learning models that do not treat source and binary representations of the software programs as individual input modalities.

Wang et al. [52] provide a review of fuzzing based on machine learning. The authors found that deep machine learning algorithms are more widely used in fuzzing than traditional machine learning algorithms, due to robust learning and expression abilities. Among the algorithms used in the reviewed papers, the two most used algorithms are LSTM and seq2seq. According to Wang et al. [52], many stages in the fuzzing process can be stated as classification problems, for which machine learning algorithms are well suited.

4.4.3 Practical Implementations

There exists a number of commercial solutions for vulnerability management that the vendors claim to utilise machine learning. A common theme appears to be the use of machine learning algorithms to prioritise vulnerabilities to remediate. Only one vendor claims explicitly to use deep machine learning.

Balbix BreachControl is claimed to leverage deep machine learning and other artificial intelligence algorithms to process information from network assets and calculate a risk score for them [53].

Qualys VMDR is claimed to utilise machine learning for prioritising the riskiest vulnerabilities on the most critical assets, but it is not known if these algorithms include deep machine learning [54].

Tenable.sc and Tenable.io are claimed to utilise machine learning for prioritising vulnerabilities to remediate, but it is not known if these algorithms include deep machine learning [55].

Kenna.VM is claimed to utilise machine learning to calculate risk scores for prioritising vulnerabilities to remediate, but it is not known if these algorithms include deep machine learning [56].

4.4.4 Open Challenges

Deep learning techniques require large data sets. In the field of vulnerability discovery, Wang et al. [52] and Zeng et al. [50] recognize the need for large-scale open datasets for training and testing.

4.4.5 Future Work

The 2016 Cyber Grand Challenge (CGC) competition sponsored by The Defense Advanced Research Projects Agency (DARPA) demonstrated the potential for automated cyber security systems that can discover, evaluate, and patch vulnerabilities in real-time. Seven teams competed in the CGC Final Event which was held on 4 August 2016 in Las Vegas.

DARPA had developed a simplified open-source operating system extension expressly for the CGC, with previously unseen binaries. Among the methods used by the teams in CGC was fuzzing.

As noted previously, deep machine learning has been applied to fuzz testing. A future vision is to have tools that can scan software for vulnerabilities and assist in patching them. On the other hand, such tools could be used for malicious purposes as well, by automatically discovering and exploiting vulnerabilities, instead of patching them [57]. This is a field of ongoing research and has been reviewed by Ji et al. [58].

4.5 SOFTWARE ASSURANCE

The Committee on National Security Systems [59] defines software assurance as the level of confidence that software functions as intended and is free of vulnerabilities, either intentionally or unintentionally designed or inserted as part of the software throughout the lifecycle [59]. The definition in NASA Technical Standard 8739.8A uses similar wording [60].

The software assurance domain ties in with other domains, particularly with the vulnerability management domain, with regards to vulnerability scanning and discovery, but also with malware detection.

4.5.1 Current Research

Three major types of tools and techniques for software analysis have been identified [49], [61]:

- **Static Analysis:** Examines the system/software without executing it, including examining source code, bytecode, and/or binaries. This group includes:
  - Attack modelling.
  - Source code analysers.
  - Binary/bytecode analysers.
  - Human review.
  - Secure platform selection.
  - Origin analyser.
  - Digital signature verification.
  - Configuration checkers.
- **Dynamic Analysis:** Examines the system/software by executing it, giving it specific inputs, and examining results and/or outputs. This group includes:
  - Application-type-specific vulnerability scanners.
  - Fuzz testers.
  - Automated detonation chambers.
- **Hybrid Analysis:** Tightly integrates static and dynamic analysis approaches; for example, test coverage analysers use dynamic analysis to run tests and then use static analysis to determine which parts of the software were not tested. This group includes:
  - Test coverage analysers.
  - Hardening tools/scripts.
  - Execution and comparison with application manifest.

- Tracking sensitive data.
- Coverage-guided fuzz testers.

Binary and bytecode analysers include traditional malware scanners.

Among the tools and techniques listed above, deep machine learning has been applied to source code analysis and fuzz testing. These applications are discussed in the section on vulnerability discovery.

Beyond these *traditional approaches* to increase software assurance, there is work done in the commercial sector that attempt to make AI-generated code [62], [63]. Whether or not this can reduce the number of bugs, and thus security vulnerabilities are until further uncertain. But we suspect this can be a direction worth pursuing, in future research.

4.5.2 Practical Implementations

Complete software assurance systems that make use of deep learning to provide a level of confidence have not been identified, but DML applications which can partially facilitate this exists through work done in vulnerability scanning and malware detection.

4.5.3 Open Challenges

The open challenges in the sections for vulnerability discovery and malware detection also applies here.

4.5.4 Future Work

We suspect future work depicted in our vulnerability management and malware detection section, can enable software assurance as a DML-application once research in those areas progress further and work in tandem. There might also be improvements, worth pursuing by researching whether AI can generate safe code.

4.6 ASSET MANAGEMENT

Best practices for cybersecurity necessitate methods to account for digital assets which make up the information environment [1], [64], [65]. Asset management refers to the practice by which organisations maintain an inventory of hardware, software, and information resources, and has long been considered an integral part of a robust cybersecurity posture [66]. Though traditionally accomplished through some combination of tools for configuration management, network management, and licence management, the proliferation of cloud computing and service-oriented technology have led way to newer solutions. Information Technology Asset Management (ITAM), Information Technology Service Management (ITSM), and Software Asset Management (SAM) tools, for instance, expose insight by which to account for and maximise the business value of technology investments [67], [68].

The need for and utility of these solutions can be characterised by their demand. Independent assessments value each of the global markets for ITAM, ITSM, and SAM tools between $1B and $5B USD annually, and cite more than two dozen technology vendors offering software tools or managed services [69], [70], [71]. These solutions instrument devices, software, or in the case of cloud services, an interface from cloud service providers. They further provide workflows to assign assets to business roles and functions. Despite the extensibility of available instrumentation and workflow features, these tools share common traits in their ability to sense, query, and

interpret data native to the assets they monitor. More distinctly, they operate as a means by which to support what are ultimately manual business processes imposed by humans.

It is through this lens the disruption of deep learning towards asset management can best be realised. Existing tools provide information to an operator who oversees a business function. While their implementation and effective use can help mitigate security risks, they require their operator to specify a set of configuration parameters. For instance, SAM tools require their operator to configure how to interpret software licensing terms and product use rights. These tools afford a degree of automation through business intelligence dashboards and workflow suggestions, which, counterintuitively, may increase the complexity of the overall solution because of the required tuning.

4.6.1 Current Research

Asset management tools can produce and record large quantities of data that, if exploited, offers novel insight into both cybersecurity and business operations. For instance, [72], [73] present techniques for automating helpdesk operations based upon the application of support vector machines. Early research into cybersecurity concerns of smart grid technology cite advanced asset management of metering infrastructure as the first step towards a smart electric grid that [74]. Recent work towards cyber defence of military systems have further proposed using distributed autonomous agents to sense and adaptively defend their environment [75]. Among the common themes of these applications are the ability to reduce human oversight, adaptively manage technology consumption, optimise resource utilisation, map dependencies between assets and digital workflows, and predict or respond to organic business risks.

Nowhere are these trends more essential than the growing adoption of the Internet of Things (IoT) in which non-traditional devices are becoming increasingly network-enabled. As computing power at the edge continues to grow, so does the potential for asset discovery applications among traditionally disadvantaged environments. Some of these applications are proposed or explored in open literature [76].

4.6.2 Practical Implementations

- IT Service Management Automation [72].
- IT Helpdesk Operations [73].
- Smart Grids [74].

4.6.3 Open Challenges

Scaling DML applications for asset management in this context will necessitate a comprehensive and interdisciplinary approach in tune with advances in edge computing, networking, etc. “Industry 4.0” refers to the trend of automation and data exchange in manufacturing [77]. Contributing digital technologies include mobile devices, IoT platforms, location detection technology (e.g., Radio Frequency Identification, Near-Field Communication), 3D printing, smart sensors, data analytics, augmented reality, wearable computing, and network-enabled robots and machines.

4.6.4 Future Work

“Smart asset management” underlies a diverse array of Industry 4.0 components and encompasses mass customisation of manufactured products, improved uptime and device availability, adaptive supply chains, and the direct interaction between parts and products. Deep learning will likely underpin aspects of smart asset

management, and much of the open literature on the topic suggests the need to explore industry specific use cases [78], [79], [80], [81], [82]. Indeed, approaches have already been proposed by which to transparently monitor the online condition and status of industrial machines [83]. Clearly, trends towards smart, ubiquitous devices will drive the demand for innovative approaches to asset management, where not only the human operator, but all manner of devices themselves, are capable of adapting to their environment in order to continually optimise themselves.

4.7 LICENCE MANAGEMENT

Licence management tools control where and how software products can run. They capture licence agreement terms in code, automate the collection of software usage, and calculate cost implications that help optimise software spending. When adopted by software vendors and integrated into their products, they help curtail software piracy and provide tailored licensing features (e.g., product activation, trial licences, subscription licences, floating licences). When adopted by end-user organisations, they help comply with software licensing agreements. Licence management capabilities are often found among SAM tools.

4.7.1 Current Research

Little work has been done in the direct application of deep learning towards licence management. Some machine learning approaches, however, have been proposed to address monitoring software licence terms and conditions. The approach aims to optimise consumption-based cost and billing models, project resource demand, and augment human performance management by aligning employee skills with software features [84]. The potential benefits are well-touted given the increasing amounts of information collected by and complexity associated with operating such systems. Given the organic integration of licence management capabilities among asset management solutions, we defer the remainder of this Section to the prior discussion on Asset Management.

4.7.2 Practical Implementations

- See Asset Management.

4.7.3 Open Challenges

- See Asset Management.

4.7.4 Future Work

- See Asset Management.

4.8 NETWORK MANAGEMENT

Network management tools include host discovery, inventory, change control, performance monitoring, and other device management capabilities. Network management tools often overlap capability with asset and configuration management tools, with added features facilitating device monitoring and configuration. Network management similarly encompasses those systems within an organisation’s boundary but may extend beyond its traditional perimeters in order to manage cloud services. In fact, the explosive growth and adoption of software, network, and virtualisation technologies have fuelled multiple markets offering an array of tools falling under the network management umbrella:

- Client Management Tools automate endpoint management tasks that include operating system and software deployment, inventory, distribution, patch management, and configuration management.
- Cloud Access Security Brokers and Cloud Workload Protection Platforms provide visibility and enforcement into data and workflows residing among cloud environments.
- Cloud Management Platforms provide for the management of public, private, hybrid, and multi-cloud resources and deployments.
- Continuous Configuration Automation Tools enable the description of configuration states, customisation of settings, software, and reporting.
- Hyperconverged or Integrated Infrastructure tools provide centralised functions to manage virtual computing, storage, and networking systems.
- Network Firewalls, Web Application Firewalls, and Secure Web Gateways provide key network security controls for inspecting and filtering network traffic.
- Network Access Control tools implement policies for controlling access to infrastructure by endpoints and devices and based on identity, location, or configuration.
- Network Automation and Orchestration Tools automate the maintenance of endpoint device configurations.
- Network Performance Monitoring and Diagnostic tools provide historical and in situ views into the availability and performance of the network and application traffic running on it.
- Software Defined Network (SDN) tools provide programmatic configuration of network devices and resources.
- Wide Area Network (WAN) Optimisation tools monitor the performance of applications running across the WAN as well as service expenses imposed by those applications.
- Unified Endpoint Management tools provide mechanisms to configure, manage, or monitor disparate devices.

4.8.1 Current Research

Many deep learning applications for cybersecurity, particularly those for event detection and malware detection, are already enabled by data which is collected or exposed by network management tools. Consequently, prior research in deep learning applications to network management have largely centred around network monitoring and event classification [85], [86], [87]. Recent work, however, demonstrates the effectiveness of deep learning-based routing for traffic control in packet switched networks [88]. Similarly, proposed applications of deep learning to network management aim to automate or optimise network management tasks without human oversight.

4.8.2 Practical Implementations

- Moving Target Defenses [89].
- Log Management [90].
- See Event Detection.
- See Malware Detection.

4.8.3 Open Challenges

Military networks comprise considerable scale and diversity. Adapting network management techniques for strategic and tactical assets will be a considerable challenge.

4.8.4 Future Work

Moving Target Defense (MTD) is an emerging area of research that stands to greatly benefit from AI-driven approaches. Where traditional network defences fail to account for the attacker's inherent advantage present due to the static nature of the environment, MTD continuously shifts the configuration of that environment, in turn reducing the success rate of cyberattacks [91]. Deep learning has already been shown to accurately classify applications with traffic that is naturally garnered by SDN controllers [92]. Emerging research has further demonstrated the value of DML towards adapting that same traffic dynamically, over time [89].

4.9 CONFIGURATION MANAGEMENT

Configuration management tools allow administrators to configure settings, monitor changes to settings, collect setting status, and restore settings as needed. Configuration management tracks the relationships between components that deliver services, rather than the assets or networks themselves. Managing configurations found among information systems and network components is an arduous task. System configuration scanning tools provide an automated capability to audit a target system and assess compliance with a secure baseline configuration. Identity and account configuration management tools enable an organisation to manage identification credentials, access control, authorisation, and privileges. Identity management systems also enable and monitor physical access control-based on identification credentials. Software configuration management tools track and control changes among source code and software builds. Similar to other security automation domains, the trend for applications of deep learning suggests a movement away from humans managing software systems and towards computers managing software systems themselves.

4.9.1 Current Research

One of the more recent and interesting practical applications of machine learning towards software configuration management is found among Linux kernel development, where maintainers use machine learning to distinguish patches that fix bugs from those which don't. In this application, the most common 10,000 words found among kernel commit messages are tagged if they are in a candidate patch. Combined with features that include code metrics for the changes in the commit, author information, involved parties (e.g., reviewers, committers), and modified files, Linux maintainers trained a neural network with commits which resulted in a 92% reduction in manual effort required to identify patches [93].

4.9.2 Practical Implementations

- Moving Target Defenses [89].
- Dynamic Service Management [94].

4.9.3 Open Challenges

Military networks comprise considerable scale and diversity. Adapting network management techniques for strategic and tactical assets will be a considerable challenge.

4.9.4 Future Work

As with Network Management, MTD techniques will enhance configuration management capabilities.

4.10 PATCH MANAGEMENT

Patch management refers to the process of identifying, locating, and applying patches to a suite of managed software, typically in an enterprise environment. Patches are generally security-oriented, aimed at fixing software or firmware vulnerabilities. As new software vulnerabilities are continuously being discovered, patch management can become a difficult and imposing task, especially for organisations with hundreds of hosts and a complex software inventory. As such, a robust patch management process is necessary in order to keep an organisation safe from malicious activity.

Patch management is complicated by a variety of challenges. Firstly, an organisation has to consider a patching mechanism that secures a multitude of hosts included work-from-home devices, non-standard devices, mobile devices, and devices with a variety of operating systems and virtualised devices. Furthermore, patches can be delivered using several different mechanisms such as manually installing the patch, directing the software to patch itself, automated, a scheduled update or a patch management tool, either a third-party tool or one provided by the operating system. Due to the fact that it is both a time-consuming process as well as vital for security, any methods for automating patch management would be extremely beneficial.

4.10.1 Current Research

Despite the importance of patch management and the desire to automate the process, little work has been done on utilising techniques from machine learning and data science to expedite patch management. This is likely due to the fact that the heterogeneity of hosts and patching mechanisms make it difficult to produce uniform datasets for training and evaluating machine learning models. Furthermore, it is unclear how machine learning could help alleviate many of the problems associated with poor patch management, such as identifying potential side effects or verifying that the patch was applied without error.

However, there does exist some work that attempts to bridge the gap between the two areas and their intersection seems to be getting some more attention. We identified three papers, all from the past year that apply some form of machine learning to patch management. All of these methods rely on data collected from the National Institute of Standards and Technology's National Vulnerability Database (NVD). The NVD contains standardised information that allows a security professional to assess the potential impact of a vulnerability, such as its CVSS. As far as we're aware, all work using machine learning for patch management published thus far prioritises patches and/or mitigation strategies based on the properties of the vulnerability that the patch is rectifying, rather than the properties of the patch itself or properties of the organisation applying that patch to their software inventory, for instance, no work has been done on considering the size of several patches and scheduling them in a manner that does not exhaust the organisation's bandwidth. There are also papers that utilise machine learning to identify the likelihood of Proof-of-Concept (PoC) exploits being developed [95] or utilised in the wild [96]; this obviously has profound impact on patching decisions but they do not utilise machine learning in the patch management process itself and thus we exclude them from this discussion.

The first of these by Zhang et al. [97] presents a system for dynamically scheduling patches in a risk-aware manner. In particular, they emphasise the fact that the risk of a vulnerability being exploited is not fixed, increasing over time as the likelihood of a Proof-of-Concept (PoC) exploit being developed and released publicly on a platform such as GitHub or Metasploit increases. In order to predict the probability of a

vulnerability being exploited, information is scraped from the NVD and converted into a feature vector, such as the CVSS score, the CWE type, the attack complexity and bigrams from the vulnerabilities title and description, 'windows server', 'execute arbitrary' and 'verify certificates' being some prevalent examples. To predict the likelihood of a vulnerability being exploited, this data is used to train a series of neural networks with each network trained to predict the exploit probability of a vulnerability on a given day. Assuming a monthly patch scheduling process, this requires roughly 30 trained neural networks. Finally, they propose a group-based scheduling process where an organisation's assets are divided into groups based on their functions, allowing patches to be applied in two phases, by determining the order of patches to be applied within each group and then determining the patching order of the groups themselves. Zhang et al. demonstrate that this methodology prioritises patches in a manner than minimises the total risk incurred by an organisation when compared with random patching, CVSS-based patching, or time-based patching. Furthermore, they claim that the number of vulnerabilities patched before their exploits are released increases substantially using their system.

A second paper by Zhang et al. [98] investigates the use of machine learning methods for automated vulnerability remediation analysis. In particular, they specify the case study of the remediation decisions made by an electric utility company where patching has a high cost as it could disrupt the power delivery service. Manually assessing vulnerabilities and assets to make remediation decisions is a time-consuming process, with strict standards being applied to utilities companies for identifying security risks. In order to relieve this workload, Zhang et al. produce a machine learning-based approach that recommends one of three possible options for an identified vulnerability: patch-now, mitigate-now-patch-later and patch-later. This is done by utilising features from the NVD to assess a vulnerability's potential impact, such as the CVSS score and attack vector, as well as features describing the importance of a company asset relative to the vulnerability, for instance, a workstation is more likely to be attacked using a browser exploit than an application server with no internet access. These features are then fed into a decision tree, trained using data collected from the electric utility company, and a course of action is recommended. This process produces a high prediction accuracy of over 97%. Zhang et al. estimate that this process can save over 30 person hours per month in their partner utility company.

Finally, McClanahan et al. [99] explore the usage of machine learning methods for locating vulnerability mitigation information in the event where a software patch is not yet available. In these cases, a security professional would have to manually search for the information, a time-consuming process. As with the other two papers, the NVD acts as McClanahan et al.'s primary repository for information regarding mitigation strategies. Their methodology involves automatically visiting the NVD page for a particular CVE and extracting all references for that vulnerability. As various vendors and security sources display their security notices in different formats and use different terms for mitigations – such as workaround or resolution – locating the information is non-trivial. McClanahan et al. use a simple NLP model, a CNN with a bag-of-words text representation, to determine if a webpage contains a mitigation strategy, automating this process. They compare this strategy to a website-specific approach, where the keywords typically used by a given website are searched for, a heading-based approach, where simple heuristics are used to try and locate the section headings of a given webpage. Their NLP approach is the least accurate of the three methods, 91.8% vs 95.2% and 94.8% respectively, but suggest that the increased generality offsets this loss in accuracy.

4.10.2 Practical Implementations

- Patch scheduling [97].
- Remediation analysis [98].
- Locating vulnerability mitigation information [99].

4.10.3 Open Challenges

Heterogeneity of hosts and patching mechanisms, make it difficult to automate tasks tied to patch management.

It is also unclear, how machine learning could improve or resolve problems associated with poor patch management

4.10.4 Future Work

In conclusion, although there is a reasonable amount of work applying machine learning methods to areas related to patch management, such as exploitability analysis, exploit discovery and vulnerability management, little work has been done applying these methods to patch management itself. The current state of the research is in its infancy, with the first papers in the area appearing as recently as last year. However, the results they present are promising; the work by Zhang et al. in particular is closely tied to an industry partner and the results they present seem to indicate that these methods are a beneficial addition to the patch management process.

For future work, a more complete pipeline could be developed; for instance, it is plausible to imagine all three papers presented here working in tandem with one another. Furthermore, research incorporating greater knowledge about the target organisation's inventory is possible; for instance, staggering patches based not only on vulnerability severity but also in a manner that does not result in the organisation's network resources being overloaded.

4.11 CONCLUSION

Among the different security automation domains, we have identified themes and suggested areas for future research. One of the top recurring themes appear to be the lack of practical implementations, i.e., with a high Technology Readiness Level (TRL). We suspect this can be due to a magnitude of different reasons, e.g., unmet performance expectations, insufficient data, subpar deep learning architectures, lack of consensus for a generic data storage and analysis solution that facilitate scalable DML applications, or infancy of research. With our initial survey we highlight future research directions and/or issues holding up further progress in every security automation domain:

- **Malware Detection:** DML applications need to deal with how malware change its statistical properties over time, e.g., due to adversarial methods (concept drift). There is also the issue regarding data sharing to accommodate for advanced malware which are unlikely to be released into the wild, and access to data in general. Moreover, research is needed into defining novel features that are able to represent software, for detection and attribution.
- **Event Management:** Insufficient DML integration with existing security controls limits the extent DML applications can be developed. In regard to operationalisation, management, and routines to facilitate labelled data collection and deep learning model development.
- **Information Management:** DLP systems can be tightly connected to network and endpoint systems and require a deep and wide understanding of systems in general. With the current IT security trends, strengthening data confidentiality such systems are facing reduced data accessibility. This is by no means a problem unique to this domain but have made development of DML applications complicated. Research opportunities therefore exist, in restoring data accessibility by e.g., deeper integration with the underlying OS. However, there are also topics that needs research into the conditions that describe whether any given data contains sensitive information, and how variations of the same data can be

identified regardless of e.g., encoding schemes. As well as how fuzzy or open rules can be represented and verified for compliance when data needed is not directly attainable without additional analysis.

- **Vulnerability Management:** Lack of consensus and access to a public and sufficiently large dataset, have been recognised as a challenge in the field of vulnerability discovery. However, there are attempts that reduce this dependency, by deploying pre-trained language models to, e.g., perform fuzz testing for software scanning to detect vulnerabilities and assist in patching them. We foresee two directions that can be pursued for further research: improving the deep learning architectures or improving datasets and their feature representation.
- **Software Assurance:** Although technology to support DML applications, exist in related domains such as malware detection and vulnerability management. We have not identified efforts that research into problems within this domain but expect such development when multiple DML applications are able to work in tandem.
- **Asset Management:** With the upcoming new wave of assets, referred to as "Industry 4.0." Which include trends of automation and data exchange in manufacturing, as well as mobile devices, IoT platforms, location device technology, 3D printing, smart sensors, augmented reality, wearable computing, and network-enabled robots and machines. We suspect that DML applications can and will aid certain aspects of this future asset management, however which aspects is still an open research question and open literature suggest the need to explore industry specific use cases.
- **Licence Management:** Same future research directions as asset management applies here, considering that Software Asset Management (SAM) takes licensing into account.
- **Network Management:** Moving Target Defense (MTD) is an emerging area of research that stands to greatly benefit from AI-driven approaches.
- **Configuration Management:** We expect techniques tied to MTD research, can benefit configuration management capabilities.
- **Patch Management:** We've identified research that tackle certain issues such as: dynamically scheduling patches in a risk-aware manner, automated vulnerability remediation analysis, and locating vulnerability mitigation information in the event where a software patch is not yet available. However, none have attempted to incorporate these to a single model and thus create a complete pipeline. This could be a venue to explore for future research.

To conclude, we have not discovered any evidence that indicate that any security domain is by and large done in terms of research for DML applications. All domains have uncharted areas of research that yet is explored, that can and is expected to undergo significant research in the future.

4.12 REFERENCES

[1] Dempsey, K.L., Johnson, L.A., Scholl, M.A., Stine, K.M., Jones, A.C., Orebaugh, A., et al. (2011). Information Security Continuous Monitoring (ISCM) for Federal Information Systems and Organizations.

[2] Verizon (2017). 2017 Data Breach Investigations Report, 10th Edition. http://www.verizonenterprise.com/resources/reports/rp_DBIR_2017_Report_en_xg.pdf Retrieved 8 Jun 2017.

[3] MalwareBytes Labs (2017). 2017 State of Malware Report. Retrieved https://www.malwarebytes.com/pdf/white-papers/stateofmalware.pdf Retrieved 8 Jun 2017.

[4] Christodorescu, M., and Jha, S. (2006). Static Analysis of Executables to Detect Malicious Patterns. DTIC.

[5] Moser, A., Kruegel, C., and Kirda, E. (2007). Limits of Static Analysis for Malware Detection. In ACSAC.

[6] Joint Chiefs of Staff (10 Jul 2012). Chairman of the Joint Chiefs of Staff Manual 6510.01B, Cyber Incident Handling Program. http://disa.mil/Services/DoD-Cloud-Broker/~/media/Files/DISA/Services/Cloud-Broker/m651001.pdf

[7] Gavriluţ, D., Cimpoeşu, M., Anton, D., and Ciortuz, L. (2009). Malware Detection Using Machine Learning. International Multiconference on Computer Science and Information Technology 12 Oct 2009, 735-741. IEEE. https://ieeexplore.ieee.org/iel5/5341930/5352681/05352759.pdf

[8] Souppaya, M., and Scarfone, K. (2013). Guide to Malware Incident Prevention and Handling for Desktops and Laptops. NIST Special Publication. 2013 Jul 22;800:83.

[9] Gibert, D., Mateu, C., and Planes, J. (2020). The Rise of Machine Learning for Detection and Classification of Malware: Research Developments, Trends and Challenges. Journal of Network and Computer Applications, 153, 1 Mar 2020. doi: 10.1016/j.jnca.2019.102526.

[10] McCallam, D., Braun, T., Delucia, M., Shearer, G., Leslie, N., Ritchey., P, Nelson, F., Yu, K., Bowman, E., Mittrick, M., and Jackson, M. (2019). Approaches to Prediction of Cyber Events: Report of the 2017 Specialist Meeting by the North Atlantic Treaty Organization (NATO) Research Group IST-145-RTG. Army Research Lab Aberdeen Proving Ground United States; 1 Jun 2019. https://apps.dtic.mil/sti/citations/AD1074564

[11] Apruzzese, G., Colajanni, M., Ferretti, L., Guido, A., and Marchetti, M. (2018). On the Effectiveness of Machine and Deep Learning for Cyber Security. 2018 10th International Conference on Cyber Conflict (CyCon). IEEE, 2018.

[12] Krych, D.E., and Acosta, J.C. (2020). Hands on Cybersecurity Studies: Uncovering and Decoding Malware Communications with Dshell. CCDC Army Research Laboratory Adelphi United States; 23 Jun 2020. https://apps.dtic.mil/sti/citations/AD1102548

[13] Dauber, E., Caliskan, A., Harang, R., Shearer, G., Weisman, M., Nelson, F., and Greenstadt, R. (2019). Git Blame Who? Stylistic Authorship Attribution of Small, Incomplete Source Code Fragments. Proceedings on Privacy Enhancing Technologies. 1 Jul 2019 (3), 389-408. https://content.sciendo.com/view/journals/popets/2019/3/article-p389.xml

[14] Yu, D., and Li, D. (2010). Deep Learning and Its Applications to Signal and Information Processing [Exploratory dsp]. IEEE Signal Processing Magazine 28(1), 145-154.

[15] Kirat, D., Nataraj, L., Vigna, G., and Manjunath, B.S. (2013). SigMal: A Static Signal Processing Based Malware Triage. In ACSAC.

[16] Nataraj, L., Kirat, D., Manjunath, B., and Vigna, G. (2013). SARVAM: Search and Retrieval of Malware. In ACSAC Workshop on Next Generation Malware Attacks and Defenses.

[17] Nataraj, L., and Manjunath, B.S. (2016). SPAM: Signal Processing to Analyze Malware. In IEEE Signal Processing Magazine.

[18] Markets and Markets (Nov 2019). Malware Analysis Market by Component (Solution (Static Analysis and Dynamic Analysis) and Services), Organization Size (SMEs and Large Enterprises), Deployment (Cloud and On-Premises), Vertical, and Region – Global Forecast to 2024. https://www.marketsandmarkets.com/Market-Reports/malware-analysis-market-108766513.html

[19] Johns, J. (2017). Representation Learning for Malware Classification. Conference on Applied Machine Learning for Information Security.

[20] Thanh, C.T., and Zelinka, I. (2019). A Survey on Artificial Intelligence in Malware as Next-Generation Threats. Mendel (25)2.

[21] Bose, S., Barao, T., and Liu, X. (2020). Explaining AI for Malware Detection: Analysis of Mechanisms of MalConv. 2020 International Joint Conference on Neural Networks (IJCNN). IEEE.

[22] Bragin, T. (Mar 2019). Schema on Write vs. Schema on Read. https://www.elastic.co/blog/schema-on-write-vsschema-on-read

[23] Splunk (2015). Real World Big Data Architecture Splunk, Hadoop, RDBMS. https://www.splunk.com/pdfs/events/govsummit/real_world_big_data_architecture_splunk_hadoop_RDBMS.pdf

[24] Ahmad, Z., Khan, A.S., Shiang, C.W., Abdullah, J., and Ahmad, F. (2021). Network Intrusion Detection System: A Systematic Study of Machine Learning and Deep Learning Approaches. Transactions on Emerging Telecommunications Technologies, page e4150. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/ett.4150

[25] Basnet, R.B., Shash, R., Johnson, C., Walgren, L., and Doleck, T. (2019). Towards Detecting and Classifying Network Intrusion Traffic Using Deep Learning Frameworks. J. Internet Serv. Inf. Secur., 9(4), 1-17.

[26] Sharafaldin, I., Lashkari, A.H., and Ghorbani, A.A. (2018). Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization. In ICISSp, 108-116.

[27] Benazera, E. (Jan 2016). Categorizing Images with Deep Learning into Elasticsearch. https://www.elastic.co/blog/categorizing-images-with-deep-learning-into-elasticsearch

[28] Chockwanich, N., and Visoottiviseth, V. (2019). Intrusion Detection by Deep Learning with Tensorflow. In 2019 21st International Conference on Advanced Communication Technology (ICACT), 654-659, IEEE.

[29] Siren (Aug 2019). Welcome Siren ML: Deep Learning for Elasticsearch (and Anything Else Siren Can Connect To). https://siren.io/welcomesiren-ml-deep-learning-for-elasticsearch-and-much-more/

[30] Splunk (2019). Announcing the Deep Learning Toolkit for Splunk with Tensorflow 2.0, Pytorch, NLP and Jupyter Lab Notebooks. https://conf.splunk.com/files/2019/slides/FN1409.pdf

[31] Splunk (2020). Beyond the Algorithms: ML & Data Science at Splunk. https://conf.splunk.com/files/2020/slides/PLA1821A.pdf

[32] Shah, SA.R., and Issac, B. (2018). Performance Comparison of Intrusion Detection Systems and Application of Machine Learning to Snort System. Future Generation Computer Systems, 80, 157-170. http://www.sciencedirect.com/science/article/pii/S0167739X17323178, doi: 10.1016/j.future.2017.10.016.

[33] Sharma, A. Zaidi, A., Singh, R., Jain, S., and Sahoo. A. (2013). Optimization of Svm Classifier Using Firefly Algorithm. In 2013 IEEE Second International Conference on Image Information Processing (ICIIP-2013), 98-202. IEEE3.

[34] Brasetvik, A. (Sep 2013). Elasticsearch from the Bottom up, Part 1. https://www.elastic.co/blog/found-elasticsearchfrom-the-bottom-up

[35] Splunk (2020). How Indexing Works. https://docs.splunk.com/Documentation/Splunk/8.1.0/Indexer/Howindexingworks

[36] Nayak, S.K. and Ojha, A.C. (2020). Data Leakage Detection and Prevention: Review and Research Directions. In D. Swain, P.K. Pattnaik, and P.K. Gupta (Eds.) Machine Learning and Information Processing, 203-212, Springer, Singapore.

[37] Trieu, Q., Tran, T., Tran, M., and Tran, N. (2017). Document Sensitivity Classification for Data Leakage Prevention with Twitter-Based Document Embedding and Query Expansion. In 13th International Conference on Computational Intelligence and Security (CIS), 537-542.

[38] Cheng, L., Liu, F., and Yao, D. (2017). Enterprise Data Breach: Causes, Challenges, Prevention, and Future Directions. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 7.

[39] Ong, J., Qiao, M., Routray, R., and Raphael, R. (2017). Context-Aware Data Loss Prevention for Cloud Storage Services. IEEE 10th International Conference on Cloud Computing (CLOUD), 399-406.

[40] Tarawneh, A., Hassanat, A.B., Chetverikov, D., Lendak, I., and Verma, C. (2019). Invoice Classification Using Deep Features and Machine Learning Techniques. IEEE Jordan International Joint Conference on Electrical Engineering and Information Technology (JEEIT), 855-859.

[41] Committee on National Security Systems (Apr 2015). Committee on National Security Systems (CNSS) Glossary. CNSSI No. 4009.

[42] Dempsey, K., Eavy, P., Moore, G., and Takamura, E. (Apr 2020). Automation Support for Security Control Assessments: Software Vulnerability Management. National Institute of Standards and Technology Interagency or Internal Report (NISTIR) 8011 Volume 4, National Institute of Standards and Technology, Gaithersburg, MD. Doi: 10.6028/NIST.IR.8011-4.

[43] Foreman, P. (2019). Vulnerability Management, Second Edition, CRC Press, Boca Raton, FL.

[44] Dempsey, K., Eavy, P. and Moore, G. (Jun 2017). Automation Support for Security Control Assessments: Volume 1: Overview. National Institute of Standards and Technology Interagency or Internal Report (NISTIR) 8011 Volume 1, National Institute of Standards and Technology, Gaithersburg, MD. Doi: 10.6028/NIST.IR.8011-1.

[45] Mell, P., Bergeron, T. and Henning, D. (Nov 2005). Creating a Patch and Vulnerability Management Program. NIST Special Publication 800-40 Version 2.0, National Institute of Standards and Technology. (*Note: SP 800-40 is superseded by the publication of SP 800-40 Revision 3, Jul 2013.*)

[46] EC-Council (2018). Module 5: Vulnerability Analysis, Ethical Hacking and Countermeasures Version 10 (Certified Ethical Hacker Version 10 Courseware), EC-Council, Albuquerque, NM.

[47] Kandek, W. (2015). Vulnerability Management for Dummies, Second Edition, John Wiley & Sons Ltd, Chichester.

[48] Dempsey, K., Chawla, N.S., Johnson, A. Johnston, R., Jones, A.C., Orebaugh, A., Scholl, M. and Stine, K. (Sep 2011). Information Security Continuous Monitoring (ISCM) for Federal Information Systems and Organizations. NIST Special Publication (SP) 800-137, National Institute of Standards and Technology, Gaithersburg, MD.

[49] Wheeler, D.A. (Oct 2017). The Software State-of-the-Art Resource, Journal of Cyber Security and Information Systems, 5(3), 48-53.

[50] Zeng, P., Lin, G., Pan, L., Tai, Y., and Zhang, J. Software Vulnerability Analysis and Discovery Using Deep Learning Techniques: A Survey, in IEEE Access, vol. 8, 197158-197172, 2020, doi: 10.1109/ACCESS.2020.3034766.

[51] Heidbrink, S., Rodhouse, K.N., Dunlavy, D.M., Cooper, A., and Zhou, X. Joint Analysis of Program Data Representations using Machine Learning for Improved Software Assurance and Development Capabilities, Sandia Report SAND2020-10150, Sandia National Laboratories, Albuquerque, NM, September 2020.

[52] Wang, Y., Jia, P., Liu, L., Huang, C., and Liu, Z. A Systematic Review of Fuzzing Based on Machine Learning Techniques. PLOS ONE 15(8): e0237749, 2020. Doi: 10.1371/journal.pone.0237749.

[53] Balbix (2018). Balbix BreachControl Product Sheet. https://www.balbix.com/app/uploads/Product-Sheet-Balbix-BreachControl-Platform.pdf (Retrieved 1 Jun 2022).

[54] Qualys (2020). Qualys VMDR eBook. https://www.qualys.com/docs/vmdr-ebook.pdf (Retrieved 1 Jun 2022).

[55] Tenable (2019). 3 Things You Need to Know About Prioritizing Vulnerabilities. https://www.tenable.com/whitepapers/3-things-you-need-to-know-about-prioritizing-vulnerabilities (Retrieved 1 Jun 2022).

[56] Jacobs, J., Romanosky, S., Adjerid, I., and Baker, W. (2020). Improving Vulnerability Remediation through Better Exploit Prediction. Journal of Cybersecurity, 6(1), tyaa015.

[57] Gillula, J., Cardozo, N., and Eckersley, P. Does DARPA's Cyber Grand Challenge Need A Safety Protocol?, Electronic Frontier Foundation Deeplinks Blog, 4 August 2016. Available online: https://www.eff.org/deeplinks/2016/08/darpa-cgc-safety-protocol

[58] Ji, T., Wu, Y., Wang, C., Xi Zhang and Zhongru Wang. The Coming Era of AlphaHacking?: A Survey of Automatic Software Vulnerability Detection, Exploitation and Patching Techniques. 2018 IEEE Third International Conference on Data Science in Cyberspace (DSC), Guangzhou, 2018, 53-60, doi: 10.1109/DSC.2018.00017.

[59] Committee on National Security Systems (CNSS) Glossary. Committee on National Security Systems, CNSSI No. 4009, April 2015.

[60] Software Assurance and Software Safety Standard, NASA Technical Standard NASA-STD 8739.8A, National Aeronautics and Space Administration, 2020.

[61] Hong Fong, E.K. and Wheeler, D.A. The Software Assurance State-of-the-Art Resource (SOAR). Institute for Defense Analyses, August 2017.

[62] Knight, W. (Apr 2021). Now for AI's Latest Trick: Writing Computer Code. Wired. https://www.wired.com/story/ai-latest-trick-writing-computer-code

[63] Source AI. https://www.sourceai.dev (Retrieved 12 Jan 2022).

[64] International Organization for Standardization. ISO/IEC 27000: 2018: Information Technology – Security Techniques – Information Security Management Systems – Overview and Vocabulary. International Organization for Standardization. (2018)

[65] Arraj, V. (2010). ITIL®: The Basics. Buckinghamshire, UK.

[66] SANS (Mar 2013). CSIS: 20 Critical Security Controls Version 4.1. http://www.sans.org/critical-security-controls/

[67] Gartner (Oct 2020). Critical Capabilities for IT Service Management Tools. https://www.gartner.com/en/documents/3991510

[68] Gartner (Jul 2020). Critical Capabilities for Software Asset Management Tools. https://www.gartner.com/en/documents/3988121

[69] Yahoo! (Nov 2020). Global IT Asset Management (ITAM) Software Industry. https://www.yahoo.com/lifestyle/global-asset-management-itam-software-181500932.html

[70] Markets and Markets (Dec 2020). Cloud ITSM Market Worth $12.2 Billion by 2025. https://www.marketsandmarkets.com/PressReleases/cloud-based-itsm.asp

[71] Markets and Markets (Mar 2018). Software Asset Management Market by Solution (License Management, Audit and Compliance Management, Software Discovery, Optimization, and Metering), Service, Deployment Type, Organization Size, Industry Vertical, and Region – Global Forecast to 2022. https://www.marketsandmarkets.com/Market-Reports/software-asset-management-market-235932482.html

[72] Gupta, R., Prasad, K.H., and Mohania, M. (Jun 2008). Automating ITSM Incident Management Process. In 2008 International Conference on Autonomic Computing 141-150. IEEE.

[73] Al-Hawari, F., and Barham, H. (2021). A Machine Learning Based Help Desk System for IT Service Management. Journal of King Saud University-Computer and Information Sciences, 33(6), 702-718.

[74] Chen, T.M. (Apr 2010). Survey of Cyber Security Issues in Smart Grids. In Cyber Security, Situation Management, and Impact Assessment II; and Visual Analytics for Homeland Defense and Security II, Vol. 7709, 77090D. International Society for Optics and Photonics.

[75] Theron, P., Kott, A., Drašar, M., Rzadca, K., LeBlanc, B., Pihelgas, M. et al. (May 2018). Towards an Active, Autonomous and Intelligent Cyber Defense of Military Systems: The NATO AICA Reference Architecture. In 2018 International Conference on Military Communications and Information Systems (ICMCIS), 1-9. IEEE.

[76] Campos, J., Sharma, P., Gabiria, U.G., Jantunen, E., and Baglee, D. (2017). A Big Data Analytical Architecture for the Asset Management. Procedia CIRP, 64, 369-374.

[77] Lasi, H., Fettke, P., Kemper, H.G., Feld, T., and Hoffmann, M. (2014). Industry 4.0. Business & Information Systems Engineering, 6(4), 239-242.

[78] Xu, H., Yu, W., Griffith, D., and Golmie, N. (2018). A Survey on Industrial Internet of Things: A Cyber-Physical Systems Perspective. IEEE Access, 6, 78238-78259.

[79] Rødseth, H., Eleftheriadis, R.J., Li, Z., and Li, J. (Nov 2019). Smart Maintenance in Asset Management – Application with Deep Learning. In International Workshop of Advanced Manufacturing and Automation, 608-615. Springer, Singapore.

[80] Nel, C.B.H., and Jooste, J.L. (2016). A Technologically-Driven Asset Management Approach to Managing Physical Assets – A Literature Review and Research Agenda for 'Smart' Asset Management. South African Journal of Industrial Engineering, 27(4), 50-65.

[81] Villalba-Díez, J., Molina, M., Ordieres-Meré, J., Sun, S., Schmidt, D., and Wellbrock, W. (2020). Geometric Deep Lean Learning: Deep Learning in Industry 4.0 Cyber-Physical Complex Networks. Sensors, 20(3), 763.

[82] Akkiraju, R., Sinha, V., Xu, A., Mahmud, J., Gundecha, P., Liu, Z. et al. (Sep 2020). Characterizing Machine Learning Processes: A Maturity Framework. In International Conference on Business Process Management, 17-31. Springer, Cham.

[83] Ansari, F., Nixdorf, S., and Sihn, W. (2020). Insurability of Cyber Physical Production Systems: How Does Digital Twin Improve Predictability of Failure Risk? IFAC-PapersOnLine, 53(3), 295-300.

[84] Punathil, G., Vipin, M.V., Paradani, S.C., and Kannan, V. (2020). U.S. Patent Application No. 16/295,640.

[85] Berman, D.S., Buczak, A.L., Chavis, J.S., and Corbett, C.L. (2019). A Survey of Deep Learning Methods for Cyber Security. Information, 10(4), 122.

[86] Li, J.H. (2018). Cyber Security Meets Artificial Intelligence: A Survey. Frontiers of Information Technology & Electronic Engineering, 19(12), 1462-1474.

[87] Xin, Y., Kong, L., Liu, Z., Chen, Y., Li, Y., Zhu, H. et al. (2018). Machine Learning and Deep Learning Methods for Cybersecurity. IEEE Access, 6, 35365-35381.

[88] Fadlullah, Z.M., Tang, F., Mao, B., Kato, N., Akashi, O., Inoue, T., and Mizutani, K. (2017). State-of-the-Art Deep Learning: Evolving Machine Intelligence Toward Tomorrow's Intelligent Network Traffic Control Systems. IEEE Communications Surveys & Tutorials, 19(4), 2432-2455.

[89] Chowdhary, A., Huang, D., Sabur, A., Vadnere, N., Kang, M., and Montrose, B. (2021). SDN-Based Moving Target Defense Using Multi-Agent Reinforcement Learning. Proceedings of the 2021 1st International Conference on Autonomous Intelligent Cyber Defense Agents.

[90] Risto, V., Blumbergs, B., and Kont, M. (2018). An Unsupervised Framework for Detecting Anomalous Messages from Syslog Log Files. NOMS 2018-2018 IEEE/IFIP Network Operations and Management Symposium. IEEE.

[91] Sengupta, S., Chowdhary, A., Sabur, A., Alshamrani, A., Huang, D., and Kambhampati, S. (2020). A Survey of Moving Target Defenses for Network Security. IEEE Communications Surveys & Tutorials, 22(3), 1909-1941.

[92] Zhang, C., Wang, X., Li, F., He, Q., and Huang, M. (2018). Deep Learning-Based Network Application Classification for SDN. Transactions on Emerging Telecommunications Technologies, 29(5), e3302.

[93] LWN.net (Sep 2018). Machine Learning and Stable Kernels, https://lwn.net/Articles/764647/

[94] Yue, Z. and Stewart, C. (2020). Poster: Configuration Management for Internet Services at the Edge: A Data-Driven Approach. 2020 IEEE/ACM Symposium on Edge Computing (SEC). IEEE.

[95] Bozorgi, M., Saul, L.K., Savage, S., and Voelker, G.M. (2010). Beyond Heuristics: Learning to Classify Vulnerabilities and Predict Exploits. In Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 105-114.

[96] Xiao, C., Sarabi, A., Liu, Y., Li, B., Liu, M., and Dumitras. T. (2018). From Patching Delays to Infection Symptoms: Using Risk Profiles for an Early Discovery of Vulnerabilities Exploited in the Wild. In 27th fUSENIX} Security Symposium (fUSENIXg Security 18), 903-918.

[97] Zhang, F., and Li, Q. (2020). Dynamic Risk-Aware Patch Scheduling. In 2020 IEEE Conference on Communications and Network Security (CNS), 1-9. IEEE.

[98] Zhang, F., Hu, P., McClanahan, L., and Li, Q. (2020). A Machine Learning-Based Approach for Automated Vulnerability Remediation Analysis. In 2020 IEEE Conference on Communications and Network Security (CNS), 1-9. IEEE.

[99] McClanahan, K., and Li, O. (2020). Automatically Locating Mitigation Information for Security Vulnerabilities. In 2020 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm), 1-7. IEEE.

Chapter 5 – CHALLENGES IN DEEP LEARNING

5.1 ADVERSARIAL ATTACKS

(The taxonomy and terminology used in this section are employed based on the NIST report [1] with minor extensions from the paper of Shafee and Awaad [2].)

The data-driven approach of Machine Learning poses some vulnerabilities in training and testing (inference) phases of ML operations. These vulnerabilities include the possibility of adversarial manipulation of training data, and the potential for adversarial exploitation of the model to affect the performance adversely. There is a research field called Adversarial Machine Learning (AML), which concerns with the design of ML algorithms that can withstand security challenges, the study of the capabilities of attackers, and the comprehension of attack consequences. AML is also interested in the attacks against Deep Learning models.

The stages in the ML pipeline define the targets of these adversarial attacks such as the *Physical Domain* of input sensors or output actions, the *Digital Representation* for pre-processing, and the ML Model. Most research in AML has focused on ML Models, especially on Supervised Learning systems.

Adversarial techniques used for performing attacks against previously stated targets may be applicable to the training or testing (inference) phases of ML operation.

1) **Training Phase Attacks:** They attempt to obtain or manipulate the training data or model itself.
    a) **Data Access Attacks:** The adversary has access to some amount of data used for training and he can use them to create a substitute model.
    b) **Poisoning (Causative) Attacks:** The adversary can alter the data or model indirectly or directly.
        i) **Indirect Poisoning**: The adversary poisons the data before pre-processing because he cannot access the pre-processed data used by the target model.
        ii) **Direct Poisoning:**
            (a) **Data Injection:** The adversary poisons the original training data by adding adversarial samples. Thereby, he can corrupt the target model by changing the underlying data distribution but without changing features and labels of the original training data.
            (b) **Data Manipulation:** The adversary poisons the original training data by modifying output labels or data itself.
            (c) **Logic Corruption:** The adversary tampers with the learning algorithm. Therefore, the learning process and the model itself is altered.
2) **Testing Phase (Exploratory) Attacks:** Adversarial attacks at the testing time do not tamper with the target model or training data but rather forces the model to produce incorrect outputs. The effectiveness of such attacks is determined mainly by the amount of information available to the adversary about the model.
    a) **Evasion Attacks:** The adversary tries to evade the model by generating adversarial samples as inputs during testing phase.

b) **Oracle Attacks:** The adversary collects and infers information about the model or training data. Even the adversary has no direct knowledge of the model itself, he can observe the model's outputs by presenting the model with inputs and train a substitute model with obtained input-output pairs. This substitute model can also be used for generating adversarial examples for Evasion Attacks.

i) **Model Extraction:** The adversary extracts the parameters (weights) of the model by observing the model's predictions.

ii) **Model Inversion:** The adversary tries to reconstruct pre-processed training data (feature vectors).

iii) **Membership Inference:** The adversary infers whether a given sample exists in the training dataset. When the adversary has complete knowledge of a sample and can reach the knowledge that the sample is used to train the target model, this indicates information leakage through the model.

iv) **Property Inference:** The adversary infers properties of the training dataset, e.g., the properties of the population which the training data was sampled from. Inferring whether there is an Indian accent in a voice training dataset that is used to train a speech recognition model is an example of property inference attack.

v) **Reconstruction Attack:** The adversary tries to reconstruct raw data using knowledge of feature vectors.

The testing phase adversary attacks are also defined accordingly to the amount of information available to the adversary about the model:

1) **Black Box Attacks:** The adversary has no knowledge about the model. He has input-output samples of training data or input-output pairs obtained using the target model.

2) **Grey Box Attacks:** The adversary has partial knowledge about the model, such as the model architecture, parameter (weight) values, training method (loss function), and training data.

3) **White Box Attacks:** The adversary has complete knowledge of the model, including architecture, parameters (weights), training methods, and training data.

Depending on whether the adversary attacks in training phase or testing phase, the countermeasure techniques are characterised:

1) **Countermeasure techniques against Training Attacks:**

a) **Data Encryption:** Encrypting data is a traditional access control measure against Data Access Attacks.

b) **Data Sanitisation:** This method identifies adversarial examples and treats them as outliers to detect and remove them from the poisoned training dataset.

c) **Robust Statistics:** This technique use constraints and regularisation techniques to provide a rectified learning model which is less sensitive to outlying training samples, rather than detecting the adversarial examples.

2) **Countermeasure techniques against Testing Attacks:**

a) **Robustness Improvements:** These techniques are deployed in the Training Phase. The adversary may defeat these defence techniques by launching Data Access or Oracle Attacks to obtain input-output pairings. These pairings are used to train a subsequent model, and this model can then be used as a White Box to craft adversarial examples. It can be difficult to defend against Evasion Attacks by an adversary capable of creating a substitute model.

i) **Adversarial Training:** The defender alters the training data by injecting inputs with adversarial examples. These examples are correctly labelled and help to minimise the classification errors caused by adversarial examples. However, this defence is useful if the adversarial examples injected for training are same kind of data exploited by an adversary. Since the technique does not generalise across different attack strategies, the model will be vulnerable to new/unknown attacks [3], [4].

ii) **Gradient Masking:** This technique reduces the model's sensitivity to small perturbations in inputs by minimising first order derivatives during the learning phase [5].

iii) **Defensive Distillation:** This technique aims to improve the resilience of a model to adversarial examples by using the knowledge of output probabilities extracted from itself. A new distilled model generalises better, and it is more robust to adversarial perturbations. This technique is useful for preventing white box attacks, but not good at preventing the black box attacks [6].

iv) **Ensemble Methods:** Multiple classifiers are trained and combined to improve robustness [7], [8].

v) **Feature Squeezing:** This technique smooths inputs to detect adversarial examples. The technique merges the samples that correspond to many different feature vectors to a single sample. Then, it compares the predictions on the original input and squeezed input and rejects adversarial examples by measuring the difference among predictions and checking them according to a threshold value [9].

vi) **Autoencoders (Detectors/Reformers):** This technique takes an input, decides whether it is adversarial or not, and change it adequately close to normal examples. This technique is called MagNet and use autoencoders to detect the adversarial examples (encoder) and reconstruct examples which are close to normal examples (decoder) [10].

b) **Differential Privacy:** It is a privacy formulation which provides a mathematical guarantee of privacy protection. The technique includes injecting a certain amount of random noise to training dataset, the model parameters (weights) during the iterations of the training algorithm, or model outputs to provide Differential Privacy. DP eliminates any potential attacks an adversary may distinguish a particular record from other records, or a particular property associated with a previously identified record, or know whether any particular record is in the dataset. It also prevents the adversary with white box access to the model from collecting any information about the model parameters [10], [12].

c) **Homomorphic Encryption:** This technique encrypts data in a form that allows computations to be carried out without decrypting data. The output prediction result will not be revealed except to the party that owns the decryption key [11], [12].

5.2 INTERPRETABLE/EXPLAINABLE AI

Artificial Intelligence (AI) has been used for many times because of their unprecedented performance when learning to solve increasingly complex computational tasks. Since it is also commonly used for decisions which affects humans' lives such as medicine, law, or defence, there comes out a need for an explanation or a justification of why such AI system has come to this conclusion.

While traditional models such as decision trees, linear and logistic regression allows for a certain degree of interpretability through the analysis of feature weights; deep neural networks are much opaquer and remains more of a black box. Furthermore, there appears to exist an inverse relationship between the performance of a machine learning algorithm and the ease of interpreting the trained model as shown in Figure 5-1.

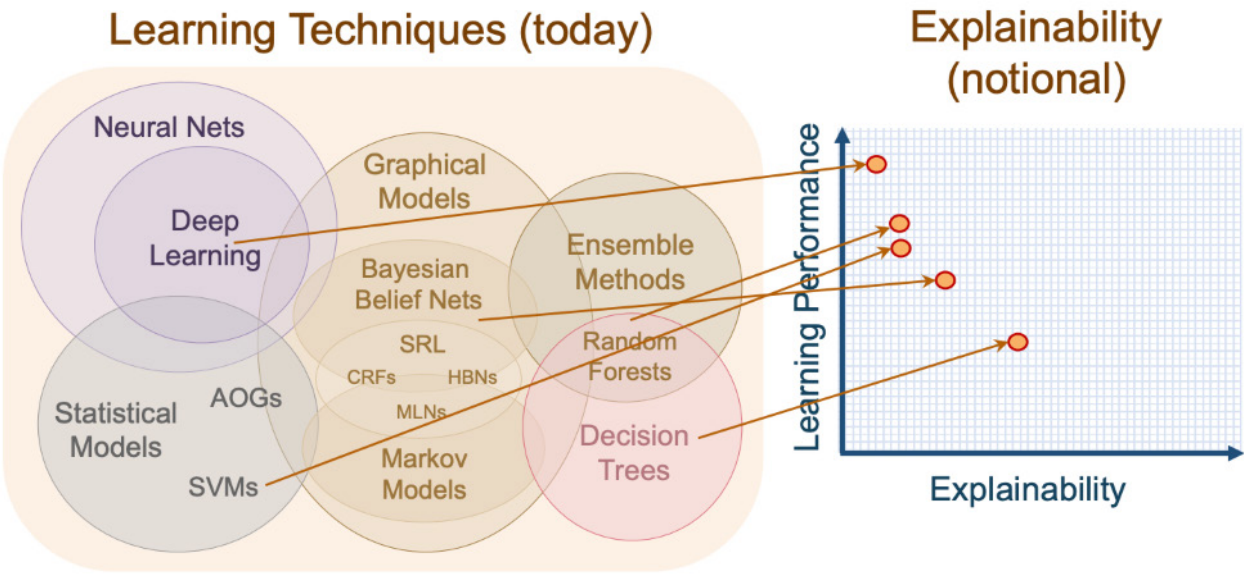


Figure 5-1: Relationship Between Learning Techniques and their Explainabilities.

In 2017, DARPA launched the Explainable Artificial Intelligence (XAI) program to address the issue of interpretability in the context of data analytics (for intelligence analysts) and as well for future autonomous systems that utilises reinforcement learning. In the DARPA's report, a suite is proposed to create such ML techniques that produce more explainable models while maintaining a high level of learning performance, e.g., prediction accuracy, and enable humans to understand, trust, and manage the emerging AI systems [13].

The literature makes a distinction between the models interpretable by design and explainable by means of external techniques. DL models are not interpretable by design; therefore, studies concentrate on external XAI techniques and hybrid methods. Arrieta et al. explains all the details of the techniques and hybrid methods that work for different types of DL models. In addition, they addressed some issues about the trade-off between interpretability and accuracy, the objectivity and unambiguity of explanations and conveying explanations which require non-technical expertise [14].

5.3 HYPERPARAMETER TUNING

Hyperparameters are the properties which control the behaviour of learning process, and they should be configured before training a model, instead of model parameters which are learned during training, e.g., weights and biases. They are important because they can have a significant impact on the performance of the model being trained. There are several techniques for finding optimal hyperparameter values such as:

- **Manual Search:** The developer chooses the values of hyperparameters based on his/her judgement and experience and decides the optimal values with trial-and-error.
- **Grid Search:** The developer defines a list of hyperparameters whose optimal values will be found. Then, he defines the range of possible values for these hyperparameters. At the end, the algorithm searches for all the possible configurations and gives the best values works for given hyperparameters. It is recommended to use when the number of hyperparameters are less than four. Even if it guarantees to find the best configuration, it is not preferable to use because it is computationally expensive.

- **Random Search:** The only difference from Grid Search is that the algorithm searches the configuration space by choosing values randomly. It is better to use when the number of hyperparameters are higher than four and it gives better results in less iterations with respect to Grid Search.
- **Bayesian Optimisation:** This technique builds a probabilistic model which tries to find optimal hyperparameter values by finding a function that learns the mapping from hyperparameter configuration to the metric of interest. It is stated that this technique gives better results in fewer evaluations than Grid Search and random search.
- **Evolutionary Algorithms / Genetic Algorithms:** Genetic algorithms can be seen as random search followed by result-driven search based on the best previous results. The details of the technique can be found in Young et al.'s paper [15].
- **Population-based Training:** This technique is a hybrid of Random Search and Hand-Tuning. It starts by training many neural networks in parallel with random hyperparameters just like Random Search. However, rather than training the networks independently, it uses knowledge from the rest of the population to refine the hyperparameters and guide computational resources to promising models, as inspired by genetic algorithms. This process of exploiting and exploring is repeated on a regular basis as the neural network population is trained. PBT can exploit good hyperparameters easily, devote more training time to promising models, and, most importantly, can adapt hyperparameter values during training, resulting in automated learning of the best configurations. The details can be found in Jaderberg et al.'s paper [16].

In addition, there are also techniques which are specific to the type of hyperparameter. Hyperparameters for Deep Neural Networks and the techniques for finding their optimal value can be listed as:

- **Learning rate:** It controls the amount that the learning algorithm updates the weights each learning iteration. It is considered as the most important hyperparameter to configure. Choosing the learning rate is challenging. If it is too small, the training process will be long and might get stuck. If it is too large, it will try to learn the model parameters too fast, and it may result in an unstable training process and a suboptimal solution. There are several ways for finding the optimal learning rate:
  - **Learning Rate Annealing:** It recommends starting with a relatively high learning rate and then gradually decaying the learning rate during training. The most popular form is step (fixed) decay which the learning rate is reduced by some percentage after a set number of training epochs. There are other forms such as exponential decay and 1/t decay.
  - **Cyclical Learning Rate:** It recommends a schedule for updating the learning rate varying between two bound values. Unlike annealing, the learning rate is updated according to a triangular update rule. The cycle length is the number of iterations until the learning rate starts with initial value, increases up to a bound value, and then decreases down to the initial value. The number of iterations during the ascent and descent is same. This technique can be used in conjunction with step decay, i.e., the upper bound will be reduced by some percentage, and exponential decay, i.e., the upper bound will be reduced exponentially [17].
  - **Stochastic Gradient Descent with Warm Restarts:** It is similar to cyclical learning rate. The annealing schedule is combined with periodic restarts to the initial high learning rate.
- **Number of epochs:** It controls the number of times that the learning algorithm will completely passes through the entire training dataset. More the epoch number is, better the accuracy of the model will be, to some point. At some point, the model will stop improving accuracy. There is no analytical way to choose the optimal value a priori, general techniques can be used.

- **Batch size:** It controls the number of samples to work through before the model's internal parameters are updated. The model converges quickly, and the algorithm requires less memory, when the batch size is chosen small. But if the batch size is chosen too small, then the accuracy of predictions will be low. There is no analytical way to choose the optimal value a priori, general techniques can be used.
- **Number of hidden layers:** It controls the representational capacity of the network. More layers are often good for learning more complex representations with relatively higher accuracy. There is no analytical way to choose the optimal value a priori, general techniques can be used.
- **Number of units in a hidden layer:** It controls the learning capacity of the network. Small number of units may cause underfitting because the model lacks complexity. By contrast, too many units may result in overfitting and increase training time. There is no analytical way to choose the optimal value a priori, general techniques can be used. However, there is a regularisation technique called dropout which prevents the large networks from overfitting because of high number of hidden units. The idea is to randomly drop units from the neural network, i.e., cut their connections with the rest of the network, during training.
- **Activation functions:** The activation function used in hidden layers controls how well the model learns the training dataset, whereas the activation function used in output layer defines the type of predictions the model can make.
    - Differentiable nonlinear activation functions are used in hidden layers, and the most used ones are Rectified Linear (ReLU), Sigmoid and Hyperbolic Tangent. ReLU is less sensitive to vanishing gradient problem. The type of network architecture often specifies the type of activation function used.
        - ReLU is mostly used for Multilayer Perceptrons and Convolutional Neural Networks.
        - Recurrent Neural Networks commonly use Sigmoid and Hyperbolic Tangent.
    - There are three activation functions mostly used for output layer: Linear, Sigmoid and Softmax. The type of prediction problem, e.g., classification or regression, is critical to choose the best activation function.
        - For regression problems, linear activation function should be chosen.
        - For binary classification (two mutually exclusive classes) and multilabel classification (two or more mutually inclusive classes), sigmoid activation function should be chosen.
        - For multiclass classification (more than two mutually exclusive classes), Softmax activation function should be chosen.
- **Weight Initialisation:**
    - When using the ReLU function for hidden layers, it is advisable to use a "He Normal" or "He Uniform" weight initialisation and scale input data to the range 0-1 [18].
    - When using the Sigmoid function for hidden layers, it is advisable to use a "Xavier normal" or "Xavier uniform" weight initialisation and scale input data to the range 0-1 [19].
    - When using the Hyperbolic Tangent function for hidden layers, it is advisable to use a "Xavier normal" or "Xavier uniform" weight initialisation and scale input data to the range -1 to 1[19].

5.4 INTEROPERABILITY CHALLENGES

Syntactic (Framework) Interoperability: In 2017, the Open Neural Network eXchange (ONNX) format was created as a community-driven open-source standard for representing deep learning and traditional machine learning models. ONNX assists in overcoming the problem of hardware dependence in AI models and allows the deployment of the same AI models to multiple HW accelerated targets. Models from many frameworks such as TensorFlow, PyTorch, MATLAB, etc. can be exported or converted to the standard ONNX format. Then, the models in the ONNX format can be run on a variety of platforms and devices (Figure 5-2).

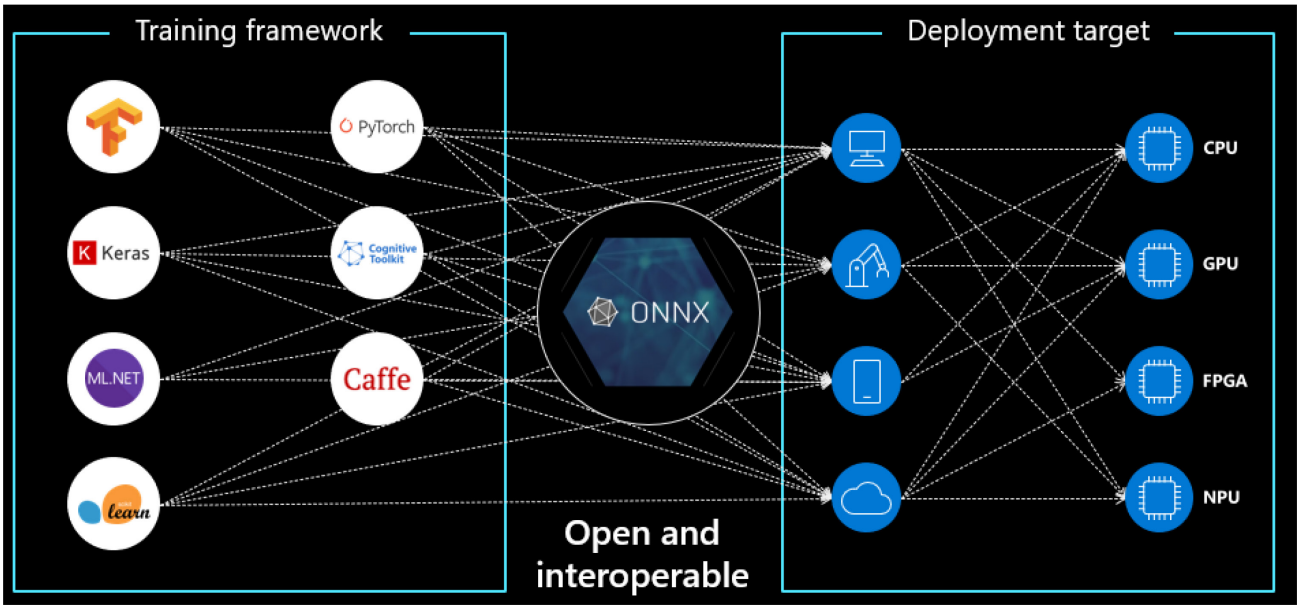


Figure 5-2: Frameworks and Platforms to Which ONNX is Applicable.

Semantic Interoperability: When data comes from a mix of sources that do not mean the same thing, it is impossible to learn the trends, forecasts, or anomalies. Semantic interoperability is the ability of computer systems to exchange information with unambiguous meaning. To this end, whether the data has been aggregated from a single source or heterogeneous sources, high-quality human-annotated data sets are needed to accurately train machine learning models.

One of the best practices to achieve semantic interoperability is to use archetypes. An archetype is a data format specification that should provide the most usable and full details possible. It provides the shared meaning of data. Semantic interoperability in AI systems requires that archetypes to be high-quality, evidence-based, structured, and designed by domain experts [20].

5.5 DATA DEPENDENCE

As compared to conventional machine learning approaches, deep learning relies heavily on vast training data because it needs a huge amount of data to understand latent patterns of data. However, in some domains, insufficient training data is unavoidable. Data collection is complicated and costly, making it incredibly difficult to build a large-scale, high-quality annotated dataset. Transfer learning is an important tool that can be used

against the problem of insufficient training data. It tries to transfer the knowledge from the source domain (training data) to the target domain (test data) by relaxing the assumption that the training data and test data must be independent and identically distributed, i.e., the samples are mutually independent and drawn from same probability distribution. In this way, the model in the target domain does not need to be trained from scratch.

Deep transfer learning studies how to effectively transfer knowledge by deep neural networks. Based on the techniques used, Tan et al. [21] classifies deep transfer learning into four categories: instances-based, mapping-based, network-based and adversarial-based:

1) **Instance-based Deep Transfer Learning:** Instances in the source domain that differ from those in the target domain are filtered out and re-weighted to form a distribution close to that of target domain. The model is trained with the re-weighted instances from source domain and origin instances from target domain.
2) **Mapping-based Deep Transfer Learning:** Instances from the source domain and target domain are mapped into a new data space. Then, all instances in the new data space are used as training set.
3) **Network-based Deep Transfer Learning:** Generally, the layers of a network before the last fully connected layer are regarded as feature extractor, and the last fully connected layer is considered as classifier/label predictor. The network is trained in source domain with large-scale training dataset. Then, the structure and weights of feature extractor of pre-trained network will be transferred to the network which will be used in the target domain.
4) **Adversarial-based Deep Transfer Learning:** This group of techniques is inspired by Generative Adversarial Nets (GAN) (Figure 5-3). An additional discriminator network called domain classifier takes extracted features from both source and target domains and tries to discriminate the origin of the features. All source and target data are fed to the feature extractor. The aim of feature extractor is to cheat the domain classifier while satisfying the classifier at the same time.

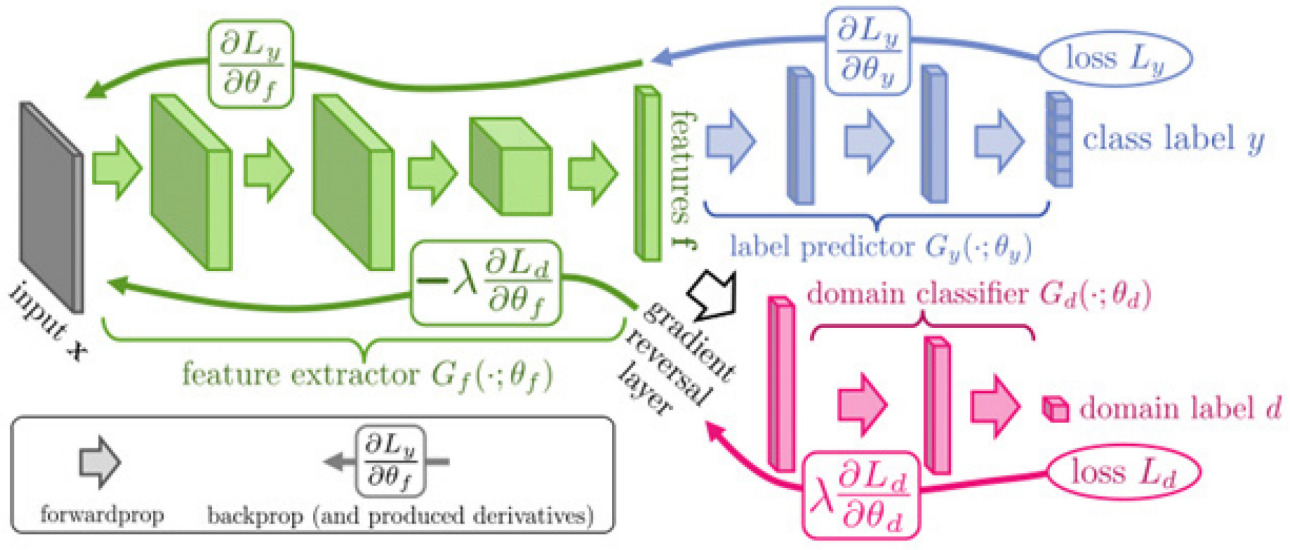


Figure 5-3: Architecture of Adversarial-Base Deep Transfer Learning.

5.6 DATA QUALITY

With low quality data, no matter how strong a machine learning and/or deep learning model is, it will never be able to do what is expected of it. The processes that affect data quality are grouped into three groups: processes that bring data into the database, processes that manipulate data inside the database, and processes that cause accurate data to become inaccurate over time without any modification. The details of the processes that degrades the data quality can be found in Ref. [22].

The process of ensuring its accuracy and consistency before using, importing, or otherwise processing data, is called *data validation*. Nowadays, data is stored in a variety of locations, including relational databases and distributed file systems, and it is available in a variety of formats. Many of these data sources lack accuracy constraints and data quality checks. In addition, most ML models today re-train on a regular basis using newly available data to maintain performance and keep up with changes in real-world data. As a result, since any team and system involved in data processing must deal with data validation in some way, it becomes a tedious and repetitive task. The demand for data validation automation is growing by the day.

One approach is the unit-test approach, proposed by Amazon Research [23]. The system provides users a declarative API which allows users to specify constraints and checks on their datasets. When the validation fails, these checks produce errors or warnings during execution. There are predefined constraints available to the users for checking data in the dimensions of completeness, consistency, and statistics. After the constraints have been defined, the system translates them into actual computable metrics. Then, the system computes the metrics and evaluates the results, and subsequently, reports which constraints succeeded, and which failed, including the constraint on which metric failed and which value caused the failure. Since new data is constantly coming, the method employs a recursive computation methodology that only considers new data since the last time step to update the metrics incrementally. Furthermore, the system proposes constraints for datasets automatically. This is achieved through the application of heuristics and a machine learning model.

Another approach is data schema-based approach, proposed by Google Research [24]. The requirements for correct data are codified in the data schema. The proposed system takes the ingested data, passes it through data validation, and sends the data to a training algorithm. The data validation system consists of three main components: A Data Analyser that computes predefined set of data statistics sufficient for data validation, a Data Validator that checks for properties of data as specified through a Schema, and a Model Unit Tester that checks for errors in the training code using synthetic data generated through the schema. The system can detect anomalies in a single batch of data (single batch validation), detect significant changes between the training and serving data, or between successive batches of training data (inter-batch validation), and find assumptions in the training code that are not reflected in the data (model testing).

5.7 CONTEXT AWARENESS

Although deep learning has enabled breaking features down progressively to identify certain characteristics through the use of multiple layers in a neural network, it has a shallow understanding of the context, in which the data origin from, where the context provides the circumstances or elements that makes out a certain event and can convey useful information for its interpretation. As a result, a model may end up being specialised for one or multiple situations recorded in the training data. Therefore, this model could be biased for similar scenarios and thus perform reasonable only on such instances. The model is able to override experience learned from training, in order to adapt to changing environments. However, this ability is constrained. The incentive to research models capable of capturing context, improves the task effectiveness through more robust, resilient, and adaptable deep learning. This enables more cost-effective usage of deep learning.

Initial efforts to remediate the bias issue, started with the work of Bottou and Vapnik [25] by the proposal of local learning. It involves separating the input space into subsets and building models for each subset. The concept itself is not novel but has gained some credibility due to complexities in applications dealing with large datasets [26]. In contrary, Mezouar et al. [27] did not find local models being a worth investment, over their global counterparts for prediction of software defects. Multi-task learning (MTL) [28] is another subfield of machine learning that can be utilised. It separates the input space into multiple tasks and leverage shared information while accounting for their differences. The aim is to improve performance of multiple classification tasks by joint learning and acquiring a shared representation. Suresh et al. [29] attempt to compare the three types of models, in the context of mortality prediction. Their work indicate that a multi-task model can outperform both a global model, as well as local models trained on separate subsets of data on both overall and per-group performance metrics. Unfortunately, it appears that no final consensus has been made on the most appropriate model to capture context. As research on techniques to perform information sharing between task-specific models, adjusting local/global models to new contexts, or how local and global models can be combined, is still active [30], [31].

5.8 CHALLENGES FOR NATO-WIDE "DEEP LEARNING IN CYBER SECURITY" APPLICATIONS

Among all the challenges mentioned above, the members of this RTG were most concerned with the possible ways of sharing knowledge. This chapter discusses the issues with two possible ways: sharing training data or sharing models.

1) **Training Data Sharing.** The data collected from NATO exercises are valuable. It would be very good to be able to make use of them. For data sharing, most probably, a database should be constructed. Semantic interoperability issues may arise when the databases of various allies are joined (See Section 6.4, Semantic Interoperability). To maintain the database's integrity, all allies should re-form their training data around a standardisation and contribute to the database that manner. This is time-consuming and error prone. Furthermore, the quality of the data is critical, and it should be reviewed before being contributed to the database (See Section 6.6). Moreover, this approach is dangerous because if the adversaries reach this database, they can poison the data. (For possible training data targeted attacks and the countermeasure techniques against them, see Section 6.1, Training Phase Attacks.)

2) **Model Sharing.** With the help of syntactic interoperability tools, it is possible to share DL models nowadays. (See Section 6.4, Syntactic Interoperability). It appears much more helpful to share feature extractors among NATO allies using Network-based Transfer Learning so that any ally can apply the derived knowledge on their test data for any task they wish (For the details of transfer learning, see Section 6.5). However, the question is, who will train the model and which data will he use? If storing data in a database is problematic, it may also be troublesome to grant one person/ally access to all NATO exercise data in order to train the model that will be shared. Normally, no such platforms exist that allow everyone to train the same DL model using their own data. However, a decentralised approach called Federated Learning seems to work in this case. It is a distributed machine learning approach in which a number of participants known as *clients* work together to train a certain machine learning model over numerous iterations. Federated learning was first proposed in [32] as a distributed training model executed by a group of mobile devices that exchange local model changes with a central server whose function it is to aggregate these updates to form a global machine learning model. A federated learning scenario consists of one central server and a set of N clients, each with their own local dataset. A subset of clients is initially chosen to get the global state of the shared model in terms of model weights. Then, based on the shared parameters, each of them carries out local computations on its

own dataset. Clients then submit the model updates (i.e., the weights learned locally based on the client's local dataset) to the server which applies these updates to its current global model to generate a new one. Then, the server again shares the global state with the clients, and this procedure is done multiple times until the server determines a particular degree of accuracy. Thus, clients do not have to share their raw data to contribute to the global model, and it will be sufficient to have enough CPU or energy resources to process only the training data that it has.

5.9 REFERENCES

[1] Tabassi, E., Burns, K.J., Hadjimichael, M., Molina-Markham, A.D., and Sexton, J. T. (2019). A Taxonomy and Terminology of Adversarial Machine Learning. National Institute of Standards and Technology – National Cybersecurity Center of Excellence. doi: 10.6028/NIST.IR.8269-draft.

[2] Shafee, A., and Awaad, T.A. (2021). Privacy Attacks Against Deep Learning Models and their Countermeasures. Journal of Systems Architecture, 114. doi: 10.1016/j.sysarc.2020.101940

[3] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout: A Simple Way to Prevent Neural Networks from Overfitting. The Journal of Machine Learning Research, 15(1), 1929-1958.

[4] Goodfellow, I., Shlens, J., and Szegedy, C. (2015). Explaining and Harnessing Adversarial Examples. ICLR 2015. Retrieved from https://arxiv.org/abs/1412.6572

[5] Papernot, N., McDaniel, P., Goodfellow, I., Jha, S., Celik, Z. B., and Swami, A. (2017). Practical Black-Box Attacks against Machine Learning. The 2017 ACM Asia Conference on Computer and Communications Security.

[6] Papernot, N., McDaniel, P., Wu, X., Jha, S., and Swami, A. (2016). Distillation as a Defense to Adversarial Perturbations against Deep Neural Networks. IEEE Symposium on Security and Privacy. IEEE. doi: 10.1109/SP.2016.41.

[7] Biggio, B., and Roli, F. (2018). Wild Patterns: Ten Years After the Rise of Adversarial Machine Learning. Pattern Recognition, 84, 317-331.

[8] Biggio, B., Fumera, G., and Roli, F. (2010). Multiple Classifier Systems for Robust Classifier Design in Adversarial Environments. International Journal of Machine Learning and Cybernetics, 1, 27-41.

[9] Xu, W., Evans, D., and Qi, Y. (2018). Feature Squeezing: Detecting Adversarial Examples in Deep Neural Networks. Network and Distributed Systems Security Symposium (NDSS) 2018. doi: 10.14722/ndss.2018.23198.

[10] Meng, D., and Chen, H. (2017). MagNet: a Two-Pronged Defense against Adversarial Examples. ACM Conference on Computer and Communications Security.

[11] Liu, Q., Li, P., Zhao, W., Cai, W., Yu, S., and Leung, V. C. (2018). A Survey on Security Threats and Defensive Techniques of Machine Learning: A Data Driven View. IEEE Access, 6, 12103-12117. doi: 10.1109/ACCESS.2018.2805680.

[12] Papernot, N., McDaniel, P., Sinha, A., and Wellman, M.P. (2018). SoK: Security and Privacy in Machine Learning. 2018 IEEE European Symposium on Security and Privacy.

[13] Gunning, D. (2017). Explainable Artificial Intelligence (xAI). Defense Advanced Research Projects Agency (DARPA).

[14] Arrieta, A.B., Diaz-Rodriguez, N., Del Ser, J., Bennetot, A., Tabik, S., Barbado, A. et al. (2019). Explainable Artificial Intelligence (XAI): Concepts, Taxonomies, Opportunities and Challenges Toward Responsible AI. Information Fusion, 58, 82-115. doi: 10.1016/j.inffus.2019.12.012.

[15] Young, S.R., Rose, D.C., Karnowski, T.P., Lim, S.-H., and Patton, R.M. (2015). Optimizing Deep Learning Hyper-Parameters Through an Evolutionary Algorithm. Workshop on Machine Learning in High-Performance Computing Environments, 1-5. Austin, Texas.

[16] Jaderberg, M., Dalibard, V., Osindero, S., Czarnecki, W. M., Donahue, J., Razavi, A. et al. (28 Nov 2017). Population Based Training for Neural Networks. Retrieved from https://arxiv.org/abs/1711.09846

[17] Smith, L.N. (2017). Cyclical Learning Rates for Training Neural Networks. IEEE Winter Conference on Applications of Computer Vision. doi: 10.1109/WACV.2017.58.

[18] He, K., Zhang, X., Ren, S., and Sun, J. (2015). Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification. IEEE International Conference on Computer Vision, 1026-1034.

[19] Glorot, X., and Bengio, Y. (2010). Understanding the Difficulty of Training Deep Feedforward Neural Networks. International Conference on Artificial Intelligence and Statistics.

[20] Potgieter, L. (10 Oct 2018). Semantic Interoperability: Are You Training Your AI by Mixing Data Sources that Look the Same but Aren't? Retrieved from KDNuggets: https://www.kdnuggets.com/2018/10/semantic-interoperability-training-ai-mixing-different-data-sources.html

[21] Tan, C., Sun, F., Kong, T., Zhang, W., Yang, C., and Liu, C. (2018). A Survey on Deep Transfer Learning. The 27th International Conference on Artificial Neural Networks (ICANN 2018). Springer. doi: 10.1007/978-3-030-01424-7_27.

[22] McGilvray, D. (2021). Chapter 1 – Causes of Data Quality Problems: https://cdn.ttgtmedia.com/searchDataManagement/downloads/Data_Quality_Assessment_-_Chapter_1.pdf Retrieved May 2021

[23] Schelter, S., Lange, D., Schmidt, P., Celikel, M., Biessmann, F., and Grafberger, A. (2018). Automating Large-Scale Data Quality Verification. The VLDB Endowment. doi: 10.14778/3229863.3229867.

[24] Breck, E., Polyzotis, N., Roy, S., Whang, S.E., and Zinkevich, M. (2019). Data Validation for Machine Learning. 2nd SysML Conference. Palo Alto, CA, USA.

[25] Bottou, L., and Vapnik, V. (1992). Local Learning Algorithms. Neural Computation, 4(6), 888-900.

[26] Grolinger, K., Capretz, M.A., and Seewald, L. (Jun 2016). Energy Consumption Prediction with Big Data: Balancing Prediction Accuracy and Computational Resources. In 2016 IEEE International Congress on Big Data (BigData Congress), 157-164. IEEE.

[27] Mezouar, M.E., Zhang, F., and Zou, Y. (2016, October). Local versus Global Models for Effort-Aware Defect Prediction. In Proceedings of the 26th Annual International Conference on Computer Science and Software Engineering, 178-187.

[28] Caruana, R. (1997). Multitask Learning. Machine Learning, 28(1), 41-75.

[29] Suresh, H., Gong, J.J., and Guttag, J.V. (2018, July). Learning Tasks for Multitask Learning: Heterogenous Patient Populations in the ICU. In Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 802-810.

[30] Ruder, S. (2017). An Overview of Multi-Task Learning in Deep Neural Networks. arXiv preprint arXiv:1706.05098. International Conference on Knowledge Discovery & Data Mining, 802-810.

[31] Nascimento, N., Alencar, P., Lucena, C. and Cowan, D. (2018). A Context-Aware Machine Learning-Based Approach. In Proceedings of the 28th Annual International Conference on Computer Science and Software Engineering (CASCON '18), pp. 40-47.

[32] McMahan, H.B., Moore, E., Ramage, D., Hampson, S., and Arcas, B.A. (2017). Communication-Efficient Learning of Deep Networks from Decentralized Data. 20th International Conference on Artificial Intelligence and Statistics (AISTATS). Retrieved from https://arxiv.org/abs/1602.05629v3

Chapter 6 – STANDARDS, LAW, AND ETHICAL QUESTIONS

Even basic versions of artificial intelligence that offers on-line shopping recommendations presents ethical implications and challenges, such as gender, racial, and cultural biases. Deep machine learning presents further challenges associated with their unpredictability and resistance to analysis. They are based upon complex black box algorithms that resist a full analysis of their functions and logic. Finding an ethical solution to this opacity is an ongoing challenge [1].

There are well-known difficulties with ensuring that deep learning AI systems are ethically compliant. These problems are mostly due to the lack of transparency of their internal hidden layers of statistical neuron logic. For a general overview of the topic, see Refs. [2], [3], [4], [5]. The use of adversarial machine learning introduces even more ethical dilemmas and issues, see for example, [6], [7], [8], [9].

Because AI and DML are still emerging technologies, legal frameworks are struggling to catch up. For an overview of legal frameworks and AI, see Refs. [10] and [11]. This impacts military operations, as well. For example, to paraphrase a US Army General in Ref. [12], "When life and death decisions were being made, the Staff Judge Advocate always got a say. Is this a just and viable course of action, given the law of armed conflict? Is this a legal response?" Those discussions take time. With the increased use of AI, DML, autonomous vehicles, and lethal autonomous weapons, there is no longer time for those discussions.

The development of AI / Human systems has its share of tragedies [13]. These usually occur because of handoff breakdowns between the AI and humans. In the Air France 447 tragedy in June 2009, a minor glitch in airspeed indicators confused the automatic pilot. The autopilot disengaged before the human pilots were ready. 228 people died. In 2003, US Patriot missile systems accidentally shot down two Allied jets, a British Tornado, and a US Navy F-18. Three pilots died. The first jet was shot down because the system's "automatic mode" was biased to fire on anything unrecognised. The second jet was shot down after automatic mode was changed to tracking only, but a sensor glitch and human miscommunication allowed the system to fire [13].

AI and DML cannot compensate for human stupidity. To the contrary: Human and computer failures usually compound each other. Thus, the legal and ethical implications of using these systems is an evolving challenge.

Below is a list of ethical dilemmas (taken partially from Ref. [1]) highlighting current challenges:

1) **Automated decisions and AI Bias.** AI algorithms and training data may contain biases because they are generated by humans. These biases prevent AI systems from making fair decisions. Biases in AI systems due to two reasons:
   a) Developers may include bias in the system without noticing.
   b) Historical training data may not represent the population accurately.
2) **Autonomous vehicles and the trolley dilemma.** As autonomous vehicles become more prolific in commercial, public, and military settings, decisions related to the trolley dilemma will also increase. In short, the trolley dilemma is: would you take action to kill one person, or take no action and watch 5 people perish?
3) **Lethal Autonomous Weapons.** Lethal Autonomous Weapons LAWs independently identify and engage targets based on programmed constraints and descriptions. There have been debates on the ethics of using weaponised AI in the military. What if a non-combatant is misidentified as an adversary?

4) **Surveillance and Security versus personal privacy.** Improved implementations of facial recognition technology and other personal identifiers (cell phones, automobile licence plates, etc.) can be abused by governments to invade and abuse personal privacy and freedoms. In contrast, it could be used to catch spies and insider threats within military units.
5) **Manipulation of human judgements.** AI-powered analytics can provide actionable insights on human behaviour. This could be used in propaganda campaigns to justify military actions, foment insurrections, or persuade populations of the need for political actions.
6) **Proliferation of deepfakes.** Deepfakes are synthetically generated images or videos in which a person or celebrity is replaced with an artificial likeness. AIs and GANs are extremely good at generating deepfakes. Creating false narratives using deepfakes can influence people's opinions, political views, and beliefs, making them easy to manipulate.
7) **Lack of transparency.** Deep learning is often described as a "black box," its knowledge is stored in its inaccessible internal levels and is difficult to analyse. So DML can make strange decisions, and they are not always explainable (e.g., adversarial attack in Section 7.d). "Explainable AI" is an area of active research.

While concentrating on how DML may help in improving cyber defence operations, keeping track on attack methods against DML-based systems [2], [6], [8], [9] and developing robust DML methods against adversarial attacks [12], it is important to also understand that adversaries are learning as well. Consequently, one has to take into account the possibility that in the following decades adversaries may invent and utilise also effective DML-based attack methods that may be both inventive and autonomous.

It is acknowledged that both the research and its applications should be conducted both ethically and legally. However, NATO and its allies must also consider how to handle adversaries that aren't bound by ethical and legal restrictions.

6.1 REFERENCES

[1] Kantarci, A. (2 Feb 2021). AI Ethics in 2021: Top 9 Ethical Dilemmas of AI. AI Multiple. https://research.aimultiple.com/ai-ethics/

[2] Bae, H., Jang, J., Jung, D., Jang, H., Ha, H., and Yoon, S. (2018). Security and Privacy Issues in Deep Learning. arXiv preprint arXiv:1807.11655.

[3] Long, B. (13 Aug 2020). The Ethics of Deep Learning AI and the Epistemic Opacity Dilemma. Blog of the APA, https://blog.apaonline.org/2020/08/13/the-ethics-of-deep-learning-ai-and-the-epistemic-opacity-dilemma/

[4] Parisi, D. and Goldman, G. (2021). AI, Machine Learning and Big Data Laws and Regulations 2021 USA. Global Legal Insights. https://www.globallegalinsights.com/practice-areas/ai-machine-learning-and-big-data-laws-and-regulations/usa

[5] Walch, K. (29 Dec 2019). Ethical Concerns of AI. Forbes. https://www.forbes.com/sites/cognitiveworld/2020/12/29/ethical-concerns-of-ai/?sh=50277fea23a8

[6] Carlini, N., and Wagner, D. (2018). Audio Adversarial Examples: Targeted Attacks on Speech-To-Text. arXiv preprint arXiv:1801.01944.

[7] Dreossi, T., Jha, S., and Seshia, S.A. (2018). Semantic Adversarial Deep Learning. arXiv preprint arXiv:1804.07045.

[8] Elsayed, G.F., Shankar, S., Cheung, B., Papernot, N., Kurakin, A., Goodfellow, I., and Sohl-Dickstein, J. (2018). Adversarial Examples that Fool Both Human and Computer Vision. arXiv preprint arXiv:1802.08195.

[9] Finlayson, S.G., Kohane, I.S., and Beam, A.L. (2018). Adversarial Attacks Against Medical Deep Learning Systems. arXiv preprint arXiv:1804.05296.

[10] Schmitt, M.N., and Vihul, L. (2014). The Nature of International Law Cyber Norms. Tallinn Papers 5.

[11] Surden, H. (2014). Machine Learning and Law, 89 Wash. L. Rev. 87. https://scholar.law.colorado.edu/articles/81

[12] Freedberg, Jr., S.J. (23 Apr 2021). Artificial Intelligence, Lawyers and Laws of War, Breaking Defense. https://breakingdefense.com/2021/04/artificial-intelligence-lawyers-and-laws-of-war-the-balance/

[13] Freedberg, Jr., S.J. (05 Jun 2017). Artificial Stupidity: Fumbling the Handoff from AI to Human Control. Breaking Defense, https://breakingdefense.com/2017/06/artificial-stupidity-fumbling-the-handoff/

Chapter 7 – MILITARY APPLICATIONS

Military operations are rooted in physical responses to industrial age crises and shaped by assumptions over scale, lethality, and reach [1]. Contemporary conflicts, however, span regional boundaries and geographical domains. The number of threats and range of actors grow in both quantity and diversity, mirrored by the number of actors with whom responses may need to be coordinated with. Adversaries that exploit cyberspace can challenge the threshold at which allies can or will react. Reliance on the cyber domain increases the importance of achieving effects in adversarial cyberspace that support military objectives. Ultimately, military operations grow more dynamic and complex.

Deep Machine Learning (DML) has become the leading source of techniques in the field of Artificial Intelligence. Predictably, DML's impact to military applications beyond cyber defence will be broad, as it provides opportunities to gain information and decision advantage within the military's operating environment. In this chapter we examine those military applications that stand to benefit and consequently reshape cyber defence beyond traditional notions of protection, deterrence, detection, and response.

7.1 COMMAND AND CONTROL

Military doctrine defines Command and Control (C2) as "the exercise of authority and direction by a properly designated commander over assigned and attached forces in the accomplishment of the mission" [2]. C2 is performed through an arrangement of personnel, equipment, communications, facilities, and procedures employed by a commander in planning, directing, coordinating, and controlling forces and operations in the accomplishment of the mission. Traditional C2 constructs include Combatant Command authority, Operational Control, Tactical Control, and Administrative Control [3], [4]. These constructs are rooted among activities carried out among the physical domain, bounded by joint operating areas, and fall increasingly short for the cyber domain.

Military doctrine further defines [1] cyber operations as encompassing cybersecurity actions to prevent unauthorised access, defensive actions taken to defeat specific threats, attack actions to create denial effects, and exploitation actions to gain intelligence [5]. As with missions conducted among traditional warfighting domains (e.g., land, air, sea, space), cyber operations are subject to certain C2 structures. Unlike those other domains, however, cyber forces may simultaneously execute missions among global, regional, and joint operating areas. Cyber operations therefore rely on centralised planning with decentralised execution and require adaptations to traditional C2 structures for detailed coordination between military units and authorities. This construct requires all parties who conduct planning, execution, and assessment to understand the fundamental actions and procedures for cyber operations. The physical and logical boundaries within which joint forces execute, and the priorities and restrictions on its use, must further be centrally identified in coordination and synchronisation among military echelons, national forces, and coalition partners.

C2 for cyber operations is largely shaped by traditional cybersecurity technologies, such as those which provide continuous monitoring of security controls for hardware, software, data, and users [6], [7], [8]. Though C2 is and will remain a fundamentally human challenge [9] common themes among emerging technologies will impact its evolution, both in the cyber and traditional warfighting domains. Advances in information technology, sensors, materials (e.g., batteries), weapons, and increased adoption of unmanned and autonomous platforms will drive evolutionary changes in C2. Computers will more increasingly connect to, and collect or share data with, other devices without human intervention or awareness. Increased computational, storage, and bandwidth

capacity on smaller scale devices will enable new analytic techniques that extract greater understanding at faster rhythms, and closer to the points of observation. Military units may further need to interact with a range of actors and work jointly to achieve mutually desirable outcomes, without any authority to direct those ad hoc partners or interoperate with their information systems. Tactical decisions may need to take place at different levels depending upon the nature of conflict. It may even be desirable to entirely remove geographical connotations from certain cyberspace missions [10].

Collectively, these factors signal the necessity for decentralisation and agility as highly desirable tenets among C2 architectures. Any novel architecture can and should support traditional hierarchies, adaptive teams within a hierarchy, and other distributed contexts while preserving situational awareness of the battlespace. These include a lack of expertise outside the cyber community, the fickle nature, timing, and equities surrounding cyber vulnerabilities, and centralisation of mission planning [11]. Emerging initiatives, such as the US Department of Defense's emerging Joint All Domain Command and Control initiative [12], reflect this concept with coordinated interplay between kinetic, electromagnetic, cyber, and information operations.

Decentralisation may further be enabled by emergent technologies for distributing and protecting data. Distributed Ledger Technologies, i.e., blockchain, are digital systems for recording the transaction of assets in which the transactions and their details are recorded in multiple places at the same time. DML has been recently proposed an integration by which to overcome practical challenges found among blockchain implementations [13]. Similarly, techniques for protecting data in use, as opposed to at rest or in transit, (e.g., secure multi-party computation, homomorphic encryption, functional encryption, oblivious RAM, differential privacy) allow useful computations to be performed on data held by other parties without revealing sensitive information about the content or structure of the data. Such techniques could allow DML processing to be securely performed by untrusted parties or allow multiple parties to compute jointly useful results without disclosing the underlying inputs. It is important to note that adversarial malware may adopt these techniques to better obfuscate their operations. Though these techniques are broadly studied in academia with a good theoretical foundation, more work is specifically needed to suit military use cases and scalability, as well as specific instances where DML can improve the utility of the application [14].

Advances in DML applications will offer opportunities to provide more capable decision support aids for planning and executing missions [15]. Novel human/machine interfaces, mixed reality synthetic environments, and remote presence capabilities will further change the way warfighters interact with each other, automated agents, machines, and robots. These technological developments collectively offer the potential to accelerate observations, orientation, decisions, and actions in complex operating environments. DML will likely improve decision making and facilitate autonomous operations through Human-Machine teaming.

7.2 SITUATIONAL AWARENESS AND MISSION ASSURANCE

Cyberspace depends upon the physical domains of air, land, sea, and space. It comprises nodes and links which perform virtual functions, that in turn, can facilitate effects in the physical domain. Cyberspace is often characterised by three interdependent layers [5]. The physical layer consists of devices and infrastructure that provide storage, transport, and processing of information. The logical layer consists of those elements of the network related to one another in a way that is abstracted from the physical network, based on the programming that drives its components. Finally, the cyber-persona layer is a view created by abstracting data from the logical layer to develop digital representations of an actor or entities operating in cyberspace.

Manoeuvring through these layers is complex and generally unobservable. Accurate and timely Situational Awareness (SA) of cyberspace is critical for success in an increasingly complex battlefield. This is especially true in tactical environments, where there are unique information processing and operating constraints. Significant ongoing research and investments by Government and Industry aim to provide tools to develop basic SA from cyber data but fall short of providing the orders of magnitude improvements needed in key metrics such as successful intrusion detection probability, false alarm rates, time to detection, speed of response, precision and predictability of effects, accuracy and timeliness of battle damage assessment, and the cognitive load on human operators. The cumulative effects of a defensive response may extend beyond the initial threat, necessitating transregional considerations as well as the coordination or synchronisation of defensive responses. These considerations, especially for the tactical battlefield, require breakthrough innovations in continuous processing and actions closer to the source, autonomic fusion of information from multiple heterogeneous networks, intelligence collection, social media, and other multimodal sources.

DML may likely be instrumental to developing approaches by which to accelerate vulnerability discovery before an adversary can exploit them. Similarly, lightweight intrusion detection systems can operate within the constraints of the tactical edge, mitigating restrictions over bandwidth and latency. Additional applications include automated fusion of data from the many heterogeneous networks that have highly distributed, federated, or hierarchical properties; Automated pattern recognition from different sources (e.g., networks and systems, intelligence, social media) and with different time scales and security sensitivities; Cyber and mission ontologies to facilitate mapping between the operating state and mission impact; and modelling and simulation solutions that allow for the automated generation of realistic data sets at scale to facilitate experimentation.

Mission assurance is a mature concept explored among many engineering areas that include high availability systems, failure analysis, and software and systems engineering [16]. DOD policy defines mission assurance as:

> *A process to protect or ensure the continued function and resilience of capabilities and assets – including personnel, equipment, facilities, networks, information and information systems, infrastructure, and supply chains – critical to the execution of DOD mission-essential functions in any operating environment or condition* [17].

Fundamental to mission assurance is insight into those resources and actions necessary to successfully achieve an objective. Mission mapping is the process of determining the dependencies between a mission and its underlying resources and procedures. In the context of cyberspace this includes information systems, business processes, and personnel roles. Cyberspace is a complex, adaptive, and contested system whose structure changes over time. Compounding factors include:

- Accidents and Natural Hazards can disrupt the physical infrastructure of cyberspace. Examples include operator errors, industrial accidents, and natural disasters. Recovery from these events can be complicated by the requirement for significant external coordination and reliance on temporary backup measures.
- Many of DOD's critical functions and operations rely on contracted commercial assets, including Internet Service Providers (ISPs) and global supply chains, over which DOD and its forces have no direct authority.
- The combination of DOD's global operations with its reliance on cyberspace and associated technologies means DOD often procures mission-essential information technology products and services from foreign vendors.

A key challenge with assuring missions that depend upon cyber infrastructure is the difficulty to understand and model aspects that are dynamic, complex, and difficult to sense directly. This includes determining what missions are active at any time, knowing what cyber assets those missions depend upon, the nature of those dependencies, and the mission impact caused by loss or compromise. The understanding of cyber terrain must factor how dependencies change over time and within the context of various missions. It requires identifying levels and complexity of dependencies between mission and cyber infrastructure; accounting for competing priorities and dynamic objectives. This insight can assure the availability of necessary resources and help assess alternative courses-of action under contested conditions.

Further, warfighters may face complex scenarios that disincentivise traditional cyber defence actions in favour of assuring missions. For instance, when computer systems are compromised, current practice dictates isolation of the compromised system. That system is then commonly rebuilt or restored from a trusted backup. Business continuity plans attempt to address operating under degraded conditions, while disaster recovery plans address worst-case scenarios. These approaches prioritise minimal profit loss and do not cater to the types of complex decisions a warfighter may face i.e., the requirement to keep a comprised system online to ensure the availability of a critical application, while the adversary uses it as a leverage point to gain further access or exfiltration of confidential information. Under conditions like this, warfighters need a clear picture of the trade-offs between each choice and the potential impacts on mission and objectives from the outcome of a chosen path. Further, unlike businesses under cyber-attack, the warfighter must contend with the notion that a cyber-attack forms part of the broader application of integrated effects and must factor the adversaries coordinated use of cyber, electronic warfare and kinetic effects. Finally, disaster recovery plans can be argued as a plan for when the war is lost. Hence the warfighter requires effective doctrine and decision support systems that call for maintaining mission continuity among denied, degraded, and contested environments.

Current approaches to mission mapping fall primarily under two categories. First, process-driven analysis is a top-down approach where subject matter experts identify both the mission space and cyber key terrain supporting that mission space. This approach produces interpretable results through business process modelling by subject matter experts, though those results tend to be static. Second, artefact-driven analysis is a bottom-up approach where logs and data from hosts and network sensors are used to draw inferences about the usage of network assets. This approach produces high fidelity decomposition through data mining, red teaming, and forensic discovery, though the results do not provide insight into alternative mechanisms for executing tasks. There currently exist an array of tools and approaches for accomplishing elemental mission mapping [18].

Artificial Intelligence (AI) has seen many applications to decision making for military missions and will continue accelerating capabilities in this problem space. Potential solutions could seek to model specific business process and make them machine-describable such that user-generated logic can ‘reason’ about those processes and assist with managing a flood of information or multiple laborious, complex, or even competing tasks and solutions sets. DML, coupled with advances in natural language processing [19], offer particular promise as traditional means of exchanging information among C2 channels include human-generated tasking orders promulgated through military message traffic.

7.3 DEFENSIVE CYBERSPACE OPERATIONS

Defensive Cyberspace Operations (DCO) comprises missions intended to preserve the confidentiality, integrity, and availability of military networks by defeating or imminent adversarial activity in cyberspace. This distinguishes DCO missions, which defeat specific threats that have bypassed or are threatening to bypass extant security measures, from traditional cybersecurity, which secures cyberspace from all threats in advance of any

specific adversarial threat activity. DCO missions are conducted in response to specific threats of attack, exploitation, or other effects of malicious cyberspace activity and leverage information from intelligence collection, counterintelligence, law enforcement, and public domain sources as required. The goal of DCO is to defeat the threat of a specific adversary and return a compromised network to a secure, functional, state. Activities include tasks native to Event Management, Incident Management, and Malware Detection. It further includes intelligence activities to help make sense of news media, open-source information, and other signals in order to assess the likelihood and impact of adversarial threats. Therefore, DML applications more traditionally rooted in intelligence collection activities hold equal utility for defensive cyberspace operations.

The growing frequency of data breaches signals an accelerated adoption of security automation concepts and capabilities [20]. Only by automating the analysis, response, and remediation of threats, do organisations stand poised to replicate the expertise and reasoning of seasoned cyber experts at scale and ensure a greater degree of protection. Two particular technologies classes stand out: Security Information and Event Management, and Security Orchestration, Automation, and Response.

Security Information and Event Management (SIEM) technology aggregates event data that includes logs and network telemetry produced by security devices, network infrastructure, systems, and applications. Data is often normalised so that events adhere to a common structure and enhanced with contextual information about users, assets, threats, and vulnerabilities. SIEM platforms facilitate network security monitoring, data breach detection, user activity monitoring, regulatory compliance reporting, forensic discovery, and historical trend analysis.

Security Orchestration, Automation, and Response (SOAR) technology enables the application of workflows to cyber event data collected by SIEM platforms. These workflows, sometimes referred to as “playbooks,”automate response actions that align with organisational processes and procedures. SOAR platforms exploit integration with complementary systems to achieve desired outcomes such as threat response, incident management, and increased automation among a wide array of network management, asset management, and configuration management tools.

Collectively, SIEM and SOAR technology enable the automation of two key stages of the security process: the information gathering and analysis, and the execution of the response. Emerging research studies the application of artificial intelligence techniques towards event detection and automated course of action recommendations amenable to both technologies [21], [22], [23], [24], [25].

As the size and scope of interconnected systems grow, applications of autonomy beyond automation will be necessary for scalable cyber defence. Connected systems of lesser criticality may be monitored by cybersecurity sensors, systems, and Security Operations Centres, whereas critical systems, such as those deployed in contested environments may require autonomous intelligent response capabilities [15].

Many mission contexts impose disadvantaged conditions where adaptive, decentralised planning and execution are highly desirable. Though the benefits and challenges of federated cyber operations have been explored [26], market forces continue to promote software-as-a-service solutions that rely upon cloud infrastructure likely unavailable in environments the DOD can expect to operate. The ubiquity of cloud computing and erosion of traditional network boundaries have fostered a dependence on an external and increasingly untrustworthy infrastructure. At the same time, this approach often offers the best economy of scale and capability.

Zero Trust is a security model and set of design principles that acknowledge the existence of threats inside and outside traditional network boundaries. The fundamental purpose of Zero Trust is to understand and control how users, processes, and devices engage with data. The zero-trust framework lays out a security vision amenable to

Enterprise networks, including cloud services and mobile devices. At the same time, zero trust remains a vision and strategy, with more prescriptive methods still emerging [27]. Among these are the Cloud Security Alliance Software Defined Perimeter framework [28], Google's BeyondCorp security model [29], Gartner's Adaptive Risk and Trust Assessment Approach [30], and Forrester's Zero Trust eXtended ecosystem [31]. Little has been done to explore the application of these design principles or the role they may play in assuring DML applications.

As the ecosystem of cybersecurity products and solutions grow increasingly diverse, achieving interoperability by which to coordinate responses at machine speed will become vital. Emerging specifications, such as OpenC2 [32], will enable command and control of cyber defence systems in a manner that is agnostic of the underlying platform or implementation. OpenC2 provides means to standardise interfaces to cyber defence systems, allowing integration, communication, and operation between decoupled blocks that perform cyber defence functions. The suite of specifications includes a semantic language that enables machine-to-machine communication for purposes of command and control of cyber defence components, actuator profiles that specify the subset of the OpenC2 language and may extend it in the context of specific cyber defence functions, and transfer specifications that utilise existing protocols and standards to implement OpenC2 in particular environments. The success of this and similar initiatives will depend upon its adoption by industry. No analogous approaches currently exist for offensive cyberspace operations, largely because of the bespoke nature of the tools employed.

7.4 SOCIAL CYBERSECURITY

Social Cybersecurity is an emerging subdomain of national security that will affect all levels of future warfare, both conventional and unconventional, with strategic consequences. It focuses on the science to characterise, understand, and forecast cyber-mediated changes in human behaviour, social, cultural, and political outcomes, and to build the cyber infrastructure needed for society to persist in its essential character in a cyber-mediated information environment under changing conditions, actual or imminent social cyber-threats." [33].

Technology enables both state and nonstate actors to manipulate the global marketplace of beliefs and ideas at cyber speed, thereby changing the battlefield at all levels of war. For instance, unexpectedly rapid advances in 'deepfake' technology, incidentally, fuelled by DML, which have the potential to change the perceived reality, news as information sources, trust among people, between people and Government, and between Governments.

Cyber defence will increasingly incorporate countermeasures by which to impede influence campaigns inexorably connected to the cyber domain. This will necessitate educating forces and perhaps even society about the decentralised nature of the modern information environment, the risks that exist, and ways and multidisciplinary means to vet the facts that we digest and allow to shape our worldview. Removing any notion of distrust between military forces and the society they swore to defend is paramount to global security.

7.5 CYBER DECEPTION

Whereas traditional approaches to cybersecurity and cyber defence engage adversaries at later stages of the cyber kill chain, cyber deception is an emerging area of research exploring the utility of engaging adversaries earlier on, specifically deceiving them [34]. Introduced decades ago, with honeypots, deceptive approaches have garnered new and reinvigorated interest among the research community as a viable approach for overturning the inherent asymmetry of cyber defence. Deceptive approaches hold potential to alter the asymmetric landscape by introducing uncertainty to the adversary. At the same time, deception capabilities may introduce added complexity.

Cyber deception, sometimes characterised as a form of Moving Target Defences, encompasses techniques across multiple system domains: networks, platforms, runtime environments, software, and data. Moving target techniques are designed to combat the homogeneity of modern systems, where systems and applications are similar enough to each other such that a single vulnerability can enable the simultaneous compromise of thousands or millions (or more) of devices. Techniques seek to introduce diversity between system setups, randomise key components of the system such that attackers cannot exploit identical characteristics, and change system components over time so that the same exploit cannot repeatedly work. Many cyber-attacks are "fragile" in that they require a precise configuration in order to succeed and moving target techniques exploit that fragility. Still, work is needed into cyber metrics and measures of effectiveness for judging the success of cyber deception and other moving target technologies and their applications to different threat models.

7.6 REFERENCES

[1] Alberts, D.S., and Hayes, R.E. (2006). Understanding Command and Control. Assistant Secretary of Defense (C3i/Command Control Research Program) Washington DC.

[2] Joint Publications (2014). Joint Publication (JP) 1-02, Department of Defense Dictionary of Military and Associated Terms, 8 November 2010 (as amended through 15 March 2014), 45.

[3] Joint Operations (2017). Doctrine for the Armed Forces of the United States. Joint Publication 1, Joint Chiefs of Staff, Washington, DC.

[4] Joint Publications (2018). Joint Publication 3-0, Joint Chiefs of Staff, Washington, DC, 10-18.

[5] Joint Publications (2018). Cyberspace Operations. Joint Publication 3-12, Joint Chiefs of Staff, Washington, DC.

[6] Dempsey, K.L., Johnson, L.A., Scholl, M.A., Stine, K.M., Jones, A.C., Orebaugh, A. et al. (2011). Information Security Continuous Monitoring (ISCM) for Federal Information Systems and Organizations.

[7] International Organization for Standardization (2018). ISO/IEC 27000: 2018: Information Technology – Security Techniques – Information Security Management Systems – Overview and Vocabulary. International Organization for Standardization.

[8] Arraj, V. (2010). ITIL®: the Basics. Buckinghamshire, UK.

[9] Lyle, D. (2014). The Rest of the C2 Iceberg. Air University Maxwell AFB AL Air Force Research Institute.

[10] Black, J., and Lynch, A. (2020). Cyber Threats to NATO from a Multi-Domain Perspective. Cyber Threats and NATO 2030: Horizon Scanning and Analysis, 126.

[11] Hossier, M. (2020). The Joint Officer in the Next War Better Know His Cyber, and Good! Methods to Integrating Cyberspace Operations Into Joint Planning. Naval War College, Newport RI, Newport United States.

[12] Hoehn, J.R. (2020). Joint All Domain Command and Control (JADC2). Congressional Research SVC Washington United States.

[13] Dinh, T.N., and Thai, M.T. (2018). AI and Blockchain: A Disruptive Integration. Computer, 51(9), 48-53.

[14] Ryffel, T., Trask, A., Dahl, M., Wagner, B., Mancuso, J., Rueckert, D., and Passerat-Palmbach, J. (2018). A Generic Framework for Privacy Preserving Deep Learning. arXiv preprint arXiv:1811.04017.

[15] Theron, P., Kott, A., Drašar, M., Rzadca, K., LeBlanc, B., Pihelgas, M. et al. (May 2018). Towards an Active, Autonomous and Intelligent Cyber Defense of Military Systems: The NATO AICA Reference Architecture. In 2018 International Conference on Military Communications and Information Systems (ICMCIS), 1-9. IEEE.

[16] Grimaila, M.R., Mills, R.F., Haas, M., and Kelly, D. (2010). Mission Assurance: Issues and Challenges. Conference: Proceedings of the 2010 International Conference on Security & Management, SAM 2010, July 12 – 15, 2010, Las Vegas Nevada, USA, 2.

[17] United States Department of Defense (21 Sep 2012). DoDD 3020.40, DoD Policy and Responsibilities for Critical Infrastructure, DoDD 3020.40P. United States Department of Defense. http://www.dtic.mil/whs/directives/corres/pdf/302040p.pdf

[18] Guion, J., and Mark R. (2017). Dynamic Cyber Mission Mapping. IIE Annual Conference. Proceedings. Institute of Industrial and Systems Engineers (IISE).

[19] Heaven, W.D. (2021). GPT-3. MIT Technology Review. https://www.technologyreview.com/2021/02/24/1014369/10-breakthrough-technologies-2021/#gpt3

[20] Mohammad, S.M., and Lakshmisri, S. (2018). Security Automation in Information Technology. International Journal of Creative Research Thoughts (IJCRT), 6.

[21] Berman, D.S., Buczak, A.L., Chavis, J.S., and Corbett, C.L. (2019). A Survey of Deep Learning Methods for Cyber Security. Information, 10(4), 122.

[22] Li, J.H. (2018). Cyber Security Meets Artificial Intelligence: A Survey. Frontiers of Information Technology & Electronic Engineering, 19(12), 1462-1474.

[23] Ferrag, M.A., Maglaras, L., Moschoyiannis, S., and Janicke, H. (2020). Deep Learning for Cyber Security Intrusion Detection: Approaches, Datasets, and Comparative Study. Journal of Information Security and Applications, 50, 102419.

[24] Nguyen, T.T., and Reddi, V.J. (2019). Deep Reinforcement Learning for Cyber Security. arXiv preprint arXiv:1906.05799.

[25] Apruzzese, G., Colajanni, M., Ferretti, L., Guido, A., and Marchetti, M. (May 2018). On the Effectiveness of Machine and Deep Learning for Cyber Security. In 2018 10th international Conference on Cyber Conflict (CyCon), 371-390. IEEE.

[26] Wehner, G., Rowell, J., Langley, J., and Mathews, J. (2018). Federated Cybersecurity Policy Arbitration. http://ceur-ws.org/Vol-2040/paper10.pdf

[27] Campbell, M. (2020). Beyond Zero Trust: Trust Is a Vulnerability. Computer 53(10), 110-113.

[28] Walker, K. (2013). Cloud Security Alliance Announces Software Defined Perimeter (SDP) Initiative. https://cloudsecurityalliance.org/media/news/csa-announcessoftware-defined-perimeter-sdp-initiative/ Retrieved Oct 2014.

[29] Osborn, B., McWilliams, J., Beyer, B., and Saltonstall, M. (2016). BeyondCorp: Design to Deployment at Google.

[30] Mar, S. (2018). Bringing Cybersecurity into The Future: Internal Auditors Should Consider Whether CARTA Is a Smarter Approach to Addressing Information Security Risks. Internal Auditor 75.1, 16-18.

[31] Kindervag, J. (2010). Build Security Into Your Network's DNA: The Zero Trust Network Architecture. Forrester Research Inc., 1-26.

[32] Mavroeidis, V., and Brule, J. (2020). A Nonproprietary Language for the Command and Control of Cyber Defenses – OpenC2. Computers & Security 97, 101999.

[33] Beskow, D.M., and Carley, K.M. (2019). Social Cybersecurity: An Emerging National Security Requirement. Carnegie Mellon University Pittsburgh United States.

[34] Wang, C., and Zhuo, L. (2018). Cyber Deception: Overview and the Road Ahead. IEEE Security & Privacy 16.2, 80-85.

REPORT DOCUMENTATION PAGE

1. Recipient's Reference	2. Originator's References	3. Further Reference	4. Security Classification of Document
	STO-TR-IST-163 AC/323(IST-163)TP/1080	ISBN 978-92-837-2397-4	PUBLIC RELEASE

5. Originator

Science and Technology Organization
North Atlantic Treaty Organization
BP 25, F-92201 Neuilly-sur-Seine Cedex, France

6. Title

Deep Machine Learning for Cyber Defence

7. Presented at/Sponsored by

Report of STO Research Task IST-163 (IWA).

8. Author(s)/Editor(s)	9. Date
Multiple	October 2022
10. Author's/Editor's Address	**11. Pages**
Multiple	126

12. Distribution Statement

There are no restrictions on the distribution of this document. Information about the availability of this and other STO unclassified publications is given on the back cover.

13. Keywords/Descriptors

Artificial intelligence; Cyber defence; Cybersecurity; Deep machine learning; Federated; Information security continuous monitoring; Machine learning; Transfer learning

14. Abstract

The main goal of the Research Task Group (RTG), Information Systems Technology (IST) 163 activity is to consolidate the NATO-wide knowledge in the field of deep ML and cyber defence, identify the gaps between civilian solutions and military needs, and collaborate with other NATO nations to use data processing, share data and pursue the transfer of the most promising technologies and applications to the military domain. The RTG activity examined the civilian and military needs and solutions including gaps and existing cyber defence techniques. The RTG examined these techniques in alignment with the National Institute of Standards and Technology (NIST) guidelines as guiding factors to compare how the current practices compare to the standards with an assessment of limitations and challenges. The research task group discussed sharing methods and models and the state of the art for sharing data across NATO nations. The technical report scrutinises the intricate utility of DML, practical implementations as well as open challenges. The Research Task Group comprises experts across the fields of data science, machine learning, cyber defence, modelling and simulation, and systems engineering.

NORTH ATLANTIC TREATY ORGANIZATION

BP 25
F-92201 NEUILLY-SUR-SEINE CEDEX • FRANCE
Télécopie 0(1)55.61.22.99 • E-mail mailbox@cso.nato.int

SCIENCE AND TECHNOLOGY ORGANIZATION

DIFFUSION DES PUBLICATIONS
STO NON CLASSIFIEES

Les publications de l'AGARD, de la RTO et de la STO peuvent parfois être obtenues auprès des centres nationaux de distribution indiqués ci-dessous. Si vous souhaitez recevoir toutes les publications de la STO, ou simplement celles qui concernent certains Panels, vous pouvez demander d'être inclus soit à titre personnel, soit au nom de votre organisation, sur la liste d'envoi.

Les publications de la STO, de la RTO et de l'AGARD sont également en vente auprès des agences de vente indiquées ci-dessous.

Les demandes de documents STO, RTO ou AGARD doivent comporter la dénomination « STO », « RTO » ou « AGARD » selon le cas, suivi du numéro de série. Des informations analogues, telles que le titre est la date de publication sont souhaitables.

Si vous souhaitez recevoir une notification électronique de la disponibilité des rapports de la STO au fur et à mesure de leur publication, vous pouvez consulter notre site Web (http://www.sto.nato.int/) et vous abonner à ce service.

CENTRES DE DIFFUSION NATIONAUX

ALLEMAGNE
Streitkräfteamt / Abteilung III
Fachinformationszentrum der Bundeswehr (FIZBw)
Gorch-Fock-Straße 7, D-53229 Bonn

BELGIQUE
Royal High Institute for Defence – KHID/IRSD/RHID
Management of Scientific & Technological Research
for Defence, National STO Coordinator
Royal Military Academy – Campus Renaissance
Renaissancelaan 30, 1000 Bruxelles

BULGARIE
Ministry of Defence
Defence Institute "Prof. Tsvetan Lazarov"
"Tsvetan Lazarov" bul no.2
1592 Sofia

CANADA
DGSIST 2
Recherche et développement pour la défense Canada
60 Moodie Drive (7N-1-F20)
Ottawa, Ontario K1A 0K2

DANEMARK
Danish Acquisition and Logistics Organization
(DALO)
Lautrupbjerg 1-5
2750 Ballerup

ESPAGNE
Área de Cooperación Internacional en I+D
SDGPLATIN (DGAM)
C/ Arturo Soria 289
28033 Madrid

ESTONIE
Estonian National Defence College
Centre for Applied Research
Riia str 12
Tartu 51013

ETATS-UNIS
Defense Technical Information Center
8725 John J. Kingman Road
Fort Belvoir, VA 22060-6218

FRANCE
O.N.E.R.A. (ISP)
29, Avenue de la Division Leclerc
BP 72
92322 Châtillon Cedex

GRECE (Correspondant)
Defence Industry & Research General
Directorate, Research Directorate
Fakinos Base Camp, S.T.G. 1020
Holargos, Athens

HONGRIE
Hungarian Ministry of Defence
Development and Logistics Agency
P.O.B. 25
H-1885 Budapest

ITALIE
Ten Col Renato NARO
Capo servizio Gestione della Conoscenza
F. Baracca Military Airport "Comparto A"
Via di Centocelle, 301
00175, Rome

LUXEMBOURG
Voir Belgique

NORVEGE
Norwegian Defence Research
Establishment
Attn: Biblioteket
P.O. Box 25
NO-2007 Kjeller

PAYS-BAS
Royal Netherlands Military
Academy Library
P.O. Box 90.002
4800 PA Breda

POLOGNE
Centralna Biblioteka Wojskowa
ul. Ostrobramska 109
04-041 Warszawa

PORTUGAL
Estado Maior da Força Aérea
SDFA – Centro de Documentação
Alfragide
P-2720 Amadora

REPUBLIQUE TCHEQUE
Vojenský technický ústav s.p.
CZ Distribution Information Centre
Mladoboleslavská 944
PO Box 18
197 06 Praha 9

ROUMANIE
Romanian National Distribution
Centre
Armaments Department
9-11, Drumul Taberei Street
Sector 6
061353 Bucharest

ROYAUME-UNI
Dstl Records Centre
Rm G02, ISAT F, Building 5
Dstl Porton Down
Salisbury SP4 0JQ

SLOVAQUIE
Akadémia ozbrojených síl gen.
M.R. Štefánika, Distribučné a
informačné stredisko STO
Demänová 393
031 01 Liptovský Mikuláš 1

SLOVENIE
Ministry of Defence
Central Registry for EU & NATO
Vojkova 55
1000 Ljubljana

TURQUIE
Milli Savunma Bakanlığı (MSB)
ARGE ve Teknoloji Dairesi
Başkanlığı
06650 Bakanliklar – Ankara

AGENCES DE VENTE

The British Library Document Supply Centre
Boston Spa, Wetherby
West Yorkshire LS23 7BQ
ROYAUME-UNI

Canada Institute for Scientific and Technical Information (CISTI)
National Research Council Acquisitions
Montreal Road, Building M-55
Ottawa, Ontario K1A 0S2
CANADA

Les demandes de documents STO, RTO ou AGARD doivent comporter la dénomination « STO », « RTO » ou « AGARD » selon le cas, suivie du numéro de série (par exemple AGARD-AG-315). Des informations analogues, telles que le titre et la date de publication sont souhaitables. Des références bibliographiques complètes ainsi que des résumés des publications STO, RTO et AGARD figurent dans le « NTIS Publications Database » (http://www.ntis.gov).

NORTH ATLANTIC TREATY ORGANIZATION

BP 25
F-92201 NEUILLY-SUR-SEINE CEDEX • FRANCE
Télécopie 0(1)55.61.22.99 • E-mail mailbox@cso.nato.int

SCIENCE AND TECHNOLOGY ORGANIZATION

DISTRIBUTION OF UNCLASSIFIED STO PUBLICATIONS

AGARD, RTO & STO publications are sometimes available from the National Distribution Centres listed below. If you wish to receive all STO reports, or just those relating to one or more specific STO Panels, they may be willing to include you (or your Organisation) in their distribution.

STO, RTO and AGARD reports may also be purchased from the Sales Agencies listed below.

Requests for STO, RTO or AGARD documents should include the word 'STO', 'RTO' or 'AGARD', as appropriate, followed by the serial number. Collateral information such as title and publication date is desirable.

If you wish to receive electronic notification of STO reports as they are published, please visit our website (http://www.sto.nato.int/) from where you can register for this service.

NATIONAL DISTRIBUTION CENTRES

BELGIUM
Royal High Institute for Defence – KHID/IRSD/RHID
Management of Scientific & Technological Research for Defence, National STO Coordinator
Royal Military Academy – Campus Renaissance
Renaissancelaan 30
1000 Brussels

BULGARIA
Ministry of Defence
Defence Institute "Prof. Tsvetan Lazarov"
"Tsvetan Lazarov" bul no.2
1592 Sofia

CANADA
DSTKIM 2
Defence Research and Development Canada
60 Moodie Drive (7N-1-F20)
Ottawa, Ontario K1A 0K2

CZECH REPUBLIC
Vojenský technický ústav s.p.
CZ Distribution Information Centre
Mladoboleslavská 944
PO Box 18
197 06 Praha 9

DENMARK
Danish Acquisition and Logistics Organization (DALO)
Lautrupbjerg 1-5
2750 Ballerup

ESTONIA
Estonian National Defence College
Centre for Applied Research
Riia str 12
Tartu 51013

FRANCE
O.N.E.R.A. (ISP)
29, Avenue de la Division Leclerc – BP 72
92322 Châtillon Cedex

GERMANY
Streitkräfteamt / Abteilung III
Fachinformationszentrum der Bundeswehr (FIZBw)
Gorch-Fock-Straße 7
D-53229 Bonn

GREECE (Point of Contact)
Defence Industry & Research General Directorate, Research Directorate
Fakinos Base Camp, S.T.G. 1020
Holargos, Athens

HUNGARY
Hungarian Ministry of Defence
Development and Logistics Agency
P.O.B. 25
H-1885 Budapest

ITALY
Ten Col Renato NARO
Capo servizio Gestione della Conoscenza
F. Baracca Military Airport "Comparto A"
Via di Centocelle, 301
00175, Rome

LUXEMBOURG
See Belgium

NETHERLANDS
Royal Netherlands Military Academy Library
P.O. Box 90.002
4800 PA Breda

NORWAY
Norwegian Defence Research Establishment, Attn: Biblioteket
P.O. Box 25
NO-2007 Kjeller

POLAND
Centralna Biblioteka Wojskowa
ul. Ostrobramska 109
04-041 Warszawa

PORTUGAL
Estado Maior da Força Aérea
SDFA – Centro de Documentação
Alfragide
P-2720 Amadora

ROMANIA
Romanian National Distribution Centre
Armaments Department
9-11, Drumul Taberei Street
Sector 6
061353 Bucharest

SLOVAKIA
Akadémia ozbrojených síl gen M.R. Štefánika, Distribučné a informačné stredisko STO
Demänová 393
031 01 Liptovský Mikuláš 1

SLOVENIA
Ministry of Defence
Central Registry for EU & NATO
Vojkova 55
1000 Ljubljana

SPAIN
Área de Cooperación Internacional en I+D
SDGPLATIN (DGAM)
C/ Arturo Soria 289
28033 Madrid

TURKEY
Milli Savunma Bakanlığı (MSB)
ARGE ve Teknoloji Dairesi Başkanlığı
06650 Bakanliklar – Ankara

UNITED KINGDOM
Dstl Records Centre
Rm G02, ISAT F, Building 5
Dstl Porton Down, Salisbury SP4 0JQ

UNITED STATES
Defense Technical Information Center
8725 John J. Kingman Road
Fort Belvoir, VA 22060-6218

SALES AGENCIES

The British Library Document Supply Centre
Boston Spa, Wetherby
West Yorkshire LS23 7BQ
UNITED KINGDOM

Canada Institute for Scientific and Technical Information (CISTI)
National Research Council Acquisitions
Montreal Road, Building M-55
Ottawa, Ontario K1A 0S2
CANADA

Requests for STO, RTO or AGARD documents should include the word 'STO', 'RTO' or 'AGARD', as appropriate, followed by the serial number (for example AGARD-AG-315). Collateral information such as title and publication date is desirable. Full bibliographical references and abstracts of STO, RTO and AGARD publications are given in "NTIS Publications Database" (http://www.ntis.gov).

ISBN ISBN 978-92-837-2397-4