The University Of Sheffield.

Artificial Intelligence Methods for Security and Cyber Security Systems

Richard Niall Mark Rudd-Orthner BSc (Hons) MSc (Dist)

This dissertation submitted in partial fulfilment
of the requirements for the degree of
Doctor of Philosophy

Department of Automatic Control and Systems Engineering (ACSE)
The University of Sheffield
September 2022

This page is intentionally blank.

To the best friend and wife: Tracey Jane Rudd-Orthner ("Boo Boo"), who has been very patient, and children Jasmine, Ben, and Olivia, but also mother Oriel, late father Aubrey, and both my elder brother Burkett (Kitt) and elder sister Gem (Gemski).

This page is intentionally blank.

ABSTRACT

This research is in threat analysis and countermeasures employing Artificial Intelligence (AI) methods within the civilian domain, where safety and mission-critical aspects are essential. AI has challenges of repeatable determinism and decision explanation. This research proposed methods for dense and convolutional networks that provided repeatable determinism. In dense networks, the proposed alternative method had an equal performance with more structured learnt weights. The proposed method also had earlier learning and higher accuracy in the Convolutional networks. When demonstrated in colour image classification, the accuracy improved in the first epoch to 67%, from 29% in the existing scheme. Examined in transferred learning with the Fast Sign Gradient Method (FSGM) as an analytical method to control distortion of dissimilarity, a finding was that the proposed method had more significant retention of the learnt model, with 31% accuracy instead of 9%. The research also proposed a threat analysis method with set-mappings and first principle analytical steps applied to a Symbolic AI method using an algebraic expert system with virtualized neurons. The neural expert system method demonstrated the infilling of parameters by calculating beamwidths with variations in the uncertainty of the antenna type. When combined with a proposed formula extraction method, it provides the potential for machine learning of new rules as a Neuro-Symbolic AI method. The proposed method uses extra weights allocated to neuron input value ranges as activation strengths. The method simplifies the learnt representation reducing model depth, thus with less significant dropout potential. Finally, an image classification method for emitter identification is proposed with a synthetic dataset generation method and shows the accurate identification between fourteen radar emission modes with high ambiguity between them (and achieved 99.8% accuracy). That method would be a mechanism to recognize non-threat civil radars aimed at threat alert when deviations from those civilian emitters are detected.

This page is intentionally blank.

ACKNOWLEDGMENTS

The experience of the Ph.D. study I have considered beneficial in my development, and this calls me to offer express thanks to those that supported it in this personal self-improvement development:

To my wife: Tracey Jane Rudd-Orthner, for all of the practical support and the *ray of sunshine* she provided while I studied working away from our home during the period of the COVID 19 pandemic.

To Professor Lyudmila Mihaylova for the supervision, encouragement, and guidance during my Ph.D. study while on that journey of self-improvement.

My thanks to Professor Hissam Tawfik and Professor Anatoliy Gorbenko for the opportunities to become more involved in IEEE conferences. Also, my thanks to Dr. Adrian Wagstaff and David Tufnell for their encouragement and belief in the research goals.

To my BAE SYSTEMS engineering lead, Vincent Shepherd, the BAE SYSTEMS programme manager, Ahmed Al-Sukran, Simon Gallagher, Michael Ratcliffe, Jim Simpson, Jonathan Sanders, my colleague, and also to Sagger Al-Sahali, Saif Shahrani, Majed Al-Robaie, Meshari Al-Harbi, Fahad Al-Mohaini, Abulahakeem Al-Shawi, Hasheim Al-Ghamdi, Mohammed Al-Amri, Abdullah Al-Hawaishal, Abdulla Aftan, Raed Al-Toukhi and of course Hassan Rashed for their general brotherhood, and support to me while in a foreign country during the COVID pandemic.

This dissertation was security graded to 'public domain' by the UK Ministry of Defence (MoD) carried out by Mass Consultants Ltd, UK.

This page is intentionally blank.

CONTENTS

LIST OF PUBLICATIONS

Journal Articles:

[J1] R. N. Rudd-Orthner and L. Mihaylova, “Deep convnet: non-random weight initialization for repeatable determinism, examined with FSGM,” *In Sensors*, vol. 21, no. 14, p. 4772, 2021, IEEE.

[J2] R. Rudd-Orthner and L. Mihaylova, “Repeatable determinism using non-random weight initialisations in smart city applications of deep learning,” *In the Journal of Reliable Intelligent Environments in a Smart Cities special edition*, vol. 6, no. 1, pp. 31-49, 2020, IEEE.

[J3] R. N. Rudd-Orthner and L. Milhaylova, “Numerical discrimination of the generalisation model from learnt weights,” *In the Journal Annals of Emerging Technologies in Computing*, vol. 3, no. 4, pp. 1-14, 2019, IAER.

[J4] R. N. Rudd-Orthner and L. S. Mihaylova, “An algebraic expert system with neural network concepts for cyber, big data and data migration,” *In the International Journal of Software & Hardware Research in Engineering (IJSHRE),* vol. 7, no. 12, pp. 42-51, 2019.

[J5] R. Rudd-Orthner and L. S. Mihaylova, “Aircraft protection for complex threat platforms through integrated EWOS application,” *In the IoP Conference Series: Materials Science and Engineering,* vol. 619, no. 012078, 2019, Institute of Physics (IoP).

Conference Papers:

[C1] R. N. Rudd-Orthner and L. Mihaylova, “Non-random weight initialisation in deep convolutional networks applied to safety critical artificial intelligence,” in *Peer Reviewed Proc. of the 13th International Conference on Developments in eSystems Engineering (DeSE)*, Liverpool, UK, 2020, IEEE.

[C2] R. Rudd-Orthner and L. Mihaylova, “Non-Random weight initialisation in deep learning networks for repeatable determinism,” in *Peer Reviewed Proc. of the 10th IEEE International Conference Dependable Systems, Services and Technologies (DESSERT-19)*, Leeds, UK, 2019, IEEE.

[C3]: R. N. Rudd-Orthner and L. Milhaylova, “Numerical discrimination of the generalisation model from learnt weights in neural networks,” in *Peer Reviewed Proc. of the International Conference on Computing, Electronics & Communications Engineering (iCCECE)*, London, UK, 2019, IEEE.

[C4]: R. N. Rudd-Orthner and L. Mihaylova, “An algebraic expert system with neural network concepts for cyber, big data and data migration,” in *Peer Reviewed Proc. of the IEEE International Symposium on Signal Processing and Information Technology (ISSPIT)*, Ajman, UAE, 2019, IEEE.

[C5] R. N. Rudd-Orthner and L. Mihaylova, “Aircraft protection of complex threat platforms through integrated EWOS application,” in *Peer Reviewed Proc. of the 18th International Conference on Aerospace Sciences & Aviation Technology (ASAT-18)*, Cairo, Egypt, 2019, Egyptian Military Technical College (MTC).

This page is intentionally blank.

LIST OF FIGURES

This page is intentionally blank.

LIST OF TABLES

This page is intentionally blank.

LIST OF ACRONYMS

2DPCA	2D Principle Component Analysis
A2/AD	Anti Access Area Denial
AC	Artificial Curiosity
ACARS	Aircraft Communications Addressing and Reporting System
ACSE	Automatic Control and Systems Engineering
AETiC	Annals of Emerging Technologies in Computing
AGI	Artificial General Intelligence
AI	Artificial Intelligence
AIS	Automatic Identification System
ANN	Artificial Neural Network
AOA	Angle of Arrival
API	Application Programming Interface
ARC	Adaptive Radar Countermeasures
ASAT	International Conference on Aerospace Sciences & Aviation Technology
ATC	Air Traffic Control
BN	Bayesian Network (or Belief Networks)
BNF	Backus–Naur form
Big Data	Extremely large datasets for pattern analysis and analytics
BLADE	Behavioural Learning for Adaptive Electronic Warfare
C2	Command and Control
C3L	Common Countermeasures Communication Language
C4I	Command, Control, Communications, Computers and Intelligence
C5I	Command, Control, Communications, Computers, Cyber and Intelligence
C4L	Common Coordinated Countermeasures Communication Language
CAA	Crossbar Adaptive Array

CAE	Convolutional Autoencoder
CBR	Case Based Reasoning
CEMA	Cyber and Electromagnetic Activities
CM	Countermeasure
SD_CMRM_Iv1	Synthetic Dataset Civil Marine Radar Modes as Images version 1
CNE	Computer Network Exploitation
CNN	Convolutional Neural Network
CNO	Computer Network Operations
CoG	Centre of Gravity
COMINT	Communications Intelligence
COVID19	Corona Virus Disease 2019
CPU	Central Processing Unit
CW	Continuous Wave
D2D	Data to Decision
DAE	De-noising Auto-Encoder
DAG	Directed Acyclic Graph
DARPA	Defense Advanced Research Projects Agency
DAS	Defensive Aid Suites
DBN	Deep Belief Networks
DCGAN	Deep Convolutional Generative Adversarial Network
DEAD	Destruction of Enemy Air Defenses
DeSE	Developments in eSystems Engineering
DESSERT	International Conference on Dependable Systems, Services and Technologies
DEX	DEcision Expert
DewL	Defensive Electronic Warfare Language
DJINN	Deep Jointly Informed Neural Network
DL	Deep Learning

DLL	Dynamic Link Library
DN	Deep Network
DNA	DeoxyriboNucleic Acid
DNN	Deep Neural Network
DOA	Direction of Arrival
DoD	Department of Defense
ESS	Explainable Expert System
EA	Electronic Attack
EA	Evolutionary Algorithm (Unused, avoiding confusion with Electronic Attack)
ECCM	Electronic Counter-Countermeasure
ECM	Electronic Countermeasure
EKB	Egyptian Knowledge Base
ELINT	Electronic Intelligence
ELISA	Natural Language Processing AI program named after Eliza Doolittle of Pygmalion
EM	Electromagnetic
EMCON	Emissions Control
EP	Evolutionary Programming (Unused, avoiding confusion with Electronic Protection)
EP	Electronic Protection
ERP	Effective Radiated Power
ES	Electronic Support
ES	Expert Systems (Unused, avoiding confusion with Electronic Support)
ESM	Electronic Support Measure
EW	Electronic Warfare
EW	Early Warning (when radar equipment type)
EWOS	Electronic Warfare Operational Support

EXAMM	Evolutionary eXploration of Augmenting Memory Models
FDOA	Frequency Deference of Arrival
FGCOM	Future Generation Computing & Technology Innovations
FMS	Flight Management System
FRUIT	False Returns Uncorrelated In Range
GA	Genetic Algorithms
GAM	Generalized Additive Models
GAN	Generative Adversarial Network
GLM	Generalised Linear Model
GPU	Graphics Processing Unit
GRU	Gated Recurrent Units
HF	Height Finding
HPRF	High Pulse Repetition Frequency
IA	Information Assurance
IADS	Integrate Air Defence System
I Channel	In-phase (Real or cos component)
iCCECE	International Conference on Computing, Electronics & Communications Engineering
ICW	Interrupted Continuous Wave
ID	Identity
IEEE	Institute of Electrical and Electronics Engineers
IF	Intermediate Frequency
IFF	Identifier Friend of Foe
IFR	Instrument Flight Rules
IJSHRE	International Journal of Software & Hardware Research in Engineering
IMINT	Imagery Intelligence
IOP	Institute of Physics

ISSPIT	International Symposium on Signal Processing and Information Technology
IW	Information Warfare
JSR	Jamming to Signal Ratio
JRFL	Joint Restricted Frequency List
KG	Knowledge Graph
kHz	Kilohertz
Km	Kilometre
KNN	K Nearest Neighbour
LDA	Latent Dirichlet Allocation
LO	Local Oscillator
LPRF	Low Pulse Repetition Frequency
LSTM	Long Short-Term Memory
M&S	Modelling and Simulation
MAE	Mean Absolute Error
MASS	Maths Analysis and Scientific Services
MEZ	Missile Engagement\Zone
MDA	Model Driven Architecture
MGAN	Mixture Generative Adversarial Networks
MGU	Minimal Gated Units
ML	Machine Learning
MLAT	Multilateration
MLE	Maximum Likelihood Estimation
MLP	Multi-Layer Perceptron
MNIST	Modified National Institute of Standards and Technology
MoD	Ministry of Defence
MODAF	Ministry of Defence Architecture Framework
MPRF	Medium Pulse Repetition Frequency

Ms	millisecond
MSE	Means Squared Error
MSSIS	Maritime Safety and Security Information System
MTARSI	Muti-Type Aircraft of Remote Sensing Images (a dataset)
MTC	Military Technical College
MTI	Moving Target Identifier
MUSIC	Multi Signal Classification
MYCIN	A suffix of antimicrobial agents (early backward chaining expert system)
N-Let	A set of N closely spaced pulses with varying with variable inter-pulse intervals
NAFv4	NATO Architecture Framework Version 4
NCE	Noise Contrastive Estimation
NCI	Non-Cooperative Identification
NIN	Network In Network
NLG	Natural Language Generation
NLP	Natural Language Processing
NM	Nautical Mile
OA	Open Architecture
ODE	Ordinary Differential Equations
PCA	Principal Component Analysis
PD	Pulse Duration (also called Pulse Width)
PDW	Pulse Descriptor Word
PFM	Power Flow Management
PhD	Doctor of Philosophy
PM	Predictability Minimisation
Pnode	Probability of a Node
PRF	Pulse Repetition Frequency
PRI	Pulse Repetition Interval

PSYOP	Psychological Operations
PW	Pulse Width (also called Pulse Duration)
Q Channel	Quadrature phase (Imaginary or sine component)
∆-RNN	Delta RNN
RBF	Radial Basis Function
RBM	Restricted Boltzmann Machine
REX	Reconstructive EXplainer
RF	Radio Frequency
RGB	Red Green Blue
RNN	Recurrent Neural Network
ROE	Rules of Engagement
RWR	Radar Warning Receiver
Rx	Receive
s	second
S-Band	2-4 GHz carrier (IEEE standard)
SA	Situation Awareness
SAM	Surface to Air Missile
SEAD	Suppression of Enemy Air Defenses
SEI	Specific Emitter Identification
SIGINT	Signals Intelligence
SSR	Sum of Squared Residuals
SGD	Stochastic Gradient Decent
SVM	Support Vector Machine
TA	Target Acquisition
TAR	Target Acquisition Radar
TDOA	Time Deference of Arrival
TOA	Time of Arrival

TT	Target Tracking
TTR	Target Tracking Radar
Tx	Transmit
UAS	Unmanned Air System
UAV	Unmanned Air Vehicle
UAE	United Arab Emirates
UGRNN	Update Gate RNN
UK	United Kingdom
UPRF	Unambiguous Pulse Repetition Frequency
US	United States
VFR	Visual Flight Rules
VGG	Visual Geometry Group
VHF	Very High Frequency
VMS	Visual Meteorological Conditions
VTS	Vessel Traffic Services
WEZ	Weapon Engagement Zone
X-Band	8-12 GHz carrier (IEEE standard), or 7-11.2 GHz in communication engineering

Chapter 1

INTRODUCTION

The chapter describes an overview of the research, its background, motivations, and applications, followed by the contributions made. Several papers published during the research are referred to in further chapters within their discussions.

1.1 Background

An area of study is Cyber and Electronic Warfare (EW) brought together in the Cyber and Electromagnetic Activities (CEMA) doctrine [1], in which CEMA brought EW together with Cyber [2] in coordination. CEMA is a cyber modernization of EW, from the collection and analysis to reprogramming into Information Warfare (IW) [3]. Although some might argue that if Alan Turing's Turing Test [4] was to be a successful electronic deception of a human, then EW has been around as long as there was a notion that deception in the electronic means was viable. Logically EW can be considered as old as electronic engineering itself. One of the first recorded uses of EW was during the American Civil War (1861-1865), when Confederate soldiers attacked and interfered with the telegraph system by providing falsified messages [5]. However, CEMA's cyber evolution has led to the incorporation of computers and networks such as C4I (Command, Control, Communications, Computers, and Intelligence) [6] and the collection, processing, analysis, manipulation, and exploitation of large amounts of information. C4I applications have used Knowledge Graphs (KG) and Machine Learning (ML) methods, but there are still challenges with information de-biasing [7]. C4I has adapted and evolved to C5I (Command, Control, Communications, Computers, Cyber, and Intelligence) with the inclusion of 'Cyber' [8], and a further challenge for C5I and CEMA is creating understanding from information that an opponent may present to conceal that understanding. Furthermore, as a step towards that challenge: an example in modern computing approaches with Artificial Intelligence (AI) uses sentiment analysis to reveal natural language understanding of human opinions [9]. With concern to unattended cyber systems, a paper by Petrovski et al. [10] proposed viability within a framework in applying ML in an aerial traffic surveillance system for anomaly and characterization of driving behaviours, establishing situation awareness. Thus the application of AI to CEMA and Information Warfare is inevitable and becomes even more appropriate for research as AI evolves toward the challenges of Artificial General Intelligence (AGI) [11].

1.1.1 Motivation

The broad motivation of the research is to apply AI techniques in EW, particularly in airborne and naval threat analysis applications [12], [13], where a method using a Bayesian Network (BN) has already been proposed [14], but as AI evolves research is required towards CEMA's Information Warfare using ML. In particular to this research are AI techniques that can be validated and machine-learnt for their applications to EW, threat analysis, and CEMA. CEMA involves processing large amounts of geographical and virtual connected data such as social media personas to inform human decisions in real-time. Currently, that real-time nature causes the pre-processing of data to meet that decision's deadline. AI methods offer a proposition to aid in formulating a decision, offer a decision-making aid, or offer potential live real-time optimal machine decisions. CEMA also has a safety and mission-critical nature, and some areas of AI lack proof, particularly in neural ML approaches. A challenge of CEMA is to decide to cause an action quickly using potentially large amounts of data; thus, the timeline can be critical for a Data to Decision (D2D) [15].

1.1.2 Application Area

The research area concerns threat analysis to countermeasure tactics in the civil domain, where civil aircraft and shipping may be engaged in error. In this area, the timeline is still short, as in the military domain, as the engagement from an aggressive threat has already potentially begun. Furthermore, a penalty for the D2D quality in the civilian domain is much less available data. In the civil domain, there is a lack of threat analysis and threat detection, countermeasure tactics response

training, or any CEMA doctrine using secure data, and this forms a further penalty for decision making in that assistance to the civil domain. An application would still need to reduce outcomes quickly to provide decision support and reduce information availability while being restricted to civilian sources. This approach may require a network as a shared infrastructure, which might build upon future AI-equipped versions of Air Traffic Control (ATC) [16], [17] and marine Automatic Identification System (AIS) [18], [19] networks. Where both of these networks have already been propositioned in research to use AI and could integrate into existing civil networks like Aircraft Communications Addressing and Reporting System (ACARS) [20], [21], [22], and Maritime Safety and Security Information System (MSSIS) [23], [24], [25], [26], which have also had proposals in research for AI incorporation. Those shared networks may provide the communication backbone for a warning to avoid or aid the evasion and defeat of a threat in coordination using high endurance autonomous platforms as part of a Civilian CEMA-like doctrine, possibly as part of the Multi Domain Operation (MDO) approach [27], [28], [29], [30], [31], [32], [33], [34]. MDO is a step beyond the concept of Joint Operations (Army, Navy, Air force, and Marines) and reaches multi-domain both within and out with those forces (air, land, maritime, space, cyberspace) [35]. MDO has also employed some AI methods in decision-making assistance [36] and is stretching toward governmental and political constraints [37].

A civilian version of the MDO approach could be via a political agreement to avoid civilian accidence. Although this research is not concerned with political standpoints or agreements, that possible implementation mechanism is suggested only to illustrate practicality. This research is also not concerned with: latency, bandwidth, the number of access endpoints, or the specific data content of those existing or future civilian accessible networks. The research is concerned with the challenge of AI methods such that they are hardened enough in proof to be reliable while assisting in threat detection and analysis towards intelligent countermeasure decisions. The hardened proofs are more of a concern in the civil domain as flight and marine operators are litigation exposed [38], [39], and operators also need to be transparent in legal processes and accident investigations. The legal standpoint for AI is urgent, as the UK Government predicts introducing driverless cars using AI in 2022-2027 [40]. However, disregarding the legal law specifics, generally, in any system, some issues of safety surround: build quality verification, making a validated error prediction, and the limited data availability for low probability scenarios that can still be catastrophic, although some development processes can seek to reduce these [41]. Thus AI methods need validated and verified predictions, but some methods in AI like ML are harder to verify, as the content is not readily understandable, particularly when high dimensional data is used [42]. Other more verifiable approaches like Symbolic AI are less able to machine-learn to adapt a solution outside of the original programming, causing the verification to be constrained to validate the use-cases expected originally [43].

1.1.2.1 Current Applications of AI Methods

Classical AI methods (called ‘Symbolic AI’) have more acceptable proofs but traditionally lack direct ML capabilities [44]. In contrast to ML, ML methods have lacked human proof. One of the reasons for that is the black box nature with instabilities of the solutions learnt, as they differ depending on a start condition and thus lack repeatable determinism [45], which complicates a black box understanding, as such neural approaches to ML can be performance measured by a prediction accuracy and loss that can differ between learning sessions [46]. Random numbers in the initial condition provide a diversity of initial numbers as a stochastic coverage and support Monte-Carlo estimation in an update [47]. Although this also impairs the analysis of the ML solutions as random numbers lack repeatable determinism. However, some methods can reuse the same random numbers for repeatability, and others may use a start condition learnt or optimized from a dataset. Random numbers are still prevalently used [48], [49], as they are not dataset tethered as a general case but also have stochastic and Monte-Carlo qualities in the diversity of the random initial condition numbers used.

Nevertheless, individual random number sequences still affect a model’s accuracy and loss after learning, which is visible over regularisation, lacking repeatable determinism. This research worked on substituting the random numbers form, which provides an alternative that is still not coupled to a dataset but is repeatable and deterministic, and removing an unintentional noise re-colourization

opportunity when the initializing noise combines with the noise in the dataset, confining the stochastic and Monte-Carlo qualities to the dataset, rather than including re-colourized noise in with the initial condition from the outset of learning. The early traditional 'Symbolic AI' techniques provide a higher level of proof but have higher abstraction levels to the computer in symbolic graph data structure methods, and they involve human involvement for analytics and adaption. More actively researched neural ML techniques provide machine adaption methods in complex networks with lower computer abstraction and less human involvement, but these are harder to verify. Interest areas of the research are techniques that allow validation and verification from ML, or Symbolic AI methods with ML, within the subject areas of 'Explainable AI' and 'Neuro-Symbolic AI.'

1.1.2.2 Application to the Case Study Area

The CEMA doctrine has the full spectrum of complexity, and within it, EW is also a broad subject, so a particular focus is EW within the civilian domain. This aspect is threat analysis with a view to countermeasure tactics. Threat analysis analyzes possible threats, how they work, and how to counter them. Threat analysis involves collections of observational evidence followed by technical analysis. The formation of rules for discriminating threats and cataloguing are applied using empirical observational evidence and a body of knowledge. These rules provide the indications for less discernible discriminating aspects from more observable evidence in the collections. Although EW has an obvious military application, it also has a civil application. It is necessary to protect civil assets operating near threats, sometimes unexpectedly, and in recent years those threats have caused losses. Threat analysis begins with identifying a threat and extends toward deploying tactics and countermeasures. Both threat analysis and countermeasures have some decision-making effects for safety-critical liabilities that must be respected, particularly in the civil domain, as legality and conformance to Air Traffic Control (ATC) and coast guard traffic rules are law requirements. As such, for the central part, the location of the decision-making authority could be placed into their controlling hands so they can coordinate with other decisions for safety.

1.1.2.3 Challenges of Applying Artificial Intelligence

Threat analysis is traditionally a human task, which supports equipment programming, detecting and recognizing a threat, and taking action or providing a cue to guide a human operator. Therefore, threat analysis is the preliminary analysis and takes time to complete and verify, making the adaption in response to unexpected changes outside of the initial analysis a problem. So it follows that if threat analysis can be adaptive to unexpected changes in threat observations with the incorporation of ML, it can be beneficial to make a judgment informed by new observations. As such, the ML needs to be verified to be safe and validated.

AI is also a broad subject, and a focus area is neural ML. So, the focus area within the research is for 'Safety-Critical AI with ML'. Many AI techniques can provide either ML or Safety-Critical AI. This research focuses on AI techniques with either safety-critical AI or ML capabilities. The two primary AI methods in this research are *'Symbolic-AI's Expert System'* method and *'ML within the Neural Network methods'*.

Symbolic AI techniques have a long track record and have a human abstraction level closer to the human consciousness of reasoning (as higher-order cognition); thus, they are easier to certify but cannot learn autonomously. ML techniques arguably have an abstraction level closer to the human subconscious (as lower-order cognition) but are harder to verify and validate. However, an existing method by Melen, Sartori and Grazioli (2015) [50] aimed to adapt an 'expert system' through time-evolving scenarios and graded the rule knowledge base for rule-selection bias changes over time using a BN. This work was limited to grading existing rules in response to operational environment changes rather than learning new rules. Conceptually, in this work, every possible rule would need to be in the knowledgebase to have an ML capability that is complete for unexpected experiences outside the original rule-based analysis.

Symbolic AI techniques lost focus in research in the mid-1990s. However, from the late-1990s, research advances in ML, firstly in speech and then image processing, began to reach closer to human performances on lower-order tasks like categorization and segmentation. Arguably, this was also because the ML techniques did not require such high levels of abstraction from the problems they

were solving and did not require human analytical involvement to create the machine representation. Also, their data requirements coincided with accessibility to larger datasets and the means of processing them, making it more practical. Furthermore, ML and Deep Networks (DN) advances were less certifiable as they had increased complexity in a node distributed form, with a subtle learnt model content that could pertain to the problem space or be an artefact of sampling in the dataset used. A method by King, Jupe, and Taylor used ML in power distribution networks for Power Flow Management (PFM) [51], and this method avoided the safety-critical validation aspects by limiting its decisions to the selection of algorithms. The safety-critical aspects were internal to those algorithms selected, and the neural network was limited to defining state-based changes between those algorithms. Another method, by Ullah et al. [52], used Artificial Neural Networks (ANN) for lightning strike prediction in building design, but that method limited its safety-critical validation to be an advisory tool. Within ANN, 'rule extraction' is an approach to understanding the learnt content of an ANN's generalization model. ANN also has some challenges when using random initializations and causes variations in the learnt weights and biases each time when training the network, affecting the system's resultant accuracy. That is to say, the initialization can cause problems for repeatable determinism and is visible over regularisation.

1.2 Research Gap

The ‘gap’ for this research was for an ML technique that can be verified and make an inference in new operating conditions or learn new operating behaviours but with safety-critical verification. Outside of threat analysis, ML with safety-critical certification has applications in other domains such as industrial processes, automotive, aircraft autonomy, and even legal liability in ‘smart cities.’ It may also assist in solving intractable problems by providing machine-learnt understanding. The application area was EW threat analysis in the aviation and naval civil domains and accessed and incorporated AI techniques. The aviation and naval domains are areas where automatic control and decision-making systems become autonomous. Advances in control systems have led to autonomous ‘unmanned flight control systems’ when flown in un-segregated airspace with civil piloted passenger aircraft. Military conflicts and the increasing use of long-range Surface to Air Missile Systems (SAMs) that are mobile and formed into Integrated Air Defence Systems (IADS) creates larger exclusion areas as an Anti-Access Area Denial (A2/AD), with the likelihood that those military systems ‘pop-up’ closer to civilian systems. The application of self-protection in the civil domain has a growing need; however, the civil domain has less access to detailed threat analysis information. Safety hardened AI approaches might protect civil domain systems. Perhaps in avoidance, mainly if the threat systems are ‘unmanned’ and ‘autonomous.’ AI methods that can be validated and verified within ML could bridge to explainable AI, and those methods also have applications beyond the case study area. Neuro-Symbolic AI is a research area where Symbolic AI and neuron methods combine to provide a verifiable approach to ML. Neuro-Symbolic AI bridges an abstraction gap between higher and lower order cognition, with higher order cognition in tasks of Symbolic AI approaches and lower order cognition in tasks of neuron ML methods.

1.2.1 Contributions

A description of the research contributions made in the research dissertation is in the following subsections as research threads. However, those research threads are also in greater detail in later chapters referencing the publications made.

1.2.1.1 Non-Random Initialization for Repeatable Determinism in Neural Networks

This research offered an alternative initialization state method for dense and convolutional neural networks with repeatable determinism while decoupling from the dataset used, making this initialization state a general case of initialization rather than coupled by dataset pre-training or pre-sampling. In dense layers, the weights after learning are more structured to the training dataset image, which is advantageous to rule extraction, as the structure has clustered areas of higher weight values, meaning it is more generalized prior to a formula extraction method's generalization phase. In convolutional networks, there are benefits of earlier learning and higher accuracy. Using the MTARSI2 [53] dataset of colour images of airplanes on runways, the first epoch difference in learning was 29.31% accuracy, which increased to 67.2% with the proposed method. In further

examination, the convolutional method retained more knowledge in transferred learning in a test case than in the existing random scheme.

1.2.1.2 Numerical Discrimination of a Neural Network

A method using several weights for separated activation strength value ranges in each dendrite-activation of a dense layer formed from a 'number line' input as a fourth data dimension after height width channels; provides a simplification of the weight network representation in a single layer. A formula extraction method after backward chaining arrives at the weight values, and those weights map from the input to output form with distortions of compression and offsets that reveal the discrete mathematical operators between the input and outputs. When reversing the operators in operation, a numerical function is exposed. The input format allows a layer to represent complex logic in a single layer, thus lowering depth, where depth can cause weight values to have vanishing gradients.

1.2.1.3 An Algebraic Expert System with a Neuron Method

Presentation of a symbolic AI method towards a Neuro-Symbolic AI method, using algebraic rules, is composed of a computational graph that virtualizes nodes of every possible calculation permutation in a neuron structure from the rule base body of knowledge. The method is modular and structured into input, hidden, and output layers of neurons with calculated Bayesian confidences, which then perfect better prediction estimates based on all the confidences in a histogram method and provide a confidence and certainty metric. An advantage in explanation is that each node has a unique receptive field and is mutually exclusive in semantics rather than in a neural network method where the semantics are overlapped and spread over nodes.

1.2.1.4 Synthetic Datasets, ELINT Disambiguate Method using Image Classification

An ELINT dataset generator method is proposed and demonstrated within a further image creation definition for an image classification method. This method helps compensate for the low availability of ELINT datasets, and an image format method used in classification achieves 99.8% accuracy in identification between 14 ambiguous civilian marine emitter radar modes. It is achieved via a synthetic dataset creator using a radar emitter mark-up language specified in Backus-Naur form (BNF) and implemented as C4L embedded into TensorFlow. The synthetic dataset creator has a high fidelity in data dimensionality, reflecting actual physics and propagation effects.

1.2.1.5 Threat Analysis, Data Availability, the Onion of Protection and Response

An alternative method to threat destruction of A2/AD IADS threats, with counters to threat system earlier in an 'extended kill chain' in an 'onion of protection' structure, is proposed. That onion of protection maps the threat's kill chain intention and data availability. This ELINT and threat analysis method provides a 'measured response' that preserves intelligence data from being unnecessarily exposed while organizing strengths, weaknesses, vulnerabilities, and opportunities into set-mappings for matches in a countermeasure tactic selection.

1.2.1.6 New datasets

A new synthetic dataset that contains the civil marine radar modes as images for image processing methods is available as (SD_CMRM_Iv1) [54]. Furthermore, the existing MTARSI dataset [55] was reclassified from the existing images into 42 categories, extended with additional images using data augmentation methods, and made available as MTARSI2 [53].

1.3 Dissertation Outline

The dissertation is structured as follows:

- Chapter 1: includes the background and contributions,
- Chapter 2: is a literature review,
- Chapter 3: is the conducted research with a methodology,
- Chapter 4: is the first research theme for an EW Threat Analysis method,
- Chapter 5: is the second research theme for an Expert Systems method, as a step toward Neuro-Symbolic AI,
- Chapter 6: is the third research theme for a Formula Extraction method as a step toward Neuro-Symbolic AI,

- Chapter 7: is the fourth research theme for a Safety-Critical AI for neural methods and with transferred learning,
- Chapter 8: is the fifth research theme for an image classification method for emitter identification using a synthetic dataset (SD_CMRM_Iv1) [54],
- Chapter 9: is the summary and conclusions,
- Chapter 10: is the bibliography,
- Chapter 11: contains further information on the Synthetic Dataset Creation model that demonstrates the numerical fidelity in the dataset.

The five research threads and reasoning for them are shown within the block diagram of chapters in Figure 1.

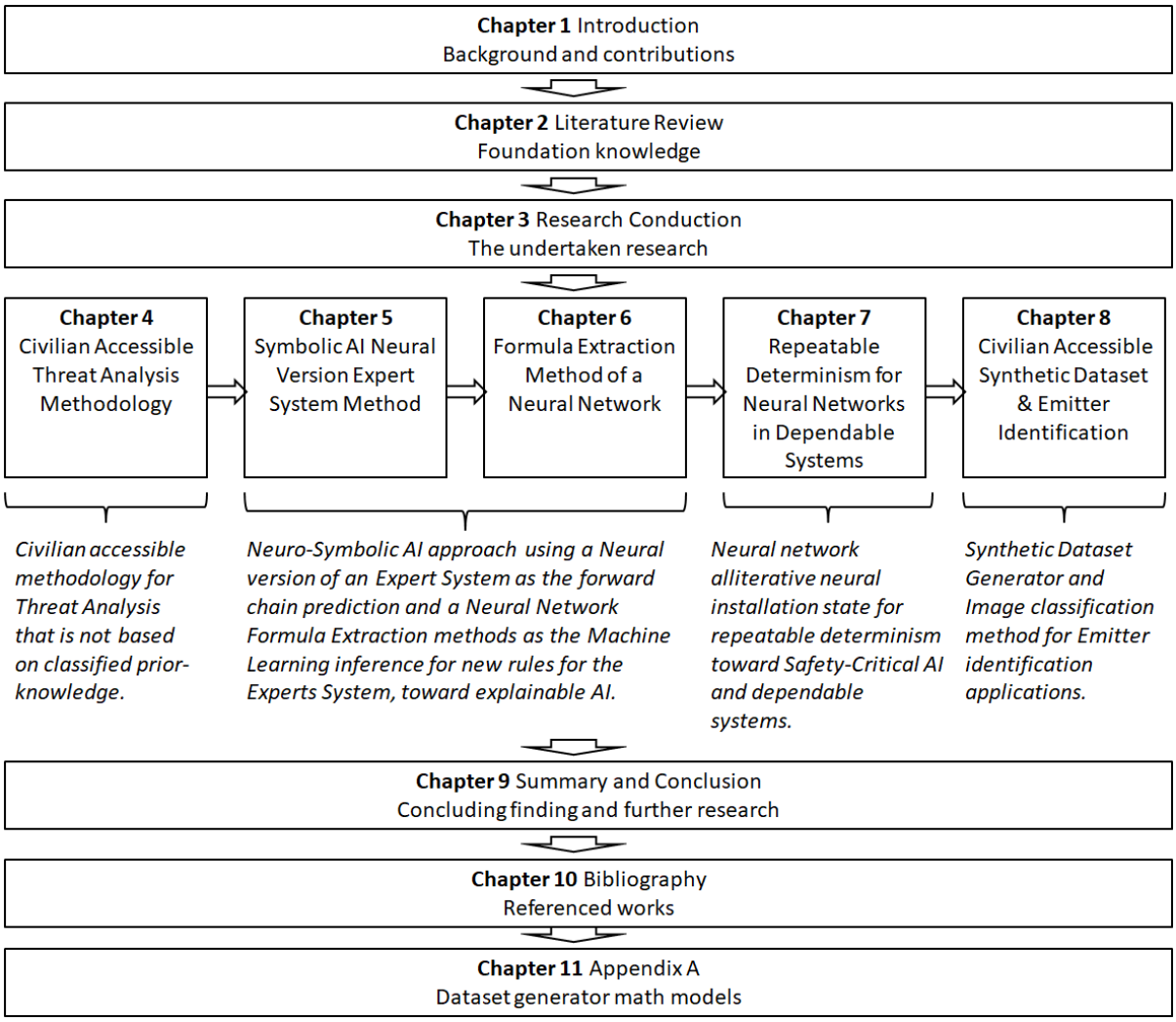


FIGURE 1 CHAPTER BLOCK DIAGRAM WITH RESEARCH THREADS

Chapter 2

LITERATURE REVIEW

This chapter contains the supporting literature, including the literature on the application area.

2.1 Cyber and Electromagnetic Activities (CEMA) Doctrine and AI

CEMA doctrine included Cyber with EW within Information Warfare as the greater area of CEMA military doctrine and now subsumes EW and Cyber as an integrated and coordinated approach for Information Warfare. EW and Cyber Operations are now overlapping components of the CEMA doctrine [56], shown in Figure 2.

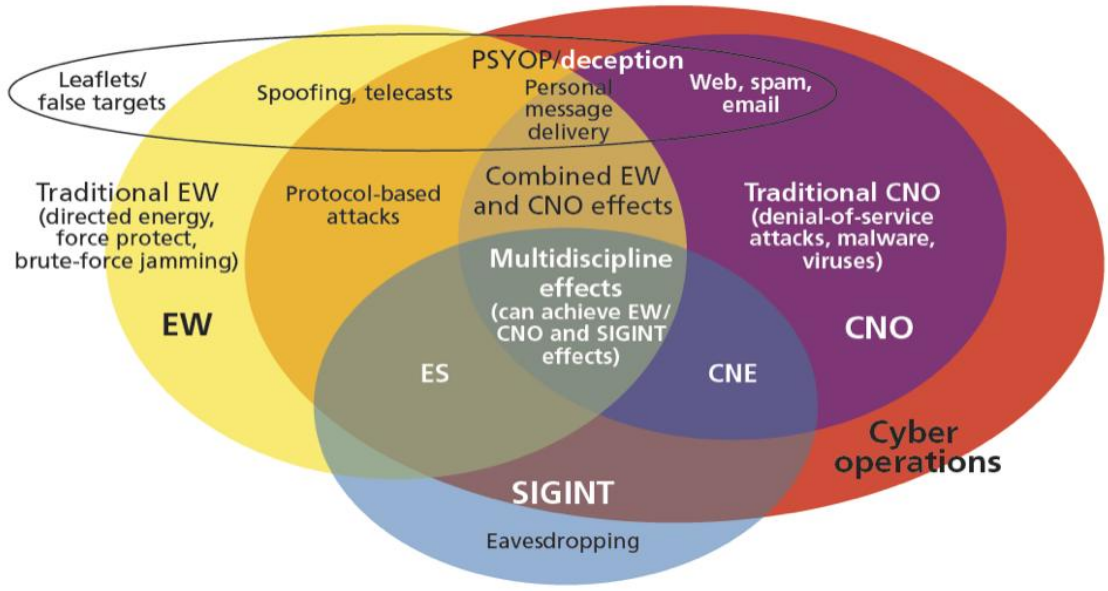


FIGURE 2 CEMA VENN DIAGRAM OF OVERLAPPING AREAS OF EW AND CYBER [56]

The CEMA Venn diagram in Figure 2 shows the area of EW overlapping with Cyber Operations and CNO (Cyber Network Operations) but also shows the EW supporting threat analysis area of SIGINT (as SIGnals INTelligence) that overlaps with EW as ES (as Electronic Support) and with Cyber as CNE (Cyber Network Exploitation). The fusion of EW and Cyber caused the SIGINT analysis to modernize and combine their approaches. Cyber and EW combine into layers from real-world geographical to virtual social connections via personas and identities as a combined intelligence product. The SIGINT analysis area was one of the challenges within the CEMA doctrine and integrated the formal military processes and reporting within the sub-branches of ELINT (ELectronic INTelligence) and COMINT (COMmunications INTelligence) with the modern processing capabilities and complex network-protocol exploitation of Cyber from geographical to social layers. This enhanced SIGINT capability between EW and Cyber could thus support collaborative planning and operations coordination. An additional advantage is that Cyber may also be less obvious and provide an ability to make a military effect that could be an alternative to the more obvious conventional and EW force effects, independently or in coordination.

2.1.1 Electronic Warfare (EW) Modernisation

There was also a modernization within EW, and Lambrechts and Sinha [57] define EW in the three branch divisions in Figure 3. Although this is quite widely accepted, not all literature is updated. This modern form subsumed the older definitions of ECM (Electronic Countermeasures), ECCM (Electronic Counter-Countermeasures), and ESM (Electronic Support Measures), which are within the more modern accepted divisions of Electronic Attack (EA), Electronic Protection (EP), and Electronic Support (ES) respectively. As such, the boundary of EW extended beyond the older EW

division's definitions predominantly based on platform self-protection. That boundary change of EW embraced the new scope of Information Warfare and is described in the book by Curtis Schleher [58].

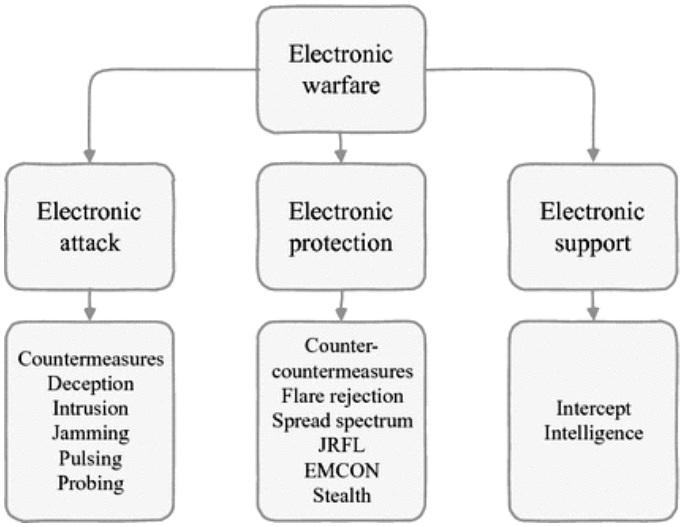


FIGURE 3 HIERARCHICAL STRUCTURE OF EW DIVISIONS [57]

The research application area is primarily within ES, intending to support EA against EP, and it allows threat detection and recognition to enable threat avoidance and countermeasures for platform protection. The division of EP is concerned with threat system hardening as protection against countermeasures and may also involve EMCON (Emissions Control) and JRFL (Joint Restricted Frequency List); this is still relevant as it affects the element of surprise and the restricted operations of equipment. When the CEMA doctrine combines with the application area, the Venn diagram areas in Figure 2 of EW, ES, and SIGINT are directly relevant to EW. The area of CNE is also directly relevant, as the modern influence caused the SIGINT area to become multi-disciplinary between EW and Cyber with a view to coordination between them.

2.1.2 Artificial Intelligence (AI)

A breakdown of AI techniques and approaches is provided by Galbusera et al. in 2019 [59], depicted in Figure 4. The breakdown of AI techniques has three main branch elements: Symbolic AI, Evolutionary Algorithms, and Machine Learning (ML).

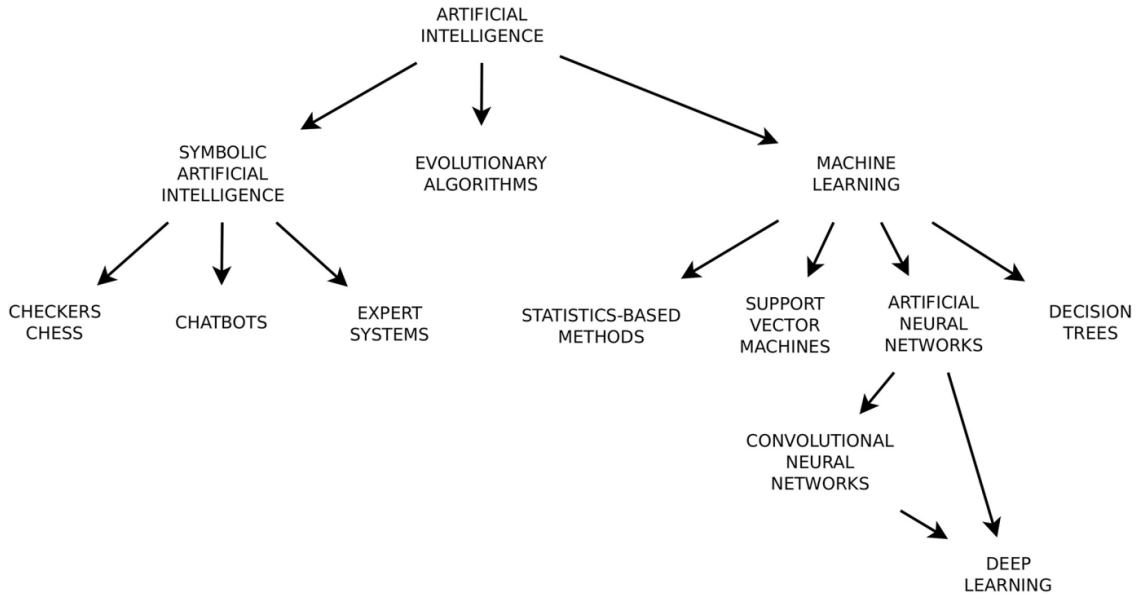


FIGURE 4 HIERARCHICAL STRUCTURE OF AI DIVISIONS [59]

In this breakdown, "Symbolic AI" (also known as Classical AI) is concerned with explicit declarative human knowledge representation [60] and, as a consequence, has higher abstraction from

the computer but is closer to human understanding [61]. By contrast, "Machine Learning" (ML) has lower abstraction to the computer and greater abstraction to the human [62]. "Evolutionary Algorithms" (also referred to as EA, but not in this research because of the confusion with the EW division name) offer an evolution of a solution, of which solution permutations can be concurrent. Concerning the application area, the branches of Symbolic AI knowledge representations, evolutionary algorithms, and ML offered promise, particularly if the abstraction gaps between the computer and the human can be bridged between Symbolic AI and ML.

2.2 EW Threat Analysis and Countermeasures with AI

Applications of AI to countermeasures have increased in prominence, and the US Department of Defense's BLADE programme aims to synthesize countermeasures based on detecting and characterizing new radio threats [63]. BLADE has a similar theme to DARPA's ARC (Adaptive Radar Countermeasures) programme [64]. Also, the interest in this area has reached into development processes in commercial entities, such as MASS Consultants Ltd [65], and anticipated research and development for block upgrades to military equipment capabilities [66]. One representation for a countermeasure in a mark-up language was called C3L. C3L is a public domain Open Architecture (OA) specification [67], [68]. C3L was also furthered as a Model-Driven Architecture (MDA) [69] in EW for EA and platform protection applications.

2.2.1 Military and Civilian Domain Threat Crossover

These approaches are in the military rather than the civil domain, but they show that the application of AI to EW is relevant in future capabilities. The publications associated with these approaches do not refer to the critical safety aspects of the techniques used. However, it is worth noting that the detailed technology in this area is not widely published owing to the security sensitivity. Brigadier General Alexus Grynkewich, in his article in 2007 [70], points out that, over several years, the build-up of military capabilities in the Pacific, Europe, Syria, and Iran introduced Integrated Air Defence Systems (IADS) for so-called Anti-Access Area Denials (A2/AD). Beyond Brigadier General Alexus Grynkewich's article on military concerns, military build-ups also represent a hazard to civil aviation, as in a Malaysian flight MH17 scenario over Ukraine in 2014 [71], or Iranian forces shooting down of Ukrainian flight PS752 over Tehran in 2020 [72]. More historically, the failure to coordinate military activities has also resulted in civilian losses, notably in the case of the unprotected British charted civilian freighter the SS Atlantic Conveyor during the Falklands war in 1982 [73]. The issue of civilian protection in the broader sense has been around for some time. An article provides many examples of civilian harm, the difficulty for civilians to protect themselves, and even their deliberate targeting [74]. Brigadier General Alexus Grynkewich (now Lieutenant General and Combined Forces Air Component Commander of the United States Central Command Southwest Asia) proposes that the future strategy of airstrikes from stand-off aircraft may need to combine with more Information Warfare and Cyber approach concepts like Data-To-Decision (D2D).

2.2.2 Research Gap in Civilian Domain Threat Analysis with AI

The gap for this research within the application area was that the use of countermeasures is reliant on EW threat analysis results, with the challenges of inaccurate and missing data, and that those threat systems can also change and evolve with their reprogramming capabilities. Thus ML that is safety-critical could provide an adaption that protects against changes in threat tactics or surprise war modes in a package that may require minor or limited specific predetermined analysis. Such an adaption may free civil domain applications from a dependency on specific classified data or restrictions in dissemination by making it less specific and more adaptable. ML may apply to the civil domain but must be safety-critical and, as such, may need to be constrained and validate-able. Moreover, Symbolic AI offers validation potential, given that the abstraction is closer to human understanding for validation and verification. When combined with safety-critical ML, this offers an adaption to cope with missing or low-quality available data and focuses on the gaps between safety-critical ML methods and Symbolic AI methods that incorporate ML.

2.3 Symbolic AI

Symbolic AI is a classical approach used in tightly bounded problems; one example: was Deep Blue (chess computer) that in 1997 succeeded in winning a chess game against Garry Kasparov. Deep Blue used a brute force method to search within nodes, arguably an 'exhaustive' method rather

than an intelligent one, but demonstrated real-time computation against a human opponent's intelligence [75]. The Symbolic AI methods are still used, with natural language processing within chatbots [76]. In Bologna's paper for 'rule extraction' in convolutional networks, methods for natural language processing have also featured in ML methods for sentiment analysis [77]. The Symbolic AI methods have been a candidate in combination with neural networks [78], which this research also aims at. Some examples are more readily humanly readable and use a declarative form, although that human may need to be an expert. These methods contain high-level abstracted representations of knowledge and involve human analytics. As NEURO-SYMBOLIC AI, symbolic AI, and neural networks are being brought closer [79].

2.3.1 Symbolic AI's Expert System

An example of a Symbolic AI technique is the expert system known as ES (but this research avoids confusion with the EW division name). Expert systems are an area where their applications have already been in systems with liability, such as medicine in the case of MYCIN and ELIZA [60], and military data fusion engines with 'imprecise' information by Rauch [80]. Vardaraju's paper [81] provided insights into AI when used as part of a development process. An expert system method was also proposed for decision support as DEX (Decision EXpert) by Bohanec et al. [82]. The intriguing part of this Bohanec et al. paper is that it symbolically represents quantities (like Low High) rather than numerical values as inputs, connects nodes in a rule structure, but calls on unity functions to apply aggregate function values from connected inputs with more primary attributes, and seems closer to a computational graph. Although the rules are if-clauses, that method has aggregation and abstraction levels in the symbols, where knowledge abstraction changes within the connected nodes. A paper by Voskoglou in 2014 measured uncertainty and used a fuzzy model for confidence from classroom experiments [83], and this is more from a human psychological approach, but then a paper by Johnson-Laird [84] experiments with logic flows from human mental reasoning in a human reasoning theory standpoint, but mapping to calculus and associations. Johnson-Laird also pointed out that there is no clear distinction between deduction, induction, and abduction in human reasoning, which can be because of different sequenced orders of learning as background knowledge. In relevance to this dissertation, abduction may serve as akin to the expert system's knowledgebase content permutations, deduction as the inference output from those widely accepted facts in that expert system knowledgebase, and induction: when based on experiences in ML, and is interesting because in safety-critical applications, the use of prior captured knowledge, like in an expert system, may have risks when that knowledge is added to by induction ML. So it follows that when an expert system is to learn, it needs an ML technique and must be validated or cross-referenced from abduction evidence, with the deduction conclusions to have confidence. That lack of extended abduction evidence was a deficiency which could be 'background knowledge' generalizations or perhaps 'common-sense,' and could have been part of the fall of Symbolic AI, and so a paper by Cook et al. [85] looked at confidence validation for rules as being heuristic rules with results from a database with image and textual content in the application of data mining.

2.3.1.1 Expert Systems and Explanation

A more historical approach is the Barzilay et al. [86] method; that method was an explanation built upon Explainable Expert System (ESS) and Reconstructive EXplainer (REX). The interest in this paper was the division of types of knowledge and that this forms a kind of inspiration that separates the domain-reasoning and communication knowledge, such that reused knowledge can be in knowledge compartments. It also asserts the need for security tracking against unintentionally revealing evidence combinations as a human counterpart might. These knowledge compartments can map onto the 'intelligence life cycle' [87] as a human reasoning pursuit in the application case-study subject area.

2.3.1.2 Expert Systems with Adaption

In a mission-critical role, a paper by Khalak et al. in 2005 [88] presented a multi-hypothesis method within the aerospace safety-critical system domain for a system in degradation. Rauch [80] looked at the probability conversion of expert system rules to probability data fusion quantities with 'imprecise' data. In an adaptive function, a Bayesian statistical analytical method for expert systems was proposed in a paper by Spiegelhalter et al. [89] to update rule probabilities and update the initial

specified probability values from the diagnostic results from the dataset. This method relies on prior analytics but includes a machine adaption element in optimization. A paper by Melen et al. [50] proposed an expert system with a learning capability to cope with environmental changes, using a rule-base controlled by a BN for the rule selection weighting adaption. The Melen et al. method was a weighting change of a BN for rule selection within the existing set of rules and conceptually would require every possible rule to be programmed for an arbitrary rule to be constructed. Therefore, this method is constrained to the original prior analytic rules, which may have been the intention. However, both Spiegelhalter et al.'s and Melen et al.'s methods were closer to a ML capability, although this is a machine adaption capability, and in that respect, the Melen et al. method is more advanced as the adaption is 'on-the-fly.'

2.3.1.3 Expert Systems and Machine Learning

There has been much research in Symbolic AI and expert systems particularly, and from the early beginnings, it solved the validation problem by including an expert. Furthermore, as the concept has developed, it brings statistical methods to confidence and provides, in some cases, multiple graded answers. The concept has graded its own rules; however, it has not generated new rules or machine-learnt them. Most of the expert system research stopped in the mid-1990s, and in the late-1990s, other techniques in ML that were furthering successes gained prominence. It may also be that Symbolic AI methods have a human understanding abstraction level far removed from accessible datasets that were becoming practical to process in ML. The ML methods furthering successes had a lower abstraction to the datasets but were harder to verify and limited to advisory roles in lower-order cognitive tasks.

2.3.1.4 Expert Systems Guard Equations and Random Forests

The expert system might have gained an ML capability by incorporating the ‘random forest’ algorithm. The random forest algorithm was proposed by Leo Breiman in 1999 [90]. The random forest is ensembles of trees rearranged through random parameter selection and sampling and then aggregated over those ensembles. A subsequent analysis in 2012 of the random forest algorithm by Gérard Biau [91] concluded that the algorithm was tolerant to sparsity and was not reliant on the number of noise variables. The expert system method interest is the rearrangement of decision trees, and decision tree rearrangement was also a subject within a paper by Shaikhina et al. [92]. However, rearrangement is not the learning of new rules but rather the grading and optimizing of their score through the rearrangement of precedence in those trees. This research dissertation establishes that the ‘guard equations’ in the rules provide a ‘valid’ set of rules for a possible machine-learnt component within an expert system. The guard equations define which rules are valid for inclusion with the final aggregated prediction. This approach allows decision trees to enhance the expert system rule knowledgebase from datasets and ML, rather than relying on the expert. That may also offer further enhanced understanding of the knowledge from a review process to verify the rule trust. An interpretation of this method is that it is an alternative approach to the Spiegelhalter et al. and Melen et al. approaches but uses random forests rather than rule probabilities with a BN. Furthermore, that approach would offer the node guards’ building, not just their rearrangement or grading, thus having some ML content.

2.3.2 Symbolic AI Summary

The area of Symbolic AI saw a rise and fall in interest and was an approach to general intelligence at one time [60]. However, the issue was the 'common sense problem,' or that common knowledge can be required to complete an understanding in adaption outside of the problem space, and this is what Johnson-Laird [84] referred to as 'abduction,' as an inference generalization with a measure of doubt. This idea conceptually made the knowledge required to extend outside the problem space, perhaps to an undefined boundary. Combined with this, the symbolic declarative nature of the knowledge representation meant that learning background knowledge requires crossing knowledge boundaries between differing human declared knowledge representations in the problem spaces and the available data available to the machine. Symbolic AI methods were overtaken in interest by ML methods, but some of those ML methods had safety certification issues. The strength of the Symbolic AI approaches, such as the expert system method, is that it is constrained but can be verified and validated by a human expert. Interest then emerged in Neuro-Symbolic AI, which seeks to combine

Symbolic AI with the ML neural network methods. Therefore, the knowledge gap crossed the abstraction boundary, and Symbolic AI approaches may have stalled rather than fallen. The Symbolic AI methods like expert systems offered a human-understandable form to be validated by an expert. Their high level of abstraction representation had difficulty bridging to the dataset and machine abstraction levels to allow the machine to learn directly.

Nevertheless, they still offered a symbolic form for humans to verify within, and when the abstraction gap can be bridged or converted between them, the Symbolic AI methods offer representations closer to human consciousness and reasoning as a representation bridge to ML. That then provides a mechanism for inductive-abductive learning (as an ML approach to a background knowledge *rule of thumb* mechanism in the context of Johnson-Laird) but is in a reviewable human communication form, which crosses the boundary between high order and low order cognitive techniques.

2.4 Machine Learning (ML) and Safety-Critical AI

From the late 1990s onwards, ML research re-invigorated in intensity, particularly in the speech and imagery domains. However, a drawback of some of the non-evolutionary ML methods is the need for large datasets. The learning process and underlying technology can create a model that may not have a realistic solution. In some cases, other methods such as fuzzy logic and expert systems were proposed in safety-critical applications instead by Ernest et al., Freitas et al., Lawson et al. [93], [94], [95], for UAV simulated air combat and feasibility in unattended space missions. Those papers did prove the demand for a solution for safety-critical applications. However, methods and techniques within ML were still relevant for adaption when datasets are known to be incomplete and thus are part of the literature review towards the application area.

2.4.1 Statistical Based Methods

Often statistical methods are also part of other methods or as a prelude to other techniques and can relate to framing, sampling, cleaning, scaling, and treating outliers before: model development, evaluation, configuration, selection, presentation, and prediction [96]. The BN method has been applicable in classifiers of ML and is also called a belief network [97]. These are parameters and a symbolic graph structure called a Directed Acyclic Graph (DAG). The DAG is used with parameters to express the joint probability interdependence of conditions towards inference predictions [98]. Within ML, a BN can be combined with neural network methods with noisy priors while also measuring uncertainty [99]. The Naive Bayes classifier was an example of a supervised method as it relied on the pre-classification in the dataset to prime the future prediction, and a BN can be most effective when the input data is independent [100]. A paper by Amor et al. uses the Naive Bayes classifier compared to decision trees applied to intrusion detection with competitive results [101]. The Bayesian theorem applied to a classification problem is in the form of Equation (1), where: the probability of the current hypothesis case assertion being true given event evidence is the likelihood of evidence *x* being 'true' given the current hypothesis case (*c*) assertion *P(x|c)*, times the prior posterior probability of the hypothesis case *P(c)* assertion being 'true,' and then normalized to the probability evidence *P(x)*. The probability evidence is the combined cases of being 'true' and the case of being falsely 'false.' This normalization of probabilities through the prior's multiplication and division of the evidence is a valuable attribute feature of Bayesian probabilities that maintains scaling in iterations of use while enhancing the probability accuracy over those iterations. It so can provide a helpful scaling over a recursive node structure too. The expression of Bayesian probability is:

$$P(c|x) = \frac{P(x|c)\cdot P(c)}{P(x)} \quad or \quad PosteriorProbability = \frac{Likelihood \cdot ClassPriorProbability}{ProbabilityOfTheEvidence} \; . \tag{1}$$

The Markov Model is a reinforcement learning method that predicts a better future state based on the current state and uses observations as the experience of that current state to statistically select a transition to a better future state based on rewards using gradient descent. It is highly applicable to gameplay and simulation, or anywhere there are sequences like speech and text [102]. However, in the basic form, it assumes that only the current state applies to that prediction, and an enhancement is the Hidden Markov Model using the Markov Chain to take sequences of hidden states into account in the prediction. Manogaran et al., in 2018, applied the Bayesian Hidden Markov Model with Gaussian Mixture clustering to DNA change detection in comparison with other existing techniques and

showed effectiveness [103]. Methods have also been dynamic with a Markov Model under a Bayesian framework as in the heard immunity paper by Haeussler et al. [104] and offered computational efficiency over other traditional dynamic methods like Ordinary Differential Equations (ODE).

2.4.1.1 Linear & Logistic Regression

In supervised learning, 'linear regression' is shown in Equation (2) [105], of which when there is one parameter has a 'straight-line graph' like form, and is known as simple regression, but can extend to multiple parameters and when it is, is known as multiple regression. Where x is the value input for prediction, applied to a prediction function $y(x)$ using the calculated intercept point ($intercept_{point}$) and a gradient ($slope_{rate}$) given the value of x and also a de-sensitization penalization term ($error_{regularisation}$). Linear regression calculates a sloped hyper-plane (gradient) from least squares as a prediction, with R^2 and P values as correlation metrics and a parameter significance measure used in optimization. By contrast, logistic regression [105] provides classifications between discrete value predictions in a curved slope between those predictions based on a likelihood measure (see Equation (3)). These methods assume the input dataset parameters are independent, and parameter independence is a regularisation subject [106]. Although regression dates back to the late 19th century by Sir Francis Galton, the applications to ML are widely known today. A novel modern application by Prosperi et al. in ML coupled regression to a rule-based expert system, offering an improvement to retrovirus prediction over the rule-based method alone [107]. Also, advances in hardware were presented in a paper by Sun et al. [108], with resistive memory (memristors) in linear and logistic regression performed in a single step by calculating the pseudo-inverse matrix of the data within the memory. The expressions of linear regression and logistic regression are:

$$(LinearRegression) \quad y(x) = intecept_{point} + (slope_{rate} \cdot x) + error_{regulaisation} \text{ , and} \tag{2}$$

$$(LogisticRegression) \quad y(x) = \frac{intecept_{point} + (slope_{rate} \cdot x) + error_{regulaisation}}{1 + e^{-(intecept_{point} + slope_{rate} \cdot x + error_{regulaisation})}}. \tag{3}$$

Generalized Additive Models (GAM) [109] is a generation of generalized linear models [110], which combines 'additive models' as a smooth function. The 'generalized linear model' allows a non-normal distribution through an iterative update method of a link function. Wood et al. [111] used this form in an 'electricity grid' load prediction application, and where the problem was intractable using the big data, the smooth terms provided 'penalized regression' with splines.

2.4.1.2 Regularisation, Variance, and Bias

Regularisation (or de-sensitization) is a method to avoid over-fitting the training dataset; it primarily attempts to reduce prediction variances while minimizing the data points' bias [112]. Principally, this is why the regressions are straight line derived rather than fitted poly-lines, such that there is a remaining cost within learning that assumes optimization has not seen all the data, which minimizes the error in prediction [113], also true when fitting curves to data, this is still based on straight lines but within sliding windows. However, the expression in 'loess-regression' prediction can still be parabolas in place of straight lines [114]. Within ML, a regression can offer a fit for prediction, but that fit may have deviations (or bias), which minimizes the bias to lower the variance between datasets, most notably between the training and validation datasets [115]. Linear regression requires that the dataset be as large as the dimensionality [116]. However, ridge regression [117] (also known as L2 regression) offers a solution to optimize sensitivity penalties, which is used in cross-validation and pertains to all the parameters. Lasso regression (also called L1 regression) is very similar, although it allows a slope of zero (dropout sparsity) and reduces or removes the influence of parameters that do not correlate [118]. A hybrid that will minimize Lasso and Ridge regression as a combined method is called Elastic Net regression [119] and arguably reduces the requirement for independent parameters in regression. Rauschenberger et al. provide an Elastic Net regression method with a stack-based combination of weights [120].

2.4.1.3 KNN, Hierarchical Clustering, and Classifiers

Clustering is an area of unsupervised learning, and K Nearest Neighbour (KNN) [121] is a prediction based on a new sample's shortest Euclidean distance from pre-clustered data. The pre-clustering performed can be by K Mean Clustering, where the value of K comes from an elbow graph

of the change in the reduction in variance using different values of *K*. That reduction in variance will change when the value of *K* passes beyond the smallest category's population. KNN differs from Hierarchical Clustering [122], a dendrogram method for reordering for clustering based on paired similarity, but in the Hierarchical case uses the Euclidean distance between the values rather than their positions. A hybrid between them in 2019 provides a Hierarchical K means clustering method by Nguyen et al. [123].

2.4.2 Support Vector Machines

Support Vector Machines (SVM) are a soft margin classifier using an affine subspace hyper-plane with a search for kernel functions to select a support vector classifier function operated in a higher order of dimensionality. That is to say; it uses a hyper-plane intersection with a set of distortion functions applied to the data to find an optimal classification boundary. Chang and Lin [124] provide a library of SVMs as LIBSVM. The SVM method is aimed at classification tasks and is tolerant to outliers and noise in the dataset, which is the reason for the 'soft margin' rather than a 'maxima margin.' The technique works well with balanced datasets (as an equal number of each classification in the dataset). However, Batuwita and Palade [125] proposed a fuzzy SVM supporting imbalanced datasets.

2.4.3 Decision Trees and Random Forests

Decision trees [126] can be readily humanly-understandable, although if unmanaged, quickly become complex, also where over-fitting may occur. Constructing decision trees from datasets uses a process to solve node precedence to minimize impurity equality using metrics like Gini. For information gain, that idea can be considered trading entropy (average logarithms of probabilities). That is to say, the decision tree node precedence in some arrangements has a more significant *information-gain* and thus lower *entropy*; this reduces the tree depth for most probabilistic combinations [127]. When applied to a dataset, over-fitting could occur if the method is exhaustive. So the introduction of stochastic sampling in bootstrapping and bagging, such that the rearrangement is more probabilistic combinations to a probabilistic sampling, with the application of the random forest algorithm as proposed by Leo Breiman in 1999 [90]. The random forest algorithm helps optimize without over-fitting the dataset, assuming that the dataset is incomplete in future predictions. Decision trees are also a helpful form that applies to the expert system method in the rule-base and raises the information gain for unseen data while reducing entropy and thus raising efficiency. Decision trees can be highbred to select between clusters as regression trees [128]; these regression nodes can be arrived at computationally by reducing the Sum of Squared Residuals (SSR) to set the node precedence and arrive at the thresholds, and this least squared residual is where the regression part of the name is derived.

2.4.4 Artificial Neural Networks (ANN)

Artificial Neural Networks (ANN) [129] can be an unsupervised learning technique, supervised learning, self-learning Crossbar Adaptive Array (CAA), and reinforcement methods too. They come in the primary types: Multi-Layer Perceptron (MLP), Convolutional Neural Network (CNN), Recurrent Neural Network (RNN), and Long Short-Term Memory (LSTM). Dense layers can be feed-forward or recurrent depending on the network type, where feed-forward is typical at the output layer of a CNN, in layers of MLP, and non-recurrent networks. A mixture of layer types is often used, separated by activation and pooling functions. When the number of hidden layers is more than one, they are said to be 'Deep,' and a recent trend was for 'Deep Networks.' In the area of dataset quality for neural networks and databases, a conference paper by Abdella et al. [130], although being an evolutionary algorithm method applied against an ML method, suggests: genetic algorithms for this application and finds that a Radial Basis Function (RBF) outperforms MLP networks in this application. Also, the application of RBF in a neural network instead of the back-propagation method was proposed by Pratiwi et al. [131] a year later in 2015.

2.4.4.1 Validation and Verification of ANN

The challenge of developing mission-critical applications focuses on development methodologies addressed by Kurd et al. [132]. Some of this effort is as part of validation assistance, and a survey paper by Zhang [133] encourages further research. Schumann et al. [134] also presented a method for a process with a validation and verification life cycle, and they define *verification* as correctness and

validation to refer to accuracy and efficiency. When applied to the traditional definitions: validation being correctness of the customer requirement capture process and verification meeting software specification, it thus implies the customer needs are accuracy and efficiency, and the model's requirement for correctness. Also, a paper by Hull et al. [135] looks at safety-critical application validation using an external testing tool and verifies number ranges with a statistical process to reduce test cases. A critical need for neural networks is datasets, and a paper by Tan et al. [136] reduces dimensionality by optimizing neural network inputs.

An example presented in the book: 'Hello World: How to be Human in the Age of the Machine' [137] of a model that provides high accuracy in classifying Wolfs and Husky. Paradoxically, that model would be using the snowy background for the classification rather than the canine in the photo. Therefore, by Schumann et al.'s definition, this example would be a failure of verification rather than validation, but these validation and verification techniques may not highlight that error. Nevertheless, there is another method in neural networks research, which is very pertinent to validation for mission and safety-critical applications, and that is the area of 'rule extraction.'

2.4.4.2 Rule Extraction

Bologna, GopiKrishna, and Hailesilassie [77], [138], [139] provide methods, surveys, and reviews of techniques using Rule Extraction. A list in Table 1 is from Hailesilassie's survey paper.

TABLE 1
SUMMARY TABLE CONTAINED WITHIN A SURVEY PAPER BY HAILESILASSIE [139]

Algorithm Used	ANN type	Algorithm Type	Extracted Rule form
DIFACONminer	MLP	Decompositional	IF-THEN
CRED	MLP	Decompositional	Decision tree
FERNN	MLP	Decompositional	M-of-N ,IF-THEN
KT	MLP	Decompositional	IF-THEN
Tsukimoto's Algorithm	MLP and RNN	Decompositional	IF-THEN
TREPAN	MLP	Pedagogical	M-of-N split, decision tree
HYPINV	MLP	Pedagogical	Hyperplane rule
BIO-RE	MLP	Pedagogical	Binary rule
KDRuleEX	MLP	Pedagogical	Decision tree
RxREN	MLP	Pedagogical	IF-THEN
ANN-DT	MLP	Pedagogical	Binary Decision tree
RX	MLP	Eclectic	IF-THEN
Kahramanli & Allahverdi	MLP	Eclectic	IF-THEN
DeepRED	DNN	Decompositional	IF-THEN

A safety-critical AI approach for ML in rule extraction is a step towards explainable AI. There are three main classifications of rule extraction approaches: decompositional, pedagogical, and eclectic. The mainstream methods are IF-THEN rules, decision trees, M of N, Binary Rules, and hyper-planes. According to Hailesilassie [139], the IF-THEN can be on a node by node basis, in typically a decompositional or eclectic classification. The form 'if and input then an output' can also apply a threshold to the IF condition to generalize. The 'M out of N' searches for Boolean expressions that may generalize and be used in decompositional and pedagogical methods and convert to IF-THEN rules. Decision trees are the most widely used for ML and data mining, as stated by Hailesilassie [139], as it provides structure and can convert to binary rules, whereas the hyper-plane is a clustering method. The pedagogical method provides structural insights from a black-box method, whereas the decompositional method creates detailed equations that are then generalized. The mainstream methods are for either MLP or RNN types of a neural network, but DeepRed extends the CRED method for Perceptron layers into Deep Neural Networks (DNN). GopiKrishna's paper [138] approaches the review from a sensitivity analysis in a pedagogical form and an opaque model (black box). Bologna [77], this method is for the convolutional network type and was published in April 2019, showing that the area of rule extraction is active. It applies to a textual sentiment analysis application in sub-networks propagated back to the input layer as the antecedent n-grams. The n-grams allowed an explanation for why the classifier worked well or badly, rather than a rule

explanation. Thrun's method [140] is from interval analysis and applies to a robot arm as a seemingly intractable problem, and this parallels EW threat analysis, where seemingly intractable problems and their solutions would need validation and verification.

2.4.5 ANN Hyper-Parameter Optimisation

The area of the configuration of hyper-parameters becomes relevant, not least the 'number of nodes' and 'learning rate,' and a paper by Baydin et al. proposed dynamically updating learning rate as part of the model optimization [141]. Also, a paper by Bergstra et al. [142] proposed random searches to find hyper-parameters in place of grid and manual forms. Nevertheless, a later paper by Bergstra et al. [143] argues that random searches for setting hyper-parameters can be efficient but insufficient in Deep Belief Networks (DBN). Probst et al. [144] looked at that problem regarding large scales for Deep Learning (DL) networks, with methods for estimating defaults and subsequent practical estimations. See section 2.4.10 for Meta-Learning use.

2.4.6 Repeatable Determinism

When testing safety-critical systems, desirable qualities are repeatability, and their assertion results are deterministic. An effect on this quality is the application of stochastic influences with random initialization states. Practical experiments showed that resultant model accuracy has variances in learning sessions with differing stochastic influences, visible above regularisation. Research experiments in this dissertation highlighted the 'shuffle algorithm' as an area of interest, and an alternative may provide further repeatable determinism by avoiding random sequences. Loshchilov et al. [145] also looked at the shuffle algorithm from a speed and performance point of view, but this research dissertation interest was from a repeatable determinism viewpoint. Misra et al. [146] is a related work, which reorders images rather than shuffles them, but is an unsupervised method and could conceptually be a technique for defining a sequence for a shuffle or indeed a 'kill chain' order in the application area. Although in the Misra et al. paper, the area is imagery, feature extraction of another type might extract sequences of signals in a radio spectrum.

2.4.7 Reducing Model and Dataset Sizes

Srivastava et al. [147] suggest dropout to prevent over-fitting and enhance regularisation. Zeiler et al. [148] method is statistical pooling for regularisation as an easy implementation method. Hinton et al. [149], supporting material shows some useful visualizations of clustering from the MNIST dataset [150], and Lin, Chen, and Wang [151] demonstrate differences in alternative pre-training strategies of reducing dimensionality through deep and shallow encoders using stacked Restricted Boltzmann Machines (RBM). Requiring smaller datasets is desirable in the application domain, as the datasets are often classified or hard to come by. However, other approaches to dataset scarcity are transferred-learning, the Generative Adversarial Network (GAN) method, and synthetic dataset generation, which are discussed later in this chapter.

2.4.8 Synthetic Dataset Generation

Due to dataset availability, research can generate datasets that contain the same or similar dimensionality of a real dataset but be available and unclassified while also in a controlled environment. The Master's degree thesis titled: 'Towards Synthetic Dataset Generation for Semantic Segmentation Network' [152] and paper [153] used synthetic datasets in the automotive domain, gained insights into unseen weather effects, and identified techniques for robust sampling for training. This method may provide unlikely observed scenarios inclusion to a dataset for training and the application area cover: war modes or increase dataset sizes for platform dynamics. Synthetic datasets are available from graphic engines and libraries [154], impact sound models [155], and Natural Language Generation (NLG) [156]. Also, in maritime surveillance, synthetic environment models can be enhanced by providing datasets to provide increased realism for generated synthetic datasets, as per Abdellaoui, Hubbard, and Duncan [157]. Jordanov and Petrov [158] used a real dataset after cleaning and transformation to compare with human judgment in an emitter recognition application, but the accessibility to these datasets in the civil domain is not assured. The dataset was also limited to the accuracy of the human judgments rather than the truth data as it can be in a synthetic dataset generator. Meta-Sim by Kar et al. [159] can automatically synthesize labelled datasets that could support a Generative Adversarial Network (GAN) method as the generative model of that network in place of the traditional generative neural method.

2.4.9 Generative Adversarial Network (GAN)

When datasets are available, they can be limited in the application area, thus building on the synthetic dataset generation and converting to a generative model trained through a GAN method. Such techniques have been used successfully in the imagery domain [160], and improvements are using style-based transfers by Karras et al. [161]. A paper by Yao et al. [162] applied it to underwater acoustic communication signals for classification purposes and found it gains a better accuracy than deep convolutional neural networks (CNN), but also without the need for detailed signal processing knowledge. Also, a paper by Zhang et al. [163] applied the GAN method to radar imagery, and that research improved a generative model's representation while providing a larger dataset. Such datasets can then be used for ML and be subject to transferred learning. More generally, the GAN method proposed in 2014 by Goodfellow et al. [164] had an optimization objective in Equation (4):

$$\min_{G}\max_{D} V(D,G) = \mathbb{E}_{x\sim P_{data}(x)}[log\, D(x)] + \mathbb{E}_{z\sim P_z(z)}[\log(1 - D(G(z)))] \ . \quad (4)$$

G is the Generator with data $P_z(z)$, and *D* is the Discriminator with data $P_{data}(x)$. Thus the optimization is the expectation, i.e., with noise removed from the Generator and Discriminator and moving in different directions using the *1-D(G(z))* in the second term, toward a global goal of minimizing the Generator and maximizing the Discriminator, and that is like a 'minimax game' form; However, the Generator objective would be equal to *-V(D, G)* strictly in a 'minimax game.' It is similar to the Noise Contrastive Estimation (NCE) and Maximum Likelihood Estimation (MLE) forms that use identical quantities, as pointed out by Goodfellow et al. in a separate comparison paper [165]. Both NCE and MLE forms use a model learnt and generator ratio in the update of the Discriminator. However, the GAN method uses a neural network with updates with ascents and descents of *V*. Because the GAN architecture of that network used an adversarial relationship between two networks, the objective is combined with the genuine data *D(x)* and generated data *G(D(z))* which are labelled and as such is a supervised method. Figure 5 illustrates a Deep Convolutional GAN (DCGAN) method in applying text to imagery from a survey paper by Esfahani et al. [166] reproduced from Reed et al. [167].

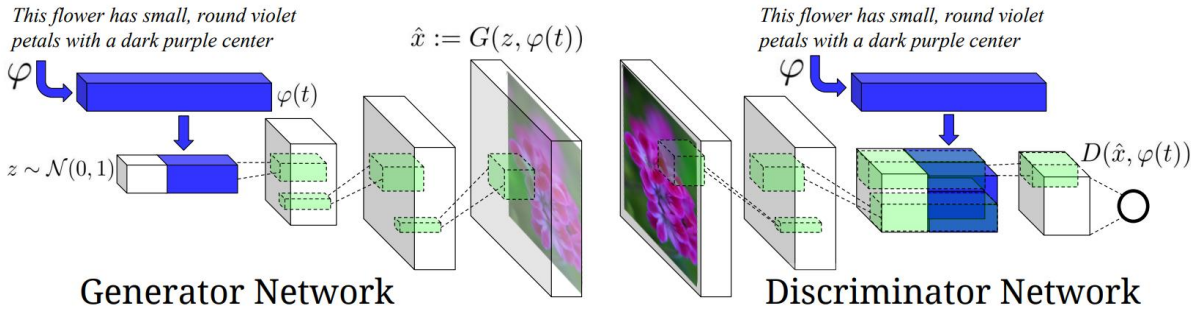


FIGURE 5 THE (DCGAN) ARCHITECTURAL METHOD [167]

In an update, the discriminator optimizes first to maximize cost, ascend the gradient, and allow convergence before updating the generator. The generator minimizes the cost as a gradient descend using only the *G(D(z))* term, as the generator is independent of the discriminator. Arguably, the *D(x)* term independence in the generator and the precedence in an update target the weakness in the generator by first optimizing the discriminator to be in advance of the generator. Furthermore, the independence also allows the generator to form different solution attempts, as the optimization is partial to the discriminator's objective. So another advantage is that the GAN method allows for more than one solution, and when compared with Mean Square Error (MSE), Lotter et al. [168] found that the GAN method could provide a more detailed prediction than the single solution methods like MSE. and that is because the GAN is not averaging to a single generalized case objective. The discriminator update precedence leads to the generator's objective (being part of the discriminator), and the result is multi-modal (or has separated clusters of solutions). The diversity of clustered solutions and their validity is where state-of-the-art currently lies. It would appear that diversity hinges on this update precedence and the partial optimization objective of the generator. The control of the generator's optimization is from the changing objectives of the discriminator, and the only part

of the objective is in the generator objectives, so it does not converge to a single solution in optimization. However, diversity can suffer from mode collapse, and previous solutions can be lost or generalized. An area of interest is increasing the number of generators per Hoang et al. [169], using dataset sampling to reduce mode collapse and increase diversity. Diversity relates to *latent space*, where a *latent variable* can provide a smooth transition between two qualities of several solutions in ML [170]. Diversity from those solutions can be mapped and found via the Latent Dirichlet Allocation (LDA) algorithm [171]. LDA has been applied to imagery, text, and music to provide the latent space variables. Thus diversity and validity in multi modes solutions is a topic of the current research [166], but the current research is also the area of high resolution. If such solutions can be valid and diverse with proper high resolution, this offers dataset creation with valid diversity and could also be a subject in transferred learning to provide comprehensive general datasets for subsequent specialization adaption.

2.4.9.1 Predictability Minimisation and GAN Methods Controversy

There was some controversy between Ian Goodfellow and Juergen Schmidhuber. Schmidhuber claims that Predictability Minimisation (PM) is related to the GAN method, and the GAN method is a 'special case' of Artificial Curiosity (AC), of which both AC and PM methods predate the GAN method. AC and PM methods are unsupervised reinforcement learning methods that play a *minimax game* between two networks, although, as pointed out by Schmidhuber, the conversion from data to encoding is inverse in the GAN method. Also, the RNN method is present in the equivalent generator network in the AC and PM methods [172]. That, however, makes the optimization objective have different quantities than the GAN method; also, the tagging of real and generated datasets at the input to the discriminator makes it a supervised or self-supervised method.

Nevertheless, considering the structure is inversely related, if the GAN discriminator provides rewards, it could view it as a reinforcement method, but that similarity would relate to any reinforcement method between two networks. Furthermore, the GAN method explicitly tags data between natural and synthetic datasets. It views the overall objective as optimization rather than reward. Hence, the update is also different as the GAN method's real data $D(x)$ and generated data $G(D(z))$ makes tagged distinctions, where the discriminator is in advance of the generator and has a partial objective in the generator, making it an impure minimax game. Whereas the AC and PM methods update predictions insight of rewards in a pure minimax game, the generator is in advance of the discriminator. That difference in update precedence and partial objective in the generator makes an essential difference to the two methods, as AC and MP are creatively generating for the discriminator judgment.

In contrast, the GAN method improves judgment to limit creativity, which arguably targets weakness. It may also be that the GAN method's generator being in a lag of the discriminator provides guided diversity as the generator converges insight of differing directions from the discriminator. That implies that, in GANs only, if there is differing sampling in several generators, increased diversity might occur, as per the findings of MGAN by Hoang et al. [169]. However, having several generators alone in the AC and PM methods may not have that effect, and arguably the equivalence might be to have several discriminators, and as such, the separated multi-mode solutions would be in different models. Still, that may also require different dataset sampling to achieve different modal solutions and, as such, would become separated generator and discriminator pairings in the AC and PM case. Schmidhuber argues that stochastically activating the encoders would re-sample in different units, thus not relying on pairings.

Nevertheless, the encoding conversion is inverse; thus, if they are related, they are opposites of each other, and the reinforcement of those two views is by Schmidhuber describing it as unsupervised and Goodfellow describing it as supervised. Those views emerge from the tagging of data and precedence of updates with the notion of the start point in the block diagrams' chain. Both methods differ in their approach. Their objective and similarity are merely using two networks in a minimax game-like strategy, but the GAN method's precedence of update and partial objective in the generator's optimization courses a more considerable distinction than the similarity between them. Furthermore, arguably, the AC method is a step further toward general intelligence as it is unsupervised.

2.4.10 Transferred Learning and Meta-Learning

Weiss et al. [173] and Tan et al. [174] provide surveys on Transferred Learning. Closer to the application domain, Huang et al. [175] provide a paper on Transferred Learning for synthetic-aperture radar image target classification. Furthermore, Zhu [176] used Transferred Learning with SVM for emitter recognition, which is very close to the application case-study area, and they used it for the same reasoning of the scarcity of datasets. Meta-Learning, by contrast, is the learning of learning, and Hospedales et al. [177], provide classified algorithm designs in a survey of methods: Meta-Optimiser, Meta-Representations, Meta-Objectives, and also applications. The neuro-evolution method [178] that GoogleAI used to learn network architectures through evolutionary algorithms is related.

2.4.11 Model Architectures

When considering model architectures, there have been significant developments. Lecun et al. in 1998 presented LeNet [179], which recognized handwritten numerical digits from black and white images. In 2012 AlexNet [180] was similar to LeNet, but it was larger and introduced ReLU, Softmax, drop-out regularisation, and max-pooling; and was also implemented on GPUs. AlexNet also won the ImageNet prize in image categorization of colour images. The Visual Geometry Group (VGG) at Oxford launched VGG in 2014 [181], which was deeper instead of wider; it also added the concept of parameterized blocks of layers. The extension in model size then led to the Network In Network (NIN) block [182] that used 1x1 convolutions instead of dense layers as an efficiency equivalence. The diversity of architecture components and when to use them led to GoogLeNet 2014 [183], which introduced the 'inception' block. The Inception block used concatenated combinations of 5x5, 3x3 convolutions, NIN, or multiple NIN blocks with max-pooling in parallel such that one of the parallel strategies would perform a benefit in the network, avoiding the block choice selection issue. In GoogLeNet version 3, 3x1 and 1x3 convolutions and later sequences of 1x7 or 7x1 convolutions provided column row sensitivity in those filters. Due to deep network sizes, Batch Normalisation offered an efficiency using mini-batch averaging with a separate mean and covariance. The Batch Normalization would turn out to have the effect of noise injection that would depend on the batch size, and as such, it can replace a drop-out layer role as a regularizer. In 2015 ResNet by He et al. [184] used residual connections like a layer skip-link route in the architecture that approximated an identity function and was like a Taylor expansion, which led to experiments on where to place the batch normalization. ResNext takes the ResNet partitioning into several channels and slices up the convolutions. DenseNet by Huang et al. [185] extended the tailor series to higher orders for mixed resolution purposes. By contrast, Squeeze Excite Net by Hu et al. [186] uses attention to focus on image locations with a global weight per channel. In 2018 ShuffleNet by Zhang et al. [187] shuffles the convolution output from the channels. MobileNet [188] would make separable convolutions within all channels.

2.5 Evolutionary Algorithms and Case-Based Reasoning

Evolutionary Algorithms [189] are stochastic search algorithms with heuristic optimization toward Genetic Algorithms (GA) and evolutionary programming [190]; they were also an inspiration from reproduction, mutation, inheritance, and selection in biological evolution [191]. Sometimes using metaheuristic (high-level partial search) methods [192] can yield a near-optimal solution in an adaption landscape, although it may not yield the optimal solution to the fitness landscape function [193]. That method also applies to intractable problems in parallel, but the solution may not be the only solution or fully optimal [194].

Case-Based Reasoning (CBR) is a step towards bridging Symbolic AI to ML [195]; it is primarily a memory-based method [196]. It is reasoning based on a previous similar experience of a case (or circumstance); as a generalization of similarity for adaption and experimentation to update the experience for a new generalization [197]. It can avoid high dependency on large prior organized datasets by constructing a generalization piecemeal from data exposed [198], and Low et al. present a multiple-retrieval method for incomplete datasets [199]. However, the performance by definition is more generalized when more problems are exposed. The method retrieves generalizations, including reasoning, re-uses and adapts them, and makes revision predictions and retention as a new generalization. It uses ensembles and stamps the definition of success or failure on the solution, and those solutions are variations. The method uses K-Nearest Neighbour and the Euclidean distance

[200]. Li et al. [201] present a fusion of evolutional fuzzy-based case-based reasoning, and using 746 publicly traded Taiwanese firms demonstrated an average accuracy of 92.36% in prediction; this supported experts' decisions. Hence, as with neural methods, the safety criticality is at a level of advisory support. CBR is a highly suitable method and may be forming a bridge between Symbolic AI and ML. Although CBR depends on prior experience, the data required need not be significantly large, but by definition, the larger the data, the better the solution will be.

When considered in the context of game-play with a platform-protection engagement software model, CBR and EA are quite well recognizable as practical solutions. However, the novelty of that approach may be limited, not providing a noticeable scientific contribution within the context of the research dissertation, and could be closer to applied science. The dissertation has chosen to research more novel methods in bridging Symbolic AI with ML, using a neural expert system combined with a formula extraction method, toward a higher potential for a contribution.

2.6 Summary and the Applicability of AI Techniques

The case study in EW threat analysis is in Electronic Support (ES) for analysis from collections, intending to support avoidance and Electronic Attacks (EA) and countermeasures, but also in the sight of the threat's own Electronic Protection (EP) hardening to those attacks. AI into EW approaches is a subject area that is growing in interest [63], [64], [65], [66]. The EW subjects within ES are threat recognition and intercept analysis, identification, cataloguing, and discrimination. Those may support countermeasure selection, tactic evolution, and optimization in forms that will reduce pre-analytics and increase adaption to new circumstances. A gap in platform protection has seen losses in the civil domain [71], [72]. Partly due to the access to threats, secure data, and threat movements, but also because of the absence of threat warning and countermeasure equipment as well as the lack of data to program them, platform protection has become more urgent with an increased likelihood of operating near to extended range military systems. High-ranking military stakeholders [70] have also identified that EW needs to incorporate more Information Warfare concepts from cyber like Data to Decision (D2D). Some published representations are in the public domain as open architectures [68] [67], [69], and these are accessible in the civil domain. The relevant areas of AI are Symbolic AI and ML. It would appear that a gap between Symbolic AI and ML is the conversion of abstraction levels, and Symbolic AI provides benefits to validation review. Others agree with this approach and are closer to a Neuro-Symbolic AI approach [60], [61]. When combined with ML, particularly neural methods, those methods need safety-critical hardening [202]. That critical safety hardening in neural methods may surround the issues of repeatable determinism and validation and verification. In repeatable determinism, the area of initialization and shuffles were of interest, and the area of rule extraction provides a bridge from ML to Symbolic AI. Symbolic AI is thus closer to validation and verification in a human reviewable form. Symbolic AI comprises many techniques but has converged into more complex expert systems and chatbots. However, Symbolic AI might represent a level of knowledge closer to a human consciousness level of reasoning and the neuron method at a sub-consciousness level. A view of why Symbolic AI methods have stalled is the acquisition of background knowledge or extended abduction knowledge in the context of Johnson-Laird [84]. When combined, two methods that might provide an ML mechanism for Symbolic AI are the expert system and the random forest algorithm method [90]. The random forest algorithm could be applied from datasets to verify or establish guard equations in rules selection. However, this may not be so far different from the methods of Spiegelhalter et al. [89] and Melen et al. [50] as an objective. Although, the method could be applied in a different form and can be used to construct the node's logic boundaries as guard equations rather than rearrange or grade the hierarchical structure of nodes.

Another approach is for mechanisms for establishing a standard format for knowledge in reuse, such that knowledge in other areas is compatible and aggregate-able. That format may have restrictions in the representation at the outset and thus may not be free textual but algebraic. Mathematical algebraic forms may still have human review-ability as a human-to-computer analytic bridge, and this also relates closer to case-based reasoning and neuron methods, which are numerical representations. Also, case-based reasoning may be a successor to expert systems, but it does not offer a humanly readable and reviewable form, thus not covering the human-computer abstraction gap. Although piecemeal learning is attractive, other approaches like the GAN method may offer

more benefits with latent parameters. The abstraction gap raised the interest in rule extraction methods in an algebraic form, with a consistent algebraic form also to be used in a Symbolic AI approach. That would have applications to understanding intractable problems and provide a human reviewable form of a model solution, thus generating validate-able and reviewable rules. That validate-able and reviewable form could also be a computationally compare-able form with existing compatible knowledge bridging the inducted abduction learning problem (as background knowledge acquisition). Such a form can be applicable, such that there is less requirement of genuine dataset accessibility, although approaches to datasets, mainly synthesized datasets, is an area of interest, also adapted through GAN methods.

The Bayesian method implies scaling in the iterative probabilities to the evidence and population and thus is extensible to problems that differ in populations of nodes, but Bayesian methods often require parameter independence. Bayesian Hidden Markov Models may offer countermeasure states as a reinforcement learning model, although this may be applied science rather than a contribution. Data clustering techniques benefit classification, and the Elastic Net Regression has a role in establishing which represented parameters are significant to the discrimination. The GAN method could apply to language parameter estimations using an accessible open architecture. A compelling area is MGANs with multi-generators to increase multi-mode diversity [169], and multi-mode diversity could relate to multi-states as latent parameters of a threat's kill chain in the application area.

This page is intentionally blank.

Chapter 3

RESEARCH UNDERTAKEN

The research undertaken is outlined in this chapter, considering the methodology for the methods developed.

3.1 Research Threads

There have been five research threads, which are:

- EW Threat Analysis, Electronic Support to Countermeasures,
- Neuro-Symbolic AI (Machine Learning Expert Systems and abstraction gap),
- Numerical Discrimination with Formula Extraction,
- Safety-Critical AI for Neural Approaches in Repeatable Determinism,
- Synthetic Emitter Dataset Generation and ELINT classification.

3.1.1 EW Threat Analysis Research

The EW threat analysis for civilian applications proposes a novel method for programming non-threats, as the data is more accessible in the civilian domain and defines alerts based on deviations from those civilian emitters. It philosophically uses available data in an onion of protection where outer layers require less threat information and inner layers more. It throttles the information requirement and reveals it by first using more accessible data. That method also provided countermeasure design considerations that directly conflict with a threat's intention through the layers mapped to an extended kill chain. That method also suggested diagramming and analysis methods based on first principle analysis with no prior knowledge. This method is sceptical of the data as reliable and tests the data with the first principle analysis to reveal the capabilities of the information towards defining their intention within a threat or non-threat system.

3.1.2 Neuro Symbolic AI Neuron-based Expert System

Missing data and imbalanced datasets are issues for neuron methods; neuron methods are less palatable to Safety-Critical AI applications. Expert Systems have been more accepted in applications closer to Safety-Critical AI but cannot learn new rules. Neuron methods, in contrast, have a learning ability but have explanation problems. The proposed method included virtualized neurons that map to individual input data and rule permutations and are semantically strong compared to a neuron method as the 'semantic' is over many neurons in the neuron methods. The method provides certainty and confidence that can adapt to the data based on the confidence and fit the body of knowledge's scope in the expert system rules. The method demonstrated is an EW threat analysis problem for antenna beamwidths. The learning capability is from a Formula Extraction method forming a Neuro-Symbolic AI method in the following research thread.

3.1.3 Numerical Discrimination for Formula Extraction

A neural network with an alternative input representation allowed more than one weight per activation and provided several weights for the activation's strength. The weights in activation strength allow a more robust input value range for low values and define more complex logic in a single layer. The method can also provide discriminating numerical operators deduced from the weights used in activation strengths. The work further reduced the number of neurons showing that the weights alone provide the function representation within the network. The formula operators extracted from weight relationships and the inputs can be reversed, revealing a hidden function, where this method's low loss and high accuracy are observations. This method is towards learning a function and extracting it to be used with the previous research thread in expert systems as a formula rule and is allied to the algebraic representation within both methods.

3.1.4 Repeatable Determinism towards Safety-Critical AI

An alternative non-random initialization state method was used to establish repeatable determinism in the neural methods, firstly in dense layers and then in convolutional layer methods. The latter used a method that demonstrated a better performance in image classification in both a test case and a challenging imbalanced colour image dataset of aircraft on runways. The method of the initialization state was more predisposed to the application of image classification and invoked earlier learning with less unlearning of the initial state. That work was also furthered with Transferred Learning using the FSGM method to provide a controlled distortion between that dataset as a control measure of dissimilarity, and the method was able to retain more of the original learning in that model and dataset.

3.1.5 Synthetic Dataset Generation and Image Classification for Emitter Identification

A synthetic dataset generator provided the (SD_CMRM_Iv1) [54] dataset and a machine learning framework with run-time programmable parameters to support future GAN work as a generative model. An image format was defined using an ANN image classification method, and the synthetic dataset generator demonstrated in an emitter identification application scored highly at 99.8% accuracy.

3.2 Methodology

The Alan Bryman book [203] is handy when considering a research methodology considering quantitative and qualitative approaches. These approaches are relevant to this research, as AI uses deductive, inductive, and transductive approaches, where the expert system method is a deductive reasoning approach, and the neural network method is an inductive reasoning approach. Bryman defines the quantitative approach as deductive, and with the attributes of epistemologically orientated to positivism, and ontologically orientated to objectivism. Bryman also defines the qualitative approach as inductive, with the attributes of epistemologically orientated to interpretivism, and ontologically orientated to constructivism. That idea means it could consider the expert system method deductive, as it uses existing established knowledge in a quantitative approach. Also, this is an objective approach as it has constrained the inputs and outputs in the rule-base, while those rules guide outputs as a positivist result. By contrast, the neural network method is inductive and qualitative, wherein it constructs a generalized model that outputs results based on interpretation from known inputs and outputs to build new knowledge. These research strategies are consistent with Aristotle and Sir Frances Bacon [204]. Also, Gill and Johnston [205] present 'deduction' as a theory tested through observation and present induction as observation to build a theory. These ideas discuss the philosophical reasoning of the expert system and neural network methods toward Neuro-Symbolic AI. Thus, that research looks at the current research output and the building blocks of two primary methods. It is targeted towards a mixed-method, combining existing knowledge building blocks of neural networks, expert systems, and experiments for understanding new knowledge.

3.2.1 Method

The Neuro-Symbolic AI approach has interests in both neuron methods for ML (an inductive approach) and Symbolic AI in the form of an expert system (a human reviewable and deductive approach). Therefore, it has both inductive and deductive techniques in the appreciation of the research area. Aside from the research problem space, the methods applied to conduct the research also have these qualities as they will use those tools.

3.2.2 Research Focus

Forming the research questions focuses on tackling issues where their findings may better understand the best or most viable approaches for contributions. The primary objective of this research is 'How can Machine Learning be applied in the mission and safety-critical field of EW threat analysis?' That question immediately led to the questions: 'What Applications of EW threat analysis can these AI techniques apply to?' and 'How can Machine Learning approaches have verification and validation with safety or mission-critical assurances?' A literature search showed growth in interest in this area, although it also identified that ML methods have challenges for certification and that Symbolic AI methods have challenges in ML. However, a new area is emerging that is highly relevant called Neuro-Symbolic AI. The area of Neuro-Symbolic AI led to the questions: 'How can Neuron approaches gain safety or mission-critical assurances?' and 'How can

Symbolic AI approaches perform machine learning?' As such, Transferred Learning and ML methods are relevant.

Figure 6 links the research questions to the research themes.

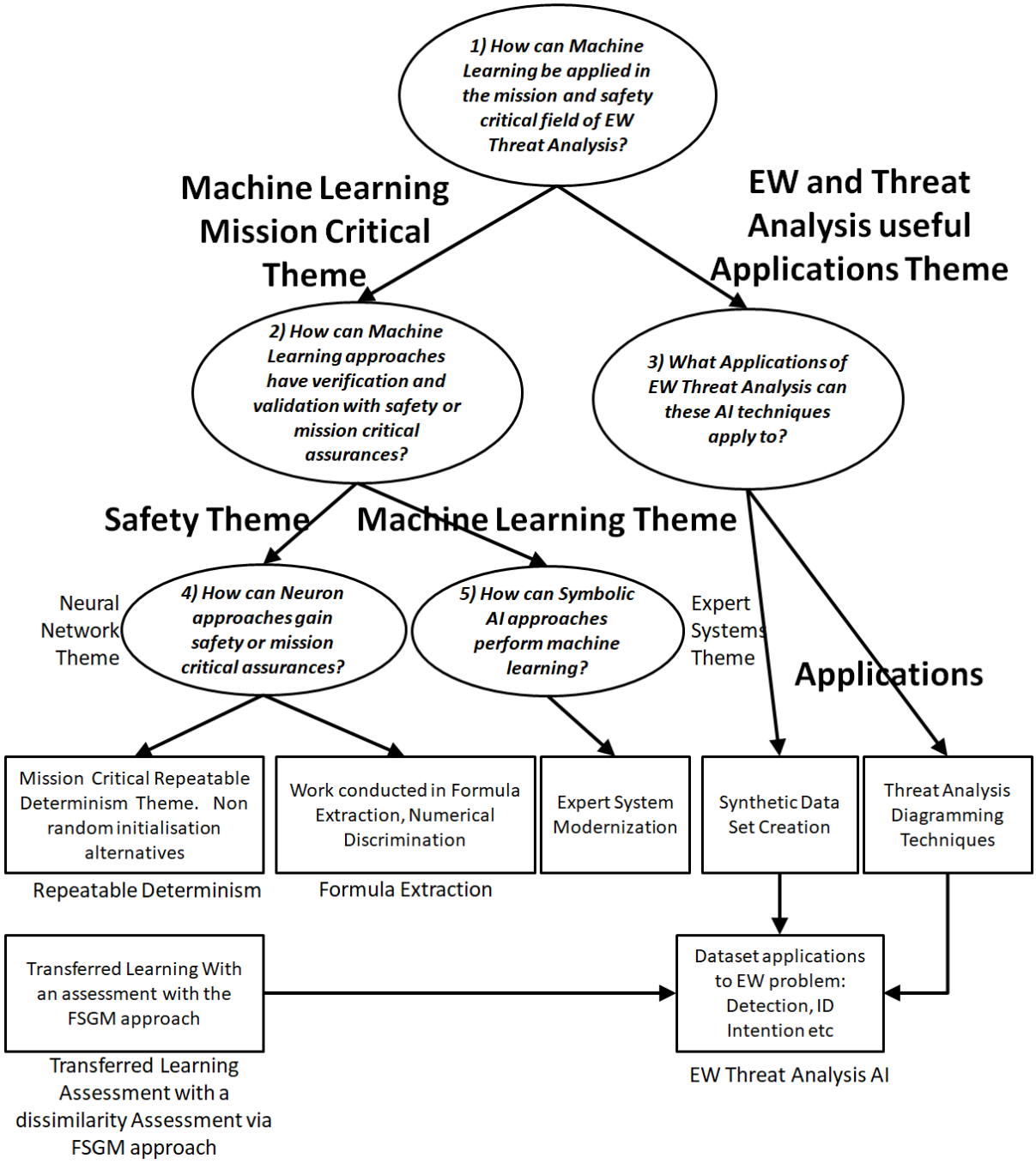


FIGURE 6 RESEARCH QUESTIONS AND RESEARCH THEMES

3.2.3 Main Methods of an Architecture

An approach to the methods could be to train a neural network, then extract the formula rules from the weights and apply them to an expert system method as the rules, where the newly learnt rules combine with the existing knowledge. This approach would be inductive and deductive, combined in a mixed-method, and arguably combining experience with knowledge. Those methods rely on a common language of knowledge that is algebraic. The neuron method with formula extraction will extract the body of knowledge for the expert system knowledge rules in a common algebraic form. These methods are data-driven, and a Synthetic Dataset Generator must provide the data source.

Figure 7 shows the research architecture linked to the application area problem.

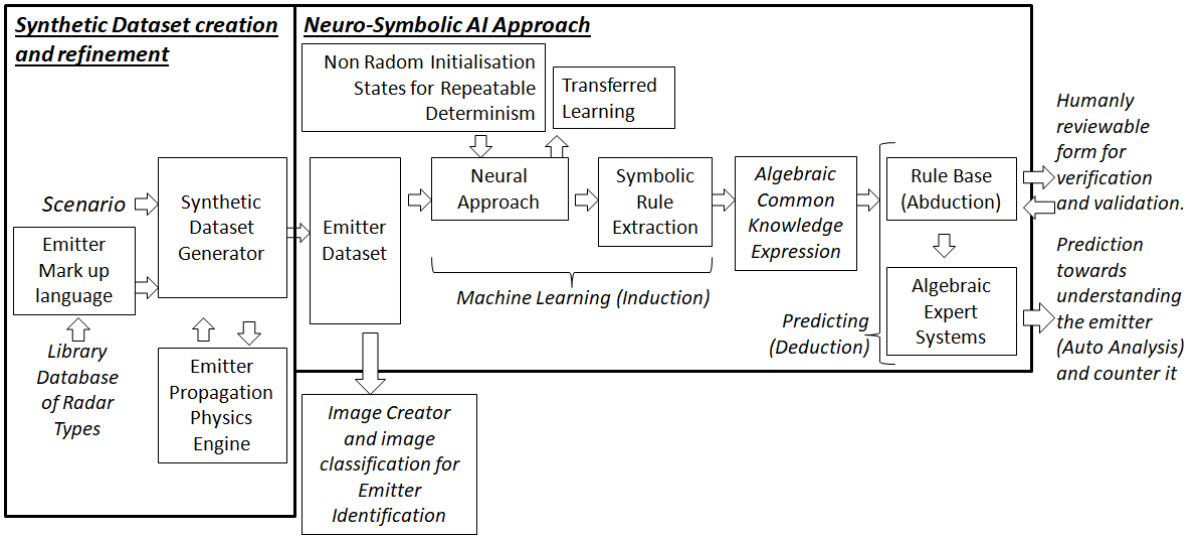


FIGURE 7 ARCHITECTURAL ORGANISATION OF RESEARCH THEMES

In the case-study application, required research areas can be deductive (i.e., understandings from established knowledge). Interest areas and gaps are deductive when combined with the literature review. There are challenges, and as part of the case study, there is a scarcity of data characterized by sparse and imbalanced datasets. Also, reducing the need for preliminary analysis could increase adaptability to environmental changes and coping with surprise. The application area also has a critical safety aspect, and although neural network methods offer an ML approach, they have both review and repeatable determinism issues. However, adding the application of a Symbolic AI approach to a neuron method aligns with the emerging research area of Neuro-Symbolic AI. Also, dataset generation is of interest but predicated on its safety-critical foundation. In the neural methods, the challenges were in repeatable determinism and are still in validation and verification; the literature review highlighted the area of rule extraction, addressed as formula extraction, although, in this case-study area also needed to address repeatable determinism in terms of random influences used. Additionally, from the literature review of the Symbolic AI approach, the expert system method also provides a captured knowledge that is humanly reviewable.

Chapter 4

EW THREAT ANALYSIS, ELECTRONIC SUPPORT TO COGNITIVE COUNTERMEASURES

This research chapter focuses on EW and threat analysis within the application area with processes and procedures applied to AI methods. There have been advances in threat technologies within the application area, and these technologies bring automation, autonomy, and faster response to threats. Threat technologies are also including AI methods as well [206], [207], but combined with this, there is an evolution of Integrated Air Defence Systems (IADS), and those advancements have hypersonic engagements, increased range, higher reach geographically, and more abilities against stealth aircraft. These problems focus commanders on combating these threats against Anti-Access Area Denial (A2AD), proposed with tactics prior to the launch and higher up the kill chain, with automation and adaption through ML.

4.1 Platform Protection and Threat Analysis

Designs of modern air and sea platforms include the integration of Command and Control (C2) and Defensive Aids Suites (DAS) in automation. These systems are modular and built on Open Architectures (OA), which incorporate complex software systems in broader roles. In the military domain, these systems rely upon mission data; without mission data, they can be less effective, as they depend upon it. That mission data uses the initial threat analysis in operational support processes to develop countermeasure protection.

Threat analysis is thus a sensitive and secretive process largely unpublished. However, with the authorization of UK MOD, a paper was presented and published [C5[1]] at the Egyptian Military Technical College (MTC) in Cairo within their library and then later published in open access by the Institute of Physics (IOP) [J5[2]]. That paper introduced how threat analysis can prepare an understanding and how diagramming methods support that understanding in Modelling and Simulation (M & S). Those diagramming methods support the analysis that can be synchronized and coordinated across platforms in cooperation, facilitating force protection.

It can use a countermeasure description language as a mark-up language called C3L which can exchange and store countermeasures in a standard interface format. The paper proposed an alternative to destructive military strike missions, and the general method is applied higher up an extended kill chain. While also being structured into layers of an onion of protection mapped to a Venn diagram of countermeasure design intentions with different data needs. As such, the model proposed makes a measured response preserving sensitive data while mapping those countermeasures to be a direct countermeasure to the threat's intention in every stage of the extended kill chain. That extension of the kill chain also naturally embraces cyber approaches in the CEMA doctrine approach.

Proposed with a similar approach to utilized civil shared networks that use AI methods in research for future systems, such as ATC [16], [17], AIS [18], [19], ACARS [21], [22], and MSSIS [24], [25], [26] networks. These networks may also coordinate countermeasure effects from cooperating platforms to assist civilian protection. Those shared networks thus provide a warning to avoid an engagement, evade the threat, or an attempt to defeat the weapon, as part of an MDO-like approach, but as a non-military force in coordination with dedicated standby capabilities, perhaps through deployment by a political agreement to avoid civilian losses. *Airstrikes* are the traditional strategy for

[1] [Cn] Published conference papers are in a separate bibliography on page xv.

[2] [Jn] Published journal articles are in a separate bibliography on page xv.

combating IADS but only apply in times of war. This research, however, embraces the A2AD and IADS problem in more non-lethal strategies as an alternative and uses the EWOS (Electronic Warfare Operational Support) higher up the extended kill chain for air and sea protection at all levels of that extended kill chain simultaneously. It is more applicable to the civil domain than military actions like the Suppression and Destruction of Enemy Air Defenses (SEAD/DEAD) as the alternative to that military strategy.

4.1.1 Complex Platforms, Stakeholders, and Modernisations

The 'projection of protection' requirement combines a diverse human stakeholder community. The communication strategies and analytical processes need to allow stakeholder types to contribute meaningfully to air and navy crews, mission production programmers, engineers, and scientists. The communication strategies are a challenge such that the contribution can be complementary; as such, diagramming and processes used in the threat analysis need to aid this communication strategy in that community.

Platforms and their threat weapon systems are increasingly more complex than their predecessors. These complexities increase relatively, and older platforms were already complex, but the pace of the complexity stretches as technology evolves with automation and autonomy. This complexity increase is due to modernization, technologies, tactic evolution, a need to reduce human decision times in faster responses, and generally a higher required performance.

Furthermore, threat platforms' roles are less confined and have broadened (as a role bandwidth). These evolutions use autonomous systems in the Integration of Air Defence Systems (IADS) and air and sea platforms with broader roles. Autonomous platforms have developed toward unattended sensing and self-organizing cooperating systems in the Marine, Air, Land, Space, and Cyber domains. It follows that matching technology is required between the DAS and the integrated threat platforms, which requires threat data availability, and that is a challenge in the civil domain.

4.1.2 Countermeasures, Threat Analysis, and Data Availability

Figure 8 shows a Venn diagram of the data availability as 'Spheres of Influence' presented in the Master's degree thesis [69]. Those Spheres of Influence are the "Protected Platform/Force," the "Defensive Constraints," and the "Threat or Weapon System"; they overlap and overlay with countermeasure design considerations of Decreased Detect-ability, Deception, Distraction, Denial, Disruption, Destruction, Design Proving, and Decoy. As stated in [69], it thus follows that the countermeasure considerations are coupled with the available data for the countermeasure tactic design. As stated in [C5] and [J5], the countermeasure consideration of a countermeasure's design implies that different data is required.

For example, when countermeasure considerations of Decreased Detect-Ability, Decoy, or Deception are in the design, the tactic can be more likely to use more protected platform/Force information and fewer data needs from the Threat or Weapon System and Defensive Constraints, as it is making a more attractive target and a less attractive Protected Platform or Force. These countermeasure considerations are soft kill deceptions and stealth using the protected platform's data. Those countermeasures are more applicable to the civil domain where that data is available.

When countermeasure considerations are Distraction, Denial, and Disruption and exploiting a weakness in the weapon system, fewer data can be available in the civil domain using more data from the weapon system.

When a countermeasure consideration is Design Proving in a trial or a laboratory environment, then the countermeasure can be exploratory from empirical experiments for equipment readiness.

Also countermeasures may not be releasable or employed via Defensive Constraints. It follows that if the countermeasure considerations are Destruction and Disruption, then a restriction in use may also be implied from the Defensive Constraints.

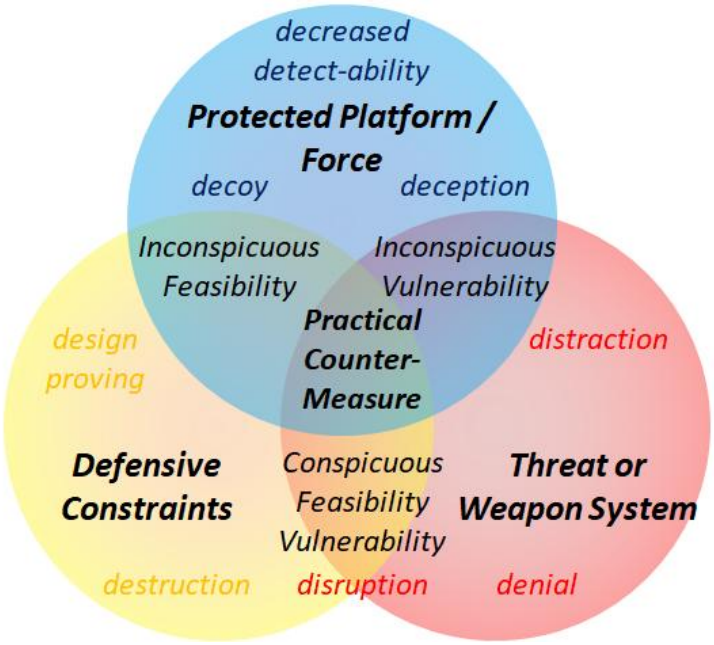


FIGURE 8 SPHERES OF INFLUENCE AND CM DESIGN CONSIDERATIONS VENN DIAGRAM [69]

Thus some kinds of countermeasures can reveal to an interceptor the detailed information available in the countermeasure design, for example, amplitude modulation timing or a frequency deviation. Knowledge of this can be used in EP by the threat. The countermeasure considerations in the design intention map to the information available, the onion of protection layers can also map to the countermeasure design considerations reserving that information to the inner layers and further along the kill chain. The threat's kill chain stages map to the onion of protection layers as counter intentions to that threat in an order that makes the countermeasure proportional to the threat's intention at that kill chains stage and, thus, a measured response. The onion of protection can make the detailed information only available within the inner layers of the onion of protection, mapping to more urgent kill chain stages. These ideas and an analysis method were presented in papers [C5], [J5], leading to an extended kill chain where data availability and managing data revealing is in an onion of protection. The onion of protection's outer layers applies to the civil domain in data accessibility. However, inner layers in that onion of protection can still use countermeasure considerations in outer layers through the onion of protection's concept that unlocks that data available in those layers as the threat engagement progresses through kill chain stages. Figure 12; presents the threat's intentions, and the countermeasure's counter-intention presents the extended kill chain.

4.1.3 Engagement Dynamics

Threat weapon systems have increased complexity and employed more technology towards their intentions, but the intentions have remained similar over time as defined roles. However, the platforms holding those weapon systems have also increased their missions, making the platform even more complex as the intention relates to several mission roles. For example, modern IADS has many role-specific missile types for intended targets [208]. Thus in civil protection, the engagement dynamics are further complicated as it may result from a miss-classification of that civil platform. As such, the protection needs to be flexible to engagement errors made in different roles that can imply different missiles intended for different targets, but where all missile types are lethal to an un-armoured un-defended civil platform. In the Falklands war between Britain and Argentina, the loss of the civil chartered SS Atlantic Conveyor was due to a lack of coordination and cooperation while protecting it [73]. Thus an error in coordination and cooperation of protection can also have further complexities when those errors occur on the protection side. Additionally, a further complication can be how sensors are employed and change ELINT emission sequences in a kill chain within an engagement. Nevertheless, those ELINT sequences may also reveal a use-case of those sensors and thus indicate the kill-chain stage intention and the equipment used.

4.1.4 Available Digital Networks and Data Links in the Civilian Domain

Air platforms: Operate under either Instrument Flight Rules (IFR) [209] monitored by ATC or Visual Flight Rules (VFR) [210] and are selected based on Visual Meteorological Conditions (VMC) or equipment failures. IFR, specified under Bravo airspace (busy airports) and restricted airspace (including war zones), as such civilian air platforms already operate under ATC monitoring when near war zones, allowing for centralized coordination. Aircraft Communications Addressing and Reporting System (ACARS) is a digital data-link service between aircraft and ground stations or satellites. ACARS interfaces with the Flight Management System (FMS) and provides flight plan and weather warning information. ACARS is also a transceiver and sends health and connection status to the network. The aircrews can receive and send messages via ACARS, and as such, ACARS can be a basis for a network for coordination.

Sea platforms: Also have digital danger warning receivers and transceiver equipment. The Navtex teleprinter [211], with a reception range of 200 NM, provides weather and other hazard notifications every 4 hours as an advanced warning. The AIS system can be a transceiver and is part of the Vessel Traffic Services (VTS) [212] with a live update. VTS is the equivalent of ATC, but for ships and has coverage in literal-water to deep-blue-sea with both terrestrial and satellite segments as a live network; as such, AIS and VTS can be the basis for a network for warnings, alerts, and coordination.

4.2 Threat Analysis for a Countermeasure Process

The military Threat Analysis approaches can be part of the Intel Life Cycle [87] and are highly data-driven with specialist collectors, exploitation equipment, and tools. The papers in Cairo [C5] and [J5] presented a threat analysis method in three main views: the Operational View, the System View, and the recommendations; this also utilized modelling and simulation for exploitation, analysis, and diagramming methods to aid the threat analysis to create a digital twin software threat model. Two views were from the Ministry of Defence Architecture Framework (MODAF) method [213]. Although in MODAF, there are seven views (Strategic, Operational, Service Orientated, Systems, Acquisition, Technical, and All Viewpoints). Arguably within threat analysis, five views are the same for any task as the task is constrained to threat analysis only. The Cairo presented 'Operational View' captures the system makeup and prioritizes components in the subsequent 'System Views.' MODAF has been replaced with NAFv4 (NATO Architecture Framework version 4) and uses architectural frameworks instead, but the 'Overarching Architecture' maps to the Cairo presented threat analysis 'Operational View' as the "What" in NAFv4. The 'Reference Architecture' maps to the 'System View' as "How" in NAFv4. The 'Target Architecture' maps to the 'Recommendations' as the "With What" in NAFv4. The general philosophy of the approach presented in Cairo is for a relevant countermeasure development and threat analysis within a compact scaled enterprise.

4.2.1 Operational View Analysis: (the mapping from NAVv4 "What")

The operational view of the threat analysis methodology presented defines what is relevant to a threat and how they are observed with sensors in data sources, chiefly in imagery, name referencing, and ELINT. These sensor observations are called the observables and catalogue the system components as observable collections. The threats have some observables that aid the recognition of threats, as observable artefacts of observation with a sensor and are called discriminators. Discriminators can thus help the employment of sensor mixes over other confusables systems to provide recognition when combined. The operational view compiles a catalogue of the alternative components and the confusables systems with observable discriminators. The structure of components also forms the capture of recognition specification and aids in filing new information in deducing if that data is relevant to the collection. In Figure 9 are radar silhouettes of some of the SNR-75 Fan Song radar variants of the SA-2 Guideline SAM threat series. Figure 9 shows that shapes of threat components can relate to variants with different capabilities as imagery observable discriminators and are highly applicable to image classification, detection, and segmentation in ML.

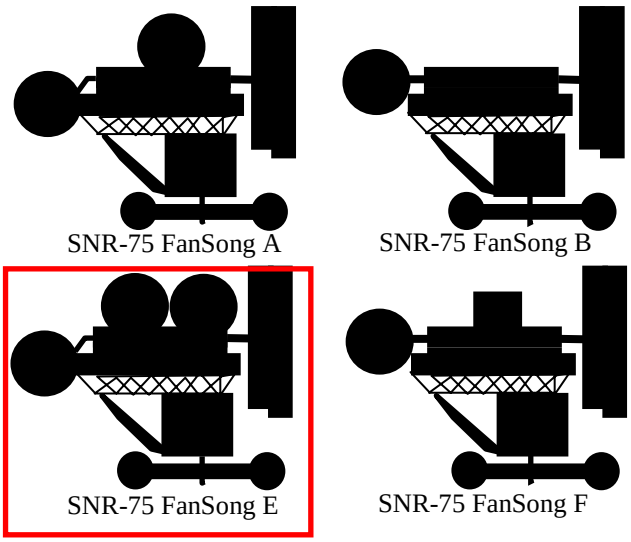


FIGURE 9 SNR-75 IMAGERY DISCRIMINATORS (LUMPS AND BUMPS) [J5]

An AI perspective provides a basis for dataset creation, employing classification, object detection, or segmentation tasks. AI methods are not limited to silhouettes and can use colour images on different angles and backgrounds. This research dissertation in Chapter 7 demonstrated a mission and safety-critical approach to convolutional neural network initialization with an application to classify aircraft on runways with the MTARSI2 dataset [53]. The MTARSI2 dataset is a challenging imbalanced dataset with variations of different lighting, shadows, time of day, aspect angle, and image resolutions. Additionally, the LSTM and RNN methods can be with Natural Language Processing (NLP) techniques from textual description sources. Also, within this research dissertation, an AI method for ELINT classification was demonstrated in Chapter 8. Combinations of these discriminators with those AI approaches can be applied to aid recognition and provide mechanisms for large-scale data processing.

The threat system components prioritize the countermeasure development order urgency rather than a priority that a countermeasure might have precedence in an engagement. Air threats are listed in the threat effector's range and classified into: dogfight, short, medium, long, and extended range. The classes are to remove range advantage from a threat and develop countermeasures employed in further ranges first. Surface threats are first listed into range and kill chain positions, lowering the development priority of lower altitude scan volume reaching systems if the threat is not a threat in takeoff or landing for an air platform. In Figure 10, the scan volumes of a threat system's components are shown with the system's functions (EW/HF/TTR), as this also shows the anticipated detection order when approaching a threat into the lethal area. If the ELINT discriminators are different within the system's components and functions (which is highly likely), then it also follows that a classification method as proposed in Chapter 8 can assert the kill chain stage and thus the intention to be countered. That method is also wideband and has higher data dimensionality than the traditional template method, providing a higher discrimination potential in classifying modes and functions.

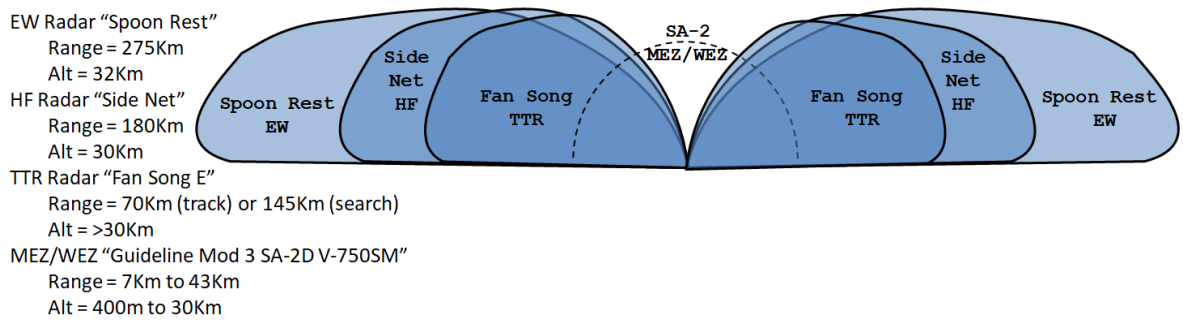


FIGURE 10 SAM SYSTEM SCAN VOLUMES AND REACHES [J5]

Jamming and ELINT detection system components can be further triage filtered for equipment capabilities in terms of their bands of operation and other restrictions. Please note that the frequency limits provided in Figure 11 do not relate to actual equipment. The method in Chapter 8 provides a further discriminator potential with the coordination in time and frequency of wideband emitters.

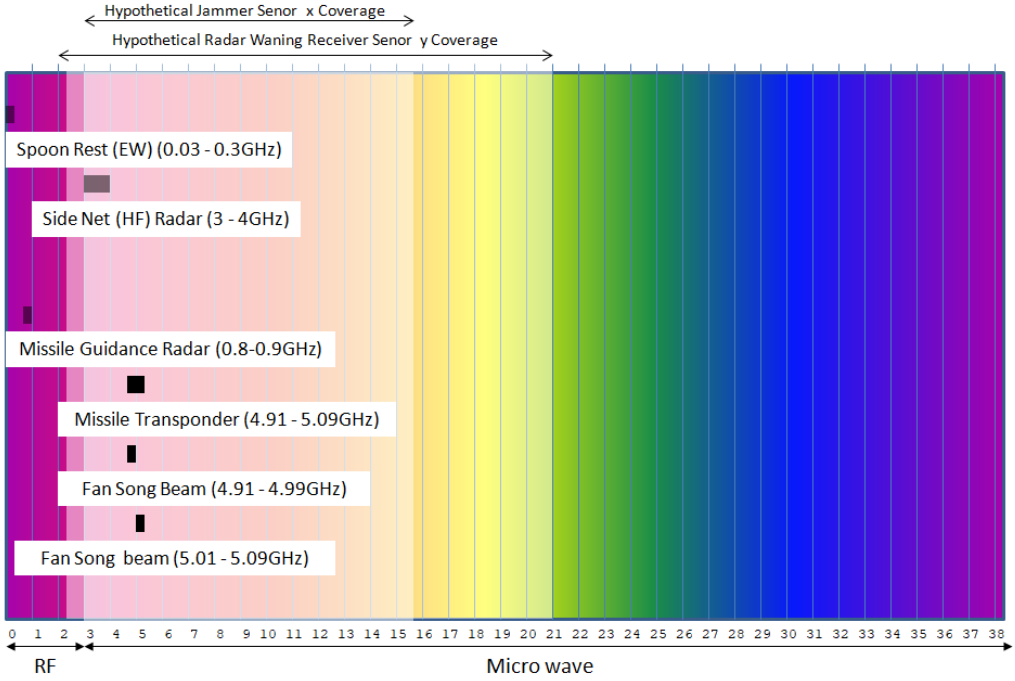


FIGURE 11 S-75 RAINBOW SPECTRUM PLOT [J5]

A list of references: [214], [215], [216], [217], [218], [219], [220], [221], [222], [223], [224] are the open sources used for the S-75 data presented including the frequencies and system understanding etc. This understanding can therefore construct the extended kill chain. In Figure 12, the extended kill chain was proposed in the Cairo paper and is an application of the ISTAR kill chain but applied to a SAM operator's point of view. Thus the SAM operator's intention at each stage is to be directly countered by a countermeasure as a direct counter-intention to the SAM operator's intention. This kill chain is extended higher up the kill chain (or earlier in the kill chain) and to more threat components of a threat embracing the nature of IADS.

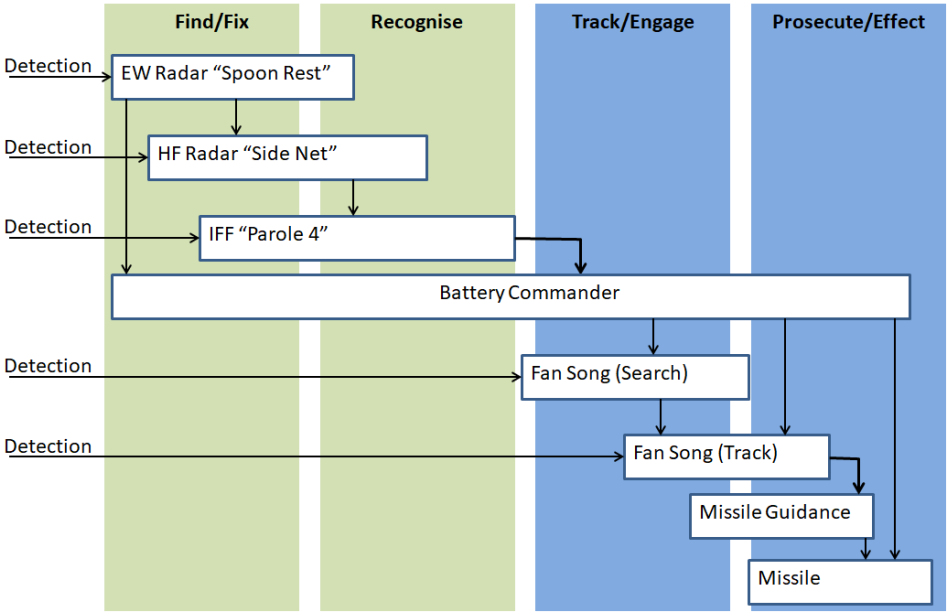


FIGURE 12 EXTENDED KILL CHAIN [J5]

- Find is the detection in partial coordinates like two-dimensional space and can relate to Early Warning functions.
- Fix is where the coordinates are completed and could relate to height finding, ranging, or accuracy in positional fixing.
- Recognise is a filtering activity to classify targets from background traffic.
- Track is when the prediction of a target is improved.
- Engage is when the engagement is decided on and may see signal changes in ELINT like illumination and idle guidance signals.
- Prosecute is the launch or release of the effector missile when guidance is active.
- Effect is the effector's terminal guidance, fusing, and detonation.

The Operational View of the presented threat analysis method has defined "what" is to be defended from, catalogued the threat components, identified their discriminators, prioritized the countermeasure development work, and finished with an extended kill chain. The threat components can now be applied to the system view in countermeasure development priority order and define "how" they work in each threat component.

4.2.2 System View Analysis: (the mapping from NAVv4 "How")

The system view of the presented threat analysis methodology considers the threat components in 'How they work' from their observables, oriented towards building a digital twin in software engagement modelling for susceptibility assessments. Commonly and repetitively, in each subsequent parametric analysis step, the ELINT parametric parameters are used with the data infilling of missing parameters from the neural expert systems method in Chapter 5 for sparse and imbalanced datasets.

Within each threat analysis step of the presented methodology, the parameters are capability tested, and a compiled set of vulnerabilities, strengths, weaknesses, and opportunities in each analysis step bring towards a countermeasure tactic selection against a mode-line and its' deduced functional intention from this presented methodology's system view. The completed captured threat parametric parameters are in the C4L emitter description language format for modelling with a digital twin using the emitter description portion of C4L of Chapter 8.

4.2.2.1 Electromagnetic (EM) Carrier Parametric Analysis

The EM carrier is fundamental to a system as it dictates the range of wavelengths that interact in the environment and has applications with atmosphere propagation while also relating to Doppler ambiguity. EM carrier has properties like bandwidth, pulsed / Continuous Wave (CW) / Interrupted Continuous Wave (ICW), agility, and coherency in terms of their emissions; these may indicate the kind of transmission device used, and thus processing as well as the expected channel plan for de-confliction against mutual interference. The standard analysis step in section 4.2.2 is applied, in which the strengths, weaknesses, opportunities, and vulnerabilities are set *Ua*, mapped to the parametric parameters combination as *Va* and possible intention *PIa*, as *A*: $Ua \rightarrow \{Va, PIa\}$.

4.2.2.2 Antenna and Beam Parametric Analysis

The antennas and beams can differ in transmission and reception, but the parametric beam parameters can also indicate the likely function from the beam orientation, geometry, and beam shape type, singular or cumulatively. The antennas and beam parametric parameters are Effective Radiated Power (ERP), polarization, and the antenna's receive gain. These are essential to how the energy interacts with the clutter environment, target, the Jamming to Signal Ratio (JSR), and the ability to detect the threat for triggering a countermeasure. From imagery, estimates of the beam shape and antenna beam gain can be estimated from the carrier and aperture areas from the method in Chapter 5. The standard analysis step in section 4.2.2 is applied, in which the deduced strengths, weaknesses, opportunities, and vulnerabilities are set *Ub*, mapped to the parametric parameters combination as *Vb* and possible intention *PIb*, as *B*: $Ub \rightarrow \{Vb, PIb\}$.

4.2.2.3 Beam Scanning Parametric Analysis

The beams scans can also differ in transmission and reception in each threat model mode, and an estimate of the threats' modal function can be from the: beam geometries, scanned volume, depth of modulation, and rate of revisit towards an angle measuring method or modal intention such as search,

acquisition, tracking, illumination, data linking, and clutter suppression. The standard analysis step in section 4.2.2 is applied, in which the deduced strengths, weaknesses, opportunities, and vulnerabilities are set *Uc*, mapped to the parametric parameters combination as *Vc* and possible intention *PIc*, as *C*: $Uc \rightarrow \{Vc, PIc\}$.

4.2.2.4 Intra-Signal Modulation Parametric Analysis

The intra-signal modulation relates to the modulations within a Continuous Wave (CW)/Interrupted Continuous Wave (ICW) or Pulsed modulation signal. The intra-signal modulations parametric parameters may include frequency excursion, frequency offset, phase coding, and amplitude modulations. They can be cumulative, contiguous, or in N-let pulse trains, providing an estimate of range resolution, bandwidth, and intra-signal processing gain, thus calculating the receiver's pass-band-filter bandwidth, tracking gate size, and parts of the processing gains can be estimated. The intra-signal modulations can apply to resolution and range given their processing gain relative to their lower observable probabilities at different ranges. The standard analysis step in section 4.2.2 is applied, in which the deduced strengths, weaknesses, opportunities, and vulnerabilities are set *Ud*, mapped to the parametric parameters combination as *Vd* and possible intention *PId*, as *D*: $Ud \rightarrow \{Vd, PId\}$.

4.2.2.5 Inter-Signal Modulation Parametric Analysis

The inter-signal modulations are the modulations that change between the Intra-signal modulations, and these could be ramp directions and bandwidths, coding or changes in the rates, or inter-signal modulation periods such as modulation timing types of staggers, jitter, switch dwell, sinusoids. Inter-signal modulations types are essential for processing, thus processing gains, mutual interference, clutter rejection features, and thus are tested with the inter-signal modulations against the instrumented range and range ambiguity, and the maximum tracking speed and Doppler ambiguity. The inter-signal modulation's repetition rates are treated as a Pulse Repetition Frequency (PRF) and classified into Low, Medium, and High PRFs (LPRF/MPRF/HPRF), defined by their Range and Doppler ambiguity. A particular case is added as UPRF when the PRF is unambiguous in range and Doppler. These inter-signal modulations are grouped and fitted to the known threat mode instrumented ranges and max tracking speeds. Grouping across modulations repetition rate types of fixed continuous, sweep, stagger, jitter, switch and dwell highlight relationships within the groupings. That relationship may relate to sub-modes as features or where further processing is dependent, as they have the same operational range and Doppler constraints. Those groups could imply different or further processing. These sub-mode features can relate to further processing required and could represent a jamming opportunity when denied. Also, sub-modes may relate to switches and buttons on the threat's panels, such as automatic and manual range tracking or high and slow speed target types.

These sub-mode classes of operator sub-intentions may have ELINT discriminators for separate countermeasure triggers as their indicators. Specific countermeasures can be reactive to the operator's switch positions during the engagement, thus directing the countermeasures to counter those operator sub-intentions. The further analysis evaluates the sub-mode and groupings with the inter-signal modulations period sequences in the groups against processing objectives: range and Doppler anti-eclipsing, MTI cancellation, False Returns Uncorrelated in Range (FRUIT) of second time around returns, observed frequencies from ambiguity for unfolding. This step and further analysis can update the groupings and their intentions. Applying the standard analysis step in section 4.2.2, where the deduced strengths, weaknesses, opportunities, and vulnerabilities are set *Ue*, mapped to the parametric parameters combination as *Ve* and possible intention *PIe*, as *E*: $Ue \rightarrow \{Ve, PIe\}$.

4.2.2.6 Mode Line Analysis

Each parametric parameter tested was loosely coupled to other parameters for programming into a modelling solution. The mode lines are the configuration combinations of the individual parametric parameter combinations. Thus they represent the combination of the parametric parameters of observations of modes. Those mode lines can relate to transmitting emissions and receiving processing in combination. They can be combined with the protected platform signatures to calculate the JSR overhead required for a countermeasure technique and the detection thresholds needed for

triggering that countermeasure, which can help down-select the viable countermeasure techniques or identify that they need combining.

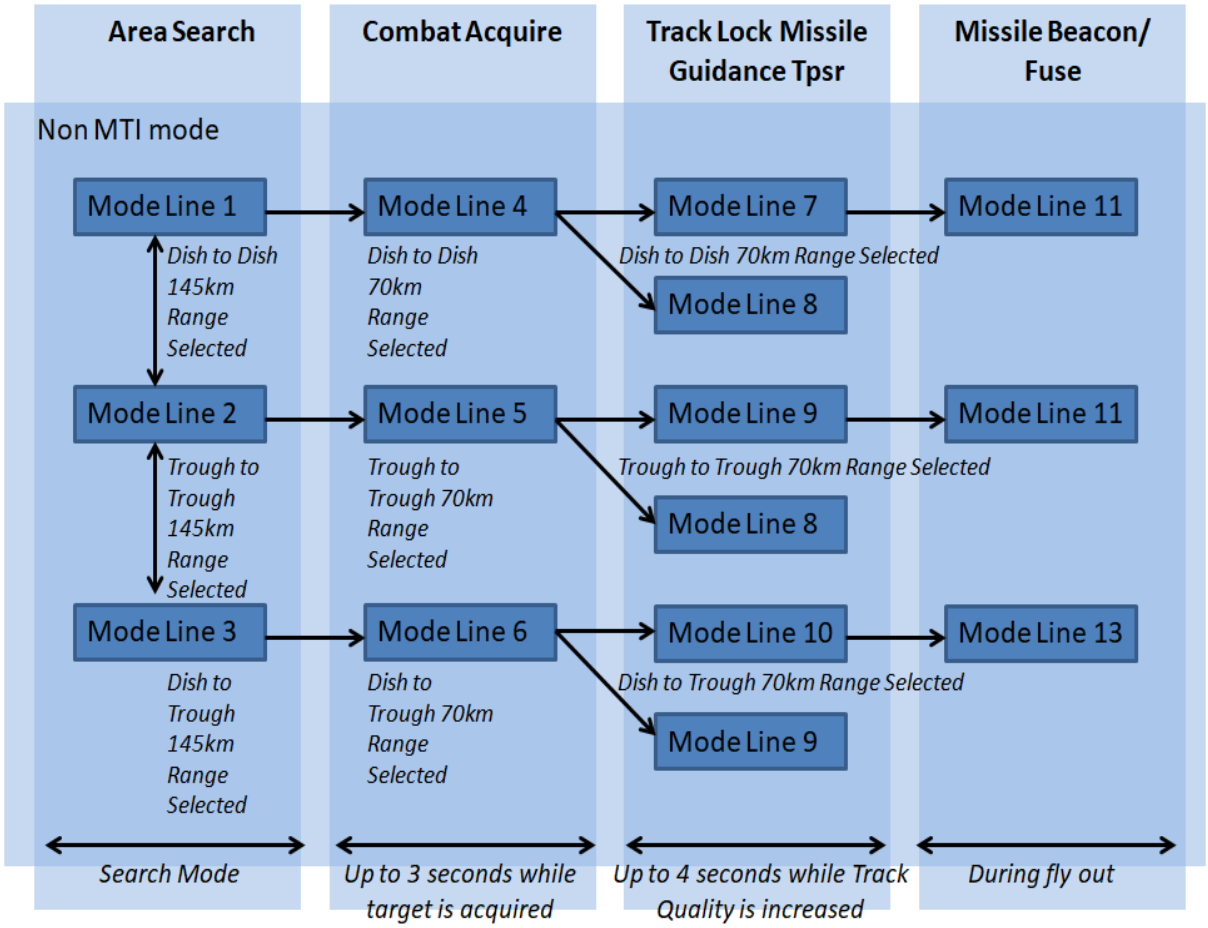


FIGURE 13 MODE LINE PERMUTATION DIAGRAM FOR AUTO TRACK AS SWIM LANES [J5]

The transition sequence analysis of the mode lines in the system-specific kill chain diagram is in Figure 13.

The ELINT parametric parameters may have missing mode line data. The neural expert systems method in Chapter 5 will infill those parameters using the method in the neural expert system; that method can also be used for the candidate countermeasure technique parameters when forming a pallet of tactics for later selection.

The combined mode line (*ML*) parameters, capabilities, and the vulnerability, strength, weaknesses, and opportunities of each analysis step towards a countermeasure tactic (*CM*) selections are made against a mode-line with its' deduced functional intention, together with the entry and exit transitions in the kill chain as in Equation (5):

$$ML = A(Va) \cup B(Vb) \cup C(Vc) \cup D(Vd) \cup E(Ve) \cup EntryTransition \cup ExitTransition \quad (5)$$

The collated individual parametric analysis results of each mode-line used and the tactics can be further down-selected from mapping mode line intention, countermeasure (*CM*) counter-intention, and the deuced system kill chain position and processing in Equation (6).

$$CM \rightarrow \{KillChain\_system, KillChain\_Extend, SignalPx\} \rightarrow \{ML \cap viable\}\} \quad (6)$$

The signal processing type (*SignalPx*) and their blocks captured for those states are in Figure 14 with a non-SNR-75 example of a system signal processing diagram.

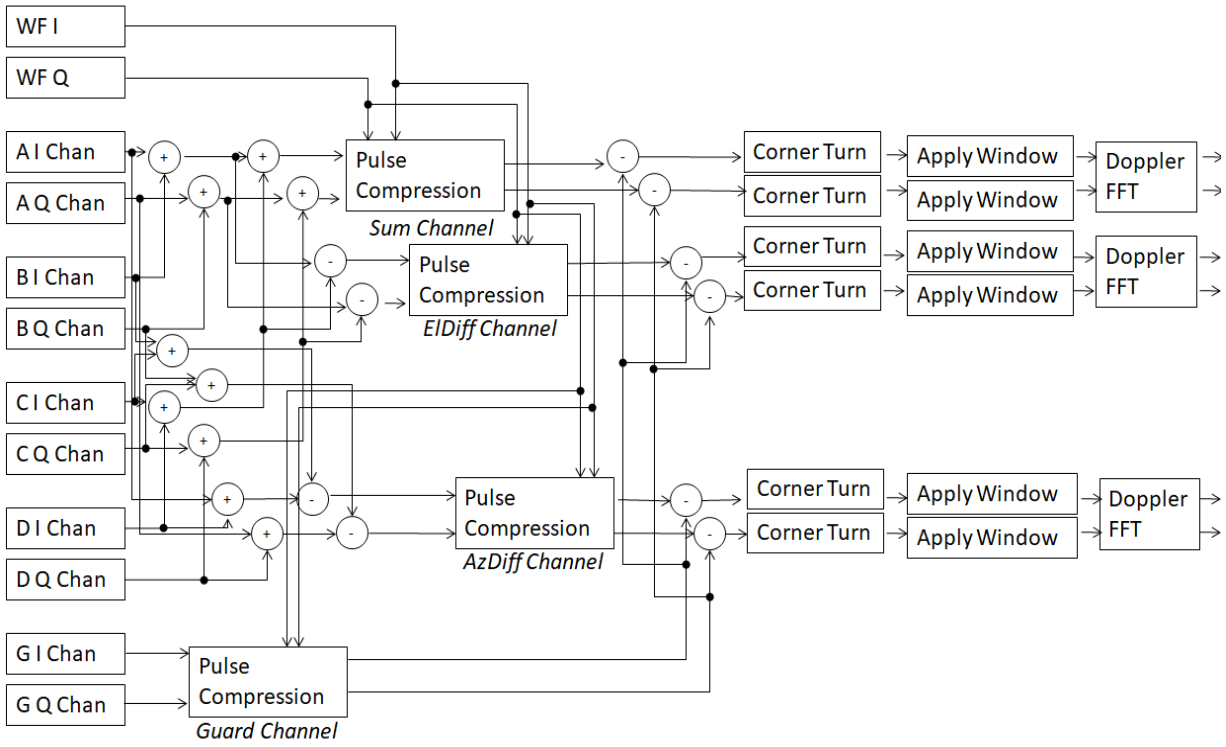


FIGURE 14 SIGNAL AND PROCESSING BLOCK DIAGRAM [J5]

Compiling those states and motivations into a UML state transition diagram for capturing in modelling software is illustrated in Figure 15. This mode line state model enriches the C4L emitter specification in Chapter 8 for the threat emitter modelling using the 'Schedule' lexicon token in the emission description language.

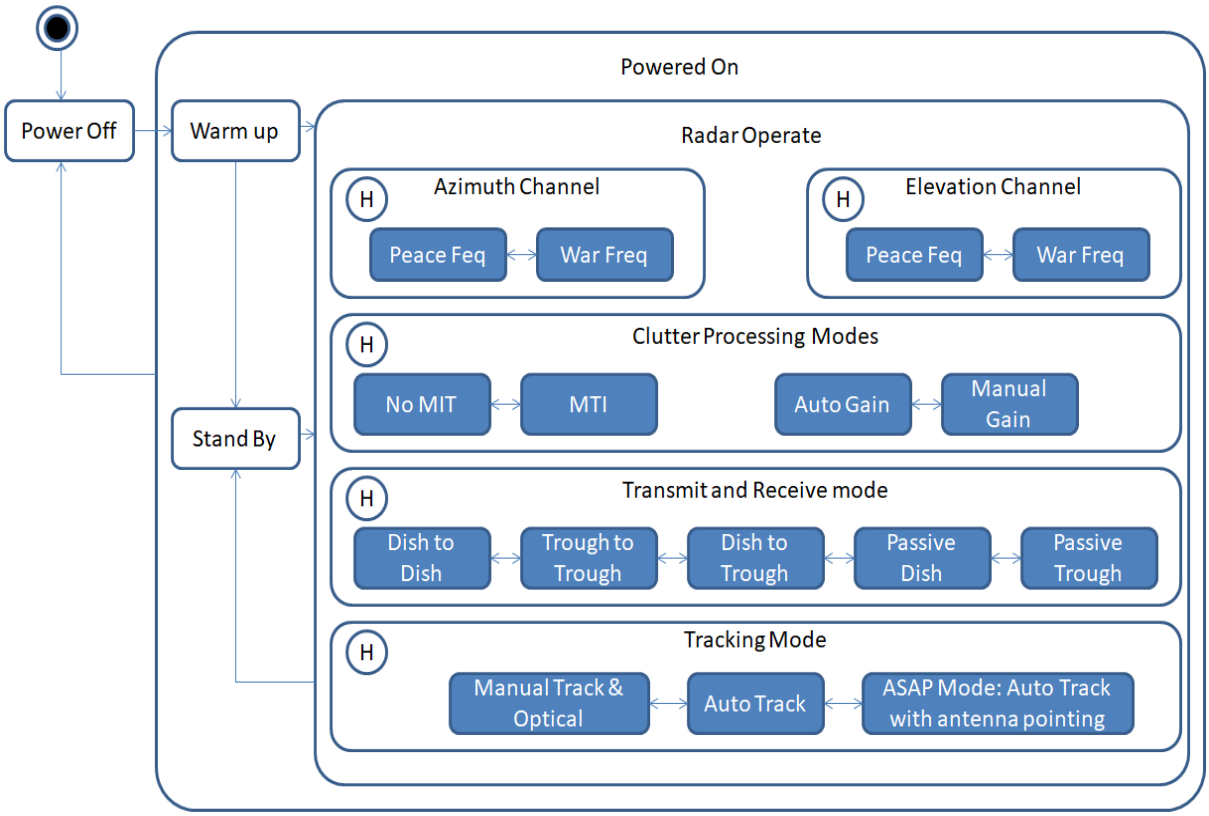


FIGURE 15 UML S-75 STATE DIAGRAM [J5]

Within an engagement, the Phases of Flights of the effector (Missile or Bullets) implies governance in motivations for state changes of intentions with those phases of flight, and again a non-S-75 example of the phases of flight diagram is shown in Figure 16, as S-75 is command guided only. The phases of flight diagram analysis also form the requirement for modelling for missile firing sequences in terms of guidance type and guidance laws, should they be different in different phases of flight or missile salvo sequences. This method also prioritizes the countermeasure tactics in the pallet of viable countermeasure techniques.

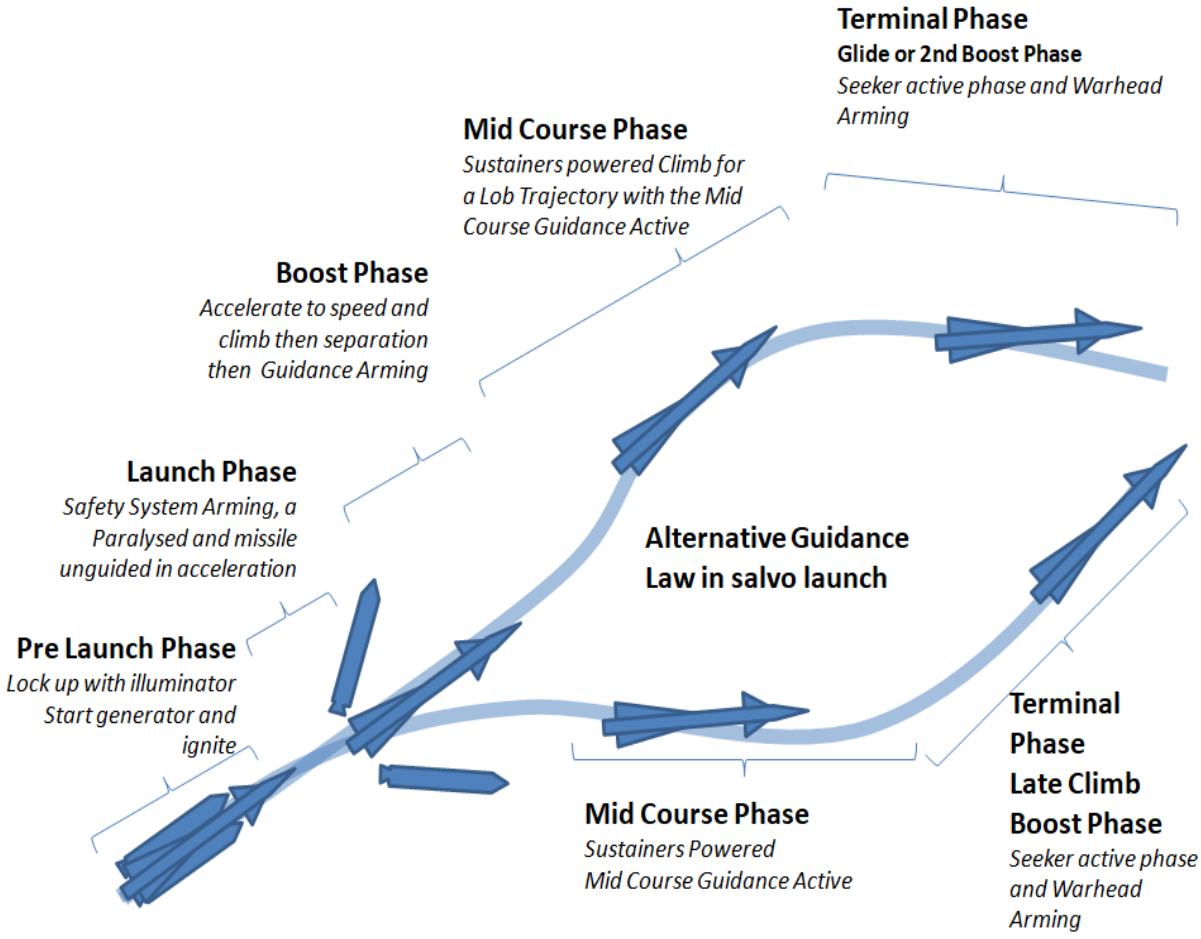


FIGURE 16 'PHASES OF FLIGHT' ILLUSTRATION [J5]

A modelling environment such as CounterWorX-PROTECT can represent the mode lines, signal processing chain, phases of flight, and state model, so the analysis and diagramming presented are realizable at that fidelity. The software model development methodology for arriving at the software model through the threat analysis presented here is part of this dissertation's research. CounterWorX-PROTECT also integrated the C3L episodic-countermeasures specification language into that threat modelling software, extended in this dissertation to include threat emitter descriptions with the countermeasures. See Chapter 8 for the synthetic dataset creation method using an emitter classification method.

4.2.3 The Onion of Protection Mapping Method

An Onion of Protection concept in Table 2 was developed with the IADS and the A2/AD problems in mind, resulting from the threat analysis and modelling using the extended kill chain presented. That method embraced the Cyber D2D concept by using the threat analysis data sparingly and directly countered the intention in every extended kill chain stage as a measured response; this aligns with the 'Spheres of Influence' (as the dominant data need) and the 'Countermeasure Design Considerations' in the technique selected for the tactic design.

TABLE 2
ONION OF PROTECTION LAYERS [J5].

Onion Layer Kill Chain Intention	Spheres of Influence (Dominant data need)	CM Design Considerations	Comment
Layer 1 Find (Outer most layer)	Protected Platform	Decreased Detectability	Using knowledge of the Protected Platform's strengths and weaknesses, the tactics focus on the kill chain intention to be less conspicuous and stealthier in Early Warning, Air Search, or Ground Control Intercept radars. The tactics are against the probability of detection and clutter suppression.
Layer 2 Fix	Protected Platform	Decreased Detectability Decoy & Deception	The tactics 'degrade' information and counter a fix in altitude to counter Target Acquisition or Height Finding radars with deceptions and decoying.
Layer 3 Recognise	Protected Platform Weapon System	Decreased Detectability Decoy & Deception Distraction Denial Disruption	The tactics cause confusion and 'delay' in assessing the identity or classification. Counter-recognition measures at target filtering based on radar modes like NCI and other discriminating behaviours.
Layer 4 Track	Protected Platform Weapon System	Decreased Detectability Decoy & Deception Distraction Denial Disruption	The tactics countermeasure track convergence (and fusion) with distraction, disruption, and denials in target acquisition or higher data rate search and acquisition modes to counter the higher quality prediction.
Layer 5 Engage	Defensive Constraints Protected Platform Weapon System	Decreased Detectability Decoy & Deception Distraction Denial Disruption Destruction	Using signal and sensor processing targeted tactics and break locks in tracking to defeat the threat, using all capabilities available in the hard and soft kill, but dependant on ROE, this is the traditional set of platform self-protection tactics to evade and defeat a threat.
Layer 6 Prosecute / Effect (Inner most Layer)	Defensive Constraints Protected Platform Weapon System	Decreased Detectability Decoy & Deception Distraction Denial Disruption Destruction	The tactics are against the seeker and sensors simultaneously and in coordination and with prioritized techniques in cooperation. These are the traditional set of platform self-protection tactics to defeat the threat towards a greater miss-distance.

4.3 Platform Protection and Threat Analysis Summary

This chapter provided a threat analysis methodology with analysis method steps for smaller-scale enterprises where that scale can employ AI methods with a shorter D2D cycle. AI methods are applied in all steps calling on other research threads within this dissertation to promote automation. The threat analysis methodology applies to the military but is openly available to the civilian domain and has lower data need at the outer layers of an onion of protection concept, supporting sparse and imbalanced datasets with AI methods. The methodology provides threat analysis with strengths,

weaknesses, opportunities, and vulnerabilities in set-mappings for countermeasure technique selections to build a proposed countermeasure tactic. The threat analysis develops a software model for capturing a threat as a digital twin, provided by the threat analysis step methods for using the proposed countermeasures.

The software modelling digital twin is exploited for countermeasure technique approaches, forming a tactic with less need for the actual system in susceptibility analysis and testing. Using the proposed extended kill chain concept with the onion of protection method is an alternative to the pure threat destruction strategy and is applicable to the civilian domain. The countermeasure design considerations map directly to the kill chain within the onion of protection and, as such, directly counter the intention of the threat. Civilian accessible networks already proposed for AI incorporation in the research were highlighted as relevant network bearers for this D2D data approach with lower data need on the outer layers of the onion of protection. The system view steps are readily implementable within a computer using Chapter 5's Expert System method. Chapter 5 presents a neuron-based algebraic form of an Expert System with calculated confidence towards the live automation aim of this methodologies approach while embracing incomplete datasets in the civilian accessible domain. Partial results were published in a peer-reviewed conference paper [C5] and a peer-reviewed journal version [J5].

This page is intentionally blank.

Chapter 5

NEURAL EXPERT SYSTEM METHOD, A STEP TOWARD NEURO-SYMBOLIC AI

Within the application area, threat data in the public domain is often incomplete, unstructured, imbalanced, and of unknown certainty, similar to Big Data and when building datasets. In this case, threat data can also have differing security caveats based on discretion and origin. Thus this proposed neuron-based Expert System allows rules in an algebraic form for threat analysis and known physics formulas to be programmed with imbalanced irregular inputs so that an estimation of the unknown values can complete and re-balance the dataset while also providing an estimate of confidence.

5.1 Symbolic AI towards Neuro-Symbolic AI

Symbolic AI methods are an area that has had much research already, but that research has primarily stopped in response to ML and DL developments diverting interest. Symbolic AI represents an AI area with closer certification for safety-critical applications. However, Symbolic AI approaches like the expert system has limited ML capability and thus cannot respond autonomously to new rules. Neuro-Symbolic AI is an area that seeks to bring Symbolic AI together with neural methods. The challenge is bridging the gap between high machine abstraction methods of Symbolic AI and high human knowledge abstraction in neural methods.

5.1.1 Modernised Neuron Based Expert System Method

This research has proposed a modernized form of an expert system, which is closer in structure to a neural network. That expert system proposed knowledge compartments to support reuse in input, hidden, and output layers like a neural network. Those layered abstractions convert units into and out with the abstracted human level, reusable while containing a hidden core representation layer. Thus the problem becomes a mapping of abstraction between layers. The proposed form used algebraic rules to couple to another research theme within this dissertation and connected to a neural network back-propagation ML technique, which that research theme then used a formula extraction method. It integrated the BN method into a ‘probability tree’ and the algebraic rules so that different input parameter populations would normalize and forward propagate, balancing the dataset. The knowledge representation was based on ‘computational graph’ structures and formed into ‘Reverse Polish Expressions’ with lambda calculus for computation. It included a histogram technique of value permutations and Bayesian probabilities to perfect the outputs based on all valid populations of rules and inputs and this virtualized the node structures as a hypothesis. The BN was initially scored by the proportional numerical distance from other valid values as the prior and combined with other hypotheses. In the histogram technique of the cumulative likelihoods of each value, it applied a centre of gravity to perfect a value based on the probability distribution of all the likelihoods; so an outliers’ influence is proportional to their numerical distance from other valid values.

5.1.2 Modernised Expert System in the Application Area

Within the application area, results are proportional to the mutual agreement in a result, given different algebraic rules of threat analysis and physics methods. Results are also still proportional in confidence given the different number of source accounts of a value, controlled by 'guard equations' of valid rule populations from decision trees. Guard equations allow the inclusion and exclusion of the threat analysis and algebraic physics rules based on sources or inferred from other rules. A resultant value perfected also provides both a confidence and uncertainty measure in that perfected value as numerical qualifiers of the assertion. The neuron approach to an expert system also has confidence and uncertainty based on the dataset, and rules applied are not inferred from a loss value of fitting like in a neural network method but are a BN. As such, confidence and uncertainty can be independent of a regularisation method based on the data presented and the knowledge contained in

the rules. However, from the related research in the formula extraction method, when deducing new rules, a loss value from a neural network method could still be applied to a rule as scaling and might be interpreted as a score of the rule accuracy in the formula extraction process. This proposed form was novel to modernize the method and map an expert system closer to a neural network. A peer-reviewed paper was presented and published [C4[3]]; then furthered and published in a journal [J4[4]].

5.1.3 Reasoning and Current Expert System Methods

Aristotle's deductive reasoning, Sir Frances Bacon's inductive reasoning [204], and transduction or transductive inference [225] can map towards computational methods like Expert Systems and ML. Commonly information within publicly accessible sources is not structured similarly, and confidence in that data is not always straightforward. From a confidence point of view, understanding the number of possible interpretations could be more important than knowing what the interpretations are, as knowing that the data is incomplete is knowledge in itself, thus relating to the readiness to make a decision. A method by Abdella et al. [226] uses a genetic algorithm and neural network mix for approximating missing data values in a database by minimizing the loss value with a genetic algorithm. However, this may not identify missing permutations in methods to infilling in values but is a fitting optimization; it risks over-fitting and perhaps incompleteness.

Additionally, humans need to be objective and unbiased in decision-making, which is not always possible in pressure circumstances. Johnson-Laird [84] considered the types of knowledge that affect a judgment and the counterclaim examples supporting it. An Expert System method called DEX is human assistance for decision-making by Bohanec et al. [82]. Measuring the confidence in decision permutations is a subject of Voskoglou's [83] experiments from a classroom mapped to a fuzzy logic model. There is also a BN method by Wiegerinck et al. [227], but Melen et al. [50] method also used a BN to grade a rule-base to adapt the individual rules influences in a time-evolving scenario. Cook et al. [85] presented a validation method of expert system rules confidences in data-mining applications within textual and imagery databases. The Barzilay et al. [86] method separated knowledge types: communication and domain knowledge for explanation purposes. Multi-hypothesis estimation was proposed by Khalak et al. [228] for a system in degradation for diagnosis and prognosis purposes. Proposed statistical probabilities decision tree rules are in a medical application by Spiegelhalter et al. [89]. Furthermore, a safety-critical application for military Data Fusion is proposed by Rauch [80]. Connected to this chapter in this dissertation is the 'formula extraction' method [C3], [J3], and is discussed in Chapter 6.

5.2 Knowledge Compartments toward the Neuron Method

Concerning the Intelligence Life Cycle [87], which form six steps as shown in Figure 17. Knowledge can be structured into the following compartments and mapped onto AI methods:

- **Raw Data**: The output of the 'Collection' stage of the Intelligence Life Cycle can map to a neural network's input layer or the input values in an expert system. It can be source unique formatted with both relevant and irrelevant information combined.
- **Information**: The output of the 'Processing' stage of the Intelligence Life Cycle. In neural networks, this is the output of the hidden layers; there is no direct translation in traditional expert systems methods unless used in libraries. It filters valuable information with defined qualities and shared values in a standard format.
- **Intelligence**: The output of the 'Analysis and Production' stage of the Intelligence Life Cycle; this is the output layer or the output knowledge in the expert system in a neural network. Contextual understanding is combined with the information to answer a specific question.
- **Metadata**: Quantities and conventions allow standardized understanding of the hidden layers and enable reuse.

[3] [Cn] Published conference papers are in a separate bibliography on page xv.

[4] [Jn] Published journal articles are in a separate bibliography on page xv.

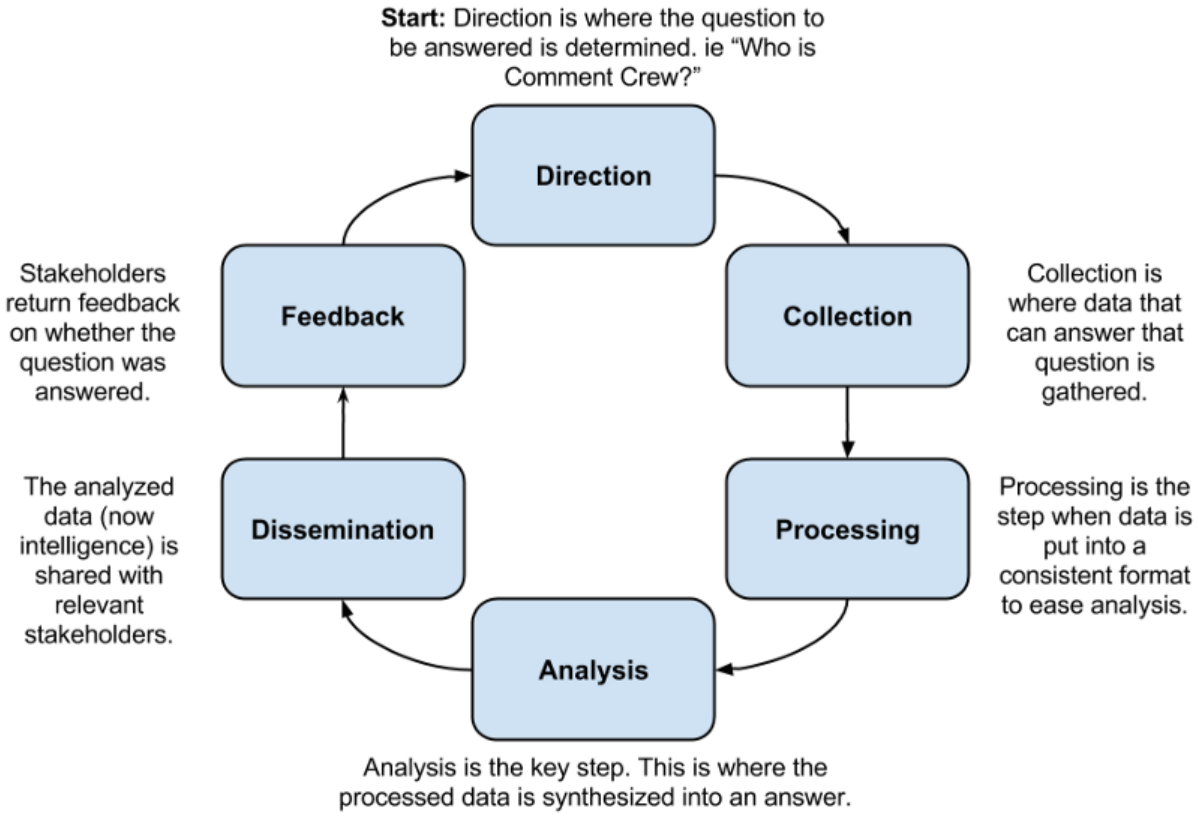


FIGURE 17 6 STEP INTELLIGENCE LIFE CYCLE [229]

The six steps of the intelligence life cycle shown in Figure 17, and the data compartments in section 5.2 benefit from a standard knowledge representation format for re-use.

5.3 A Language Method to an Algebraic Knowledge Representation

The information representation is algebraic, such that rules of thumb and physics are formula rules; this means that the body of established knowledge is in a form that can re-use the knowledge compartments described in section 5.2 and represent both atoms and axioms commonly.

Some knowledge can be selected based on a 'valid' body of knowledge; the guard equations can increase or reduce the scope. When a scope of knowledge is selected, every rule (in scope) can calculate all the permutations using the known input values to infill missing parameters with perfected values. Thus this represents the whole in-scope relevant body of knowledge towards the results balancing the dataset concerning the whole applicable body of knowledge in every case. Data sources can have different exclusions for commercial or security reasons, so the security caveats track through all value permutations. The sources and rule confidence can bias the confidence and prediction result. Using the formal Backus-Naur form (BNF) for a language syntax as published [J4] are expressed in Equations: (7), (8), (9), (10), (11), (12), (13), (14), (15), (16), (17), (18), (19) and (20). Equation (7) defines "Knowledgebase" as a list of 'Axioms' or 'Atoms' separated by semi-colons.

<KnowledgeBase> ::= <AxiomAtom> ";" [<KnowledgeBase>] ; (7)

Equations (8) and (9) define that an Axiom or Atom is an assignment-equality to an expression and allows the attributes of a *"When"* clause for an equation guard, a *"Security"* caveat definition, and a *"Confidence"* expression weighting.

<AxiomAtom> ::= <Symbol> "=" <Expression> ["When" <Expression>] ["Security" <BIT_FIELD>] ["Confidence" <Expression>] ; (8)

<Symbol> ::= <Identifier> ; (9)

Equations (10), (11), (12), (13), (14), (15), (16), and (17) define an expression in BODMAS precedence of brackets and comparative operators being later in evaluation then numerical operators.

<Expression> ::= <Term> [<Relation> <Expression>] ; (10)

<Term> ::= <Factor> [<Operator> <Expression>] ; (11)

<Operator> ::= "+" | "-" ; (12)

<Factor> ::= <Quantity> ["*" | "/" <Expression>] ; (13)

<Relation> ::= "<" | "<=" | "==" | ">=" | ">" | "!=" ; (14)

<Quantity> ::= <Value> ["&&" | "||" <Expression>] ; (15)

<Value> ::= [<Operator>] <Parameter> ; (16)

<Parameter> ::= <Symbol> | <Function> | <Literal> | "(" <Expression> ")" ; (17)

Figure 14 illustrates the ordering of the expressions to realize the associative nature of operators and their BODMAS precedence; please note that root and powers are defined as functions in this language and thus raised in the BODMAS precedence.

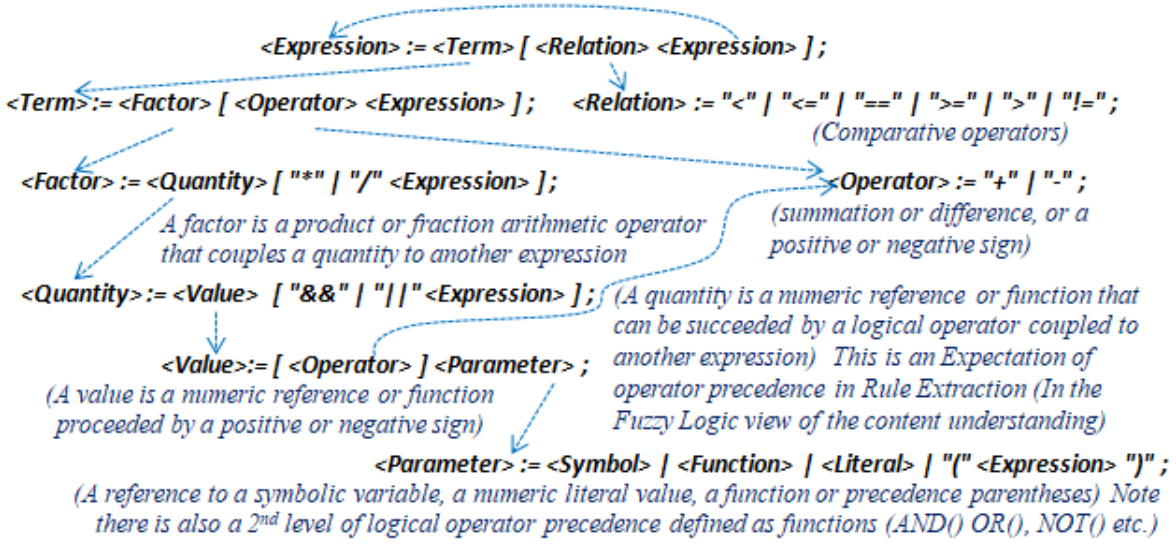


FIGURE 18 HIERARCHICAL BNF EXPRESSIONS OF THE KNOWLEDGEBASE LANGUAGE

In Figure 18, the logical operators (quantity) and comparative operators (relation) might appear to have the wrong precedence from the math operator convention. However, there are also logical operators AND(), OR(), and NOT() as defined functions. As such, there are two logical operator precedence levels. The first precedence is as a function which is the math convention, and the other is the logical operator precedence. In Chapter 6: Section 6.1.1, one interpretation of neuron content and networks: is an expectation that fuzzy logic gates are built within dense layers and may form fuzzy sets towards fuzzy rough sets. A single layer's node can form a fuzzy logic gate such as an 'AND' and 'OR' gate from the summation of the weight activations when applied to the bias threshold and activation function. In a single layer's node, these are 'AND' gate or 'OR' gate, which can elaborate further to more complex logic such as XOR, NAND, and NOR gates, when the network becomes deeper. The language expression thus is organized in the BNF for a single pass compilation and includes a provision for an expectation of a rule extraction complexity in precedence. Equations (18) (19) and (20) define the remaining BNF definitions as primitive types and function calls.

<Literal> ::= <Real> | <Integer> ; (18)

<Function> ::= <Symbol> "(" [<ParameterList>] ")" ; (19)

<ParameterList> ::= <Expression> ["," <ParameterList>] ; (20)

This language thus can be used to represent both atoms and axioms as equation formulas with multiple alternative declarations for irregular population permutations of atom evidence using the same symbol name but with the multiple alternative hypothesis axiom for the body-of-knowledge.

Then rules and evidence values can be mixed with a scope selection as the known-valid-body-of-knowledge via a Guard Equation in the *"When"* clause.

5.4 Knowledge Representation Method

Expert systems provide a structure of knowledge in a rule form and require an engineering approach to create. A difficulty is that they are handcrafted and tend not to be very well transferable. The expert system rules are symbolic, but a formula form provides transferability. The formula form of the expert system also provides the infilling of known unknown information of a dataset, and missing information can be learnt from other datasets and captured as rules. Confidence in data is often problematic because it may reflect a notion of the data completeness rather than the knowledge itself. It can reflect what is known to be unknown and what is known to be known. It also may be in error as it may not qualify the unknown knowns or unknowns. Outside of computational approaches, there is a reluctance to score Confidence highly, an artificial bias, and that bias can be prejudice in belief in an individual's knowledge completeness or competence rather than the evidence itself. However, in a computational method, the rigorous approach can reassure competence subject to the view of the current body of knowledge being complete while not repeating knowledge in different but equivalent forms. In ML, residual errors in fitting can be extracted from the losses and applied as a probability of fitting to data seen, and cross-validation can offer another loss value as an expectation of fitting unseen data. However, that expectation is that the unseen data matches the dimensionality because it splits a dataset. Also, philosophically that method is artificial as regularisation is a deliberate biasing; it can be more explicit that losses are an accuracy error of a solution rather than a probability of expectation. In the method presented, Confidence is a score of convergence between separate rules and values in the body of knowledge with all known values and can be regardless of their accuracy or trust bias. Thus, considering every possible permutation with the current body of knowledge makes Confidence reflect the divergence of other possible outcomes proportionally from the complete known body of knowledge. Every example of an answer is without prejudice except in convergence. Thus the representation with the presented method builds a computational graph by parsing the language and forming a computational graph structure of nodes formed into 'Reverse Polish Expressions' with virtualized nodes for every combination of all possible outcomes. The structure is data compartmentalized into input encoder, output decoder, and hidden layers like a neural network, but where the input and output layers constrain the abstraction in the hidden layer, as it could be from a rule or formula extraction method that deduced the rule.

5.4.1 Building the Computational Graph

When executed from the leaf nodes to the root node, a depth-first then breadth node recursion order is used to compute each node in a computational graph and converts to a reverse Polish expression for evaluation when adding each input as an equation compiled into the same computational graph structure, the captured knowledge is complete in one structure. When many input values are required, many equation assignments use the same symbol name as fact atoms. Rules are loaded into the same structure to compose a single tree for the knowledgebase using three parts: a symbol table, a hierarchical node structure, and a value list. When unset, the use of the default values of *0x00000000* and *1.0* in the security caveat and confidence weighting value apply in the absence of values. In evaluation, the hierarchical nodes are dependent on the node type for their computation, and the machine operation is different in node type cases:

Literal value nodes use the symbol table for the value as in Equations (21), (22), and (23).

$$v = v_{SymbolTable}, \quad \textit{as the value from the symbol table.} \tag{21}$$

$$C = C_{SymbolTable}, \quad \textit{as the confidence from the symbol table (default 1.0).} \tag{22}$$

$$S = S_{SymbolTable}, \quad \textit{as the security caveat from the symbol table (default 0x00000000).} \tag{23}$$

Operator nodes are a list of values from the child node (Below) and the child node's same level node (*Below.Same*) to calculate value combinations. Confidences are the product of the operand values' confidences. Security caveats are the binary OR of the operands; see Equations (24), (25), and (26).

$v = v_{Below}\ (Operator_{Sematic})\ v_{Below.Same}$, *as the value applied to the operator's semantic function.* (24)

$C = C_{Below} \cdot C_{Below.Same}$, *as the product of the confidences.* (25)

$S = OR(S_{Below}, S_{Below.Same})$, *as the binary OR of the security caveats in a bit field.* (26)

Function nodes are the same as the operator's process but allow more operands at child same level nodes.

Symbol nodes collate all the values for every matching symbol name in the tree; Confidence is the minimum fraction ratio between each other, scaled by the population and the symbol's confidence, and Security Caveats are from the matching symbol values as in Equations (27), (28), and (29).

$v_{[0\ldots i]} = v_{Search_{[0\ldots i]}}$, *as the values from a search of the matching symbols.* (27)

$minFract(A, B) = min\left(\frac{A}{B}, \frac{B}{A}\right)$, *as the minimum fraction between to values.* (28)

$C_{[0\ldots i]} = \left(\frac{\prod_{j=0}^{n}\left(minFract(v_{[i]}, v_{[j]})\right)}{n}\right) \cdot C_{thisNode}$, *as the confidences for the value set.* (29)

The compartmentalized knowledgebase structure supports input values first, so the computational graph values are resolved before computing the missing values.

5.4.2 Estimation of Values

Consistent with probability trees, the addition operation combines individual confidence likelihoods associated with each value permutations' prior through the computational graph's output, which the histogram combines into a joint probability. The histogram is not the number of values that fit within a value interval as a bin but the summation of the cumulative confidences of all individual prior confidence associated with each value permutation; this converts the likelihood from value instance permutations to the joint probability of a value as a probability density function. Given the probability density function, a centre of gravity can arrive at a sub-bin resolution perfected value. In Equations: (30), (31), (32), (33), (34), (35), (36), (37), (38) this process is described. The number of value permutations is *nV*, restricted to the *nH (as the histogram bin limit)*, which is the denominator of a resolution ratio of the scale between the maximum and minimum value of the permutation values *($v_{[0\ldots nV]}$)* from the computational graph's output.

$\Delta v = max(v_{[0\ldots nV]}) - min(v_{[0\ldots nV]})$, *as the numerical difference of the value.* (30)

$\rho = \frac{\Delta v}{max(min(nV, nH) - 1, 1)}$, *as the bin resolution of the histogram.* (31)

$l = \left\lceil\left(\frac{\Delta v}{\rho}\right)\right\rceil$, *as the number of histogram bins.* (32)

$\sigma\lambda = \sum_{i=0}^{l}\left(i \cdot \rho + min(v_{[0\ldots nV]})\right) \cdot hist_{[i]}$, *as the sum of the product of histogram values.* (33)

$\sigma x = \sum_{i=0}^{l}\left(i \cdot \rho + min(v_{[0\ldots nV]})\right)$, *as the sum of the histogram bin index range.* (34)

$\sigma Confidence = \sum_{i=0}^{l}\left(hist_{[i]}\right)$, *as the sum confidence probability.* (35)

$WeightedValue = \frac{\sigma\lambda}{\sigma Confidence}$, *as the centre of gravity for the perfected weighted* value. (36)

$Certainty = 100 \cdot \left(\frac{\sigma\lambda}{\sigma Confidence \cdot \sigma x}\right)$, *as a certainty probability measure.* (37)

$Uncertainty = \left(\frac{\sigma Confidence}{\sigma\lambda \cdot \sigma x}\right) \cdot p$, *as an uncertainty metric in the value scales as a variance measure.* (38)

The *σConfidence* is thus the total probability in the *WeightedValue* given all other possible values. The *WeightedValue* is the *centre of gravity* given the convergence agreement in separate axiom rules and input atom facts and is reflective of every possible permutation in the body of knowledge. The *Certainty* is a probability of accuracy given all other possible values. *Uncertainty* is a variance measure concerning the *WeightedValue for* modelling for stochastic analysis variations.

5.5 Body of Knowledge and Valid Scope

As the resultant perfected value is from the confidence probability density function in a histogram, which includes every permutation of possible answers within the body of knowledge, the proper scope of the body of knowledge is essential. The different gains could be calculated for different antenna type assumptions when calculating the beamwidth of an unknown antenna type (as *Z*) applied to the application area. The convergence method provides a value reflective of all the possible outcomes of the unknown antenna type. When an antenna type is not unknown but ambiguous, the resultant values thus represent the scope of uncertainty to apply rules of the relevant body of knowledge. So as there is more or less ambiguity in the antenna type, the body of knowledge is sampled and thus reflects the body of knowledge that is valid. Using the $k\lambda/D$ formula that calculates beamwidth for different Antenna Types, the knowledgebase representation is in Equations (39), (40), (41), (42), and (43).

*BeamWidth = 70.0 * Lamda / D When OR(AntType == Z AntType == UniformParaReflect);* (39)

*BeamWidth = 60.0 * Lamda / D When OR(AntType == Z AntType == UniformParaDish);* (40)

*BeamWidth = 57.0 * Lamda / D When OR(AntType == Z AntType == InUniformParaReflect);* (41)

*BeamWidth = 66.468 * Lamda / D When OR(AntType == Z AntType == TelescopeReflect);* (42)

*BeamWidth = 1.33 * Lamda / D When OR(AntType == Z AntType == Yagi);* (43)

The above values of *k* are from the following references [230], [231], [232], [233]. The whole body of knowledge is in scope when the beamwidth calculates with an unknown antenna type (Z). In that case the *InUniformParaReflect* is 1.82 degrees with 0.3% confidence, *UniformParaDish* is 1.9 degrees with 0.33% confidence, *TelescopeReflect* is 2.13 degrees with 0.29% confidence, *UniformParaReflect* is 2.24 degrees with 0.25% confidence and *Yagi* is 0.04 degrees with 3.9e-6% Confidence. The *Yagi* is an outlier with little confidence, but it barely affects the weighted value of 1.94 degrees and has a lower effect on the 34% certainty. However, it does affect the total confidence and is reduced by 1.18%. This impact on the total confidence reflects the human intuition that knowledge of an outlier causes doubt, but the certainty accuracy and weighted value are little affected. When adding the atom evidence for a quoted example value, the perfected value is 2.07 degrees nearer the main cluster; the weighted value estimated is steered with that evidence but still includes the entire body of knowledge, resulting in confidence and certainty reducing slightly. The weighted value becomes 2.05 degrees with 1.16% Confidence and 30% Certainty. Finally, suppose the antenna type is a Uniform Parabolic Reflector (*UniformParaReflect*). In that case, the coverage of the body of knowledge reduces. The weighted value is 2.15 degrees with a confidence of 92.5% and a certainty accuracy of 50%; as the selected rule and data example are close numerically confirming each other, the confidence increases significantly. As a human intuition, this reflects a human bias in trust when a closer match in a value occurs: between an example answer and a body of knowledge rule; this raises the confidence in the prediction trust and thus accuracy for the deduced value. When desired in the application, a confidence term can be added to the rules and data examples to promote additional bias towards data examples or rules.

5.6 The Computational Detail

Each input and rule and all permutations of computations are within virtualized node neurons, so the value and rule combinations are known for security tracking. Therefore, that estimation is based on the scope of threat analysis and known physics but limited by guard equations. As such, irregular populations of values and all possible rules can be made in a computation, thus filling the scope of uncertainty but limited by the data and known rules (known data values, known threat analysis, and known physics).

The alternative input values (*val*) arrangement is in a matrix. Those values derived may have been from irregular input values or several alternative rules.

In this illustrative example, there are three values *(n)* within that matrix (*val*) see Equation (44):

$$val \coloneqq \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}. \tag{44}$$

The minimum fraction function *(minFract)*, which provides the fraction of numerator and denominator combination of the input values (*A* and *B)* that is less than one, is in Equation (28).

The initial idea was that a dot product form (inner-product) probabilities [234], as in quantum mechanics, could generalize Euclidean vector space for comparison in dimensionality scales, and the fraction matrix *(minFract$_{Mat}$)* would be as follows in Equation (45). Note that the zeros represent the numerator and denominator combination comparisons, which would make a comparison with itself, and are invalid. The matrix multiplication with a matrix of one's takes the dot-product (inner product) to collapse the matrix to an un-normalized probability for each value based on the numerical fractional distance from all permutations of other valid values, cumulatively:

$$minFract_{Mat} = \begin{bmatrix} 0 & minFract(val_0, val_1) & minFract(val_0, val_2) \\ minFract(val_1, val_0) & 0 & minFract(val_1, val_2) \\ minFract(val_2, val_0) & minFract(val_2, val_1) & 0 \end{bmatrix} \times \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 83\% \\ 116.7\% \\ 100\% \end{bmatrix}. \tag{45}$$

From those example results in Equation (45), without scaling, it can be seen: for the first value *(val$_0$)*, the value is 0.833, which is 0.5 + 0.333, as the value 1.0 is one-half of the value 2.0, and one-third of the value 3.0. For the second value *(val$_1$)*, the value is 1.167, which is 0.5 + 0.667, as the value 1.0 is one-half of the value 2.0, and the value 2.0 is two-thirds of the value 3.0. For the third value, 3 *(val$_2$)*, the value is 1.0, which is 0.333 + 0.667, as the value 1.0 is one-third of the value 3.0, and the value 3.0 is two-thirds of the value 2.0. Therefore it provides a summation of the numerical scaling difference between the values of the numerical factor as a cumulative sum. The *minFract$_{Mat}$* matrix is then normalized *(P$_{norm}$)* in Equation (46) to the population size *(n)* (as the evidence). The scale is one 6th, as six is 2×3 with two numbers compared in a population of three values, and if there were a population of four values, then the scalar value would be one 12th as three numbers compared between a population of four, and so on, and is defined as:

$$P_{norm} \coloneqq minFract_{Mat} \cdot \frac{1}{(n-1) \cdot n} = \begin{bmatrix} 13.9\% \\ 19.4\% \\ 16.7\% \end{bmatrix}. \tag{46}$$

The example shows a one 6th scaling of the *minFract$_{Mat}$* probabilities as *(P$_{norm}$)*. Those probabilities scaled by the node probability for this variable, defined in the knowledgebase, allows weighting strength to be assigned for the rule, and to illustrate this in this example, in Equation (47), it will be assumed to be 0.75 as (*P$_{node}$)*:

$$P \coloneqq P_{norm} \cdot P_{node} = \begin{bmatrix} 10.4\% \\ 14.6\% \\ 12.5\% \end{bmatrix}. \tag{47}$$

These probabilities now represent 75% of the dot product (as an inner product probability co-alignment) of the numerical fractional scale difference between other values. They are also scaled by the population of the valid answers and thus represent the cumulative probability of value convergence to other valid values proportionally. The histogram of cumulative confidences as the bin value assignment provides a likelihood of each bin value rather than the voting population count. The histogram then represents the cumulative likelihoods as a joint probability within the distribution; this is also proportional to an outlier value's confidence influence as the minimum fractional scale difference rather than a majority voting count.

Nevertheless, as the research wants the network to operate over an arbitrary node structure iteratively, and the inner product concept is not as efficient as a probability tree when used with the histogram, as the node's confidences are not just multiplied longitudinally (depth) and then only summed in the histogram laterally (breadth). So, to be consistent with probability trees and also with the summation being more optimally once in the histogram, the *minFract$_{Mat}$* matrix becomes, as in Equation (48):

$$minFract_{Mat} = \begin{bmatrix} 1 \cdot minFract(val_{0,} val_1) \cdot minFract(val_{0,} val_2) \\ minFract(val_{1,} val_0) \cdot 1 \cdot minFract(val_{1,} val_2) \\ minFract(val_{2,} val_0) \cdot minFract(val_{2,} val_1) \cdot 1 \end{bmatrix} = \begin{bmatrix} 16.7\% \\ 33.3\% \\ 22.2\% \end{bmatrix}. \quad (48)$$

From the example results, the first value *(val_0)* for the value 1.0 is 0.167, which is 0.5 × 0.333, as the value 1.0 is one-half of the value 2.0, combined with the scaling of one-third of the value 3.0. The second value *(val_1)* for the value 2.0 is 0.333, which is 0.5 × 0.667 as the value 1.0 is one-half of the value 2.0, combined with the scaling of value 2.0 as two-thirds of the value 3.0. The third value *(val_2)* for the value 3.0 is 0.222, which is 0.333 × 0.667, as the value 1.0 is one-third of the value 3.0, combined with the scaling of the value 3.0, which is two-thirds of the value 2.0; and this, therefore, provides a mutual numerical scaling difference between the values in terms of the numerical fraction and combined with all other factors and is mutually interdependent. The $minFract_{Mat}$ matrix is now only multiplied longitudinally and only summed in the histogram laterally. The population normalization becomes Bayesian-like and is more straightforward, as in Equation (49), as the scaling is occurring in the multiplication of fractions, that then can be P_{node} scaled in Equation (50), as before, and are as follows:

$$Bayes \coloneqq minFract_{Mat} \cdot \frac{1}{n} = \begin{bmatrix} 5.6\% \\ 11.1\% \\ 7.4\% \end{bmatrix} \text{ and} \quad (49)$$

$$P \coloneqq Bayes \cdot P_{node} = \begin{bmatrix} 4.2\% \\ 8.3\% \\ 5.6\% \end{bmatrix}. \quad (50)$$

The relative probabilities between the values 1.0, 2.0, and 3.0 ($val_{0\text{-}2}$) of the example are still representative of their mutual factor difference combined with all other combinations in each node, but rather than a dot product addition in each node, there is a consistency with probability trees, the longitudinal product is through nodes and lateral summation in a histogram. Those probabilities represent mutual scaling in every node, and the accumulation is then in that subsequent histogram. The histogram uses the minimum and maximum limits in the range 1.0-3.0 of ($val_{0\text{-}2}$) with a bin size of 1. Then instead of assigning the population that matches the bin limits, the value's confidences are added (summated) for each bin of values that fall within each bin limit —Furnishing a histogram of cumulative confidences of values concerning all the value likelihoods proportionally. A centre of gravity deduces a sub-bin resolution centroided weighted value as a perfected estimate based on the evidence likelihood, relative outlier's likelihood, and numerical position. The centre of gravity centroided weighted value is the *perfected estimated value* based on all evidence's likelihoods in the body of knowledge, as in Equation (51). Equations (52) and (53) are the *certainty* and *confidence* of that value. The difference between *confidence* and *certainty* is: that *certainty* is a question of accuracy and possibility. It is proportional to an outlier's position and likelihood, and confidence is a measure of a belief given all the values cumulative agreement in that value as a probability, defined as:

$$CoGValue \coloneqq \frac{(bin_0 \cdot val_0 + bin_1 \cdot val_1 + bin_2 \cdot val_2)}{\sum(bin_{0..2})} = 2.07692, \quad \textit{(perfected estimate value),} \quad (51)$$

$$CoGCert \coloneqq \frac{\left(\frac{bin_0 \cdot val_0}{\sum val} + \frac{bin_1 \cdot val_1}{\sum val} + \frac{bin_2 \cdot val_2}{\sum val}\right)}{\sum(bin_{0..2})} \cdot 100 = 34.6154\% \quad \textit{(certainty in that value) and} \quad (52)$$

$$Conf \coloneqq \sum(bin_{0..2}) \cdot 100 = 24.0741 \quad \textit{(belief or confidence).} \quad (53)$$

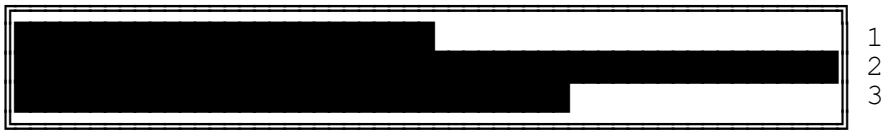
Before the P_{node} scaling in this example, the cumulative sum is 24.1%, instead of 50%, as it would have been with the dot product method, as it includes all other mutual combinations that have scaled the comparisons. Nevertheless, $\sqrt{0.241}=0.49$ (49% or ~50%), as the dot product (inner product) method is equivalent via a square root. The perfected value comes from the centre of gravity, using the values applied as likelihoods. The certainty in that figure is with scaling from the sum of the likelihood values. In Figure 19, P_{node} scaling at the output does not affect the certainty or the perfected value but only the confidence in those results. However, setting an input node's confidence

value (P_{node}) like an input sample value (in *val*) would affect the perfected value and the certainty and confidence in that result. This arrangement provides an extensible mechanism to operate across nodes structures, where the multiplication of probabilities occurs in the nodes, and the summation of those probabilities occurs outside of the node structure in a modified histogram technique, i.e., it accumulates the node confidences into the histogram bins as likelihoods, rather than merely the population. The node confidences are centre of gravity centroided to provide a sub-bin resolution perfected value and certainty (accuracy), combined with the cumulative confidence of belief.

```
1, (5.55556%) Confidence,
2, (11.1111%) Confidence,
3, (7.40741%) Confidence,
HISTOGRAM
```


```
Weighted value 2.07692 (24.0741% Confidence / Certainty 34.6154%)

When scaled by the probability of Pnode it becomes:
1, (0.0416667%) Confidence,
2, (0.0833333%) Confidence,
3, (0.0555556%) Confidence,
HISTOGRAM
```




```
Weighted value 2.07692 (0.180556% Confidence / Certainty 34.6154%)
```

FIGURE 19 EXPERT SYSTEM PROTOTYPE'S HISTOGRAM WITH THE CENTRE OF GRAVITY AGGREGATION

Thus, the node structure may combine values and rules in the nodes, and as this is an algebraic method, operators and functions increase in the population of value combinations. Those population combinations would then be subject to the confidence probability method, as previously described, using the Bayesian probability tree and centre of gravity histogram technique. In Equations (54) and (55), a node's inputs *(valA* and *valB)* in an operator-type node can be applied from the input vectors (*valA* and *valB*) to produce all combinations as a matrix, defined as:

$$valA \coloneqq \begin{bmatrix}1\\2\\3\end{bmatrix} \times [1 \quad 1 \quad 1] = \begin{bmatrix}1 & 1 & 1\\2 & 2 & 2\\3 & 3 & 3\end{bmatrix}, \textit{and} \tag{54}$$

$$valB \coloneqq \begin{bmatrix}1\\1\\1\end{bmatrix} \times [1 \quad 2 \quad 3] = \begin{bmatrix}1 & 2 & 3\\1 & 2 & 3\\1 & 2 & 3\end{bmatrix}. \tag{55}$$

So the input values *valA* and *valB* are applied as vectors to a matrix for both the values *(val)* and the probabilities *(*P*a* and P*b)* in those values (in Equations (56) and (57)) and are defined as:

$$\mathrm{P}a \coloneqq \begin{bmatrix}1\\0.5\\0.75\end{bmatrix} \times [1 \quad 1 \quad 1] = \begin{bmatrix}1 & 1 & 1\\0.5 & 0.5 & 0.5\\0.75 & 0.75 & 0.75\end{bmatrix}, \textit{and} \tag{56}$$

$$\mathrm{P}b \coloneqq \begin{bmatrix}1\\1\\1\end{bmatrix} \times [1 \quad 0.5 \quad 0.75] = \begin{bmatrix}1 & 0.5 & 0.75\\1 & 0.5 & 0.75\\1 & 0.5 & 0.75\end{bmatrix}. \tag{57}$$

As such, all the permutations of the inputs are as in Equation (58), and in this example, the node operation type is a divide operation, defined as:

$$y \coloneqq \begin{bmatrix} \left(\frac{valA_{0,0}}{valB_{0,0}}\right) & \left(\frac{valA_{0,1}}{valB_{0,1}}\right) & \left(\frac{valA_{0,2}}{valB_{0,2}}\right) \\ \left(\frac{valA_{1,0}}{valB_{1,0}}\right) & \left(\frac{valA_{1,1}}{valB_{1,1}}\right) & \left(\frac{valA_{1,2}}{valB_{1,2}}\right) \\ \left(\frac{valA_{2,0}}{valB_{2,0}}\right) & \left(\frac{valA_{2,1}}{valB_{2,1}}\right) & \left(\frac{valA_{2,2}}{valB_{2,2}}\right) \end{bmatrix} = \begin{bmatrix} 1 & 0.5 & 0.333 \\ 2 & 1 & 0.667 \\ 3 & 1.5 & 1 \end{bmatrix}. \quad (58)$$

The probabilities calculations are as previously described. Other node types like equality or a constant value are more straightforward as they have one input-to-output relationship in a vector. Function nodes with more than two parameters use an ordered set for the parameter lists but are identical in operation, although a function replaces the operator.

5.6.1 Dealing with Zero and Negative Numbers in the Inputs

A drawback of this method is comparing input values of zero or a mix of signs in the input values. This drawback is an issue with input values such as differences but not absolute values more than 0 like lengths or power measurements. For example: when the magnitude is not essential, only the relative numerical difference in a number line is; then, to resolve this, a relative data type that will pre and post-offset the magnitude is required. As such, for that data type, then consider the input values *(val)* to include zero and mixed signs in the number set as in Equation (59):

$$val \coloneqq \begin{bmatrix} -1.0 \\ 0.0 \\ 1.0 \end{bmatrix}. \quad (59)$$

Then a value offset *(val_{offset})* can be calculated, and in this example, the value two as the smallest input value (-1.0) is two number units different from the ratio 1.0, as in Equation (60):

$$val_{offset} \coloneqq 1 - \min(val) = 2. \quad (60)$$

That offset *(val_{offset})* can be added to the input values *(val)* as an adjusted value set *($val_{adjusted}$)* and then applied as before, with those adjusted values as in Equation (61):

$$val_{adjusted} \coloneqq \text{val} + val_{offset} = \begin{bmatrix} 1.0 \\ 2.0 \\ 3.0 \end{bmatrix}. \quad (61)$$

That offset *(val_{offset})* can then be removed, from the established perfected value *(CoGValue)*, at the output as *($CoGValue_{readjusted}$)*, as in Equation (62):

$$CoGValue_{readjusted} \coloneqq CoGValue - val_{offset}. \quad (62)$$

The relative data type, therefore, provides pre and post-offsets and allows the incorporation of numbers that would be relative by converting to an absolute scale. Where that scale begins at the smallest value, all other input values will be of a relative magnitude and thus naturally provide relative confidence and certainty without further scaling or offsets. Other data types are also future work: for angles with continuous measurements, which reset at values.

5.7 Summary of Modernised Expert System Method

The initial method using the inner product (dot product) is nominally equivalent to the Bayesian method and relates by a square root of the inner product being nominally equivalent to the Bayesian method. The Bayesian method is computationally less expensive as the addition operation occurs once in the final histogram rather than in every node in the dot product. The Bayesian method is also a more intuitive figure and is arrived at directly without the square root term. The method is consistent with probability trees replacing the probability tree's final summation in breadth (latitude) with a histogram and centre of gravity method when perfecting a better-estimated value. The expert system has a software prototype built in the application area with a knowledge base. In the application area, it can manipulate source raw data and, from more 'observable' raw data, create estimates for less 'observable' information via a rule set, and those rules are from textbooks, analysis, transforms, and equations as an established body of knowledge. The modernized expert system has a virtualized node structure and knowledgebase organization (input, hidden, and output layers) closer to a neural network structure. The method fills in the best values for missing data and forms the best

value from multiple and differing input values and alternative rules. So it is highly applicable to data migration and knowledge conversions. The method uses a Bayesian-like probability tree with measured confidence in a value propagated through the nodes rather than a probability in a rule. Thus, it provides a confidence metric and value commensurate with all possible alternative results. The technique converts BN probability trees into neural network structures. It also constructs the BN probability trees from algebraic rules in an input language. That input language can be an open architecture interface from an existing neural network rule extraction technique or the formula extraction technique presented in this dissertation.

Each variable amount of input values virtualizes nodes in that neural network structure. Those virtualized node semantics pertain to just one combination of possibilities rather than in a neural network where each node contains many and mixed possibility combinations. The node semantic makes it easier to certify as the nodes' semantics are directly derivable. The knowledge structure organization aids reuse, with compartments of input, hidden, and output knowledgebase structures, as a convenience for data-migration applications and the reuse of knowledgebase rules toward extended abduction knowledge reuse. Extended abduction knowledge or 'background knowledge' generalizations are closer to 'common-sense' in reuse as they are numerical and algebraic. The ultimate aim is to enhance the rule-base by experience from a neural network extracted from this dissertation's formula extraction research. Partial results were published in a peer-reviewed conference paper [C4] and a journal version [J4].

Furthermore, one current research approach toward general intelligence is a middle ground called Neuro Symbolic-AI, termed Broad-AI as a distinction from the single method AI approaches such as Symbolic AI, Evolutionary AI, and ML, known as Narrow AI. As such, Broad AI is a combination of Narrow AI methods. In IBM Watson Lab's opinion, Broad AI is the precursor to General AI and is multi-domain and multi-mode distributed and explainable [235]. IBM Watson Lab has invested in this area using the combined Neuron ML, Symbolic-AI, and language generation methods. Combining the following chapter's method with this chapter's method also forms a neuro-Symbolic AI method. This research dissertation uses a modified expert system and a neural network numerical extraction method but with different methods from Watson's lab. Watson's lab's application is a visual explanation, coupling image processing with symbolic AI and language generation. In comparison, the methods in this dissertation are an algebraic neuron-based expert system with a formula extraction method of a neural network.

Chapter 6

FORMULA EXTRACTION TOWARDS NEURO-SYMBOLIC AI

The subject of this chapter is related to rule extraction as part of an Explainable AI theme but is more specifically Formula Extraction. This research supports the neural expert systems method presented in Chapter 5, but this chapter forms the backward chaining element of a Neuro-Symbolic AI method when combined with Chapter 5. The outcome of this chapter's method is a new formula rule learnt from a neural network to provide a new rule in the neural expert system method of Chapter 5. The content of a neural network is still not well understood. However, research has progressed in Rule Extraction using perceptron and Recurrent Network layers.

6.1 Neural Network Content

Intuition for the content of a neural network can fall into two main theories of interpretation: the Fuzzy Logic view and the Numerical Function view. These are valid interpretations, but the proper interpretation can be a mixture. A perceptron dense layer network structure is presented in Figure 20 to illustrate the symbols, nodes, and indexing to examine these interpretations.

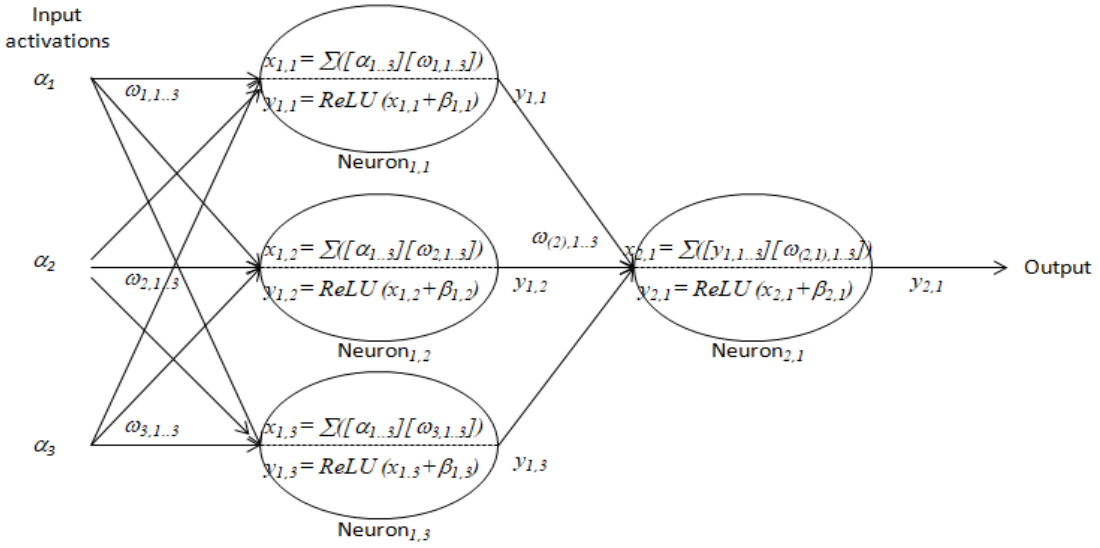


FIGURE 20 DENSE LAYER NEURAL NETWORK NODE STRUCTURE

6.1.1 Weighting, Scaling, and Dissimilarity

A single node's ω (weight) is associated with an input activation (α) and a node Bias (β). An activation function such as *ReLU* is associated with the nodes and thus the node output. The node output is the sum of activations pre-scaled by the weights, which is a form of loose normalization for the value variations, as the weights are initialized initially to variations within limits derived from the population of the activations at the outset. However, the weight scaling will adapt the value variations in back-propagation, but the initial condition avoids saturation by normalization scaling. Although the weight initialization has a variety of values scaled up to an initialization value limit, the weights are normalization variations based on a normalization limit. The distribution in use also affects those variation limits values. In some forms, such as He et al. and Glorot/Xavier initialization, the calculated variation limits are from the number of nodes, activation population, and distribution used [236]. That means the more prominent the weight scale limit, the smaller number of nodes or activations. The variation within the limit values supports the scale limits for the more significant

number of activations at different influence strengths. However, the primary intention for the distribution under the limit is to provide different start conditions for activations, such that weights have different and more unique updates to a dataset as unseen prior to learning. The maximum weight limit applied to an activation scale should there be full activation; thus, it can be more likely to retain scaling without saturation at the output. The set of quasi-mean average limited start points with distributions for the weight values allows dissimilarity of nodes in updates, up to a normalization scale from the outset of learning given a dataset unseen but a known architecture. The weight updates given a dataset shall yield an adjusted mapping of weight values. The weight updates lead to some interpretations of what those adjustments to the weights from the dataset mean.

6.1.1.1 The Fuzzy Logic and Fuzzy Rough Sets Interpretation

A neuron's (n) output in a fuzzy logic-gate interpretation will form an 'AND' or 'OR' gate combination (as an activation selection combination of the inputs). The node's activation function and bias (β) threshold provide an output logical non-linearity function to provide a more robust logic 0 or 1 state at the output of each node gate; these are non-linearity functions as they convert the linear input into a logic output when combined with the bias (β) threshold. Without them, nodes may collapse in the update and not form a gate logic exclusively, removing logic representation potential. As such, a single dense layer (a broad network) is limited in those gates' complexity, for example, no eXclusive OR (XOR). When adding layers for a deeper network, then an 'XOR' can be formed as: (XOR = AND (OR, AND), and this greater complexity of fuzzy logic aids linear separation in classification (i.e., it allows more than one classification boundary). Another advantage of a deeper network allows numerical scaling of large numbers or small magnifying activations (NOR or NAND), and the reuse of activated node structures can reuse functions within them. Thus the network breadth (broad) provides input activation selections with a quasi-averaging set, and the depth provides more complex logic terms (for greater linear separation) combined with the opportunity for nodal reused logic structures. That view extends across neurons ($n_{x,1..3}$) in breadth as rough sets of knowledge regions where those regions approximate classification boundaries. The concept of rough sets uses approximations for subsets of inclusion, exclusion, and possibly included sets toward approximating a crisp set of the upper and lower approximations. Fuzzy sets are 'uncertainty sets' that have a degree of membership, and the interpretation follows that fuzzy logic groups are further grouped in a deeper layer ($n_{2,1}$), forming Fuzzy Rough Sets from other layers.

6.1.2 The Numerical Function Interpretation

Each node's input is a vector of activations (α), and each activation is the product of a uniquely associated weight (ω) to that activation. Then the weight activation products ($\alpha\omega$) are accumulated, and a bias (β) for that node offsets the value before the output and the activation function. Therefore, when only one weight is one, and all the other weights are zero, this forms a straight-line graph computationally as $y=mx+c$ (or $y=\alpha\omega+\beta$). As the more significant form is $y=\sum(\alpha\omega)+\beta$, this means that more than one activation (α) and weight (ω) product sums up prior to the bias offset. So in the straight-line graph interpretation, several straight-line graphs with the same or similar bias threshold (β) requirement enrich the representations proportionally from the activation they pertain to with a normalization effect in the weights (see Weighting and Scaling 6.1.1). The straight-line graph form also provides a sub-node resolution between individual nodes and is more continuous than discrete at the output. The straight-line graph form reduces the number of nodes required to represent a function and has a linear function related to a set of activations with an activated receptive field.

The network node breadth thus provides node variations in proportion to the input influences of activations and thus allows different activation sets to select between straight-line graph combinations from activated nodes. As those are selected based on the same activated inputs, a complex function becomes reactive to those inputs with different bias selection thresholds in a sub-node resolution. If excluding the activation function, then when layers are added in-depth, the form will become $y=\sum((\sum(\alpha\omega)+\beta)\omega)+\beta$, and as the biases are zero after initialization, this simplifies to an accumulation of weighted activated inputs. At the outset of learning, the weight scales are by the populations of inputs and outputs and are thus a quasi-mean average like normalization in the initial condition. The activation function is thus required to avoid a linear input to output relationship causing an equivalence simplification in Back Propagation. So a non-linearity function is used to separate the

layers to not collapse to an equivalence simplification and is $y=ReLU(\sum(ReLU(\sum(\alpha\omega)+\beta)\omega)+\beta)$. The *ReLU* or other non-linearity function also provides a non-linearity to a logic scaling between the layers. However, the non-linearity function also provides the complex logic separation forms and decomposition for reused node structures mentioned in the Fuzzy Logical Interpretation section 6.1.1.1. In particular, the *ReLU* function acts with the bias value (β) as a threshold on the output switching on or off further activations in depth. So generally, a network can be (but is not guaranteed to be) like a math function that calls another primitive function in different layers in-depth, where each function representation is a set of nodes that have straight-line graph combinations selected from the weight of the activation to that layer. A neural network combines a logic function and a numerical, straight-line graph set. These are then selected from input weighted activations to a neuron and can provide a continuous representation output between the discrete input weighted activations by the straight-line graph combinations.

6.2 Input Representation to a Network

Traditionally an input representation for a non-imagery computational method can be, as shown in Figure 21, an example of an ANN Fractional PID Adaptive Controller Design [237]. In that application, *r* is the input sample, *y* is the last output, and *e* is the error. The output values are the controller design coefficients, integrals, and differential indexes. Each scalar value's input is individually and uniquely defined.

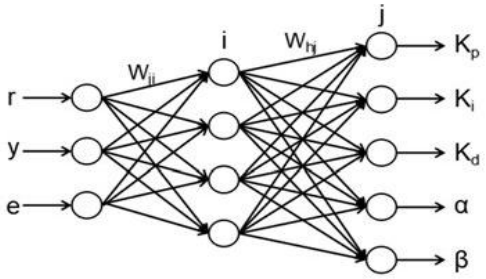


FIGURE 21 NEURAL NETWORK STRUCTURE IN THE PAPER [237] WITH SCALAR INPUTS

Each input value will be the product with the activation with a weight value as strength to a node, and then each node's non-linearity function and bias threshold can be applied for the output to the next layer. Other nodes or inputs in breadth provide proportions of activations with other weights, and each node's bias threshold and non-linearity function can be a product in the subsequent layer. This reliance on the subsequent layer implies that depth is required. However, selecting the node and input activation via the bias implies that breadth is also required. Also, in an imagery format, *r, y* and *e* could be pixels, and the network coding map could also have that image's height, width, and channels. Nevertheless, given that all nodes are inter-connected in dense layers, their actual position is not significant in the network, only that they originated from a pixel and thus have independence. However, this is not true in convolutional layers, as the weights are filters swept across the image rather than mapping directly to the input image pixels.

6.2.1 Value Scale Representation for Strength-Based Activation Weight Set

As the values of *r, y* and *e* have a single weight (ω) for all the values of the activation strength. There is another way to include more weights coupled to restricted value ranges of *r, y,* and *e,* allowing weights to have a unique weight for an activation value strength range. A number-line as an input format de-complicates the node structures for interpretation while including a quadratic quality, as the straight lines can have different weight values map to different activation value strength ranges of a neuron.

6.2.1.1 Number Line as an Input Representation

Thus, representing an arbitrary function within a network and extracting it causes an input representation consideration. The presented form in Figure 21 represents the algebraic function input parameters as numerical values activating the nodes. However, when combined with the weight, using that form would require different weights based on the activation value, meaning that low values can be weaker weighting strength regardless of their semantic prior to the bias (β) and the

subsequent layer in depth. Ideally, that might require more than one weight per activation with a restricted activation value range. An alternative input representation can provide a set of weight groups that can be activation value restricted and independent; this is a vector number line such that the position in the vector is the coded value with more than one weight for the encoding into that vector position based on the value as it is positional. That means that the weight value pertains to separated input variables' value range in a restricted number range instead of the whole number range of an input variable. The activation strengths in the input vector can be in up to two positions at the coded position where those two positions are adjacent to represent the vector's sub-resolution position of the value and add up to 1 in all cases. There will be at least one weight value for each input representation value case, and the minimum activation value of at least one of the activations will range between 0.5 and 1.0 in all value cases. So there is a substantial activation value guaranteed compared to possible faint image pixel values or low scalar value inputs. In the numerical interpretation in section 6.1.2, the straight-line graph combinations of the node will provide sub-resolution linearity between the nodes along with the input value range. That form is also like a monochrome image encoding of a single row of an image where the network could have a prediction based on pixel positions, so as such has a high expectation of working. More rows can be added to the input to add more algebraic function input parameters, but the number line has to be the same length for each parameter as it is a matrix. So for the flexibility of input ranges and the possible different weights activations resolution, the first layer of a dense layer network would be a flattening layer, so the added parameters are a single contiguous vector of all the vector parameters combined. Thus the number of weights and parameter ranges can be varied per algebraic function input. An illustration of the number line numerical representation is in Figure 22, with three-parameter symbols using three contiguous number lines of proportional sub-resolution positions to encode the values to a position.

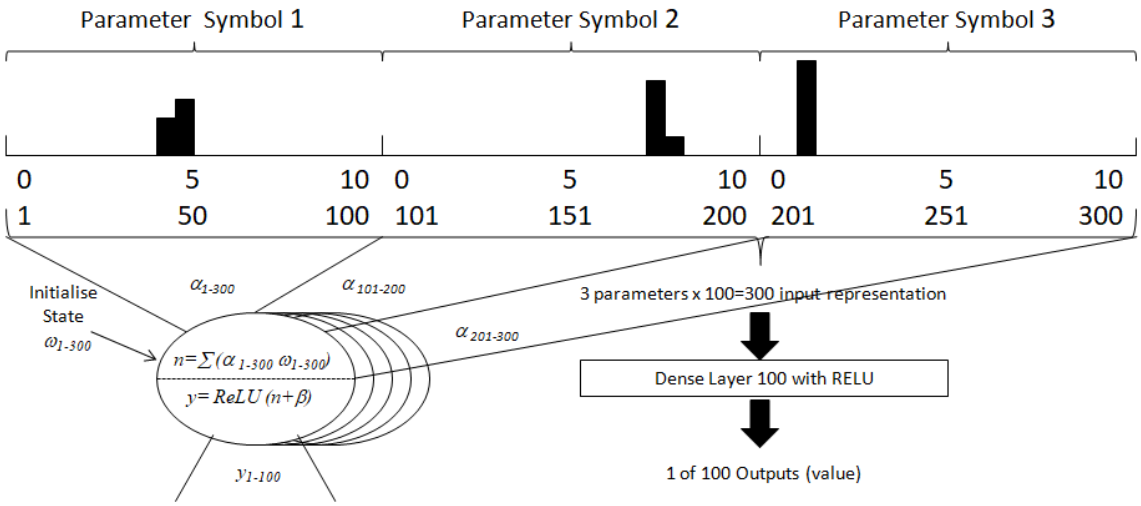


FIGURE 22 NUMBER LINE INPUT REPRESENTATION

In Figure 22, the format is similar to the categorization output format but at the input. However, the sub-resolution positions are not a categorization feature as the categorization outputs are independent and may not represent a linear scale between them.

6.2.1.2 Weight Values at the Start Point of Learning

Traditional weight initialization schemes use random numbers with limits on the distribution derived from the population of activation inputs and sometimes the number of nodes in that layer. The limits differ depending on the distribution between the most common distributions, which are 'normal' and 'uniform.' The initialization limits vary between the Glorot and He initializations. Glorot takes both the activation and node population into account, and He only the activation population, of which the He initialization results in higher numerical limit values and has less dropout potential in deep networks and is the more advanced method. Random numbers provided coverage and variation of values in the initial condition and were envisaged as a Monte Carlo or stochastic analysis approach to overcome incomplete and imbalanced dataset expectations. However, a

drawback is that random initialization states affect the resultant learning accuracy and cause a variation in learning accuracy from learning session to learning session that is visible over regularisation, and this is the subject of Chapter 7. The random forms also may have a further unintentional noise re-colourization effect when multiplying the activations and weights, i.e., the noise from the weights colours with the noise from a sensor's activations. Chapter 7 presents non-random forms for dense and convolutional layers, using Glorot or He limits with linear ramps, sinusoidal slopes, and non-random reordering. These forms provide repeatable determinism from learning session to learning session. The intuition for why they work in dense layers is that the layer nodes are connected fully between the layers, so the order of the nodes is not significant, only the amount in value variation on offer within the defined limit values. The intuition for why they work in convolutional layers is that the non-random reordering shuffle alternative to the random numbers is more predisposed to the application area.

Nevertheless, in this chapter, the weights are positionally encoded based on an activation value, and the initial weight values are 1.0. The method is fair for all activation value ranges at the outset when translated from the input representation. The parameter values become encoded into up to two positions where the summation of those positions is 1.0; this also provides repeatable determinism in the formula extraction method. A more mature method could use the repeatable determinism initialization state in Chapter 7 and then subtract the initial weight values after the back-propagation when deducing the formula; however, this input may not be necessary for illustration. However, for a clear illustration of the method, the initial weight values are 1.0.

6.2.1.3 Input Representation Encoder

The next step was to examine a possible formula extraction method with a single dense layer. To be compatible with the Symbolic AI expert systems method in Chapter 5, as a symbolic algebraic form of a neural network expert system mix, this method of formula extraction required an input representation to represent values for a formula. The input representation is a vector and is not unlike the image pixels of an image when flattened. See Equation (63) to Equation (67) for the encoding of the value (v) into a number line (NL) of the length NL_n, and in the full activation value range between V_{min} to V_{max}.

$$\rho = \frac{(V_{max} - V_{min})}{(NL_n - 1)}, \quad \textit{as the resolution of the number line.} \tag{63}$$

$$\Delta_v = \frac{v - V_{min}}{\rho} \quad \textit{as the offset ratio for value and position.} \tag{64}$$

$$\alpha = \lfloor \Delta_v \rfloor , \quad \textit{as the left hand index in the number line.} \tag{65}$$

$$\beta = \alpha + 1 , \quad \textit{as the right hand index in the number line.} \tag{66}$$

$$NL_{[\alpha,\beta]} = \{1 - (\Delta_v - \alpha), \quad \Delta_v - \alpha\}, \quad \textit{value assignment into the number line (NL).} \tag{67}$$

6.2.1.4 Input Representation Decoder

See Equations (68), (69), (70), and also including Equation (63) for decoding the number line (NL) in a value (v) captured within that input representation. The decoder can verify the input activation encoding and conceptually might be like an activation function for the conversion back to scalar representations in the next layer.

$$K = \{0,1 \ldots NL_n\} \cdot \rho + V_{min}, \quad \textit{as the weight vector for a centre of gravity.} \tag{68}$$

$$NL_{CoG} = \{NL \cdot K\}, \quad \textit{as the weighted centre of gravity vector.} \tag{69}$$

$$v = \frac{\left(\sum_{i=0}^{NL_n - 1} (NL_{CoG}[i])\right)}{\left(\sum_{i=0}^{NL_n} (K_{[i]})\right)}, \quad \textit{as the value from a centre of gravity calculation.} \tag{70}$$

The encoding method at the input is similar to categorization coding generally used at the output layer, but this scheme can have values between the tensor vector elements proportionately, representing a continuous number line in a discrete resolution. The generated dataset has a range of

values from 0 to 9.9999 in steps of 0.0001, which applies to the simple dense layer neuron model in Figure 23.

6.3 The Formula Extraction Architecture

The formula extraction method's model architecture is in Figure 23, which is the weights' back-propagation training model.

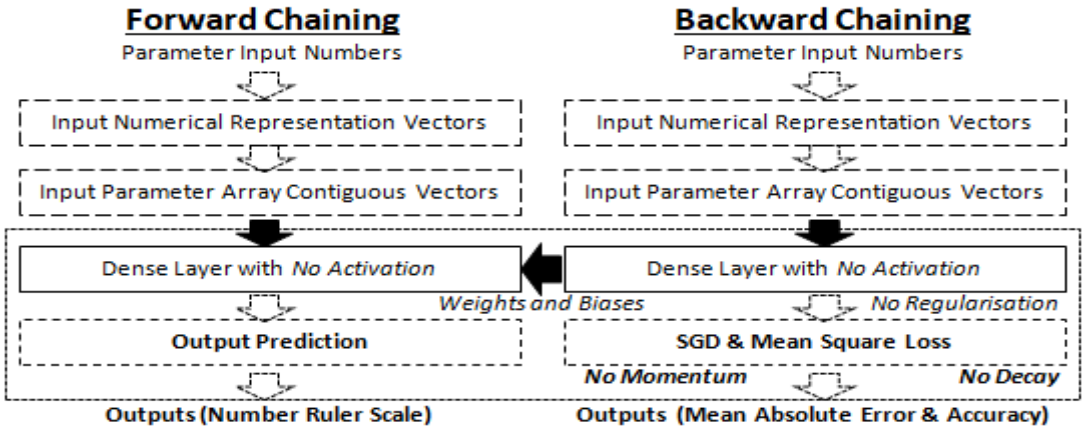


FIGURE 23 SIMPLE NEURAL NETWORK MODEL FOR FORMULA EXTRACTION [J3]

To demonstrate the method, a simple function $y = sin(x) \times x$ forms a generated dataset, where y is the output and x is the input 0 to 9.9999 in steps of 0.0001. Figure 24 shows the input in Blue, in Red is the pre-calculated expected y values and in green dashed is the actual predicted output, with no shuffle in Figure 24 left, and the shuffle enabled in Figure 24 right both with a Learning Rate of 1.0.

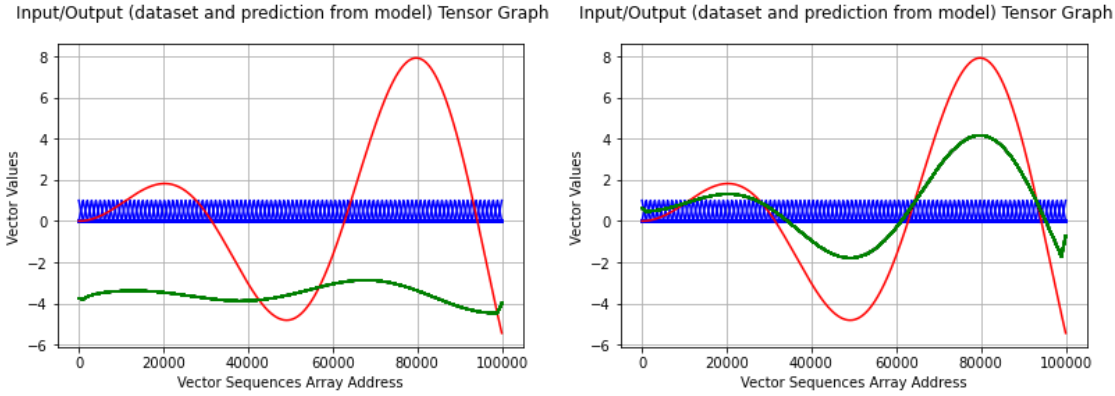


FIGURE 24 MODEL OUTPUT & EXPECTATION LEFT UN-SHUFFLED, RIGHT SHUFFLED [J3]

When un-shuffled (Figure 24 left), the order of update has caused a skew in the difference between the expected results (in red) and the model prediction output (in green); there is also a lower amplitude and an offset in those values. When the shuffle is used (Figure 24 right), there is a reduction in the offset and skew as the updates do not have an update direction implied in the dataset like a sweeping filter, and this shows that dataset order is significant and when the dataset order is varied this provides less skew, offset and more significant amplitude response. Although there is an improvement in the skew and offset (Figure 24, right), the amplitude is still reduced compared to the pre-calculated expected results (in red). The dataset could have more updates in higher resolution of the input, but also other methods: Momentum in the gradient descent, additional epochs, and further adaption of the Learning Rate can be applied. The last two methods (epoch and Learning Rate) depend less on the model architecture in future models. Increasing the Epochs will increase the updates shown in Figure 25 left. However, keeping a single epoch and setting the learning rate to 10.0 will increase the influence of each of the current number of updates, as shown in Figure 25 right.

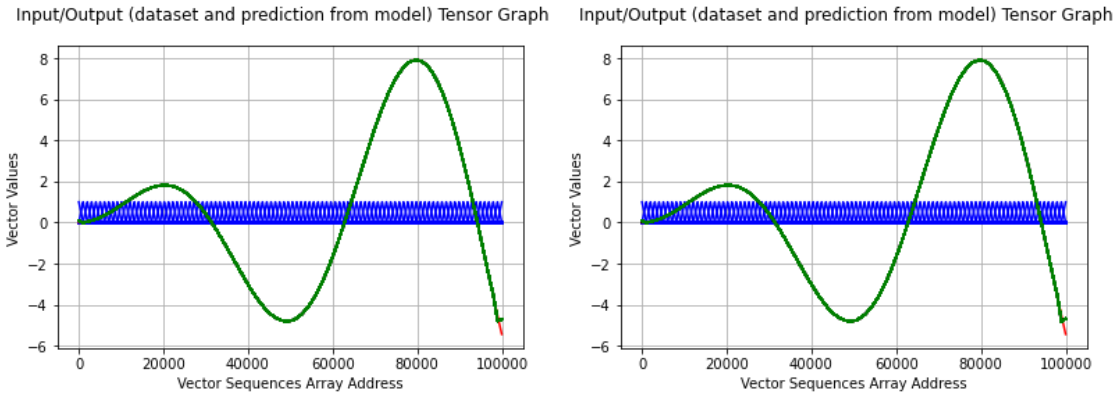


FIGURE 25 MODEL OUTPUT & EXPECTATION LEFT [LR=1 10 EPOCH] RIGHT [LR=10 1 EPOCH] [J3]

Figure 25, left and right, are identical and both show that the input representation is working at this point, and the learning creates the model prediction function (in green) within a network that matches the expected pre-calculated dataset values (in red but eclipsed by the green line). More updates (i.e., epochs) of minor influences (i.e., lower learning rate) will provide a higher accuracy but with a risk of over-fitting as the back-propagation nudges in Gradient Descent are minor; thus, that method is favourable as over-fitting is desirable to extract the formula. There is still a minor discontinuity in both results at the higher end of the value range; this will reduce as more parameters to the formula are added. Although learning creates a prediction function, the regularisation is also unset as the method wishes to over-fit the model for extraction purposes as it reduces the x to y influence in regression. Also, the activation function is disabled as the full linear value range is required, and this is effectively regression, where that regression is applied to different value ranges along the input representation number line. Traditionally in an ANN, the absence of the activation function can collapse the layers as there can be numerical equivalence in single layers; traditionally, a single layer cannot solve more complex logic like the XOR as there is a single layer weight for all strengths of activation. However, in this input form, it can, as weights connect to the activation strength value ranges directly and exclusively. Weights of different node activations can modify activations, and the weight can be sensitive to only some input activation value ranges. This method uses a different input representation for a formula extraction to yield a newly learnt rule supporting the algebraic expert system method in Chapter 5 as a component of a Neuro-Symbolic AI method.

6.3.1 Discriminating Division and Multiply Operator Relationships

Further extending the function to two input parameters: x, and z, where the prediction output is $y=sin(x)\times z$, and the z input is $z=x/2$. Then the resultant learnt model prediction is as in Figure 26 left, again with the model prediction in the green dashed line and the predetermined function expectation in the red line, where the green line prediction exactly overlays the pre-calculated expectation values.

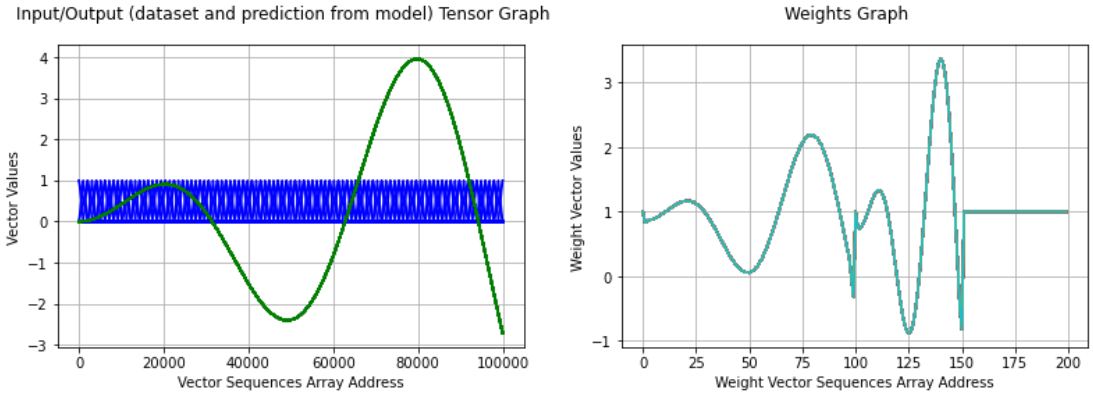


FIGURE 26 RESULTANT WEIGHT TENSOR FOR FORMULA EXTRACTION WITH 2 INPUT PARAMETERS [J3]

Furthermore, the weight vector from the two-parameter input model in Figure 26, left and right, shows the predicting function weights for parameters x and z. The prediction function y value

mapping is in both parameters, but parameter *z* has been compressed as the number line input only used half the vector as per the *z=x/2*. Thus given the input representation, a divide operator has a compression in the weights connected to that input range used. The expectation is that divide operators compress that representation relatively given the number range used, and multiply operators will expand it. Note that the un-used vector still contains the initialization weight value of 1.0. In this case, the relationship of *y* to *z* is half that of *y* to *x* as derived from the weight observations.

6.3.2 Discriminating Addition and Subtraction Operator Relationships

Further extending the inputs with a parameter *v*. the function parameters now become *v=z+4*, where *z=x/2* and *y=sin(x)×z+v*. Figure 27 left has the expected *y* values in red, the input vectors are in Blue, and the Greenline is the model prediction. The weights after learning are in Figure 27, right, and the same relationship to *x* can be seen in compression but is offset by the value 4, as per the value in the *v* parameter's addition operation.

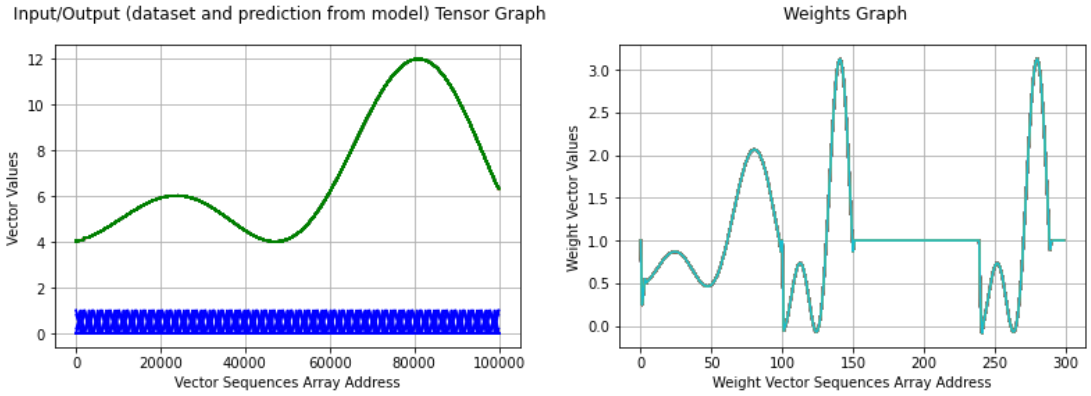


FIGURE 27 RESULTANT WEIGHT TENSOR OF FORMULA EXTRACTION WITH 3 INPUT PARAMETERS [J3]

Intuitively the resultant weights from the simple neural model, when exposed to that dataset, are subject to regression *(least squares)* optimized with Stochastic Gradient Descent (SGD), and the numerical input representation caused the model optimization to distribute the influence of the input parameters, towards the output over many weight values. These weights individually represent the inputs to output, allowing the weights to characterize the math operator used between those inputs. However, some machine learning features were disabled for clarity (regularisation and activation), as this model would not be used to predict unseen data, nor did it want the number representation to be limited. It used a single layer, anticipating that the method repeats per layer and fewer layers might be required, and the layer type was dense.

6.3.3 Optimising Learning Rate and Number of Nodes

When a common denominator reduces the Number of Neurons and Learning Rate, it sustains the loss measurements, as shown in Table 3 with the same values coloured in the same colour. Note that the high number of significant digits in the results allows for the grouping of the exact result values.

TABLE 3
LEARNING RATE AND NUMBER OF NEURONS OPTIMISATION

Division Factor	No of Neurons	Learning Rate	Mean Square Error	Mean Absolute Error
1	100	1.0	1.6964304450084455e-05	0.0015970466192811728
2	50	0.5	1.6964357200777158e-05	0.001597068621776998
4	25	0.25	1.6964357200777158e-05	0.001597068621776998
5	20	0.2	1.6964289898169227e-05	0.001597053837031126
10	10	0.1	1.6964289898169227e-05	0.001597053837031126
20	5	0.05	1.6964289898169227e-05	0.001597053837031126
25	4	0.04	1.6964319001999684e-05	0.00159705919213593
50	2	0.02	1.6964319001999684e-05	0.00159705919213593
100	1	0.01	1.6964282622211613e-05	0.0015970563981682062

Table 3 demonstrated that the method is robust to different Numbers of Neurons in different architectures. The neuron reduction works because the input format representation exploits extra weights for activation strengths (as value ranges) connected to each neuron, as per section 6.1.2. Therefore, the number of neurons is less critical than the number of activation weights as the linear regressions operate over the different numerical ranges as a weighting of those segmented activation value ranges. The discrete weighted activations provide a set of straight-line graphs to a neuron segmented to different activation value ranges from the input format. The relationship between the Learning Rate and the Number of Neurons when there is a common denominator is that there are fewer neurons to update, there are fewer classification boundaries possible, and that lowers the resolution making the learning step in update more significant in gradient descent, so the more minor learning step couples to the sustainment of the performance. Figure 28 left shows the prediction function in green and the expected *y* value in red eclipsed by the prediction with only one neuron.

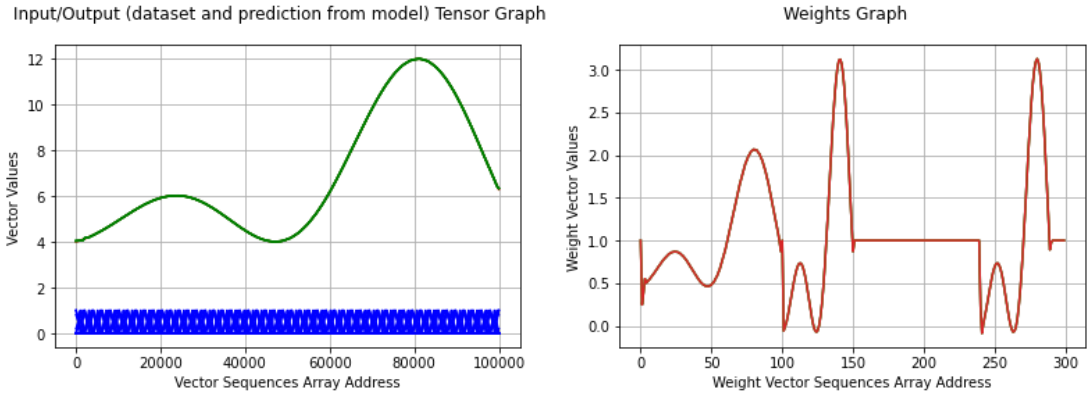


FIGURE 28 RESULTANT WEIGHT TENSOR FROM FORMULA EXTRACTION WITH 1 NEURON

The numerical operators are still interpretable as the number of activations is unchanged. Only one neuron is needed in this case, as this method exploits the weights attributed to different value strengths. As such, the classification boundaries are also paired to the activation strengths coding in the input representation.

6.3.4 Revealing Further Learnt Content

The learnt weights indicate the deduction of the mathematical relationships between the inputs to outputs. However, when those deduced mathematical operators were reversed in operation and applied to the prediction outputs, the residual of a sine function remains (See Figure 29) that was deliberately not in the inputs, so exposing a learnt mathematical function in that layer that was present in the training data but not in the inputs to that layer.

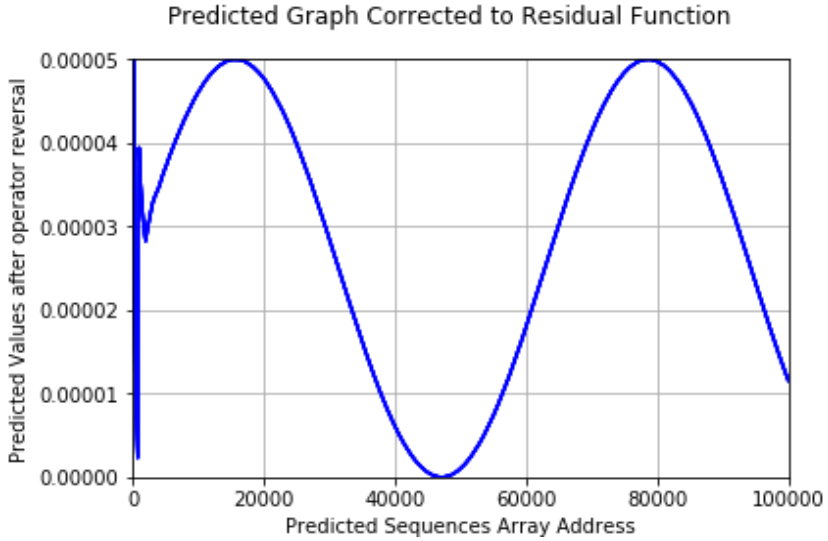


FIGURE 29 RESIDUAL OF THE LEARNT SINE FUNCTION IN PREDICTION [J3]

6.4 Summary of Numerical Discrimination

A variation of simple mathematical operators was combined and, in this case, used primitive add and divide operations; this demonstrated that a symbolic formula representation with a high degree of accuracy is possible. However, the weights could also establish the relationships between *y* and the *x*, *z*, and *v* inputs. Add and subtract operators displaced the learnt *y* pattern in the weights, multiplying and dividing stretched and compressed them. This method observed a very high degree of accuracy, with low losses, as there was no noise and regularisation. Also, the residual of the hidden function was recognized when reversing the deduced math operations.

Confined to dense layers, this was early work towards a formula extraction method in support of Neuro-Symbolic AI, but rather than a rule extraction, it is an alternative method. That is to say, the output is naturally generalized rather than a dense set of logic equations in some rule extraction methods that need to be generalized. The method forms a decompositional approach, which would operate layer by layer, examining weight relationships to the layer inputs and exposing available residuals of learnt mathematical functions not present in the inputs, but only in the learnt weights for prediction outputs. In the context of Neuro-Symbolic AI, regression and SGD can be utilized in a different form using a numerical representation of input symbol number ranges, and those weights discriminate the input-to-output relationship. Further work is required for more complex math functions like exponential and trigonometric functions, perhaps using convolutional layers or other filters as recognizers and discriminators. However, this work establishes a repeatable deterministic experiment environment and a rule extraction thread to solve intractable problems with a neural network. The new rules can then be a subject for the neural expert system method in Chapter 5. Partial results were published in a peer-reviewed conference paper [C3[5]] and a journal version [J3[6]].

6.4.1 A Benefit to ANN by Augmenting with the Input Representation

More generally, the input representation might enhance image processing by reducing model depth, as model depth can increase the potential for vanishing gradients. The method makes a colour image 4 Dimensional, where those dimensions are: height, width, colour channels, and activation strength value ranges. That activation strength dimension thus captures some logic and numerical representation, which would otherwise require model depth for other value ranges of an activation input by having more than one weight per activation associated with different number ranges. The activation strength dimension would be prior to the activation function used. The activation strength dimension might also benefit the low activation values as the activation used in small values will be between 0.5 and 1.0, which might avoid vanishing gradients. The extra weights required might also be omitted from the population count in the initialization scheme limit calculation as proven with a single initialization value. An augmented method to ANN might use the current node He et al. or Glorot Xavier limited value for that whole weight dimension of each input activation weight set. Finally, that dimension can be more interpretable with primitive add and divide operations. The representation may cause a network to be more interpretable in Safety-Critical AI towards explainable AI. However, repeatable determinism for safety-critical aspects of the application area and accuracy in analysis support is addressed further in Chapter 7.

[5] [Cn] Published conference papers are in a separate bibliography on page xv.

[6] [Jn] Published journal articles are in a separate bibliography on page xv.

Chapter 7

SAFETY-CRITICAL AI FOR NEURAL METHODS AND TRANSFERRED LEARNING

Explaining the content of a neural network is challenging to establish neural networks as palatable in the application area, but some insights are in Chapter 6. Neural networks also need a repeatable and deterministic quality, so testing and qualification are repeatable. A significant impact on this is random weight initializations, and an alternative non-random scheme with the same or better performance as the existing random weight initializations is required. This chapter examines non-random weight initialization schemes for dense and Convolutional layer types. The chapter will also employ the adversarial attack with the Fast Sign Gradient Method (FSGM) for evaluating the weight initialization impact in transferred learning between two datasets, where one dataset has a controllable distortion via the FSGM approach relative to the other.

7.1 Safety-Critical Aspects

Both within and out of the application area, safety-critical AI aspects apply and thus also relate to applications in human life like Smart Cities [238], [239], and these applications also have public or legal liabilities [240], particularly in hazard avoidance [241]. A general goal of AI is to reach a performance that is better than a human baseline and is free from human error [242], [243], [131]. Thus AI that can be trusted is a challenge [244], both in verification and validation and within the processes used [245]. In many cases, AI applications have been used as decision assistance rather than decision making, avoiding that liability [52], [246], [247], [248], [249], [250], [251].

Leaving aside the legal and ethical standpoints of a machine making decisions, technologically, a variation of performance results in repeated sequences of learning sessions is a problem for safety and quality. It suggests that more than one valid solution exists and questions how many more possible solutions there are and which ones are both optimal and safe. Indeed it questions if there is catastrophic content in some of them.

Another approach is to have a repeatable result from every learning session, which allows for higher investment in a solution's qualification testing while permitting a regression testing approach to be more acceptable. Thus repeatable determinism is desirable when combined with other approaches as part of safety-critical AI, but repeatable determinism also aids other safety-critical AI development and testing approaches. With this in mind, using random number initializations that vary in each solution and yield a variance in visible results over regularisation can be a problem. A repeatable and deterministic non-random number initialization is desirable to focus critical safety approaches to fewer or single solutions in testing. Two main random number variations used are the epoch shuffle and the random number initialization state.

The question of random numbers and hidden structures are questioned and explored by Fang et al. compared to quantum mechanics [252] and Duch et al. in initialization [253]. However, the research aim in this chapter is an alternative to the random number initialization, such that there is a single solution that is singularly deterministic and repeatable.

7.1.1 Unexpected Random Number Source

It is common to assume that seeding the random number generators would provide a repeatable result; this was not the case in some frameworks using CPUs. An unexpected problem arose when experimenting with the tools. When running on a CPU, the model accuracy varied whenever training the model even if there was no change, and also, the random number generator was seeded. More concerning, when using non-random initialization values: for the weights as part of the research, shuffles were disabled, but the variations in accuracy continued, threatening repeatability. The CPU

and multi-task scheduling events problem was solved using *'processor affinity'* and *'real-time priority'* in critical code sections, denying the task scheduler and avoiding the internal 80bit extended precision floating-point register to not be rounded on task scheduling events. Critical code sections made CPUs available in applications with repeatable determinism where GPUs are unavailable; a GPU might be too power-intensive in some Smart Cities applications. This solution also provided greater Information Assurance (IA) in AI development by avoiding a rounding truncation and increasing model accuracy.

7.1.2 Current Initialisation Schemes

Currently, there is the everyday use of, Glorot/Xavier and the He et al. initialization schemes [254], where the method suggested by He et al. is regarded as the more advanced method, but where both are not coupled to the dataset, providing signal gain normalization based on the input activation populations only and attains a distinct minimum quicker.

There has been some questioning of whether random initialization is a thing of the past [255], which states that random numbers cause an unlearning of the initial condition, and an asymmetry could be an advantage if it matches the after-learning state. Both Glorot/Xavier and the He et al. initializations use random numbers. However, those two schemes relate mainly to set limits to the random numbers given different random distribution and activation functions rather than the random numbers themselves.

Time evolution is a subject of RNN and LSTM networks, but outside of ANN, Melan et al. [50] used a BN to grade rules progressively in learning sessions. Learning of an initial condition with pre training or learning is a theme but couples the initialization state to the dataset specifically and still requires unlearning of an original initial condition. More ideally, the initial condition would be a general case and thus use more of the dataset from the outset of learning rather than investing some of it for deriving that initial condition.

There has been an appetite for mission-critical applications of ANN for some time, and examples are uncrewed air vehicles [93], crewless space missions [94], and unattended communication fault diagnoses [95]. As such, an alternative to the random numbers used in Glorot/Xavier and the He et al. initialization schemes is desirable such that learning when unattended can have a repeatable deterministic outcome should retraining be required remotely or progressively.

7.2 Repeatable Determinism: Dense Layer Networks

There were two publications for dense layer models, and the first publication [C2[7]] used fixed value limits derived from the baseline model; this would show that a non-random form was viable but was slightly underperforming in the accuracy score with the Glorot/Xavier baseline. However, resolving the unexpected random number source problem provides repeatability. The second publication [J2[8]] extended that work using the Glorot/Xavier initialization limits. i.e., replacing the random numbers only, and achieving an equal performance to the baseline in direct comparison, showing the non-random form can be equivalent.

7.2.1 A Familiar and Well Understood Baseline Model

Initially using a dense layer network with the MNIST dataset [150] of black and white handwritten numbers, the model architecture in Figure 30 used TensorFlow, Keras, and NumPy to form the experiment model for further experiments. The dataset and model architecture is familiar, trusted, and well understood by researchers.

[7] [Cn] Published conference papers are in a separate bibliography on page xv.

[8] [Jn] Published journal articles are in a separate bibliography on page xv.

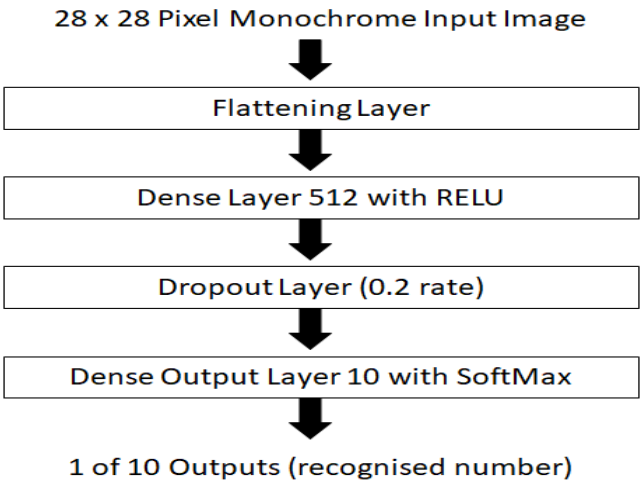


FIGURE 30 ARCHITECTURE OF THE BASELINE MODEL [J2]

7.2.2 Experiments and Method

In the more advanced MLP research in the journal version [J2], several experiments and an experimental model were defined in Figure 31, forming the experiment method. The baseline performance would be the experiment control case, with a further variation of initial weight state defined in four classes: Class one is the experiment control shown in green; Class two uses Fixed Values at limits shown in blue; Class three uses Linear Ramps, testing number ranges with a fixed slope shown in orange, and finally Class four used Sinusoidal slopes to make a variation to the slope gradient shown in red. Please note that all the measurements are from a cross-validation dataset.

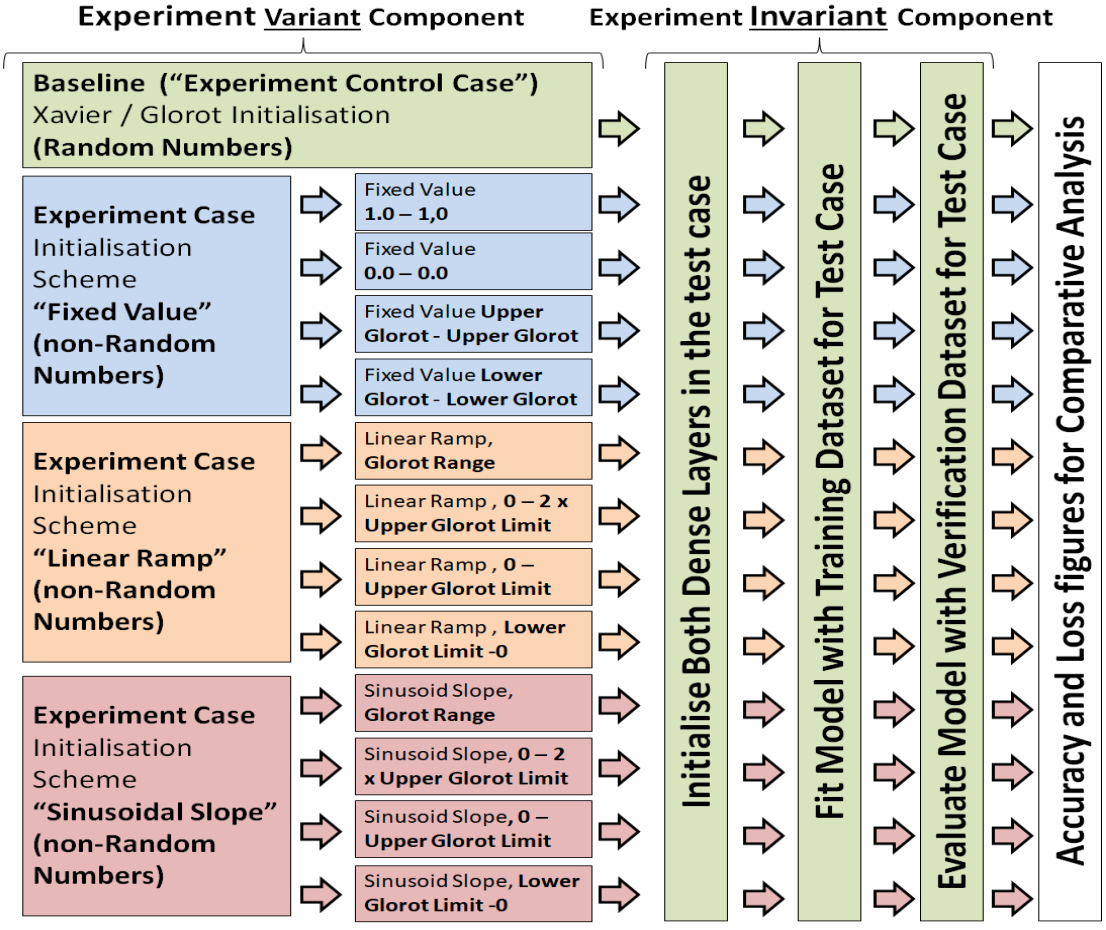


FIGURE 31 ARCHITECTURE OF THE EXPERIMENT'S DESIGN [J2]

The model results are in Table 4 using the ten epochs of that model and run over ten separated learning sessions, showing the accuracy variation due to the random number start conditions and the unexpected random source caused by task scheduling events rounding the internal extended floating-point register.

TABLE 4
BASELINE RESULTS WITH A RANDOM INITIAL CONDITION WITH 10 EPOCHS [J2]

10 Epochs Run Each	**Loss**	**Accuracy**
1st Random Learning Session	0.06106613632314256	98.18
2nd Random Learning Session	0.06175447308695293	98.16
3rd Random Learning Session	0.07186600035531446	97.72
4th Random Learning Session	0.06600431568695349	98.18
5th Random Learning Session	0.06500834331280785	98.13
6th Random Learning Session	0.06586962914280885	98.01
7th Random Learning Session	0.07172874092692509	97.96
8th Random Learning Session	0.08020385432066396	97.65
9th Random Learning Session	0.0815817079940578	97.71
10th Random Learning Session	0.07415228985190625	97.94
Mean Average	0.069923549	97.964
Variance	5.19614E-05	0.042737778
Standard Deviation	0.007208428	0.206731173

Table 4 shows the accuracy variance due to different random number sequences is about 0.043 and a standard deviation of 0.2 around a mean of 97.964% accuracy. Those results achieved about ~98% of that model's stated accuracy. As the first epoch after learning is the epoch of interest, being the epoch after the initialization, the results include a single epoch with no dataset shuffling to disregard that random effect too, and thus Table 5 also shows the results for a single epoch un-shuffled.

TABLE 5
BASELINE RESULTS WITH A RANDOM INITIAL CONDITION IN A SINGLE EPOCH UN-SHUFFLED [J2]

Run	**Loss**	**Accuracy**
1st Random Learning Session	0.1266106634631753	95.91
2nd Random Learning Session	0.1216393306143582	96.21
3rd Random Learning Session	0.13143637651763856	95.62
4th Random Learning Session	0.1323663795016706	95.74
5th Random Learning Session	0.12944038207307457	95.83
6th Random Learning Session	0.13047181819714607	95.69
7th Random Learning Session	0.13344295675437898	95.7
8th Random Learning Session	0.13349654669184238	95.58
9th Random Learning Session	0.12230887789316476	96.14
10th Random Learning Session	0.12589706211015583	95.93
Mean Average	0.128711039	95.835
Variance	1.92267E-05	0.045316667
Standard Deviation	0.004384824	0.212877116

In Table 5, the accuracy variance and standard deviation are similar; however, around a ~2% lower mean of 95.835% accuracy, and as the learning, after initialization is with disregard to epoch random shuffle sequences, this forms the 'baseline experiment' control performance case of the learning.

7.2.2.1 Fixed Value Scheme

Expecting that fixed weight values will not be as high performing as the initial condition, like the weights, when set to the same value, in back-propagation, many nodes may calculate the same nudged values; as such, the update lacks diversity in the network, and caused the duplication of node updates, lowering the network's efficiency. Each of the results over ten separated learning sessions is in Table

6. Note that the variance in the results is due to the unexpected random number source caused by the task scheduler events rounding the CPU's internal floating-point extended precision register.

TABLE 6
FIXED VALUE SCHEME OF WEIGHT INITIALISATION [J2].

Experiment	Loss and Accuracy			Comment
Fixed value 1.0		Accuracy	Loss	This scheme is the lowest score. However, it still may have some applications to reserve a network area for later use, like unused input vectors.
	Mean	**10.1%**	14.49016949	
	Var	0	0	
	StdDev	0	0	
Fixed value 0.0		Accuracy	Loss	Low performing and compares with the negative number experiment. However, it may have some applications for a network area to be disregarded.
	Mean	**11.52%**	2.301160769	
	Var	0	5.88128E-15	
	StdDev	0	7.66895E-08	
Fixed value Upper Glorot limit		Accuracy	Loss	Although low performing, the highest score shows that the Glorot value has benefit, although only using that value is under-utilized the network.
	Mean	**28.156%**	1.791479329	
	Var	0.124671111	8.40606E-06	
	StdDev	0.353087965	0.00289932	
Fixed value Lower Glorot limit		Accuracy	Loss	It compares with the zero number experiment and may conflict with the use of ReLU in the first dense layer.
	Mean	**11.35%**	2.301160767	
	Var	0	3.49831E-15	
	StdDev	0	5.91465E-08	

As expected, a Fixed Value scheme is low-performing, as a fixed number used in the weights lacks diversity in updates. However, the green row shows that the upper Glorot value is the highest performing, even with that lack of diversity. In comparison with the research in Chapter 6, which used a fixed-value initialization state of 1.0 successfully, that success in Chapter 6 was because of the different input representation and the use of multiple weights per input, assigned to different activation strengths, of which these results use the traditional single weight per input for all activation strengths; as such, the outcome is not so successful as Chapter 6.

7.2.2.2 Linear Ramp Scheme

The Linear Ramp provides a diversity of weight values with a range of weight values in the network. The distribution is uniform as per the random form in the baseline. The experiment tests a number range with the same or similar slopes but is a fixed slope.

TABLE 7
LINEAR RAMP SCHEME OF WEIGHT INITIALISATION [J2]

Experiment	Loss and Accuracy			Comment
Ramp through Glorot range.		Accuracy	Loss	The result is only 1.5% lower accuracy from the baseline in this case of substituted random numbers.
	Mean	**94.269%**	0.184886392	
	Var	0.276676667	0.00024044	
	StdDev	0.526000634	0.015506117	
Same Slope as the Glorot range but slid up to positive numbers.		Accuracy	Loss	The result is 4% lower accuracy from the baseline in this case.
	Mean	**91.475%**	0.268064027	
	Var	0.024094444	9.9364E-06	
	StdDev	0.155223853	0.003152205	
Change in slope but in positive and Glorot limited.		Accuracy	Loss	The result is 3% lower accuracy from the baseline in this case..
	Mean	**92.448%**	0.239309289	
	Var	0.105462222	9.02308E-05	
	StdDev	0.324749476	0.009498988	
Same slope as the above experiment but negative values and -Glorot limited.		Accuracy	Loss	Accuracy is low performing, and the ReLU activation function may have affected learning from the outset by being lower than the bias threshold.
	Mean	**11.35%**	2.30116078	
	Var	0	2.33796E-15	
	StdDev	0	4.83524E-08	

In Table 7, the linear ramp scheme within the Glorot range is the highest performing and is in green. Although the unexpected noise source is still causing variance in those results as the task scheduler event is still rounding the internal extended precision floating-point register.

7.2.2.3 Sinusoidal Slope Scheme

The sinusoidal slope scheme provides a variation in the steps in values within the number range; as such, it varies the gradient in the weight diversity and the steps within that diversity. Table 8 provides results from different number ranges and varying gradients.

TABLE 8
SINUSOIDAL SLOPE SCHEME OF WEIGHT INITIALISATION [J2].

Experiment	Loss and Accuracy			Comment
Sinusoid slope in Glorot Range.		Accuracy	Loss	The result is almost the same score as the same number range with the equivalent linear ramp experiment.
	Mean	**94.886%**	0.168972311	
	Var	0.022937778	8.76335E-06	
	StdDev	0.151452229	0.002960295	
Sinusoid slope from twice the Glorot upper limit to 0.		Accuracy	Loss	The result is almost the same score as the same number range with the equivalent linear ramp experiment.
	Mean	**91.413%**	0.279573134	
	Var	0.01189	2.41043E-05	
	StdDev	0.109041277	0.004909613	
Sinusoid slope from Glorot upper limit to 0.		Accuracy	Loss	The result is almost the same score as the same number range with the equivalent linear ramp experiment.
	Mean	**92.628%**	0.243604778	
	Var	0.08944	7.93523E-05	
	StdDev	0.29906521	0.008907991	
Sinusoid slope from 0 to lower Glorot limit.		Accuracy	Loss	The result is almost the same score as the same number range with the equivalent linear ramp experiment.
	Mean	**11.35%**	2.301160704	
	Var	0	6.39136E-15	
	StdDev	0	7.9946E-08	

In Table 8, the Glorot range is again the highest performing and slightly higher than the equivalent linear ramp, but the task scheduler event still compromises the internal extended precision floating-point register.

7.2.3 Avoiding a Misleading Conclusion

Using the two highest-scoring schemes of sinusoidal slope and linear ramp but with ten epochs and the shuffle enabled, a comparison to the original baseline performance is in Table 9 and Table 10.

TABLE 9
SINUSOIDAL SLOPE SCHEME OF WEIGHT INITIALISATION 10 EPOCHS SHUFFLED [J2]

Run	**Loss**	**Accuracy**
1	0.06873708092225715	97.99
2	0.07566913830568082	97.75
3	0.06941359058758244	97.81
4	0.07690233801202849	97.75
5	0.07229105311079184	98
6	0.07870250816526823	97.79
7	0.06857179706634488	97.98
8	0.07224223068275024	97.86
9	0.07307772935463581	97.75
10	0.07484171458326745	97.87
Mean	0.073044918	97.855
Var	1.22188E-05	0.010494444
StdDev	0.003495545	0.102442396

TABLE 10
LINEAR RAMP SLOPE SCHEME OF WEIGHT INITIALISATION 10 EPOCHS SHUFFLED [J2]

Run	Loss	Accuracy
1	0.06948850467810408	97.93
2	0.0810321348624304	97.66
3	0.07320999270802131	97.81
4	0.0818435779891268	97.49
5	0.06348531504337443	97.95
6	0.07764026021502214	97.67
7	0.08206962382048369	97.53
8	0.07047365378377726	97.86
9	0.07122972569263075	97.9
10	0.06634688437929144	97.93
Mean	0.073681967	97.773
Var	4.42831E-05	0.029801111
StdDev	0.006654552	0.172629983

Table 9 and Table 10 both present the highest performance in run five, shown in green, and the sinusoidal slope's mean average is higher with lower variance and standard deviation. However, the maximum and minimum performances between the sinusoidal slope and the linear ramp are overlapped, as shown in Figure 32, left and right.

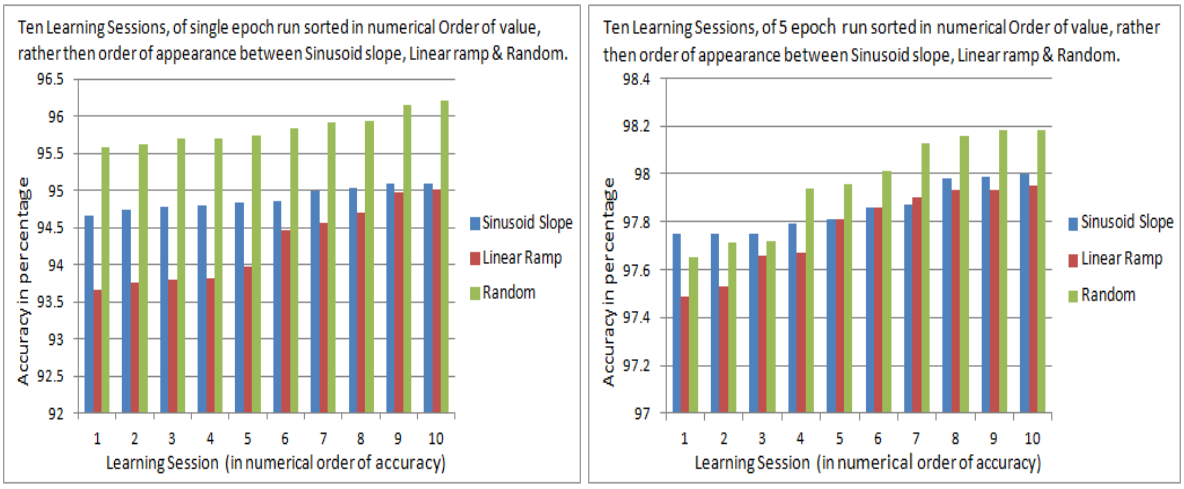


FIGURE 32 ACCURACY OF HIGHEST SCORING SCHEMES, SINGLE EPOCH LEFT 10 EPOCHS RIGHT [J2]

From Figure 32, left and right, these results could be misleading to conclude the random number initialization sequences are superior probabilistically, although other non-random sequences could be equivalent. However, the probabilistic content in what is to be deterministic content means that the known variances put a different perspective on the results.

Figure 32 actually shows the vulnerability sensitivities of those methods to the unexpected random number source. If that unexpected random number source is a 'numerical instability' or rounding corruption, then the random number initialization sequences are less sensitive to it, which is what Figure 32 shows. When termed as a 'numerical instability' rather than a noise source, the notion of a corruption or an IA (Information Assurance) threat is imperative to resolve that numerical instability to gain the true clarity of the results.

As such, the task scheduler was denied via real-time priority [256] with a single processor affinity [257] to preserve the integrity of the 80-bit internal floating-point register [258]. Upon which the results became deterministic and repeatable in every separate learning session. The sinusoidal slope, linear ramp, and random number scheme are re-measured within Table 11.

TABLE 11
RE-MEASURED SINUSOID, LINEAR AND RANDOM: WITHOUT UNINTENTIONAL RANDOM SOURCE

Type	**Loss**	**Accuracy**
Sinusoidal slope	0.06988054851347116	97.93
Linear Ramp	0.06633297475341242	98.05
Random Numbers	0.061059941675240405	98.05

Table 11 shows that the linear ramp achieves an equal score to the random form shown in green without the numerical instability. The intuition for this is that the order in which the weights are in the network is not significant with the fully connected dense layers. As the random form is also uniformly distributed, it matches the distribution of the linear ramp and thus is equivalent.

7.2.4 Summary of MLP Dense Layers

The research initially used a pure dense layer network and non-random schemes for weight and bias initializations. That research found that the number range rather than the gradient was necessary and showed an almost equal performance using a linear ramp between the values -0.05 and +0.05. That number range would match the test case model, and subsequently, the journal paper version [J2] used the Glorot/Xavier limit values instead, making it adaptive to other models and achieving an equal accuracy result in a non-random initialization. Also, accuracy was arrived at in the first learning session and was repeatable, whereas the random form had variations and took several learning sessions to achieve the highest score. Intuitively the substitution of a linear ramp for a random form achieving equality in performance is understood by the nature of dense layers being: fully and feed-forward connected, meaning that the node order is not relevant, only the numerical steps and the distribution. That supports the finding that a gradient change was not beneficial; however, avoiding the value zero and Glorot/Xavier limit values would also avoid potential dropout and saturation.

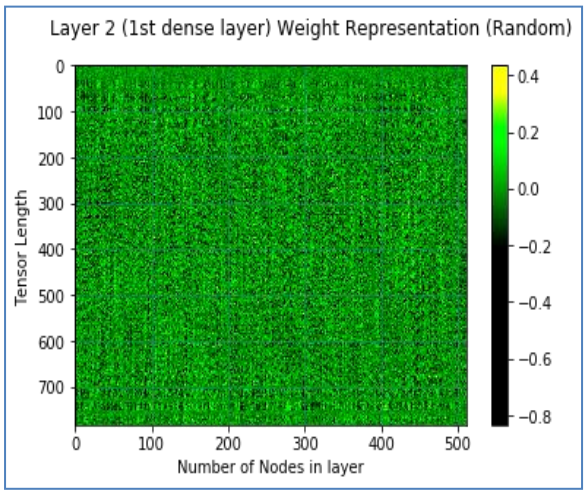


FIGURE 33 ORIGINAL RANDOM SCHEME, WEIGHTS AFTER LEARNING [J2]

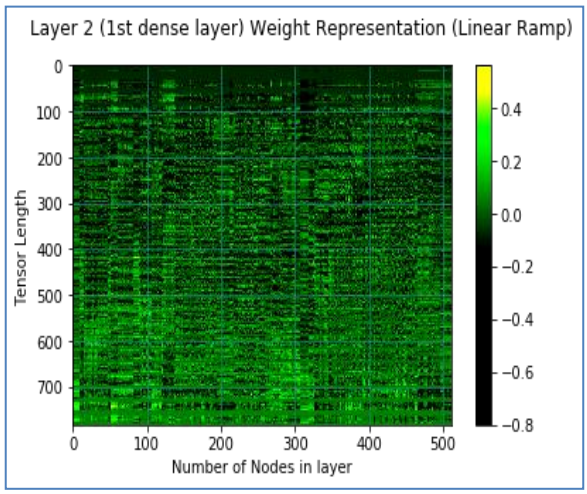


FIGURE 34 NON-RANDOM SCHEME, WEIGHTS AFTER LEARNING [J2]

Additionally, the non-random initialization state provided a structure to the weights, which may benefit subsequent rule extraction, and grew the weights in a structured form (See Figure 33 and Figure 34 for comparisons). Figure 33 shows the weights after learning with the random scheme, and Figure 34 shows the weights after learning but with the non-random linear ramp scheme. Both schemes achieve equal model accuracy and are equivalent, although perhaps reordered, but Figure 33 (right) shows that the non-random scheme has a structure along the '*Number of Nodes*' axis and correlates with the '*Tensor length*' axis at pixel positions. That correlation may have value to rule extraction, as the weights are linear aligned, and as such, the node semantics index to adjacent nodes that pertain to pixels, and in the non-random form, the weights are aligned and clustered, making the generalization step easier in a rule extraction method. Blumenfeld et al. [259] also asserted that random numbers are unnecessary for the initialization state in a paper with experiments of zeroing weights in a convolutional network.

7.3 Repeatable Determinism: Convolutional Networks

The previous dense layer work was limited to MLP networks and dense layers, and repeated research is for convolutional layers. In the imageNet challenge, convolutional layers reached prominence with better than human performance in Alex Krizhevsky's paper in 2021 [180]. Convolutional layers have more significant complexities as the weights are differently used and are affected by the image size and the filter dimensions. Convolutional layers operate in a sliding window operation, making a weight order significant to receptive fields. The dense layer work would not be directly applicable, and a new method would be required. The safety-critical challenges still exist in AI [240], [260], [242], [243] as applied to applications: ship classification for riverside monitoring [261], traffic accident detection from social media [262], driverless cars [263] and in research at least there is an appetite for safety-critical applications for aircraft taxiing [264] and situation awareness in autonomous ships [265].

7.3.1 Current Related Methods

In 2017, an alternative method to initialization, proposed by Seyfioğlu et al. [266], outperformed the random method but with two methods selected based on dataset size. Used in radar microDoppler where only small datasets are available, they applied transferred learning and unsupervised Convolutional AutoEncoder (CAE). Their findings were that both methods were superior to random methods with CAE on larger datasets (greater than 650 samples) and transferred learning on smaller datasets. Also, in 2017, Seuret et al. [267], in document analysis, outperformed the random initialization method, using Principal Component Analysis (PCA) parameters to initialize neural layers from an auto-encoder. These methods fit the initial state to the dataset and are thus coupled to the dataset, making them less of a general case.

Later in 2019, Zhang et al. [268] highlighted the area of initialization as an active research topic and proposed a modification to other initialization methods to limit values using FIXUP Initialization. The claim is that FIXUP allows 10,000 layers without normalization but with the proper regularization. Again in 2019, Humbird et al. [269] proposed a method that sampled a normal distribution in the bias values but still employed random numbers in the weights. This method is called Deep Jointly Informed Neural Network (DJINN) and uses decision trees searching for the "warm start" condition and a dataset in back-propagation. Ferreira et al. [270] used a De-noising Auto-Encoder (DAE) to classify tumour samples through dataset sampling, a data sample convergent method to weight initialization.

In 2020, Wang et al. in 2020 [271] proposed a convolutional networks initialization method, 2D Principle Component Analysis (2DPCA), adjusting the weight difference values to promote back-propagation. This method uses samples of the dataset as a convergent dataset method and avoids random numbers. In the area of fundus lesions images, Ding et al. [272] proposed a shuffle leapfrog algorithm method with random Gaussian forms in update and initialization. The method contains random numbers in an initially Gaussian distribution and then optimizes with the shuffle leapfrog algorithm.

Later again, in 2021, in neuroevolution, Lyu et al. assessed Xavier and Kaiming (also known as He) during mutation and crossover operations with two neuroevolution Lamarckian weight inheritance methods [273]. Lyu et al. find that Lamarckian weight inheritance is superior in crossover and mutation operations. Lyu et al. identify creation, offspring, and mutation as weight initialization points for use with their Evolutionary eXploration of Augmenting Memory Models (EXAMM) neuroevolution algorithm. This method generated network types: RNN: Δ-RNN, Gated Recurrent Units (GRU), LSTM, Minimal Gated Units (MGU), and Update Gate RNN (UGRNN).

7.3.2 Inspiration for this method

Part of the inspiration for this method was the connection to Hubel and Wiesel's work in brain anatomy [274], [275] and their experiments on cats and spider monkeys under light anaesthetic while stimulating the retina with images of spots and stripes. Generally accepted, there is a connection with spots and stripes forms in early layers of convolutional networks as part of hierarchical feature extraction [276], [277]. As such, spot and stripe forms also result after learning within image classification applications. The assertion thus follows that if an initial state is to be closer to the learnt

state generically, then it may have the ability to outperform the current random initialization state. Furthermore, if it is non-random in nature, it can also have a repeatable deterministic quality for safety-critical image classification applications.

7.3.3 The initial baseline model

The initial baseline dataset is the MNIST dataset [150] for direct comparison with the previous work in MLP networks. Torres's initial baseline model is a Convolutional network version [278], with a stated accuracy of ~97%. Another Convolutional network model for the MNIST dataset is Kassem [279] which has a higher stated accuracy of ~99%, but the high number of epochs (50) is a dominant effect of the random shuffle. Also, interest is primarily in the first epoch after initialization, and 99% does not provide much headroom to show an improvement. Convolutional networks used the weights in a more complicated inherited form see Figure 35, for the weights and image size effects in the test case with the layer types.

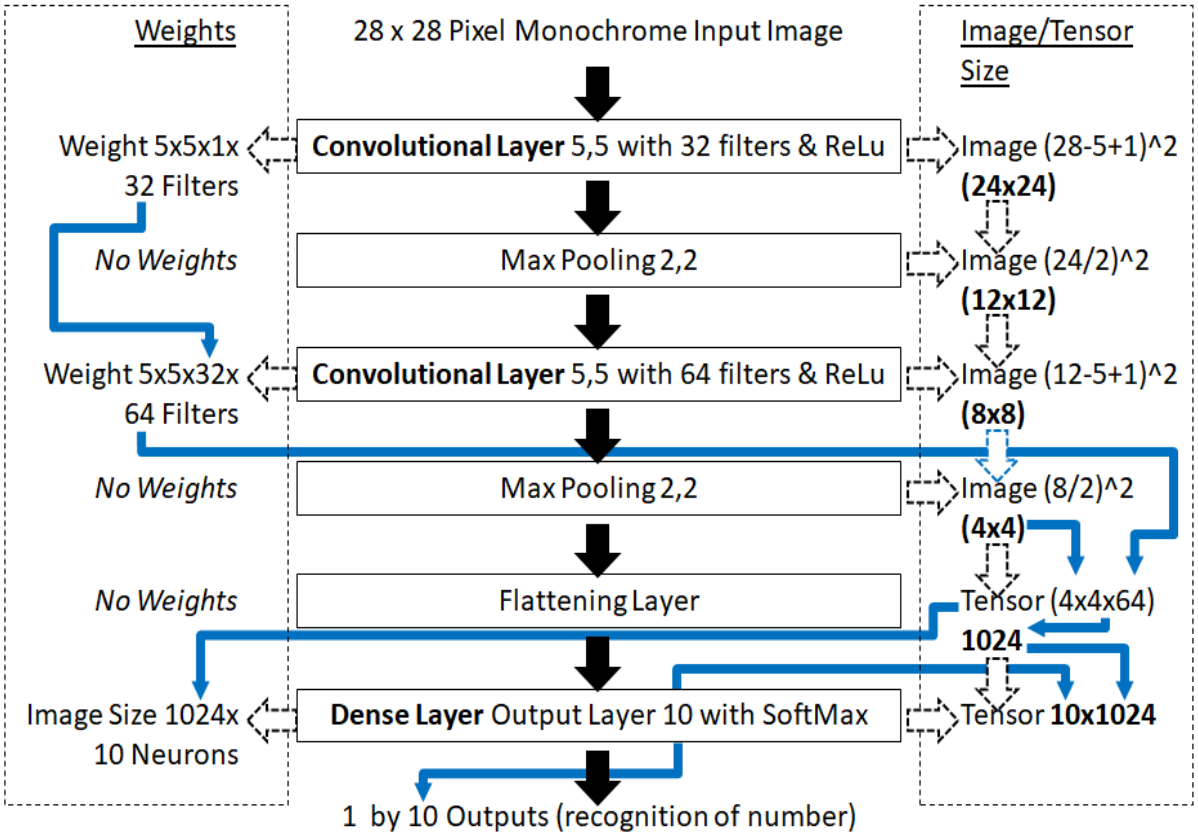


FIGURE 35 FILTER AND IMAGE SIZE TO EFFECT THE NUMBER OF WEIGHTS [J1]

The weight sizes using the layer inheritance in Figure 35 collate into Table 12, and three layers with weight values are associated. These are the two convolutional layers and the last dense layer in green.

TABLE 12
WEIGHTS AND IMAGE SIZE PARAMETERS: BY LAYER IN THE TORRES BENCHMARK MODEL.

Layer	Filter/Pool/Neurons	Depth	Image/Tensor Size		Weights
Input 28x28x1	N/A	1 (B/W image)	28x28	(748)	N/A
Conv Layer 1	5 by 5 by 32 filters	1	24x24	(576)	800
Max Pooling	2 by 2	32	12x12	(144)	N/A
Conv Layer 2	5 by 5 by 64 filters	32	8x8	(64)	51200
Max Pooling	2 by 2	64	4x4	(16)	N/A
Flatten Layer	N/A	1	1x(4x4x64)	1024	N/A
Dense Layer	10	1	10x1024	(10240)	10240

These three layers' weight values calculate the Glorot /Xavier limits as per the Torres baseline model as in Equations (71), (72), and (73). However, for later analysis of He et al. robustness, the He et al. calculated limits are in Equations (74), (75), and (76).

$$ConvLayer1 = \sqrt{\frac{6}{(5\cdot5\cdot1 + 5\cdot5\cdot1\cdot32)}} = 0.08528029, \quad \textit{Glorot/Xavier limit for Layer 2,} \tag{71}$$

$$ConvLayer2 = \sqrt{\frac{6}{(5\cdot5\cdot1\cdot32 + 5\cdot5\cdot64)}} = 0.05, \quad \textit{Glorot/Xavier limit for Layer 4,} \tag{72}$$

$$DenseLayer = \sqrt{\frac{6}{(4\cdot4\cdot64 + 10)}} = 0.07617551, \quad \textit{Glorot/Xavier limit for Layer 7,} \tag{73}$$

$$ConvLayer1 = \sqrt{\frac{6}{(5\cdot5\cdot1)}} = 0.48989795, \quad \textit{He et al. limit for Layer 2,} \tag{74}$$

$$ConvLayer2 = \sqrt{\frac{6}{(5\cdot5\cdot32)}} = 0.08660254, \quad \textit{He et al. limit for Layer 4, and} \tag{75}$$

$$DenseLayer = \sqrt{\frac{6}{(4\cdot4\cdot64)}} = 0.07654655, \quad \textit{He et al. limit for Layer 7,} \tag{76}$$

7.3.4 The Proposed Method

Utilized here with modification to allow odd number vector lengths, a least adjacent dataset non-random shuffle algorithm was previously published as part of this research [C3], [J3], and that algorithm had attractive properties to this application and provided spots, stripes, and curved patterns. It is also in tune with the Hubel and Wiesel intuitions [274], [275] as an inspiration for its use. When using the non-random shuffle, the algorithm outputs set *y* of reordering from an input set *x*. With zero indexed subscripts addressing in α and β, Equation (77) defines a recursive function for *nFilter* iterations *LayerNo* in Equation (78). Equations (79) and (80) define the subscript sets α and β for the reordering within each recursive shuffle iteration, where *n* is the standard subscript set size.

$$y = shuffle(x, nFilter_{LaterNo}), \tag{77}$$

$$shuffle(x_{\{\beta\}}, i) = \begin{cases} shuffle(x_{\{\alpha\},} i-1) & if\ i > 1 \\ x_{\{\beta\}} & otherwise \end{cases}, \tag{78}$$

$$\beta_{\{0..n-1\}} = \left\{ \beta \in N \left| \begin{array}{c} \left\{0 \le 2\beta \le 2\left(\left\lfloor \frac{n}{2} \right\rfloor - 1\right)\right\} \cup \\ \left\{1 \le (2\beta+1) \le \left(2\left(\left\lfloor \frac{n}{2} \right\rfloor - 1\right)+1\right)\right\} \cup \\ \left\{= \begin{cases} (n-1) & if\ n(mod\ 2) \ne 0 \\ \{\} & otherwise \end{cases}\right\} \end{array} \right. \right\}, \tag{79}$$

$$\alpha_{\{0..n-1\}} = \left\{ \alpha \in N \left| \begin{array}{c} \left\{0 \le \alpha \le \left\lfloor \frac{n}{2} \right\rfloor - 1\right\} \cup \\ \left\{n-1 \le n-1-\alpha \ge \left\lceil \frac{n}{2} \right\rceil\right\} \cup \\ \left\{= \begin{cases} \left\lfloor \frac{n}{2} \right\rfloor & if\ n(mod\ 2) \ne 0 \\ \{\} & otherwise \end{cases}\right\} \end{array} \right. \right\}. \tag{80}$$

The length of the tensor is in Equation (81). Each value within this subscript rearrangement is based on a layer type as in Equation (82), where *t* is the layer type and *l* is the limit value calculated by Glorot/Xavier or He et al. initialization limits. The layer type is required because a finding will be that linear ramps work best with dense layers and sinusoidal slopes with convolutional layers. Values of *m* and *cnt* are calculated from filters or image size, depending on the layer type in Equations (83), (84), (85), and (86). The set limits are in Equations: (87), (88), (89), (90), (91), and (92). See Equations (93) and (94) for the indexing order in the layer types.

$$InitTensor_{Length} = m \cdot maxFilters \ , \tag{81}$$

$$valSet(cnt, m, l, t) = \begin{cases} cos\left(\frac{cnt}{m-1}\pi\right) l & if\ t = Convolutional \\ \frac{cnt}{m-1} 2l - l & if\ t = Dense \end{cases} , \tag{82}$$

$$m = Height_{Filter} \cdot Width_{Filter} \cdot Depth_{Filter} \quad if\ t = Convolutional \ , \tag{83}$$

$$m = Height_{Image} \cdot Width_{Image} \cdot Depth_{image} \quad if\ t = Dense \ , \tag{84}$$

$$cnt = \{0..m-1\}, index\ as: cnt_{(Width_{Filter}, Height_{Filter}, Depth_{Filter})} \quad if\ t = Convolutional, \tag{85}$$

$$cnt = \{0..m-1\}, index\ as: cnt_{(Width_{image}, Height_{image}, Depth_{image},)} \quad if\ t = Dense \ , \tag{86}$$

$$nFilter = \{nFilter \in N \mid 0 \leq nFilter < maxFilters\} \quad if\ t = Convolutional \ , \tag{87}$$

$$nNeurons = \{nNeurons \in N \mid 0 \leq nNeurons < maxNeurons\} \quad if\ t = Dense \ , \tag{88}$$

$$nDepth = \{nDepth \in \mathbb{N} \mid 0 \leq nDepth < maxDepth\} \ , \tag{89}$$

$$nHeight = \{nHeight \in N \mid 0 \leq nHeight < maxHeight\} \ , \tag{90}$$

$$nWidth = \{nWidth \in N \mid 0 \leq nWidth < maxWidth\} \ , \tag{91}$$

$$nSet = \{nSet \in \mathbb{N} \mid 0 \leq cnt < m-1\} \ , \tag{92}$$

$$InitTensor = {}_{(nHeight, nWidth, nDepth, nFilter)} \quad if\ t = Convolutional \ , \tag{93}$$

$$InitTensor_{(Activations, Neurons)} = {}_{((nHeight, nWidth, nDepth), nNeurons)} \quad if\ t = Dense \ . \tag{94}$$

As a disruption to the zero index location in both sets, the subscript location is in reverse order in every second iteration see Equation (96), doubling the number of filters on offer and disrupting the first value index. Equation (97) applies the iterative shuffle on each pair. For convenience, this is a transposed matrix see Equations (95) and (98).

$$TsprMat_{(nFilter,\ nDepth,\ nWidth,\ nHeight,)} = T_{(initvalues_{nHeight,\ nWidth,\ nDepth,\ nFilter})} \ , \tag{95}$$

$$flipMat = \begin{cases} TsprMat_{(nFilter,(m-1)..0)} & (nFilter + 1)\ (mod\ 2) = 0 \\ TsprMat_{(nFilter,0..(m-1))} & otherwise \end{cases} , \tag{96}$$

$$shuffleMat(nFilter) = shuffle(flipMat_{(nFilter,0..(m-1))}, \left\lfloor \frac{nFilter}{2} \right\rfloor) \ , \tag{97}$$

$$TsprMat2[nFilter] \ = \ shuffelMat_{(Depth,Width,Height)} \ . \tag{98}$$

For illustration, using the *cnt* values rather than the *valSet* function response as the c*nt* values are the indexes, then in the case of filters = 5, channel depth = 4, width = 3 height = 2, that example is in Equations (99), (100), and (101) in each stage:

$$set_{of_{cnt}} = \left\{ \begin{Bmatrix} \{0,0,0,0,0\}, \\ \{6,6,6,6,6\}, \\ \{12,12,12,12,12\}, \\ \{18,18,18,18,18\} \end{Bmatrix}, \begin{Bmatrix} \{1,1,1,1,1\}, \\ \{7,7,7,7,7\}, \\ \{13,13,13,13,13\}, \\ \{19,19,19,19,19\} \end{Bmatrix}, \begin{Bmatrix} \{2,2,2,2,2\}, \\ \{8,8,8,8,8\}, \\ \{14,14,14,14,14\}, \\ \{20,20,20,20,20\} \end{Bmatrix} \right\} \left\{ \begin{Bmatrix} \{3,3,3,3,3\}, \\ \{9,9,9,9,9\}, \\ \{15,15,15,15,15\}, \\ \{21,21,21,21,21\} \end{Bmatrix}, \begin{Bmatrix} \{4,4,4,4,4\}, \\ \{10,10,10,10,10\}, \\ \{16,16,16,16,16\}, \\ \{22,22,22,22,22\} \end{Bmatrix}, \begin{Bmatrix} \{5,5,5,5,5\}, \\ \{11,11,11,11,11\}, \\ \{17,17,17,17,17\}, \\ \{23,23,23,23,23\} \end{Bmatrix} \right\} , \tag{99}$$

$$set_{of_{cnt}} = \left\{ \begin{Bmatrix} \{0,23,0,23,0\}, \\ \{6,17,6,17,6\}, \\ \{12,11,12,11,12\}, \\ \{18,5,18,5,18\} \end{Bmatrix}, \begin{Bmatrix} \{1,22,1,22,1\}, \\ \{7,16,7,16,7\}, \\ \{13,10,13,10,13\}, \\ \{19,4,19,4,19\} \end{Bmatrix}, \begin{Bmatrix} \{2,21,2,21,2\}, \\ \{8,15,8,15,8\}, \\ \{14,9,14,9,14\}, \\ \{20,3,20,3,20\} \end{Bmatrix} \right\} \left\{ \begin{Bmatrix} \{3,20,3,20,3\}, \\ \{9,14,9,14,9\}, \\ \{15,8,15,8,15\}, \\ \{21,2,21,2,21\} \end{Bmatrix}, \begin{Bmatrix} \{4,19,4,19,4\}, \\ \{10,13,10,13,10\}, \\ \{16,7,16,7,16\}, \\ \{22,1,22,1,22\} \end{Bmatrix}, \begin{Bmatrix} \{5,18,5,18,5\}, \\ \{11,12,11,12,11\}, \\ \{17,6,17,6,17\}, \\ \{23,0,23,0,23\} \end{Bmatrix} \right\}, \tag{100}$$

$$set_{of_{cnt}} = \left\{ \begin{Bmatrix} \{0,23,0,23,0\}, \\ \{6,17,4,19,20\}, \\ \{12,11,6,17,4\}, \\ \{18,5,10,13,21\} \end{Bmatrix}, \begin{Bmatrix} \{1,22,3,20,23\}, \\ \{7,16,2,21,1\}, \\ \{13,10,9,14,19\}, \\ \{19,4,8,15,5\} \end{Bmatrix}, \begin{Bmatrix} \{2,21,1,22,3\}, \\ \{8,15,5,18,22\}, \\ \{14,9,7,16,2\}, \\ \{20,3,11,12,18\} \end{Bmatrix} \right\} \left\{ \begin{Bmatrix} \{3,20,23,0,12\}, \\ \{9,14,19,4,8\}, \\ \{15,8,17,6,16\}, \\ \{21,2,13,10,9\} \end{Bmatrix}, \begin{Bmatrix} \{4,19,20,3,11\}, \\ \{10,13,21,2,13\}, \\ \{16,7,14,9,7\}, \\ \{22,1,22,1,22\} \end{Bmatrix}, \begin{Bmatrix} \{5,18,5,18,5\}, \\ \{11,12,11,12,11\}, \\ \{17,6,17,6,17\}, \\ \{23,0,23,0,23\} \end{Bmatrix} \right\}. \tag{101}$$

To illustrate the algorithm steps for clarity, Equation (99), Equation (100) and Equation (101), show the main three algorithm reordering components included as piecemeal steps. Equation (99) is the indexing without the shuffle or the vector reversing alternation. Equation (100) is the indexing reordering with the vector reversing alternation but without the shuffle, and Equation (101) is the indexing reordering with both the shuffle and vector reversing alternation.

The maximum number of filters on offer is in Equation (102), but in practice, in some filter geometries, the number of filter permutations can repeat early in aliasing, which is a further research subject.

$$n_{filter} = 2 \cdot (Height_{filter} \cdot Width_{filter} \cdot Depth_{filter} - 1) \tag{102}$$

As each layer's receptive field mapping depends on the last layer, see Figure 45, and with dense layers, the algorithm varies slightly depending on the layer prior.

7.3.4.1 Comparison with the Random Scheme

The research work re-used a variation of the non-random shuffle [C3], [J3], as it had properties of rotating lines and creating stripes that would be consistent with the expectations of feature extraction [280], [276], and is also consistent with observations from Hubal and Wiesel with cats and spider monkey experimentations [274], [275]. These results would exceed the baseline random initialization target using a non-random numbers set.

Figure 36 shows the generated filter initialization weights in a convolutional network before and after learning as used in that work.

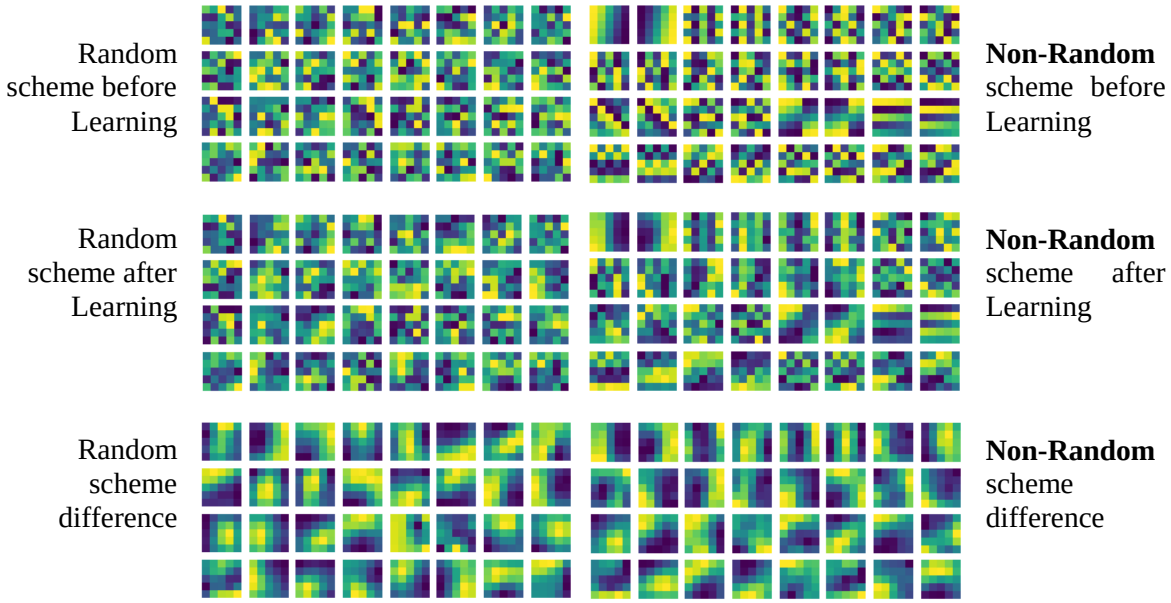


FIGURE 36 CONVOLUTIONAL LAYER FILTER INITIALISATION NUMBER SET SEQUENCES [J1]

When the non-random method runs against the benchmark model, with the model in successive numbers of 1 to 5 epochs, the results are in Table 13 from cross-validation. The most significant improvement is in the first epoch, which might be expected with a better initialization method, being the epoch after initialization. As the more advanced limit calculation method, Table 14 shows further improvements in the interests of robustness to He et al. initialization. Table 14 also compares to the previous Table 13 Glorot/Xavier results and compares with the random initialization using He et al. limits. In all cases, the non-random replacement for the random numbers added benefit.

TABLE 13
NON-RANDOM WEIGHT (GLOROT/XAVIER LIMIT) RESULTS IN CONVOLUTIONAL NETWORK [J1]

Epochs	Accuracy *(Cross-val.)*	Loss *(Cross-val.)*	Gains over existing (random) method
5 Shuffled	**97.5%**	**0.085728347**	**+0.599%** *(Cross-validation gain)*
4 Shuffled	97.11%	0.097854339	N/A
3 Shuffled	96.85%	0.114757389	N/A
2 Shuffled	95.96%	0.141269892	N/A
1 Shuffled	**93.77%**	**0.230065033**	**+2.642%** *(Cross-validation gain)*
1 No Shuffle	**93.28%**	**0.230725348**	**+3.705%** *(Cross-validation gain)*

TABLE 14
NON-RANDOM WEIGHT (HE ET AL. LIMIT) RESULTS IN CONVOLUTIONAL NETWORK [J1]

Epochs	**He et al. (Non-Rnd) measure**		**He with proposed method gains over:**	
	Accuracy	**Loss**	**Glorot (Non-Rand)** [Table 13]	**He (Rnd)**
5 Shuffled	97.55%	0.082669578	+0.05%	+0.7%
4 Shuffled	97.19%	0.093996972	+0.08%	+0.91%
3 Shuffled	96.97%	0.10997723	+0.12%	+1.49%
2 Shuffled	96.15%	0.134461805	+0.19%	+1.83%
1 Shuffled	**94.11%**	**0.214723364**	**+0.34%**	**+5.13%**
1 No Shuffle	**93.57%**	**0.217569217**	**+0.29%**	**+4.27%**

The non-random method is higher-performing with convolutional networks with repeatable determinism benefits for mission and safety-critical applications [240], [260], [242], and [243]. That work used Glorot/Xavier limits, but the work is robust with He initialization [254]. The non-random

scheme would also induce earlier learning, and in the first epoch, which is the epoch after initialization, the losses in learning reduce quicker in Figure 37 (right).

7.3.4.2 Invocation of Earlier Learning

The reasoning for this benefit would be that stripes and curved forms are more allied to image classification. Figure 37 showed the loss in learning with the shuffled dataset compared to the random scheme left and the non-random scheme right, both using the Glorot/Xavier limits. Figure 38, again using the Glorot/Xavier limits, showed the loss in learning when a shuffle is used, with the random scheme left and the non-random scheme right. The non-random scheme's loss has lowered earlier in the learning, at the 100 batch mark. Thus, the non-random scheme uses more of the dataset more effectively and is robust to shuffling.

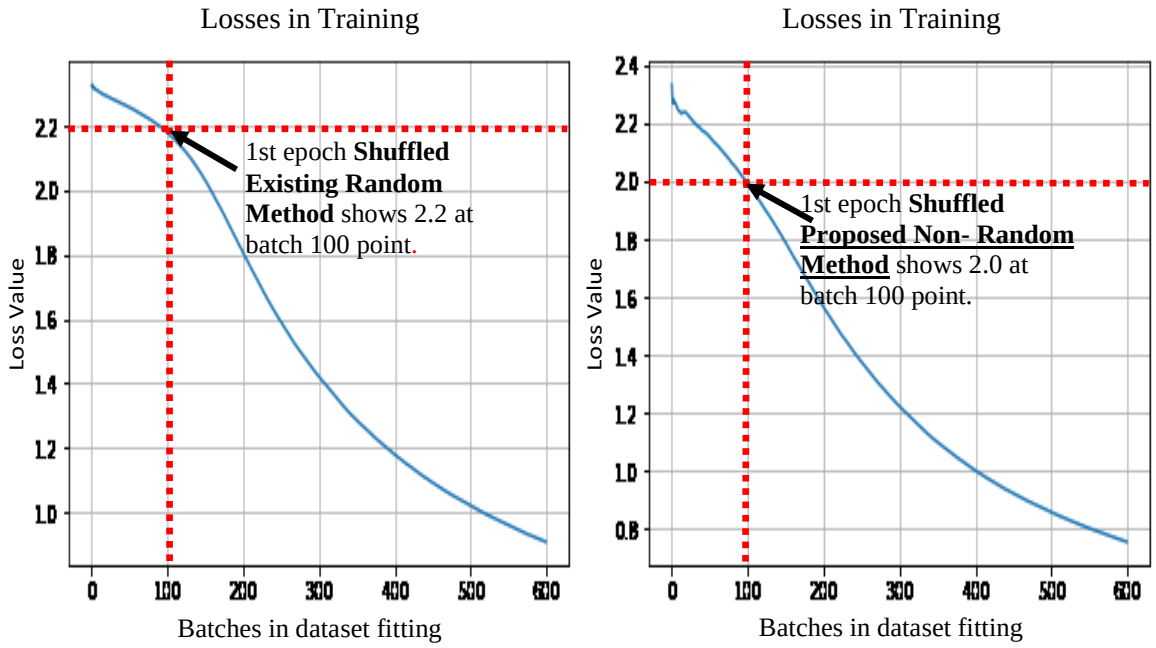


FIGURE 37 LOSSES 1ST EPOCH: RANDOM (LEFT) AND NON-RANDOM (RIGHT) WHEN SHUFFLED [J1]

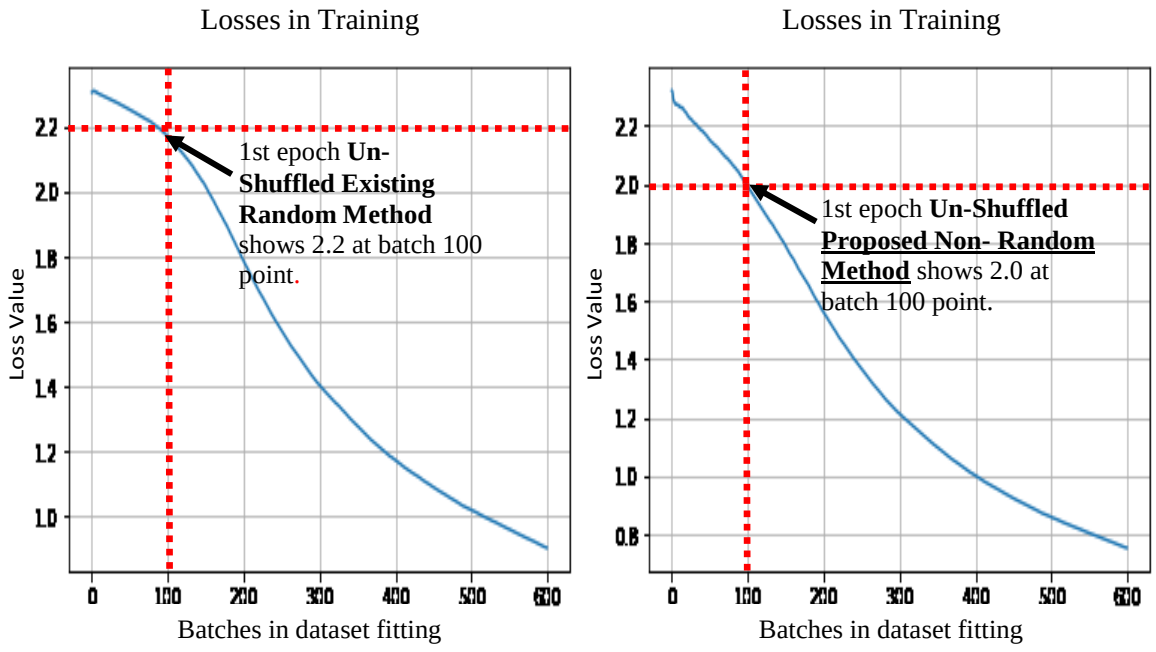


FIGURE 38 LOSSES 1ST EPOCH: RANDOM (LEFT) AND NON-RANDOM (RIGHT) WHEN UN-SHUFFLED [J1]

The losses in Figure 37 (right) and Figure 38 (right) show that the loss has reduced quicker with the non-random form, demonstrating that in the period after initialization that there is less unlearning of the initialization state, where the initial state's influence is more decisive than later learning. Subsequently, the loss is still lower but by less margin, as the effect of the initialization state's influence is less in subsequent learning.

Partial results were published in a peer-reviewed conference paper [C1[9]] and a peer-reviewed journal version [J1[10]] with Transferred Learning and FSGM Adversarial Attack as an analytical method.

7.3.5 Transferred Learning and FSGM

This work with convolutional networks was extended, using convolutional and dense layers. Using the Fast Sign Gradient Method (FSGM) with transferred learning introduces a controlled distortion causing dissimilarity between the transferred learning and the subsequent new learning. The FSGM approach is convenient for providing a controlled distortion for progressive dissimilarity.

The FSGM approach was proposed by Ian Goodfellow [281], [282] as an adversarial attack to cause a miss-classification and usefully has a strength to that attack in an Epsilon (ε) value (See Equation (103)).

$$x' = x + \varepsilon \cdot sign\left(\nabla_x J(\theta, x, y)\right) \quad . \tag{103}$$

The FSGM adversarial attack can cause distinct scaling of perturbing images. Equation (103) is modified from its usual form to Equation (104) and includes image clipping of the perturb images to be fairer on the perturb images concerning the actual images; this would be as if both actual and perturb images were value clipped in pre-processing.

$$x' = max\big(min\big(x + \varepsilon \cdot sign\left(\nabla_x J(\theta, x, y)\right), 1.0\big), 0.0\big) \quad . \tag{104}$$

7.3.5.1 The Analytical Method using FSGM in Transferred Learning

The Torres [278] model combines with the Theiler method [283] to model defence from the FSGM perturbation adversarial attack. The two model architectures combine as an architecture framework for transferred learning assessment, with test points 1, 2, and 3 shown in Figure 39.

Theiler's model architecture adapts to the Torres model by modifying the number of epochs in the later learning model defence by considering the Torres model number of epochs instead. The ratio of back-propagation is similar between the transferred learning model and the adaption learning, respecting Theiler's method. The Torres model also uses higher-performing He et al. initialization limit values.

[9] [Cn] Published conference papers are in a separate bibliography on page xv.

[10] [Jn] Published journal articles are in a separate bibliography on page xv.

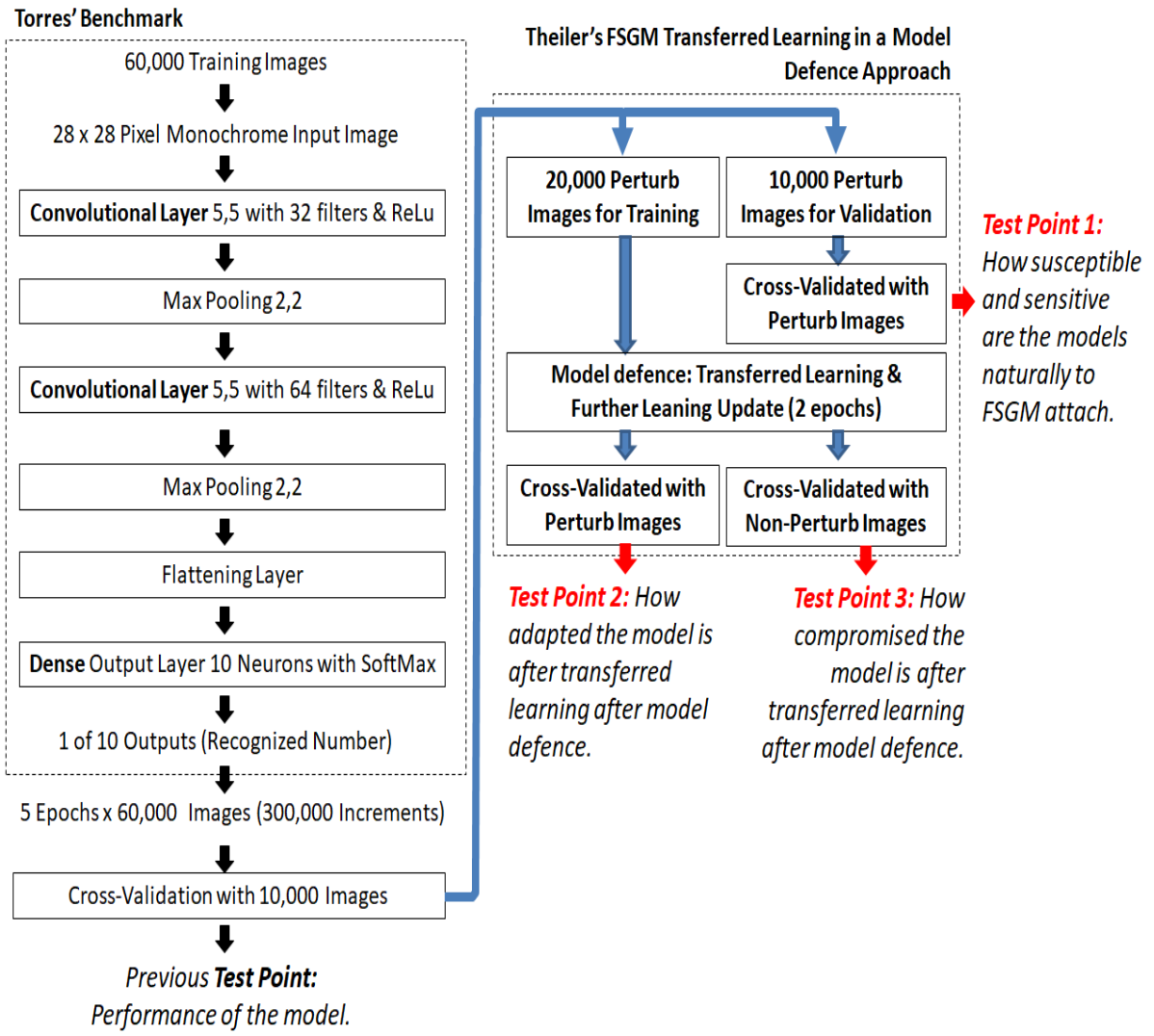


FIGURE 39 EXPERIMENT MODEL FOR AN ANALYTICAL METHOD WITH FSGM [J1]

Test point 1 is cross-validated with the perturbed images before the model defence, and this is how successful the FSGM attack is on the model before the defence.

Test point 2 is after the transferred learning model defence and cross-validated with the perturb cross-validation dataset and shows how the model adapts to the new dataset.

Test Point 3 is also after transferred learning model defence but with the original non-perturbed cross-validation dataset.

This model architecture thus forms the experiment model. The FSGM epsilon (ε) value is varied in increments on each experiment as the test variation, and those experiments are in two configurations. Configuration one is the random initialization, and the other is the non-random initialization. A research question in this experiment is: If a non-random initialization invoked earlier learning, will the total learning be more of the discriminatory content rather than the noise in the image, and will that aid transferred learning to transfer more helpful content. Also, a hypothesis is that the non-random scheme may have removed an unintentional random noise source in the learning. When random numbers are in the weight initialization, they combine with the noise in the dataset and re-colour with the noise when the activation and weight multiply within the dot-product of a convolutional filter. It also can be noted that Schwinn et al. [284] proposed noise injection as a method for defence from FSGM attacks. However, Schwinn et al. require a 'dataset coupling' learning regularization step, which is not the approach in this research section of the dissertation.

7.3.6 Perturbation Datasets

To verify that the generated perturbation dataset is correct, Figure 40 shows the first 20 images of the perturbation datasets, with the left using the random initialization of the model and the right using the non-random initialization.

FIGURE 40 PERTURBED IMAGES: LEFT RANDOM, RIGHT NON-RANDOM INITIALISATION [J1]

In Figure 40, the epsilon value increases from 0.0 to 1.0 from the upper to lowest row in steps of 0.5. The images coloured in green are the correctly classified images with their original tag, and red is the images that did not classify to their original tag.

As such, with zero perturbation influence ($\varepsilon=0$), all images in the row are classified correctly, and in lower rows, with an increasing value of (ε), the images are less well classified, with less human discernible discrimination in the images as well.

7.3.6.1 Test Point 1 How Susceptible is the Method to Dissimilarity

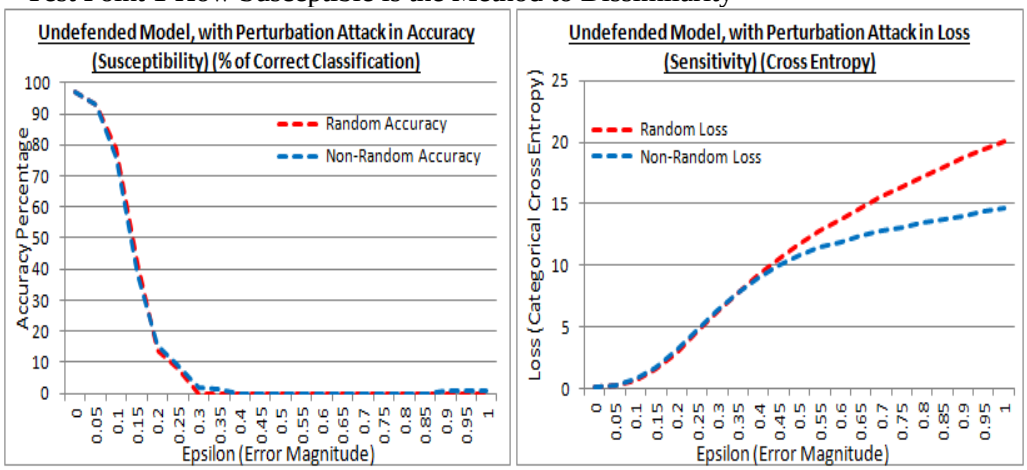


FIGURE 41 TEST POINT 1: ACCURACY AND LOSS WITHOUT TRANSFERRED LEARNING [J1]

7.3.6.2 Test Point 2 How Adapted the Model is after Transferred Learning

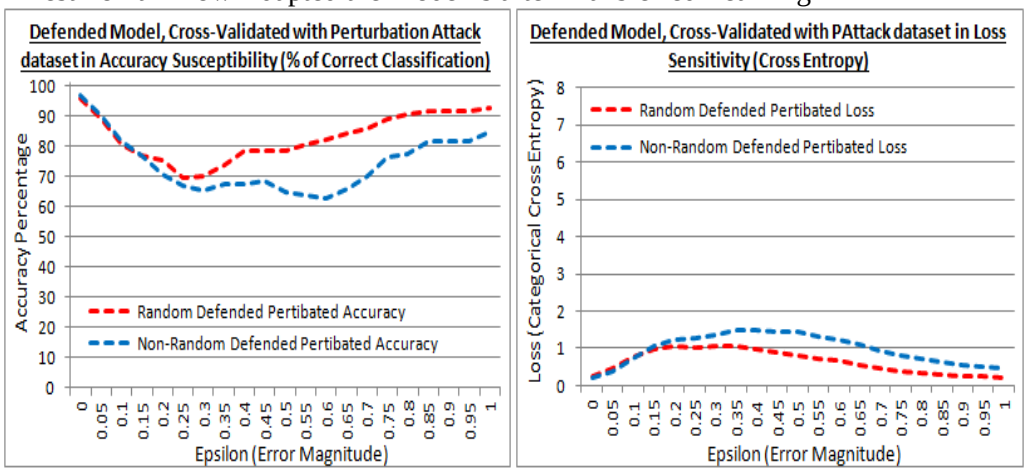


FIGURE 42 TEST POINT 2: ACCURACY AND LOSS WITH TRANSFERRED LEARNING [J1]

7.3.6.3 Test Point 3 How Compromised is the Model after Transferred Learning

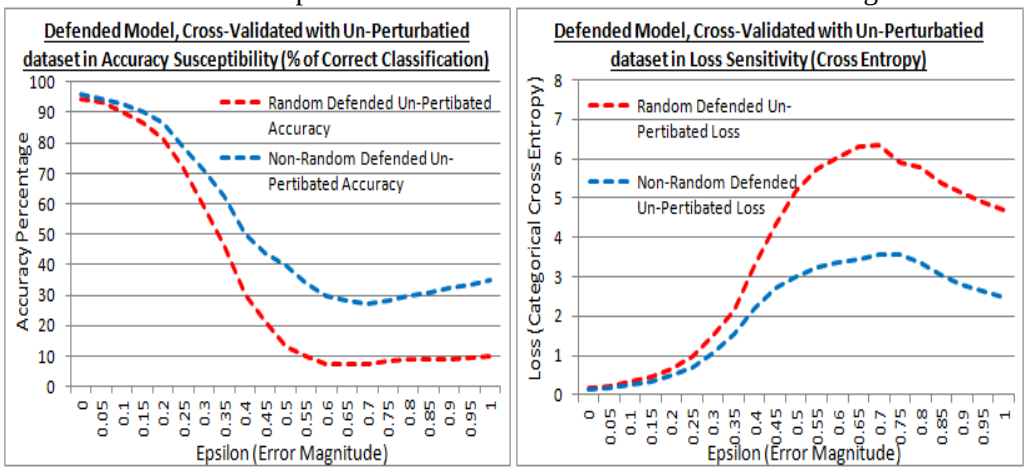


FIGURE 43 TEST POINT 3: ACCURACY AND LOSS WITH TRANSFERRED LEARNING [J1]

7.3.7 Transferred Learning Findings

In these experiments, the only difference is using the model's non-random initialization state with the transferred learning method. Test point 1 cross-validated the model with a perturbation attack dataset. When applying the FSGM method with increments of the Epsilon value, the accuracy falls off in a similar slope in both the non-random and random schemes, as shown in Figure 41. Figure 41

shows that both initialization methods are affected by the FSGM attacks, although the non-random scheme has a lower loss at higher epsilon values. Test point 2 cross-validated the model with a perturbation attack dataset after transferred learning. With larger epsilon value increments, the accuracy decreases with a higher loss in the non-random method, as shown in Figure 42. The transferred learning has less adapted in the non-random form. Test point 3 cross-validated the model with the original cross-validation dataset after transferred learning. With larger epsilon value increments, the accuracy is more significant with a lower loss in the non-random method, as shown in Figure 43. The non-random form has retained more of the earlier learning in the transferred learning.

7.3.7.1 A Random Epsilon Value Dataset

After assessing both initialization schemes with uncontrolled epsilon values, i.e., random epsilon values, the findings are the same, as shown in Table 15.

TABLE 15
RANDOM EPSILON DATASET RESULTS [J1]

Initialization method used prior to model defense and transferred learning.	Non-Attack cross-validation dataset		Attack cross-validation dataset	
	Loss *(Cross Val)*	Accuracy *(Cross Val)*	Loss *(Cross Val)*	Accuracy *(Cross Val)*
Proposed (Non-Random) Method	**0.9854**	**67.01%**	1.3331	61.82%
Existing (Random) Method	1.2736	61.05%	**0.8366**	**78.41%**

The proposed non-random scheme still has higher accuracy and lower loss when cross-validated with the original cross-validation dataset. Thus it shows more significant retention of the original learning after transferred learning, whereas the random initialization scheme had a more noticeable adaption to the attack dataset at the expense of the original learning. When the controlled epsilon value steps are compared, the effect is more substantial with higher *epsilon* values. The intuition is that the earlier learning in the non-random form is more fitted and has used the dataset more effectively in the original data learning.

7.3.8 Colour Images and a Dissimilar Model Architecture

Convolutional networks in image classification operate on colour images with more than one colour channel. Aligned with the application area, the MTARSI dataset [55] of different aircraft on runways is a representative challenge. This dataset is quite challenging as it has varying light, aspect angle, backgrounds, and image resolutions in an imbalanced dataset. The MTARSI dataset was developed further with data augmentation and classification into 42 categories as MTARSI2 [53] and was made available to other researchers. For image examples from MTARSI2, see Figure 44.

FIGURE 44 MTARSI2 DATASET EXAMPLES [53]

The model used for the MTARSI2 dataset is in Figure 45 and features three convolutional layers and two dense layers. As such, the receptive field of the layer is different depending on the layer prior, and Figure 45 also includes three cases where the algorithm differs.

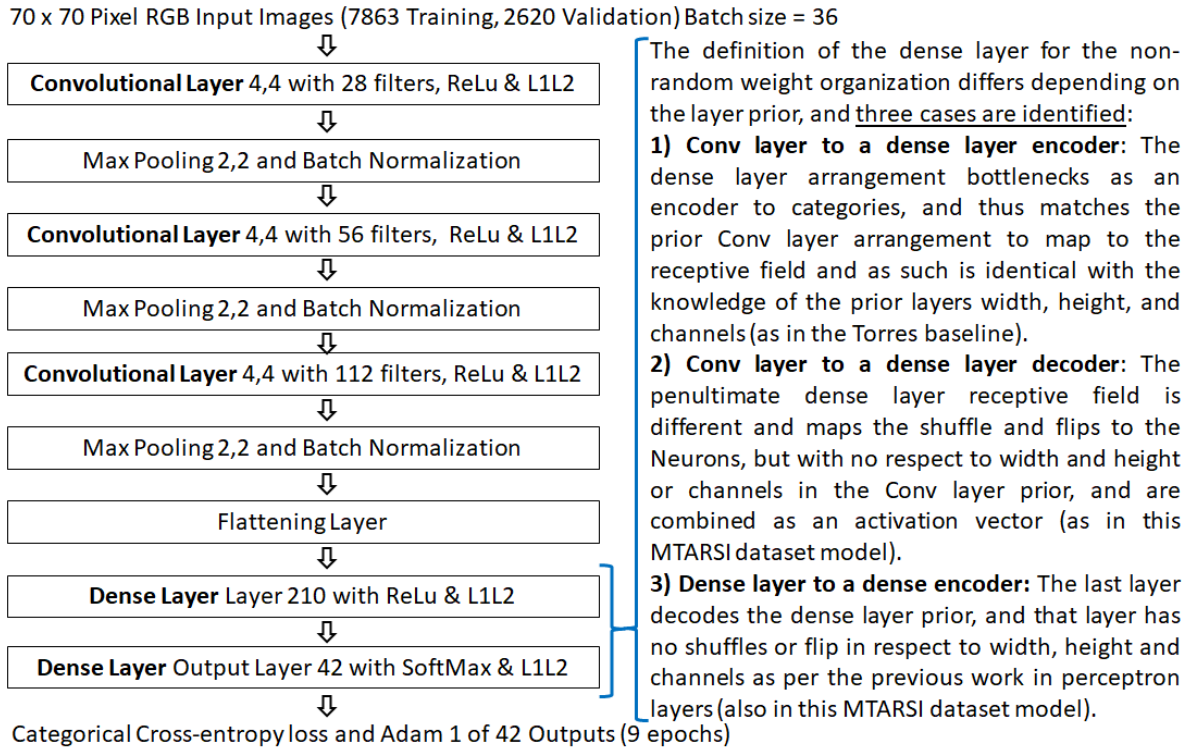


FIGURE 45 COLOUR IMAGE MODEL IN DISSIMILAR ARCHITECTURE AND MTARSI2 DATASET

After nine epochs, the model accuracy fluctuates in cross-validation accuracy and thus is testing an over-fitting recovery from that state rather than the after initialization state, so nine epochs are the limit. The L1L2 regularisation (where L1 is Lasso and L2 is Ridge regression, and L1L2 is Elastic Net regression) was selected based on the image discriminators not having pre-determined importance as there is an unknown classification and two general classes in the dataset.

Table 16 contains the cross-validation-dataset results from the MTARSI2 dataset and the model architecture in Figure 45 and shows a benefit with the non-random method shown in green. The first epoch accuracy difference is in bold increased from 29.31% with the traditional random method to 67.2% with the proposed non-random method.

TABLE 16
COLOUR IMAGE MTARSI2 DATASET RESULTS

Model Results at Test States	Accuracy in the first epoch of learning and also a single epoch in transferred learning)	Accuracy after nine epochs, and also four epochs in transferred learning.
Non-Radom Initialisation Cross-Validation on the original image dataset	**67.2%**	86.03%
Random Initialisation Cross-Validation on the original image dataset	**29.31%**	84.81%

7.3.9 Summary of Convolutional Networks

The non-random initialization scheme promotes the palatability of neural networks in roles closer to the mission and safety-critical applications. The scheme achieves repeatable determinism, as the quality of dependable systems in testing and deployment. The non-random initialization replaces the random numbers and complements both Glorot/Xavier and He at al. initializations. The inspiration

for the non-random form was from Hubel and Wiesel's work [274], [275] in brain anatomy and used striped, spotted, and curved forms that are generally accepted to be relevant to hierarchical feature extraction. The non-random scheme invoked earlier learning by being more allied to image categorization from the outset of learning.

Further interest in the non-random convolutional initialization is that the start condition, having more stripes, is less resolution-specific than the spotted forms of the random form. As such, it might be that earlier learning is more able to get underway irrespective of the kernel geometries. Also, that might impact model architectures originating from a multi-resolution motivation like Inception and ResNet.

7.4 Summary of Safety-Critical AI

A non-random scheme can be equivalent in MLP networks but with the advantage of structured weights. However, in convolutional networks, it is superior. Repeatable determinism is a quality of a dependable system for Safety-Critical AI, and as such, ML is closer to palatability. The convolutional networks, the non-random method, invoked earlier learning and had advantages in transferred learning, retaining 22% more accuracy (as 31% - 9% = 22%) and also with earlier learning. However, more work is required for different applications and extending the number of available filters. The non-random method offered better accuracy and was further assessed in transferred learning with the FSGM adversarial attack to provide a controlled dissimilarity between two datasets. The findings were that the non-random scheme retained more original learning ~31% instead of ~9%, particularly with higher epsilon values (i.e., with higher distortions, as a more significant dissimilarity between the datasets. The non-random scheme can be transferable to new learning while retaining earlier learning. If the MNIST dataset [150] included a hexadecimal category, then the new initialization scheme would remember more of the numbers of the original learning than the existing initialization scheme, rather than just being a warm start condition. The method was also applied to colour images in a different model architecture and with a more challenging dataset than the original analysis (MTARSI2), and also found an advantage of accuracy with the non-random initialization scheme accuracy increasing from 29.31% to 67.2% in the first epoch.

Chapter 8

SYNTHETIC EMITTER DATASET GENERATION AND EMITTER IDENTIFICATION

Firmly within the application area, this research proposed an alternative method to emitter identification using image classification. There are few or no publicly accessible datasets; this research also proposed and implemented a synthetic emitter dataset generator, integrated into an AI framework and supported by a mark-up language for emitter behavioural descriptions; see Figure 46 for that proposed architecture. The method presented here is for emitter identification with image classification and would be after a previous de-interleaving process, with generated images generated from this method as a training dataset source.

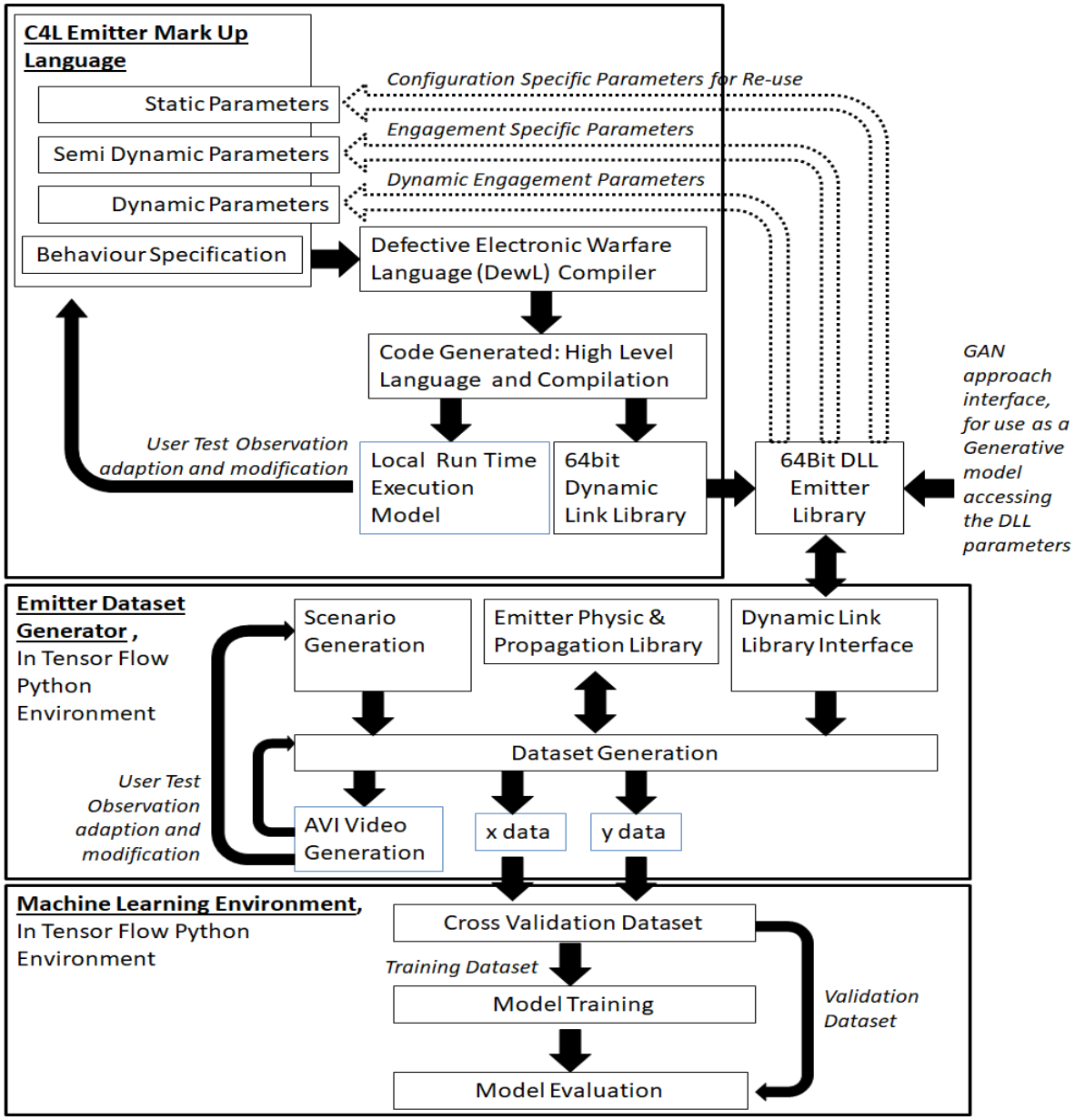


FIGURE 46 SYNTHETIC EMITTER DATASET GENERATOR ARCHITECTURE

8.1 Architecture for Synthetic Data Generation for Emitter Classification

This architecture could also be a generative model of a GAN to perfect emitter behaviours to similar fidelities or create other datasets for discriminators in their adversarial generalization. The Synthetic Dataset Generator has complexity and modularity. Figure 46 shows the architecture in three significant components to understand the design and is also described in the following sections.

8.1.1 Architectural Component: C4L Emitter Mark-Up Language

The architecture for the Dataset Generator is data-driven by the emitter mark-up language through a C3L upgrade to C4L made as part of this research. The C3L baseline allows concurrent activities of elements scheduled without the user requiring any experience of process synchronization or message passing paradigms. It has constructs to allow specification elements to schedule: concurrently, sequentially, selectively, or iteratively. When updated to C4L within this research, the language allows the system states of parameter modulations as a specification for a radar emitter and receiver model. The language allows parameter adaption at runtime for modelling reactive behaviours and can use simultaneous or state-full sequences with asynchronous updates that are code generated into a single process thread for data integrity for IA. Thus, the scheduled elements added to C4L allow emitters or radars to describe behaviour specifications that also separate the modulations for transmitting and receiving emissions and scans. C4L cross-compiles to code-generate in a 3rd generation programming language called c, and that c code is highly portable and is used to generate a 64bit DLL, with an Application Programming Interface (API) as a standard interface for other modelling systems. That interface then links to an Emitter Dataset Generator within the python and TensorFlow AI framework permitting neural network development of methods.

8.1.2 Architectural Component: Emitter Dataset Generator

The dataset generator uses a scenario linked to the radar emitter and receiver states and modes model within the DLL; that then applies to a python physics library for free-space propagation, also developed in this research. The physics library applies to transmitting and receiving beams, scans, Pulse, and PRI modulations with propagation losses and noise for high bandwidth I/Q data. That high bandwidth I/Q data is used to generate still colour images in a dataset format for image classification. A video generated from those images is for the user's review in developing the C4L specification. The dataset synthesizes y values as the categories; these are the radar and operation identities tags. The x dataset is the images of those radars and modes of operation.

8.1.3 Architectural Component: Machine Learning Environment

The machine learning environment thus has a dataset of images and categories that can be split into separate training and validation datasets and apply the python and TensorFlow neural network framework to that dataset.

8.2 Related Work in Synthetic Datasets

Many ML methods require available datasets. In new developments, this is not always the case, and synthetic datasets can be required to prove the viability of a technology. Synthetic datasets are available from graphic engines [153], impact sound models [155], and Natural Language Generation (NLG) [156]. Synthetic datasets [157] can also provide the incorporation of low probability test cases towards proving a Safety-Critical outcome. There is work related to radar emitter analysis for a machine learning method [158]. A language method by Hoag et al. [285] used XML and features in a survey of general techniques [286]. However, this paper presents a programming language with a BNF and contains both Data and Operations. In this application, complexities exist in the physics, the emitter sequencing, concurrency, and control logic. The method of Greig et al. provides high fidelity I/Q data for air and surface targets [287], but the presented method in this paper's contribution is for ELINT analysis through an emitter behaviour language, although it does extend to radar data generation too. Another method is the Generative Adversarial Network (GAN) [159], but although it ideally requires existing real data for the discriminator, a synthetic dataset generator such as proposed in this chapter could be a generator in a GAN method too.

8.3 Traditional Methods to Emitter Identification

Emitter identification begins with recognizing an emitter type in the operating environment of radar pulses overlapped and interleaved in time. The approach separates the pulse chains using de-

interleaving methods [288], [289]. Processing streams of Pulse Descriptor Words (PDWs) captured from a Radar Warning Receiver (RWR), Electronic Support Measures (ESM), or ELectronic INTelligence (ELINT) equipment sensor that records: Time of Arrival (TOA), Pulse Width (PW), and Amplitudes from the receiving antennas to generator PRI pattern that matches with templates and the received signals for recognition. These methods may use more than one antenna amplitude to measure a Direction of Arrival (DOA). That DOA can use methods such as Multi Signal Classification (MUSIC) [290], Angle of Arrival (DOA), interferometry or Time Difference of Arrival (TDOA), and Frequency Difference of Arrival (FDOA), where TDOA and FDOA use the multilateration (MLAT) method for geo-location and other use kinematic ranging or triangulation. As such, a signal's template match identification can plot on an angle, where that signal's identification may match to a template of an emitter in a particular emitter mode and provides a warning of lethality, allegiance, and threat level of an emitter as a display distance, as a Situation Awareness (SA) of that emitter. Where ESM types of equipment provide more accurate estimates than RWR equipment, some more advanced methods in ELINT equipment can identify a particular emitter in a Specific Emitter Identification (SEI) process. These types of equipment emphasize mission data and the templates for matching with threats, which has a security limitation when applied to civilian applications.

8.4 Applying the Research to a Civilian Application

Aside from using early kill chain approaches with less sensitive data, the method proposed in this research within Chapter 4 could be closer to a negative correlation approach. It proposes using the identification of civilian emitters, where they can be distinct from military emitters, and the secure data limitation does not apply. The mission data templates thus contain civilian accessible data, and the deviations of intercepted emissions from those mission data templates provide the warning of a non-civilian emitter. Tracking civilian emitters complicates the mission data as many emitters look similar, obeying the same industry standards. Military emitters can be more diverse and have more distinctions, making the templates for matching more detailed and increasing the data requirement. However, the method demonstrated here uses the high dimensionality discrimination potential of the neural ML methods. The generated dataset demonstrated is within civil marine radars in X-Band (9.2-9.5 GHz) and S-Band (2.9-3.1 GHz) [291] and also within different radar modes and configurations to boat fits but within the same industry standards. Marine radars in X-Band are very common at (~9.4 GHz) and contribute to an area of the spectrum called the busy band, which generally is a well-known ambiguous problem in identification. As such marine radar modes in the X and S-bands are in the dataset for the method demonstration that will also differ in the parameters scan-speed, PD, and PRI.

8.4.1 Image Dataset Construction

The construction of images used in the ML method from ELINT emitter data specifications used a three-channel image and captured Pulse Duration (PD), PRI, carrier frequency, power, timing, and phase timing, with also the modulation of those parameters within a single image over time, and this has the potential for a high level of discrimination. See Figure 47 for examples of the constructed images from different emitter radar modes.

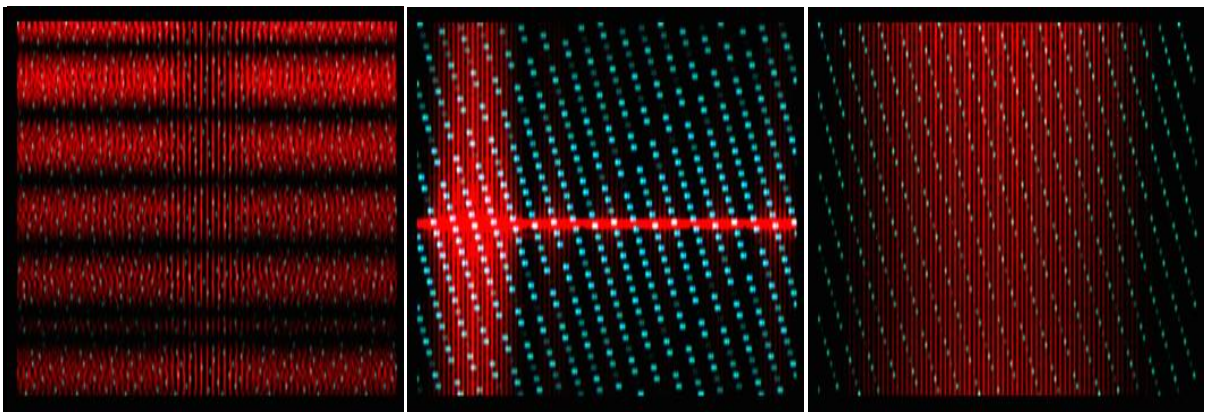


FIGURE 47 THREE IMAGES GENERATED IN THE DATASET GENERATOR ARCHITECTURE

This method took some experimentation as the size of an image could quickly become large, so low rate sampling over long durations is employed in constructing the image, and the geometry required to have pixel clusters that map directly to the discriminate-able emitter information. Each channel has an independent kernel filter until combined, as each channel is a 2D convolution rather than 3D, so the filter encoding in the channels also has some independence to be respected in radar emissions. The existing template methods can use frequency and PRF modulation for matching, so coding two channels with the I/Q pulse phase provides PD and PRI modulations, while the third channel is the Fourier transform, providing frequency and power and scanning and beam shaping discriminate-able emitter information. In Figure 47, the machine learning dataset images are RGB, where the Red channel is a spectrogram, and the Blue and Green channels are I/Q channels. The images have a fast and slow time duration definition, where the fast time is *10ms*, as the duration for a sample rate of 50kHz (in this LPRF example), for the row values in the image and synchronized to the first pulse. *10ms* is longer than many anticipated Pulse Repetition Intervals (PRI), providing no blank lines in the image on continuous operation. The slow time duration is *1 second* over all the rows in the image and is the period to experience a scan modulation. That image format provides an image where the first pulse is in the top left corner and has a cascade of pulses in the rows and columns depending on the PRI and pulse width modulations. These will vary in the Blue and Green channels, depending on the power and phase content in those pulses. The Red channel is the Fourier transform of the rows and therefore shows frequency domain modulations from the pulses and carrier together with beam scan gains. The images use a convolutional network, so the three colours become channels and are subject to filters in feature extraction in the layers. They will be passed on to subsequent convolutional layers and combined later in the network in a dense layer, and those channels provide overlaps in the discriminator dimensionality information of the emitters. From a collected group of 46 radar modes from manuals, the 14 selected had a diminutive level of discrimination, being from the same radar sets in different configurations, so many parameters are the same. They would require more parameters in a template in the traditional method. See Table 17 for a list of the radar modes with their respective classification tags.

TABLE 17
MARINE RADAR MODES FOR IDENTITY CLASSIFICATION AND DISCRIMINATION [J1]

Radar Mode / Classification Tag	RF (GHz) and Agility	Tx Coherent	ERP (dBw)	PD and Excursion (us/MHz)	PRI (us)	Beamwidth / Side Lobes (deg/dBi)	Scan Speed (rpm)
S-band Kelvin Hughes SharpEye 24NM Fast Scan	3.1-2.9 non agile 8 chans	Non-Coh.	52.3	128/ 5 FMOP	434.782 6 Fixed	1.9 Az 26 El -30 Az -20 El	46
S-band Kelvin Hughes SharpEye 24NM Slow Scan	3.1-2.9 non agile 8 chans	Non-Coh.	52.3	128/ 5 FMOP	434.782 6 Fixed	1.9 Az 26 El -30 Az -20 El	24
S-band Kelvin Hughes SharpEye 48NM Fast Scan	3.1-2.9 non agile 8 chans	Non-Coh.	52.3	128/ 5 FMOP	847.457 6 Fixed	1.9 Az 26 El -30 Az -20 El	46
S-band Kelvin Hughes SharpEye 48NM Slow Scan	3.1-2.9 non agile 8 chans	Non-Coh.	52.3	128/ 5 FMOP	847.457 6 Fixed	1.9 Az 26 El -30 Az -20 El	24
S-band Kelvin Hughes SharpEye 96NM Fast Scan	3.1-2.9 non agile 8 chans	Non-Coh.	52.3	128/ 5 FMOP	1562.5 Fixed	1.9 Az 26 El -30 Az -20 El	46
S-band Kelvin Hughes SharpEye 96NM Slow Scan	3.1-2.9 non agile 8 chans	Non-Coh.	52.3	128/ 5 FMOP	1562.5 Fixed	1.9 Az 26 El -30 Az -20 El	24
X-band Kelvin Hughes SharpEye 96NM Fast Scan	9.48-9.22 0.3MHz Agile	Non-Coh.	52.3	40/ 5 FMOP	1562.5 Fixed	0.45 Az 26 El -30 Az -30	44

						El	
X-band Kelvin Hughes SharpEye 96NM Slow Scan	9.48-9.22 0.3MHz Agile	Non-Coh.	52.3	40/ 5 FMOP	1562.5 Fixed	0.45 Az 26 El -30 Az -30 El	22
X-band Kelvin Hughes 1262 SharpEye 24NM Fast Scan	9.48-9.22 0.3MHz Agile	Non-Coh.	52.3	40/ 5 FMOP	434.782 6 Fixed	0.45 Az 26 El -30 Az -30 El	44
X-band Kelvin Hughes 1262 SharpEye 24NM Slow Scan	9.48-9.22 0.3MHz Agile	Non-Coh.	52.3	40/ 5 FMOP	434.782 6 Fixed	0.45 Az 26 El -30 Az -30 El	22
X-band Kelvin Hughes 1262 SharpEye 48NM Fast Scan	9.48-9.22 0.3MHz Agile	Non-Coh.	52.3	40/ 5 FMOP	847.457 6 Fixed	0.45 Az 26 El -30 Az -30 El	44
X-band Kelvin Hughes 1262 SharpEye 48NM Slow Scan	9.48-9.22 0.3MHz Agile	Non-Coh.	52.3	40/ 5 FMOP	847.457 6 Fixed	0.45 Az 26 El -30 Az -30 El	22
RayMarine HD	9.405 25MHz Agile	Coh.	36	0.9/ 0 Fixed	1219.5 10% Jitter	4.9 Az 25 El -35 Az -35 El	24
RayMarine Quantum	9.354-9.446 92MHz Agile	Non-Coh.	13	14.7/ 32 FMOP	1086.95 10% Jitter	4.9 Az 20 El -35 Az -35 El	24

These radar modes were then captured into C4L for synthetic dataset generation as the time-varying parameters of radar behaviour. An example is in Figure 48, where the setting of semi-dynamic parameters is at the beginning of a simulation (with default parameter values provided). The "Schedule" command implies that what is within the blocks statement is run in sequence or concurrently. The assignments make a calculation either sequentially or concurrently. The *TxScanConfig* command is configured for a single circular radar scan and is the exit criteria *(as ScanGen)* of the *SurvScan* Schedule. Concurrently there is a *TxModConfig* for the PRI generation, which will repeat until the *SurvScan* exit criterion is complete. The *"Assignment"* statements within the *SurvScan concurrent* schedule command provide varying pulse parameters within the scan.

```
Schedule RayMarineQuantum Sequenced
{
   Parameter SemiDynamic RF              = ((9.354+9.446)/2);
   Parameter SemiDynamic RFAgilityBW     = 91;
   Parameter SemiDynamic TxPower         = 13.0;
   Parameter SemiDynamic RxAmp           = 10.0;
   Parameter SemiDynamic PulseDuration   = 14.7;
   Parameter SemiDynamic PulseExcustion  = 32;
   Parameter SemiDynamic PRIValue        = 1.08695;
   Parameter SemiDynamic PRIJitterPerc   = 10;
   Parameter SemiDynamic TxNonCoherent   = 0.0;
   Parameter SemiDynamic RxCoherent      = 1.0;
   Parameter SemiDynamic ScanSpeed       = 144;
   Parameter SemiDynamic AzBeamWidth     = 4.9;
   Parameter SemiDynamic ElBeamWidth     = 20;
   Parameter SemiDynamic AzSideLobeSupp  = 35;
   Parameter SemiDynamic ELSideLobeSupp  = 35;
```

```
Assignment CarrierRF                    = (RF * 1000000000.0);
Assignment RFExcustion                  = (PulseExcustion * 100000.0);
Assignment PulseWidthVal                = (PulseDuration/1000000.0);
Assignment PulseRep                     = (PRIValue/1000.0);
Assignment TxStart                      = (-RFExcustion/2);
Assignment TxEnd                        = (+RFExcustion/2);

Assignment PulseBandwidth               = (1.0/PulseWidthVal);
Assignment ButterWorthFiltRatio         = (sqrt(2.0));
Assignment RxBandwidth                  = (PulseBandwidth*ButterWorthFiltRatio);
Assignment RxStart                      = (min((-RxBandwidth/2),TxStart));

Assignment RxEnd                        = (max((+RxBandwidth/2),TxEnd));
Assignment ScanPos                      = 0;
Assignment ScanInc                      = ((PulseRep/ScanSpeed));
Assignment ScanPeriod                   = (360.0/ScanSpeed);

 SurvScan Emitter On Radar SensorProgram BoatRadar Concurrent ExitCriteria (ScanGen)
{
   Assignment PhaseVal      = ((random() *360.0-180.0)*TxNonCoherent);

   Assignment TxPhaseVal    = (PhaseVal*TxNonCoherent);
   Assignment RxPhaseVal    = (PhaseVal*RxCoherent);

   Assignment PRI       = (PulseRep+(random()*(PulseRep*(PRIJitterPerc/100.0))));
   Assignment RF        = (CarrierRF + (random()*(RFAgilityBW*1000000.0)));


   ScanGen Sensor ChanNum 1
      TxScanConfig Repeat Start 1 WaveNum 0
      PointAngle Azimuth Position Start (-180) End (180-ScanInc) Rate (ScanSpeed) Elevation
      Position Start (0) RollAxis Position Start (0) x Position Start (0) y Position Start (0) z Position
      Start (0) ScanCentre Horizon ElOnAzMount HorizonStablised Beamwidth Azimuth
      Position Start (AzBeamWidth) Elevation Position Start (ElBeamWidth) RollAxis Position
      Start (0) Suppression Position Start (AzSideLobeSupp) Position Start (ElSideLobeSupp) Pol
      Linear Position Start (90) Purity Position Start (100);


   Syncroniser Sensor ChanNum 1
      TxModConfig Repeat Start 1 WaveNum 0
      Carrier Position Start (RF) GapDuration Position Start (0) Duration Position Start
      (PulseWidthVal) StartRF Position Start (TxStart) EndRF Position Start (TxEnd) Phase
      Position Start (TxPhaseVal) Gain Position Start (TxPower) RepetitionInterval Position Start
      (PRI) TargetAcquire EWSearch;


   InAction TimeDelay (ScanPeriod);

};

}
```

FIGURE 48 EXAMPLE C4L SCRIPT FOR GENERATING TIME-VARYING RADAR BEHAVIOUR

8.4.2 Neural Network Model Architecture in Emitter Classification

The neural model in Figure 47 is simple, and is not unlike the Torres model used in Chapter 7 with the MNIST dataset [150], but with adapted convolutional layer kernel filters and an RGB colour image input.

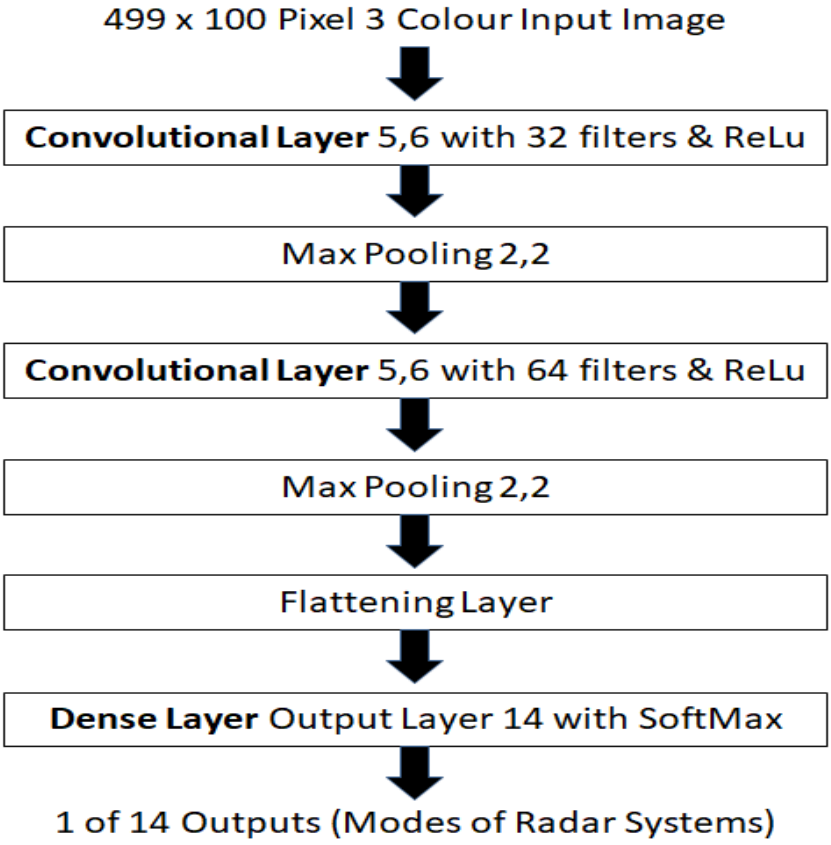


FIGURE 49 EMITTER IDENTIFICATION MODEL ARCHITECTURE

See Appendix A to examine the higher dimensionality fidelity used in the dataset generation, and also developed as part of this research.

8.5 Summary of Synthetic Dataset Generation with Emitter Identification

When testing the synthetic emitter dataset generator against emitter ambiguity, the dataset generated (SD_CMRM_Iv1) [54], when applied to an image classifier, achieved 99.8% accuracy in identity classification between the 14 emitters, which contained the same or similar exchanged parameters; this is a very high level of disambiguation. Some might describe it as unprecedented disambiguation compared to traditional template methods.

The high dimensionality in the ML methods in learning a template can provide a high level of disambiguation. At the same time; also, the matching PRI patterns from PD and PRI modulations are more trivial in this image format as the individual pixels in the Blue and Green channels are sensitive to the PRI modulations being present or not, while also capturing progressive scan beam and side-lobe gains in the Red Channel. Additionally, the scan speed, beamwidth, and PRI time base provide discrimination of radar modes with fast or slow scan outfitting configurations. In this respect, the proposed image format has the potential for higher dimensionality templates derived from an ML method, where that method is more palatable when combined with the Chapter 7 findings in repeatable determinism.

This method is broadband as the images cover 16 GHz of the spectrum as 2-18 GHz in the spectrogram and can also capture other CW, ICW, or Pulsed emitters that might work in combination as a discriminator of a mode. That broadband quality is equally relevant to identifying military radar groups of IADS and civilian emitter networks on aircraft or port approaches, mainly when the

emissions are time or scan synchronized. However, this method is currently an ML method for emitter identification using image classification and would need to extend to object detection and segmentation methods for an ML de-interleaving method. Furthermore, the dataset generation method could be a standard with exceptionally high disambiguation potential.

Chapter 9

Summary and Conclusions

This chapter forms the summary and conclusions with the opportunities for further research.

9.1 Summary

The research dissertation is in Cyber and Electromagnetic Activities (CEMA) for applications of AI to civilian platform protection when civilian platforms are operating near threat systems. In this field of work, the certainty and accuracy of the information available are challenges for practicality. Part of this research offered a method with a BN inside an Expert System in an ANN format for data infilling and perfecting better values based on the evidence. Another part of the research offered greater palatability of neural ML approaches with methods for repeatable determinism and explanation, and that research proposed a non-random initialization method and a formula extraction method. Some Threat analysis applications within case studies were demonstrated and led to new datasets, but those applications also demonstrated an advantage in those methods.

More specifically, Chapter 4 examined the application area of platform protection with a method to use an onion of protection mapped to kill chain stages. The kill chain stages provided management of data sources while unlocking countermeasure types with different countermeasure design considerations. That chapter also proposed some diagramming and analytical processes based on first principle analysis as a civilian accessible method, which tested the data for its capability to assert the emitter functions in a lower data trusting approach. The sensing and processing equipment was to be within existing shared civilian network infrastructures. A centralized control of the network can also offer access to private aviation and yachting with minimal additional equipment. The threat analysis method used set mappings for countermeasure tactic selection and was to combat the data availability and the lack of publication of methods with a method accessible to the civilian domain that was UK MoD cleared for publication.

In support of the Threat Analysis method proposed, Chapter 5 proposed a neuron-based expert system method that can apply rules from emission intercepts to infill missing parameters, apply analysis, or derive values for a parameterized countermeasure defined in C4L. That method was algebraic for data transferability; it could provide better-perfected values based on evidence combined with a body of knowledge within the rules. The Expert System was ANN structured with virtualized nodes for hypothesis permutations and provides confidence and certainty values based on that known body of knowledge. Compared with mandraulic approaches to threat analysis, confidence in an answer is often a human heuristic approach or based on a source-trust rather than the rigorous computation of the evidence's match to known physics and existing knowledge. This method was one part of an approach to a Neuro-Symbolic AI approach.

Also aligned to the Neuro-Symbolic AI approach, Chapter 6 provided a formula extraction method to understand neural network content using a different input data representation. That method complements the neuron-based expert systems method as the other part of the Neuro-Symbolic AI approach. The method increases the number of weights for different weights for activation value strengths. When back-propagated, this method reduces the complexity of the representation in a single layer as it allows activations strengths assigned to strength-based unique weights. That method used back-propagation, regression, and SGD, providing a high level of accuracy in the prediction and would then be extracted from the distortions in the weights after ML.

Chapter 7 brought repeatable determinism with a non-random initialization method for dense and convolutional layers in image classification with neural network methods. An advantage of the method for dense layers was structured weights towards rule extraction generalization while providing comparable performance to the existing random method but with repeatable determinism between the learning sessions. An advantage of the method for convolutional layers was earlier learning with less unlearning of the initial state. That method had stripes, curves, and spots in a Hubel and Wiesel

intuition [274], [275] toward feature extraction; that method also had higher retention in transferred learning, for example, with repeatable determinism between the learning sessions. Combined within a CNN, those methods would improve accuracy in the first epoch from 29% to 67% with a challenging, complex imbalanced colour image dataset related to the application area.

In Chapter 8, synthetic dataset generation has advantages in that the dataset can be balanced and includes low likelihood observation test cases that probabilistically would not be in real datasets from ELINT collection. With the lack of emitter datasets accessible in the civilian domain, a synthetic generation method was proposed that extends the C3L language baseline for countermeasure tactics to C4L to include new emitter signal descriptions. A demonstration of the synthetic dataset for emitter identification made with an image formation method in image classification had very high levels of disambiguation and achieved 99.8% accuracy. The demonstrated accuracy was in a problem area where the existing template methods would have a higher level of ambiguity and thus low accuracy in identification. So the level of disambiguation is an unprecedented performance in comparison and shows a promise towards a future approach to identify the emitter mode and the system, which also can map on to a possible kill chain or identify that emitter as a background emitter.

9.1.1 Research Questions, Answered

Chapter 3 focused on research aims in the form of research questions mapped onto research threads answered in this section.

9.1.1.1 How can machine learning be applied in the mission and safety critical field of EW threat analysis?

As noted in Chapter 4, the mission and safety-critical nature of EW threat analysis requires a measured assurance. As threat analysis deals with uncertain and inaccurate information, thus key to this is a measure of confidence in those results. See Chapter 5 for the confidence calculation and optimizing value estimating method within the virtualized neuron-based expert system method. In the military, the employment of AI is emerging, as are trials for ground truth, but aside from those methods, the employment of AI requires repeatability and determinism expectation measures in those results. See Chapter 7 for a repeatable and deterministic method in ML with advantages in accuracy and transferred learning. As part of the EWOS life cycle, ML can provide an influential tool for dealing with large-scale data. However, the data is less available in the civilian domain and requires data convergence (or 'data fusion') from multi-sources and imbalanced inputs. Chapter 4 provides the first principle analytical steps to threat analysis where data is not trusted. Chapter 5 provides the confidence estimating and optimizing method where a body of knowledge of rules of thumb and known physics can be applied to test the data for correlation in a machine reasoning deductive approach.

Furthermore, Chapter 6 provides a method for establishing new rules in an inductive machine reasoning approach, where the loss in that method may translate to accuracy in that inductive machine reasoning approach in terms of certification. Chapter 6 also explains learning in layers in a decompositional method. Chapter 7 demonstrated the repeatable determinism from the ML learning sessions required in fitting and prediction when fielded. Chapter 8 provides dataset generation where datasets can be scarce in the civilian domain.

9.1.1.2 What applications of EW threat analysis can AI techniques apply to?

There are many applications where AI methods can apply in threat analysis or CEMA, but with the scope of the research conducted: Chapter 4 provides insights into the threat analysis methods and challenges for the civilian domain and provides a civilian accessible methodology for threat analysis. When applying the virtualized neuron method to an expert system in Chapter 5, an example of calculated beamwidth was demonstrated with varying scopes to the body of knowledge, based on the uncertainty of the antenna type providing a perfected prediction and confidence and certainty in that prediction. This method also applies to infilling other missing database or dataset parameters. It also provided a perfected value with a confidence value based on the body of knowledge in a 'valid' scope. Chapter 6 provided the link for establishing new rules for the body of knowledge via a formula extraction method. The body of knowledge can be initially filled with existing knowledge in a deductive approach and then augmented with the formula extraction method for new rules in an ML

inductive approach. Also, in section 4.2.1 (Operational View) of Chapter 4, the analysis starts with cataloguing discriminators, and image classification is a benefit. The results shown in Chapter 7 demonstrated 86% accuracy classification of aircraft on runways in colour images using the MTARSI2 dataset [53]. Also, in Chapter 8, image classification used the generated (SD_CMRM_Iv1) [54] ELINT dataset and demonstrated emitter identification accuracy of 99.8% in emitter classification of an emitter system's identification and its' mode of operation.

9.1.1.3 How can machine learning approaches have verification and validation with safety or mission-critical assurances?

Using Schumann et al. [134] definition of verification and validation for AI approaches. In verification, Chapter 7 provided a non-random initialization state with repeatable determinism, and as such, the network has a stable, testable output. Removing an unexpected computational numerical stability noise source avoids re-colourizing noise from the sensor when combined in the activations and weights. Within validation, Chapter 7 also provides a weight order in MLP networks that is more translatable to the input activation as they correlate with pixel position and neurons and may benefit rule extraction methods. Also, in convolutional networks, the receptive field was derivable, and the non-random initialization accounted for it. That initialization state was also more aligned to image classification, and the non-random initialization method arrived at a higher learnt accuracy. In both verification and validation, Chapter 6 also explains the learning. Using an input representation and extra weight values for value ranges simplified the representation and lowered the models' depth to avoid dropout potentials. That input representation also provided higher numerical values with weights organized uniquely for value ranges. When combined with faint pixels or low numerical values scalar inputs, the numerical value is higher when combined with the weight and is more uniformly fair in numerical representation across activation strength value ranges.

9.1.1.4 How can neuron approaches gain safety or mission-critical assurances?

Safety or mission-critical assurance is a broad subject, but concerning this research: In Chapter 7, a non-random initialization method provided repeatable determinism in every separated learning session as an advantage for testing. Removing a numerical instability provided repeatable determinism with higher IA. In MLP networks, the non-random initialization learnt weights resultant are clustered and aligned to neighbours, which can be an advantage to rule extraction, simplifying a generalization step. In convolutional layers, non-random initialization provides earlier learning and, with less unlearning of the initial state, as the non-random method is more aligned to the application of image classification. The non-random initialization state's accuracy is higher with the same deterministic and repeatable result in every separated learning session. In Chapter 6, a formula is extracted from a network layer, providing a level of explanation of the learning. It used an input representation method with weights assigned to input value number ranges and, as such, could be commutable to general neural networks, reducing the model depth and simplifying the representation for later extraction. In Chapter 8, a synthetic dataset generator allowed a high discrimination rate as an alternative to template matching for emitter identification as a motivation to gain safety or mission-critical assurance; this employed image classification with images of emitters that are still humanly interpretable. The synthetic dataset generator provides an ability to control the balance of datasets by understanding regular observations and controlling the proportions of safety cases.

9.1.1.5 How can Symbolic AI approaches perform machine learning?

In Chapter 5, is a virtualized neuron method in an expert system where those virtualized neurons have a unique semantic derived from the imbalanced and irregular input of atom facts and axiom rules. Those combinations of graded atoms and axioms aim at the confidence of the agreement as applied in all permutations and provide a better value with a confidence metric. The rules were algebraic and formed a body of available knowledge. The confidence and value steering pertained to the matching agreement in the available 'body of knowledge' and updated the estimate. In Chapter 6, a formula extraction method to a neural network forms new algebraic rules from back-propagation. Combining both methods is a basis for the Neuro-Symbolic AI approach, where the symbolic AI method in Chapter 5 combines with the ML and formula extraction method in Chapter 6.

9.2 Conclusions

AI generally offers immense potential in the application area, of which ML methods have been more demanding to certify given the black-box nature of the cross-validation accuracy and limited understanding of the content, also compounded by the lack of repeatability in learning sessions combined with a diversity of solutions to know which are safe. This research provided a single solution initialization method via an optimized initialization state that is repeatable and deterministic, which also benefits accuracy and learnt model retention in a transferred learning case. Required further work is in the shuffle algorithm used for the non-random initialization state in convolutional layers to resolve aliasing in the number of filters on offer, which occurs with some filter dimensions.

A Symbolic AI method called the expert system has been more palatable in advisory roles, as they are reviewable. This research examined this method with a virtualized neuron-based form with algebraic rules. That form also generated confidence that estimated better values given the rules 'valid' body of knowledge. This method uses an ML approach for the expert system to form new algebraic rules via a formula extraction method. The formula extraction method exploited regression, SGD, and back-propagation with an input representation that used multiple weight values in activation strengths of layer input value ranges. This form simplified the representation into a shallower network as it could represent more in a single layer, given that number ranges of activations are represented uniquely in a layer. The number of nodes was less critical as the learnt weights were in those dendrite activations. That method made numerical discriminations of math operators concerning the input relationships. That method also could be integrated into the mainstream ML approaches to extend the input dimensions and include activation strength as an input dimension along with height, width, and channels. The method demonstrated with the TensorFlow AI framework has a higher expectation of successful integration. Required further work is for more complex numerical relationship discrimination and integrating the two methods. However, further work could exploit the virtualized neurons in the modified expert systems method, forming a method closer to Neuro-Symbolic AI.

The non-random initialization methods provided repeatable determinism in dense and convolutional layer types, where the repeatable determinism quality is more palatable to safety and mission-critical applications in testing. Combined with a CNN, the approaches provided higher accuracy with earlier learning. The convolutional layer method is Hubel and Wiesel inspired [274], [275], plus observations of feature extraction. The dense layer method had correlated pixels and neurons in the weight structure, which has an advantage for generalization in rule extraction. The convolutional layer method has an aliasing issue with some window dimensions, which is a research subject for increasing the number of filters on offer across all window geometries.

Image classification of aircraft on runways was demonstrated with repeatable determinism in a challenging imbalanced dataset varying in light aspect, viewing angle, and image resolution using the MTARSI2 dataset [53], with an increase in the accuracy from 29% to 67% in the first epoch with this initialization method. Overall, the cross-validation accuracy was 86%, and further research is for higher classification performance. Also, Image classification in an emitter identification application where a synthetic dataset generator generated ambiguous civilian marine radar emitters using the same emission standards (SD_CMRM_Iv1) [54], and as employed in an image classification method, shows a very high level of disambiguation at 99.8% accuracy. Further work is required to exploit further AI approaches in object detection and segmentation into this method toward de-interleaving. However, this method demonstrated identification between fourteen radar modes, which may derive kill chain positions from those radar modes and indicate civil emitter emissions. Further work may also look at the robustness against the adversarial attack approaches such as FSGM.

A method of threat analysis using first principle analytical steps and further research should apply more AI methods to those steps. The expert system method can be employed to automate those steps, while the rule extraction method learns new rules and characterizes unknown emitters. The threat analysis method also provides set mappings from the steps toward making countermeasure tactic selections. The threat analysis method is published here with the UK MOD authority as a civilian accessible method in the scarcity and absence of publishing of other threat analysis methodologies, which could be more classified.

C3L countermeasure specification language was adapted to emitter descriptions and used in the synthetic dataset generator as C4L. C4L provides an unclassified dataset source to compensate for the lack of publically accessible datasets in ELINT. That synthetic dataset generator can also allow more AI methods for generating images toward a mechanism for warning and as a trigger for countermeasures. C4L also provides live parameter updates allowing live adaption, and future work will use parameter fitting in a GAN approach.

9.2.1 Further Research

A list of different further research themes is provided below from the research conducted in this dissertation:

9.2.1.1 Non-Random Initialization method

The non-random initialization method used for convolutional layers and networks exhibits numerical aliasing issues with some filter dimensions, which causes a filter sequence to repeat earlier than other filter dimensions; this means fewer filter permutations are on-offer in some cases. Therefore, that means that some filter dimensions have better performance and some less if numerical aliasing occurs when some filter permutations are required. Further research is required to resolve aliasing in the shuffle algorithm limiting the number of kernel filters on-offer used in the non-random convolutional layer initialization method. Other sequences of inspirations in other applications may also exist, where the data is not imagery but is sound, textual, radar, or ELINT, but where another predisposed warm-start format may be applicable and where it is still not dataset coupled. Further work may also look at the impact on model architectures such as Inception and ResNet with the non-random initialization state, as that initialization state may be less resolution coupled at the outset as it has more stripes across the kernels rather than only speckles and spots at the kernel resolution in the random form.

9.2.1.2 Formula Extraction method and a 'Condensed Network'

The formula extraction method must examine more weight distortion indicators of more complex numerical operators and functions. Also, in a more general case, the extra weights for the activation strengths could be integrated and tested with more dataset types of imagery, sound, video, and text. Low-level activation values may have importance, such as the numerical value activation within the network has a higher numerical value: 0.5 to 1.0 in at least one of the values used in the numerical representation. This focus may lead to a network requirement that is 'condensed' rather than 'deep' as this format can represent more in a single layer and within a single node of that layer. That may be in a form already predisposed to an extractable form.

9.2.1.3 ELINT Imagery Classification method

The (SD_CMRM_Iv1) [54] dataset may also be a subject for object detection and segmentation methods to perform emitter de-interleaving when more than one emitter overlaps with other emitters. There may also be extra complexities in congested broad-spectrum analysis, which may cause a 4D image where the fourth layer of the image is a frequency band as the fourth dimension after height, width, and colour channels. Additionally, with the GAN approach, research for a production process for producing C4L descriptions, where the parameters of the C4L are the stimulus for the generator, and the discriminator provides judgment from captured samples, which may provide a humanly readable format for a human validation approach. That leads to research into the effect and protection of adversarial attacks like FSGM. Furthermore, the MTARSI2 dataset [53] and model may include research for further categorization, data augmentation, noise injection, orientation, stretching, compressing, and translation of the images towards achieving higher classification accuracy.

9.2.1.4 Furthering the Neuro-Symbolic AI method

The expert system virtualized neurons for each permutation of the rules in the body of knowledge and evidence inputs available toward perfected values and confidences. The formula extraction method intends for the backward chaining ML method to establish new algebraic rules as the Neuro-Symbolic AI approach. That backward chaining ML method provides many weights indexed on activation weights' strength as an input dimension as an array of weights of what would traditionally be a single weight. Further research will characterize the weight array values to provide a matching function realized as an algebraic rule.

9.2.1.5 Development of the Threat Analysis method

The threat analysis approach is a methodology that has low trust in the data provided. That is to say, it is not reliant on high confident data sources but is inferred based on testing the data for its strengths, weaknesses, opportunity, and vulnerabilities within the contexts of the data provided. The methodology is automatable by testing the data presented to it to assert possible functions and then combining them in groupings as a complete understanding. When considered in a kill chain requirement, further research may be in the set mappings to establish missing functions and observations with a collection viewpoint.

9.2.1.6 Countermeasure Generation and Selection

Countermeasure techniques captured in C4L are defined in schedules of elements to be adaptable to a changing engagement. Those C4L techniques have parameters that allow the countermeasure technique to configure for a specific threat and protected platform engagement. When they are configured and combined, they form a specific countermeasure tactic. In further research, an adversarial or software annealing approach might optimize the parameters for a tactic. Furthermore, automatic countermeasure technique selection based on the threat kill chain stage's intentions, and the set mappings from the threat analysis method may provide countermeasure technique selection for the tactic generation, and further research may look at approaches to matching and optimizing tactics with fallback approaches compliant to the onion of protection's content.

BIBLIOGRAPHY

[1] F. D. Coonfield, "Cyber electromagnetic activities within the mission command warfighting function: why is it important and what is the capability?," *Army Command and General Staff College Fort Leavenworth,* vol. 62, 2013.

[2] C. H. Cheng and J. Tsui, "3 overall view of electronic warfare," in *An introduction to electronic warfare; from the first jamming to machine learning techniques,* River Publishers, 2021, pp. 39-40.

[3] R. Di Pietro, S. Raponi, M. Caprolu and S. Cresci, "New dimensions of information warfare. in: new dimensions of information warfare," in *Advances in Information Security,* vol. 84, Cham, Switzerland, Springer, 2021.

[4] B. Gonçalves, "Can machines think? the controversy that led to the turing test, 1946-1950," *PhilSci Archive,* 2021.

[5] J. P. Bowne and M. T. Thurbon, Electronic warfare, 1st ed., London, UK: Brasssy's Inc, 1998.

[6] E. Azimirad and S. R. Ghodsinya., "Review of data fusion models and architectures for air targets threat assessment in C4I system," *C4I Journal 4,* no. 3, pp. 13-34, 2021.

[7] E. Paparidis and K. Kotis, "Knowledge graphs and machine learning in biased C4I applications," *arXiv Preprint, arXiv:2106.09258,* 2021.

[8] D. M. Dermanelian, "Command, control, communications, computers, cyber, and intelligence (c5i) sustainment management policy," US department of homeland security: United States coast guard, Washington, USA, 2021.

[9] M. Mekhail, J. Salminen, L. Plé and J. Wirtz, "Artificial intelligence in marketing: topic modeling, scientometric analysis, and research agenda," *Journal of Business Research 124,* pp. 389-404, 2021.

[10] A. Petrovski, P. Rattadilok and S. Petrovskii, "Intelligent measurement in unmanned aerial cyber," in *Peer Reviewed Proc of the International Conference on Engineering Applications of Neural Networks,* Aberdeen, UK, 2019.

[11] S. McLean, G. J. Read, J. Thompson, C. Baber, N. A. Stanton and P. M. Salmon, "The risks associated with artificial general intelligence: a systematic review," *Journal of Experimental & Theoretical Artificial Intelligence,* pp. 1-17, 2021.

[12] J. Brett, G. Gamble, W. Reid, S. Smith, A. Woolley and G. Yiannakopoulos, "Integrated survivability analysis of naval platforms in high threat environments," in *Peer Reviewed Proc. of the Pacific International Maritime Conference,* Sydney, Australia, 2017.

[13] J. Matuszewski, "The analysis of modern radar signals parameters in electronic intelligence system," in *Peer Reviewed Proc. of the 13th International Conference on Modern Problems of Radio Engineering, Telecommunications and Computer Science (TCSET),* 2016.

[14] X. T. Nguyen, "Threat assessment in tactical airborne environments," in *Peer Reviewed Proc. of the Fifth International Conference on Information Fusion. FUSION 2002. (IEEE Cat.No.02EX5997),* 2002.

[15] J. Preden, J. Kaugerand, E. Suurjaak, S. Astapov, L. Motus and R. Pahtma, "Data to decision: pushing situational information needs to the edge of the network," in *Peer*

Reviewed Proc. of the IEEE International Multi-Disciplinary Conference on Cognitive Methods in Situation Awareness and Decision, 2015.

[16] J. A. Scardina, P. Ryberg and A. Gerstenfeld, "Future ATC automation aids based upon AI technology,," *Peer Reviewed Proc. of the IEEE,* vol. 77, no. 11, pp. 1625-1633, 1989.

[17] C. Downing, J. Breitenbach and E. McCauley, "Building the data-centric air traffic management system of the future," *Journal of Air Traffic Control,* 2020.

[18] B. L. Young, "Predicting vessel trajectories from AIS data using R," Defence Technical Information Centre, 2017. [Online]. Available: https://apps.dtic.mil/sti/citations/AD1046595. [Accessed 24 September 2021].

[19] K. Dogancay, Z. Tu and B. Ibal, "Research into vessel behaviour pattern recognition in the maritime domain: past, present and future," *Digital Signal Processing, 103191,* 2021.

[20] P. W. Lemme, "ACARS/VHF transceiver interface unit (AVIU)". US Patent US5920807A, 13 November 1995.

[21] S. Chen, S. Zheng, L. Yang and X. Yang, "Deep learning for large-scale real-world ACARS and ADS-B radio signal classification," *IEEE Access,* vol. 7, pp. 89256-89264, 2019.

[22] A. B. Garcia, R. F. Babiceanu and R. Seker, "Artificial intelligence and machine learning approaches for aviation cybersecurity: an overview," in *Peer Reviewed Proc. of the 2021 Integrated Communications Navigation and Surveillance Conference (ICNS),* 2021.

[23] Volpe, Center;, "Maritime safety and security information system (MSSIS)," U.S. Department of Transportation, 1 April 2021. [Online]. Available: https://www.volpe.dot.gov/infrastructure-systems-and-technology/situational-awareness-and-logistics/maritime-safety-and. [Accessed 4 February 2022].

[24] F. Natale, M. Gibin, A. Alessandrini, M. Vespe and A. Paulrud, "Mapping fishing effort through AIS data," *PloS one,* vol. 10, no. 6, p. e0130746, 2015.

[25] L. Etienne, E. Alincourt and T. Devogele, "Maritime network monitoring," *Maritime Networks,* pp. 214-233, 2015.

[26] M. S. Syms, A. W. Isenor, B. Chivari, A. DeBaie, A. Hogue and B. Glessing, "Building a maritime picture in the era of big data: the development of the geospatial communication interface+," in *Peer Reviewed Proc. of the 2021 International Conference on Military Communication and Information Systems (ICMCIS),* 2021.

[27] J. M. Reilly, "Multidomain operations: a subtle but significant transition in military thought," *Air Force Research Institute Maxwell AFB United States,* 2016.

[28] S. Townsend, "Accelerating multidomain operations," *Military Review Online Exclusive,* 2018.

[29] C. Bartels, T. Tormey and J. Hendrickson, "Multidomain operations and close air support," *Military Review,* 2017.

[30] J. Robertson, J. M. Fossaceca and K. W. Bennett, "A cloud-based computing framework for artificial intelligence innovation in support of multidomain operations," *IEEE Transactions on Engineering Management,* 2021.

[31] N. Flack and M. Reith, "Self-directed learning tools in USAF multi-domain operations education," in *Peer Reviewed Proc. of the European Conference on Cyber Warfare and Security.,* Reading, UK, 2019.

[32] N. Flack, C. Voltz, R. Dill, A. Lin and M. Reith, “Leveraging serious games in air force multi-domain operations education: a pilot study.,” in *Peer Reviewed Proc. of the ICCWS 2020 15th International Conference on Cyber Warfare and Security*, 2020.

[33] G. Cirincione and D. Verma, “Federated machine learning for multi-domain operations at the tactical edge,” in *Peer Reviewed Proc. of the Volume 11006, Artificial Intelligence and Machine Learning for Multi-Domain Operations Applications*, Baltimore, USA, 2019.

[34] S. A. Atkins, “Multidomain observing and orienting: ISR to meet the emerging battlespace,” *Air & Space Power Journal,* vol. 32, no. 3, pp. 26-45, 2018.

[35] A. Feickert, “Defense primer: army multi domain operations (MDO),” *Congressional Research SVC,* 2021.

[36] T. South, “This 3-star army general explains what multi-domain operations mean for you,” Army Times, 11 08 2019. [Online]. Available: https://www.armytimes.com/news/your-army/2019/08/11/this-3-star-army-general-explains-what-multi-domain-operations-mean-for-you/. [Accessed 2021 10 16].

[37] M. W. Burgoon, “Multi-domain operations: the historical case,” *US Army School of Advanced Military Studies Fort Leavenworth United States,* 2019.

[38] L. England and S. Phippard, “It’s the robot’s fault! AI and legal liability in aerospace,” Royal Aeronautical Society, [Online]. Available: https://www.aerosociety.com/news/it-s-the-robot-s-fault-ai-and-legal-liability-in-aerospace/. [Accessed 24 September 2021].

[39] N. Gardner, “How will AI impact maritime Law?,” Thetius, [Online]. Available: https://thetius.com/how-will-ai-impact-maritime-law/. [Accessed 24 September 2021].

[40] R. J. Kemp, “Regulating the safety of autonomous vehicles using artificial intelligence,” *Communications Law,* 2019.

[41] R. Martin, S. Milz and P. Mader, “Development methodologies for safety critical machine learning applications in the automotive domain: a survey,” in *Peer Reviewed Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPT) Workshops,* 2021.

[42] M. Yap, R. L. Johnston, H. Foley, S. MacDonald, O. Kondrashova, K. A. Tran, K. Nones, L. T. Koufariotis, C. Bean, J. V. Pearson, M. Trzaskowski and N. Waddell, “Verifying explainability of a deep learning tissue classifier trained on RNA-seq data,” *Scientific Reports 11, 2641,* 2021.

[43] C. Berghoff, P. Bielik, M. Neu, P. Tsankov and A. von Twicke, “Robustness testing of AI systems: a case study for traffic sign recognition,” in *Artificial Intelligence Applications and Innovations. AIAI 2021. IFIP Advances in Information and Communication Technology,* vol. 627, Cham, Springer, 2021.

[44] L. M. Augusto, “From symbols to knowledge systems: A. Newell and H. A. Simon's contribution to symbolic AI,” *Journal of Knowledge Structures and Systems,* vol. 2, no. 1, pp. 29-62, 2021.

[45] J. Radua and A. F. Carvalho, “Route map for machine learning in psychiatry: Absence of bias, reproducibility, and utility,” *European Neuropsychopharmacology,* vol. 20, pp. 115-117, 2021.

[46] J. Brownlee, “Why do I get different results each time in machine learning?,” Machine Learning Mastery, 17 August 2020. [Online]. Available: https://machinelearningmastery.com/different-results-each-time-in-machine-learning/.

[Accessed 23 September 2021].

[47] I. K. Chen, M. D. Klimek and M. Perelstein, “improved neural network Monte Carlo simulation,” *SciPost Phys,* vol. 10, no. 023, pp. 2009-07819, 2021.

[48] D. Chijiwa, S. Yamaguchi, Y. Ida, Y. Umakoshi and T. Inoue, “Pruning randomly initialized neural networks with iterative randomization,” *Advances in Neural Information Processing Systems,* vol. 6, no. 34, 2021.

[49] F. Kausar, P. Aishwarya and G. K. Shyam, “Understanding and study of weight initialization in artificial neural networks with back propagation algorithm,” *Information Technology in Industry,* vol. 9, no. 1, pp. 1443-1450, 2021.

[50] R. Melen, F. Sartori and L. Grazioli, “Modeling and understanding time-evolving scenarios,” *Journal of Systemics Cybernetics and Informatics,* vol. 13, no. 5, pp. 62-67, 2015.

[51] J. E. King, S. C. Jupe and P. C. Taylor, “Network state-based algorithm selection for power flow management using machine learning,” *IEEE Transactions on Power Systems,* vol. 30, no. 5, pp. 2657-2664, 2015.

[52] I. Ullah, M. N. Baharom, H. Ahmad, F. Wahid, H. M. Luqman, Z. Zainal and B. Das, “Smart lightning detection system for smart-city infrastructure using artificial neural network,” in *Peer Reviewed Proc. of the Wireless Personal Communications Conference 2018,* 2019.

[53] R. Rudd-Orthner and L. Mihaylova, “Multi-type aircraft of remote sensing images: MTARSI2,” Zenodo, 30 June 2021. [Online]. Available: https://zenodo.org/record/5044950#.YcWalmDP2Ul. [Accessed 30 June 2021].

[54] R. N. M. Rudd-Orthner and L. Mihaylova, “AI synthetic dataset of 14 civil marine radar modes as images (SD_CMRM_Iv1),” University Of Sheffield, Sheffield, UK, 2022.

[55] W. Zhize, “Multi-type aircraft of remote sensing images: MTARSI,” Zenodo, 18 May 2019. [Online]. Available: https://zenodo.org/record/3464319#.YcWZU2DP2Ul. [Accessed 30 June 2021].

[56] M. Senft, “Convergence of cyberspace operations and electronic warfare effects,” The Cyber Defence Review US Army, 2016. [Online]. Available: https://cyberdefensereview.army.mil/CDR-Content/Articles/Article-View/Article/1136055/convergence-of-cyberspace-operations-and-electronic-warfare-effects/. [Accessed 1 October 2021].

[57] W. Lambrechts and S. Sinha, “SiGe based re-engineering of Electronic Warfare Subsystems,” in *SiGe-based r-engineering of Electronic Warfare Subsystems. Signals and Communication Technology*, Cham, Switzerland,, Springer International Publishing, 2017, p. 3 Fig 1.1.

[58] C. Schleher, Electronic warfare in the information age, Norwood, MA, USA: Artech House Inc, 1999.

[59] F. Galbusera, G. Casaroli and T. Bassani, “Artificial intelligence and machine learning in spine research,” *JOR Spine,* vol. 2, no. 1 e1044, 2019.

[60] R. Singh, “Rise and fall of symbolic AI,” Medium, [Online]. Available: https://towardsdatascience.com/rise-and-fall-of-symbolic-ai-6b7abd2420f2. [Accessed 24 May 2020].

[61] B. Dickson, “What is symbolic artificial intelligence?,” TechTalks, [Online]. Available: https://bdtechtalks.com/2019/11/18/what-is-symbolic-artificial-intelligence/. [Accessed 26 May 2020].

[62] A. Santoro, D. Barrett, A. Morcos, T. Lillicrap and F. Hill, “Measuring abstract reasoning in neural networks,” Deepmind, [Online]. Available: https://deepmind.com/blog/article/measuring-abstract-reasoning. [Accessed 26 May 2020].

[63] G. Seffers, “Smarter AI for electronic warfare: cognitive EW and other technologies are set to deduce the right countermeasures in the moment,” Signal magazine, [Online]. Available: https://www.afcea.org/content/smarter-ai-electronic-warfare. [Accessed 23 May 2020].

[64] M. Iriarte, “DARPA's adaptive radar countermeasures project moves to phase 3,” Military Embedded Systems, [Online]. Available: https://militaryembedded.com/radar-ew/signal-processing/darpas-adaptive-radar-countermeasures-project-moves-to-phase-3. [Accessed 23 May 2020].

[65] B. Tottingham, “EW countermeasures developed with artificial intelligence in mind,” Armada International, [Online]. Available: https://armadainternational.com/2018/06/ew-countermeasures-developed-with-artificial-intelligence-in-mind/. [Accessed 23 May 2020].

[66] S. Cole, “Cognitive electronic warfare: countering threats posed by adaptive radars,” Military Embedded Systems, [Online]. Available: http://mil-embedded.com/articles/cognitive-electronic-warfare-countering-threats-posed-by-adaptive-radars/. [Accessed 23 May 2020].

[67] R. N. Rudd-Orthner, “AOC EW Saudi Arabia (3rd): EW Saudi Arabia (3rd),” TangentLinks, [Online]. Available: http://tangentlink.com/wp-content/uploads/2014/07/6.-A-Formal- Countermeasure-Language-a-Common-Generic-Architectu-Richard-Rudd-Orthner.pdf. [Accessed 5 December 2018].

[68] R. N. Rudd-Orthner, “A formal countermeasures language: a common generic architecture as a technology enabler for future electronic warfare capability,” p. 105, 2013.

[69] R. N. Rudd-Orthner, “To investigate a collaborative environment enabling stakeholders, engineers, scientists and mission dataset technicians to cooperate effectively over disciplined boundaries in the field of countermeasures and platform self-protection,” MSc thesis, Dept. of Computer Science, Univ. of Lincoln, Lincoln, UK, 2015.

[70] A. Grynkewich, “The future of air superiority, part III: defeating A2/AD,” WarOnTheRocks, [Online]. Available: https://warontherocks.com/2017/01/the-future-of-air-superiority-part-iii-defeating-a2ad. [Accessed 24 September 2018].

[71] Dutch Safety Board;, “Dutch safety board: Buk surface-to-air missile system caused MH17 crash,” Onderzoeksraad, [Online]. Available: https://www.onderzoeksraad.nl/en/page/6932/dutch-safety-board-buk-surface-to-air-missile-system-caused-mh17-crash. [Accessed 22 September 2019].

[72] D. L. Josephs, “Iran missile shot down Ukraine-bound Boeing airliner, officials say,” CNBC, [Online]. Available: https://www.cnbc.com/2020/01/09/trump-says-he-has-doesnt-believe-the-boeing-plane-crash-in-iran-was-due-to-mechanical-error.html. [Accessed 15 April 2020].

[73] L. K. Privratsky, Logistics in the Falklands war, UK: Pen and Sward Military Ltd, 2014, p. 123.

[74] B. Jose and P. Medie, “Civilian self-protection and civilian targeting in armed conflicts: who protects civilians?,” oxfordre.com, [Online]. Available:

https://oxfordre.com/politics/view/10.1093/acrefore/9780190228637.001.0001/acrefore-9780190228637-e-216. [Accessed 23 May 2020].

[75] L. Greenemeier, "20 Years after deep blue: How AI has advanced since conquering chess," Scientific American, [Online]. Available: https://www.scientificamerican.com/article/20-years-after-deep-blue-how-ai-has-advanced-since-conquering-chess/. [Accessed 26 May 2020].

[76] E. Verani, "Symbolic AI vs machine learning in NLP (natural language processing)," Inbenta, [Online]. Available: https://www.inbenta.com/en/blog/symbolic-ai-vs-machine-learning/. [Accessed 26 May 2020].

[77] G. Bologna, "A simple convolutional neural network with rule extraction," *Applied Sciences,* vol. 9, no. 12, p. 2411, 2019.

[78] C. Nicholson, "Symbolic reasoning (symbolic AI) and machine learning," Pathmind, [Online]. Available: https://pathmind.com/wiki/symbolic-reasoning. [Accessed 26 May 2020].

[79] G. Lawton, "Neuro-symbolic AI seen as evolution of artificial intelligence," SearchEnterpriseAI, [Online]. Available: https://searchenterpriseai.techtarget.com/feature/Neuro-symbolic-AI-seen-as-evolution-of-artificial-intelligence. [Accessed 26 May 2020].

[80] H. Rauch, "Probability concepts for an expert system used for data fusion," *AI Magazine,* vol. 5, no. 3, pp. 55-60, 1984.

[81] R. Varadaraju, "A survey of introducing artificial intelligence into the safety critical system software design process," *Univ. of Northern Iowa,* 2011.

[82] M. Bohanec and V. Rajkovič, "DEX: an expert system shell for decision support," *Sistemica,* vol. 1, no. 1, pp. 145-157, 1990.

[83] M. G. Voskoglou, "Measuring the uncertainty of human reasoning," *American Journal of Applied Mathematics and Statistics,* vol. 2, no. 1, pp. 1-6, 2014.

[84] P. Johnson-Laird, "Mental models and human reasoning," *Peer Reviewed Proc. of the National Academy of Sciences,* vol. 107, no. 43, pp. 18243-18250, 2010.

[85] C. D. Cooke, C. A. Santana, T. I. Morris, L. DeBraal, C. Ordonez, E. Omiecinski, N. F. Ezquerra and E. V. Garcia, "Validating expert system rule confidences using data mining of myocardial perfusion SPECT databases," *Computers in Cardiology 2000,* vol. 27 (Cat. 00CH37163), pp. 785-788, 2000.

[86] R. Barzilay, D. McCullough, O. Rambow, J. DeCristofaro, T. DeCristofaro and B. Lavoie, "A new approach to expert system explanations," *International Workshop on Natural Language Generation,* 1998.

[87] RecordedFuture, "What the 6 Phases of the Threat Intelligence Lifecycle Mean for Your Team," RecordedFuture, [Online]. Available: https://www.recordedfuture.com/threat-intelligence-lifecycle-phases. [Accessed 15 January 2020].

[88] A. Khalak and E. Wemhoff, "Multi-hypothesis estimation approach to diagnosis and prognosis of degrading systems," in *Peer Reviewed Proc. of the IEEE Aerospace Conference,* 2005.

[89] D. Spiegelhalter, A. Dawid, S. Lauritzen and R. Cowell, "Bayesian analysis in expert systems," *Statistical Science,* vol. 8, no. 3, pp. 219-247, 1993.

[90] L. Breiman, “Random forests,” *Machine learning,* vol. 45, no. 1, pp. 5-32, 2001.

[91] G. Biau, “Analysis of a random forests model,” *The Journal of Machine Learning Research,* vol. 13, pp. 1063-1095, 2012.

[92] T. Shaikhina, D. Lowe, S. Daga, D. Briggs, R. Higgins and N. Khovanova, “Decision tree and random forest models for outcome prediction in antibody incompatible kidney transplantation,” *Biomedical Signal Processing and Control,* vol. 52, pp. 456-462, 2019.

[93] N. Ernest and D. Carroll, “Genetic fuzzy based artificial intelligence for unmanned combat aerial vehicle control in simulated air combat missions,” *Journal of Defense Management,* vol. 6, no. 1, 2016.

[94] R. Freitas Jr, T. Healy and J. Long, “Advanced automation for space missions,” *The Journal of the Astronautical Sciences,* vol. 1, no. 1, pp. 1-11, 1982.

[95] D. Lawson and M. James, “Sharp: a multi-mission AI system for spacecraft telemetry monitoring and diagnosis,” *Telematics and Informatics,* vol. 6, no. 3-4, pp. 221-236, 1989.

[96] J. Brownlee, “10 Examples of how to use statistical methods in a machine learning project,” Machine Learning Mastery, [Online]. Available: https://machinelearningmastery.com/statistical-methods-in-an-applied-machine-learning-project/. [Accessed 26 May 2020].

[97] N. Kote, M. Biba and E. Canaj, “Bayesian networks: a state-of-the-art survey,” in *Peer Reviewed Proc. of the 3rd International Conference on Recent Trends and Applications in Computer Science and Information Technology*, 2018.

[98] D. Soni, “Introduction to Bayesian networks,” Medium, [Online]. Available: https://towardsdatascience.com/introduction-to-bayesian-networks-81031eeed94e. [Accessed 27 May 2020].

[99] A. Siahkoohi, G. Rizzuti and F. Herrmann, “A Deep-Learning Based Bayesian Approach to Seismic Imaging and Uncertainty Quantification,” *Conferece Proc. of the EAGE 2020 Annual Conference & Exhibition Online,* vol. 2020, no. 1, pp. 1-5, 2020.

[100] R. Gandhi, “Naive Bayes classifier,” Medium, [Online]. Available: https://towardsdatascience.com/naive-bayes-classifier-81d512f50a7c>. [Accessed 27 May 2020].

[101] N. B. Amor, S. Benferhat and Z. Elouedi, “Naive Bayes vs decision trees in intrusion detection systems,” in *Peer Reviewed Proc. of the ACM symposium on Applied computing*, 2004.

[102] A. Dejeu, “'What is a Markov model' to 'here is how Markov models work,” Hackernoon.com, [Online]. Available: https://hackernoon.com/from-what-is-a-markov-model-to-here-is-how-markov-models-work-1ac5f4629b71. [Accessed 27 May 2020].

[103] G. Manogaran, V. Vijayakumar, R. Varatharajan, P. M. Kumar, R. Sundarasekar and C. H. Hsu, “Machine learning based big data processing framework for cancer diagnosis using hidden Markov model and GM clustering,” *Wireless Personal Communications,* vol. 102, no. 3, pp. 2099-2116, 2017.

[104] K. Haeussler, A. Van-Den-Hout and H. Baio, “A dynamic Bayesian Markov model for health economic evaluations of interventions in infectious disease,” *BMC Medical Research Methodology,* vol. 18, no. 1, 2018.

[105] M. H. Katz, Multivariable analyses: a practical guide for clinicians, 2nd ed., Cambridge,

UK: Cambridge Univ. Press, 2006, pp. 38-44.

[106] J. Brownlee, “Linear regression for machine learning,” Machine learning mastery, [Online]. Available: https://machinelearningmastery.com/linear-regression-for-machine-learning/. [Accessed 27 May 2020].

[107] M. Prosperi, A. Altmann, M. Rosen-Zvi, E. Aharoni, G. Borgulya, F. Bazso, A. Sonnerborg, E. Schulter, D. Struck, G. Ulivi, A. Vandamme, J. Vercauteren and M. Zazzi, “Investigation of expert rule bases, logistic regression and non-linear machine learning techniques for predicting response to antiretroviral treatment,” *Antiviral Therapy*, vol. 14, pp. 433-442, 2009.

[108] Z. Sun, G. Pedretti, A. Bricalli and D. Ielmini, “One-step regression and classification with cross-point resistive memory arrays,” *Science Advances*, vol. 6, no. 5, p. eaay2378, 2020.

[109] A. Walia, “Generalized additive models,” DataScience+, [Online]. Available: https://datascienceplus.com/generalized-additive-models/. [Accessed 2 August 2020].

[110] Y. Kida, “Generalized linear models,” Medium, [Online]. Available: https://towardsdatascience.com/generalized-linear-models-9cbf848bb8ab. [Accessed 2 August 2020].

[111] S. Wood, Y. Goude and S. Shaw, “Generalized additive models for large data sets,” *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, vol. 64, no. 2, pp. 139-155, 2014.

[112] J. Shadbolt and J. G. Taylor, Neural networks and the financial markets: predicting, combining and portfolio optimisation, London, UK: Springer Science & Business Media, 2002, pp. 57-59.

[113] A. Chandar, “Concept of regularization,” Medium, [Online]. Available: https://towardsdatascience.com/concept-of-regularization-28f593cf9f8c. [Accessed 1 August 2020].

[114] S. Prabhakaran, “Loess regression and smoothing with R,” R-statistics.co, [Online]. Available: http://r-statistics.co/Loess-Regression-With-R.html. [Accessed 27 May 2020].

[115] Nzumel, “When cross-validation is more powerful than regularization,” Win Vector LLC, [Online]. Available: https://win-vector.com/2019/11/12/when-cross-validation-is-more-powerful-than-regularization/. [Accessed 1 August 2020].

[116] L. Rangarajan and P. Nagabhushan, “Linear regression for dimensionality reduction and classification of multi dimensional data,” in *Peer Reviewed Proc. of the International Conference on Pattern Recognition and Machine Intelligence. PReMI 2005, Lecture Notes in Computer Science*, Berlin, 2005.

[117] K. Kim, “Ridge regression for better usage,” Medium, [Online]. Available: https://towardsdatascience.com/ridge-regression-for-better-usage-2f19b3a202db. [Accessed 1 August 2020].

[118] S. Bhattacharyya, “Ridge and Lasso regression: L1 and L2 regularization,” Medium, [Online]. Available: https://towardsdatascience.com/ridge-and-lasso-regression-a-complete-guide-with-python-scikit-learn-e20e34bcbf0b. [Accessed 1 August 2020].

[119] A. Mandal, “Ridge, lasso and elastic net regularization,” Medium, [Online]. Available: https://towardsdatascience.com/ridge-lasso-and-elastic-net-regularization-7861ca575c64. [Accessed 1 August 2002].

[120] A. Rauschenberger, E. Glaab and M. Van-de-Wiel, “Predictive and interpretable models via the stacked elastic net,” *Bioinformatics,* 2020.

[121] O. Harrison, “Machine learning basics with the K-nearest neighbour algorithm,” Medium, [Online]. Available: https://towardsdatascience.com/machine-learning-basics-with-the-k-nearest-neighbors-algorithm-6a6e71d01761. [Accessed 1 August 2020].

[122] C. Patlolla, “Understanding the concept of hierarchical clustering technique,” Medium, [Online]. Available: https://towardsdatascience.com/understanding-the-concept-of-hierarchical-clustering-technique-c6e8243758ec. [Accessed 1 August 2020].

[123] H. Nguyen, X. Bui, Q. Tran and N. Mai, “A new soft computing model for estimating and controlling blast-produced ground vibration based on hierarchical k-means clustering and cubist algorithms,” *Applied Soft Computing,* vol. 77, pp. 376-386, 2019.

[124] C. Chang and C. Lin, “LIBSVM. a library for support vector machines,” *ACM Transactions on Intelligent Systems and Technology,* vol. 2, no. 3, pp. 1-27, 2011.

[125] R. Batuwita and V. Palade, “FSVM-CIL: fuzzy support vector machines for class imbalance learning,” *IEEE Transactions on Fuzzy Systems,* vol. 18, no. 3, pp. 558-571, 2010.

[126] P. Gupta, “Decision trees in machine learning,” Medium, [Online]. Available: https://towardsdatascience.com/decision-trees-in-machine-learning-641b9c4e8052. [Accessed 1 August 2020].

[127] A. Sharma, “4 Simple Ways to Split a Decision Tree in Machine Learning,” Analytics Vidhya, 12 July 2020. [Online]. Available: https://www.analyticsvidhya.com/blog/2020/06/4-ways-split-decision-tree/. [Accessed 12 Aug 2022].

[128] K. Vala, “Tree-based methods: regression trees,” Medium, [Online]. Available: https://towardsdatascience.com/tree-based-methods-regression-trees-4ee5d8db9fe9. [Accessed 1 August 2020].

[129] N. Chauhan, “Introduction to artificial neural networks (ANN),” Medium, [Online]. Available: https://towardsdatascience.com/introduction-to-artificial-neural-networks-ann-1aea15775ef9. [Accessed 1 August 2020].

[130] M. Abdella and T. Marwala, “The use of genetic algorithms and neural networks to approximate missing data in database,” *IEEE 3rd International Conference on Computational Cybernetics, ICCC 2005,* Vols. 17, No 3, no. 3, pp. 558-571, 2005.

[131] M. Pratiwi, Alexander, J. Harefa and S. Nanda, “Mammograms classification using gray-level co-occurrence matrix and radial basis function neural network,” *Procedia Computer Science,* vol. 59, pp. 83-91, 2015.

[132] Z. Kurd, T. Kelly and J. Austin, “Developing artificial neural networks for safety critical systems,” *Neural Computing and Applications,* vol. 16, no. 1, pp. 11-19, 2006.

[133] G. Zhang, “Neural networks for classification: a survey,” *IEEE Transactions on Systems, Man, and Cybernetics,* vol. 30, no. 4, pp. 451 - 462, 2000.

[134] J. Schumann, P. Gupta and S. Nelson, “On verification & validation of neural network based controllers,” in *Peer Reviewed Proc. of the Engineering Applications of Neural Networks (EANN),* 2003.

[135] J. Hull, D. Ward and R. Zakrzewski, “Verification and validation of neural networks for safety-critical applications,” in *Peer Reviewed Proc. of the American control conference*

(IEEE Cat. No.CH37301), 2002.

[136] S. Tan and M. Mayrovouniotis, "Reducing data dimensionality through optimizing neural network inputs," *AIChE Journal,* vol. 41, no. 6, pp. 1471-1480, 1995.

[137] H. Fry, Hello world: how to be human in the age of the machine, New York, USA: W.W Norton & company, 2018, p. 87.

[138] T. GopiKrishna, "Evaluation of rule extraction algorithms," *International Journal of Data Mining & Knowledge Management Process (IJDKP),* vol. 4, no. 3, pp. 9-19, 2014.

[139] T. Hailesilassie, "Rule extraction algorithm for deep neural networks: a review," *International Journal of Computer Science and Information Security (IJCSIS),* vol. 14, no. 7, pp. 376-381, 2019.

[140] S. Thrun, "Extracting rules from artificial neural networks with distributed representations," *Advances in Neural Information Processing Systems (volume 7),* vol. 7, 1995.

[141] A. Baydin, R. Cornish, D. Rubio, M. Schmidt and F. Wood, "Online learning rate adaptation with hypergradient descent," *arXiv Preprint, arXiv:1703.04782,* 2019.

[142] J. Bergstra and Y. Bengio, "Random search for hyper-parameter optimization," *Journal of machine learning research,* vol. 13, pp. 281-305, 2012.

[143] J. Bergstra, R. Bardenet, Y. Bengio and B. Kegl, "Algorithms for hyper-parameter optimization," in *Peer Reviewed Proc. of the Advances Neural Information and Processing Systems,* 2011.

[144] P. Probst, A. L. Boulesteix and B. Bischl, "Tunability: Importance of hyperparameters of machine learning algorithms," *Journal of Machine Learning Research,* vol. 20, no. 53, pp. 1-32, 2019.

[145] I. Loshchilov and H. Hutter, "Online batch selection for faster training of neural networks," *arXiv Preprint, arXiv:1511.06343,* 2015.

[146] I. Misra, L. Zitnick and M. Hebert, "Shuffle and learn: unsupervised learning using temporal order verification," in *Peer Reviewed Proc. of the European Conference on Computer Vision,* Cham, Switzerland, 2016.

[147] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from over-fitting," *Journal of Machine Learning Research,* vol. 15, no. 1, pp. 1929-1958, 2014.

[148] M. Zeiler and R. Fergus, "Stochastic pooling for regularization of deep convolutional neural networks," *arXiv Preprint, arXiv:1301.3557,* 2013.

[149] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science,* vol. 313, no. 5786, pp. 504-507, 2006.

[150] Y. LeCun, C. Cortes and C. J. Burges, "The MNIST database of handwritten digits," yann.lecun.com, [Online]. Available: http://yann.lecun.com/exdb/mnist/. [Accessed 9 April 2022].

[151] J. Lin, F. Chen and D. Wang, "Data compression based on stacked RBM-AE model for wireless sensor networks," *Sensors,* vol. 18, no. 4273, 2018.

[152] S. Khan, "Towards synthetic dataset generation for semantic segmentation networks," *MASc Thesis, Electrical and Computer Engineering Depart., Univ. of Waterloo,* 2019.

[153] S. Khan, B. Phan, R. Salay and K. Czarnecki, “ProcSy: procedural synthetic dataset generation towards influence factor studies of semantic segmentation networks,” in *Peer Reviewed Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2019.

[154] R. Fisher, “Cvonline: image databases,” Homepages.inf.ed.ac.uk, [Online]. Available: http://homepages.inf.ed.ac.uk/rbf/CVonline/Imagedbase.htm. [Accessed 8 June 2020].

[155] A. Sterling, J. Wilson, S. Lowe and M. Lin, “ISNN: impact sound neural network for audio-visual object classification,” in *Peer Reviewed Proc. of the European Conference on Computer Vision (ECCV)*, 2018.

[156] O. Vinyals, L. Kaiser, T. Koo, S. Petrov, I. Sutskever and G. Hinton, “Grammar as a foreign language,” in *Peer Reviewed Proc. of the Advances in Neural Information Processing Systems 28 (NIPS 2015)*, 2015.

[157] N. Abdellaoui, P. Hubbard and P. Duncan, “An enhanced synthetic environment for maritime surveillance,” Apps.dtic.mil, October 2005. [Online]. Available: https://apps.dtic.mil/dtic/tr/fulltext/u2/a640449.pdf. [Accessed 1 May 2020].

[158] N. Petrov, I. Jordanov and J. Roe, “Radar emitter signals recognition and classification with feedforward networks,” *Procedia Computer Science 2013*, vol. 22, pp. 1192-1200, 2013.

[159] A. Kar, A. Prakash, M. Liu, E. Cameracci, J. Yuan, N. Rusiniak, D. Acuna, A. Torralba and S. Fidler, “Meta-sim: learning to generate synthetic datasets,” in *Peer Reviewed Proc. of the IEEE International Conference on Computer Vision*, 2019.

[160] D. Bau, J. Zhu, H. Strobelt, B. Zhou, J. Tenenbaum, W. Freeman and A. Torralba, “GAN dissection: visualizing and understanding generative adversarial networks,” arXiv Preprint, arXiv:1811.10597, 2018.

[161] T. Karras, S. Laine and T. Aila, “A style-based generator architecture for generative adversarial networks,” in *Peer Reviewed Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[162] X. Yao, H. Yang and Y. Li, “Modulation identification of underwater acoustic communications signals based on Generative Adversarial Networks,” in *Peer Reviewed Proc. of the OCEANS 2019 - Marseille*, Marseille, France, 2019.

[163] C. Zhang, X. Yang, Y. Tang and W. Zhang, “Learning to generate radar image sequences using two-stage generative adversarial networks,” *IEEE Geoscience and Remote Sensing Letters*, Vols. 17, no. 3, pp. 401-405, 2020.

[164] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville and Y. Bengio, “Generative adversarial nets,” in *NIPS*, 2014.

[165] I. J. Goodfellow, “On distinguishability criteria for estimating generative models,” *arXiv Preprint, arXiv:1412.6515*, 2014.

[166] S. N. Esfahani and S. Latifi, “A survey of state-of-the-art GAN-based approaches to image synthesis,” in *Peer Reviewed Proc, of the 9th International Conference on Computer Science, Engineering and Applications (CCSEA 2019)*, 2019.

[167] S. Reed, Z. Akata, X. Yan, L. Logeswaran, B. Schiele and H. Lee, “Generative Adversarial Text to Image Synthesis,” *Peer Reviewed Proc. of The 33rd International Conference on Machine Learning*, vol. 48, pp. 1060--1069, 2016.

[168] W. Lotter, G. Kreiman and D. Cox, “Unsupervised learning of visual structure using

predictive generative networks," *arXiv Preprint, arXiv:1511.06380,* 2015.

[169] Q. Hoang, T. D. Nguyen, T. Le and D. Phung, "Multi-generator generative adversarial nets," *arXiv Preprint, arXiv:1708.02556,* 2017.

[170] D. M. Blei, A. Y. Ng and M. I. Jordan, "Latent Dirichlet allocation," *Journal of Machine Learning Research,* vol. 3, pp. 993-1022, 2003.

[171] D. M. Blei, A. Y. Ng and M. I. Jordab, "Latent Dirichlet Allocation," *Journal of Machine Learning Research,* vol. 3, no. Jan, pp. 993-1022, 2003.

[172] J. Schmidhuber, "Generative adversarial networks are special cases of artificial curiosity (1990) and also closely related to predictability minimization (1991)," *Neural Networks,* vol. 127, pp. 58-66, 2020.

[173] K. Weiss, T. M. Khoshgoftaar and D. Wang, "A survey of transfer learning," *Journal of Big Data,* vol. 3, no. 9, 2016.

[174] C. Tan, F. Sun, T. Kong, W. Zhang, C. Yang and C. Liu, "A survey on deep transfer learning," in *Peer Reviewed Proc. of the International Conference on Artificial Neural Networks. ICANN 2018,* Cham Switzerland, 2018.

[175] Z. Huang, Z. Pan and B. Lei, "Transfer learning with deep convolutional neural network for SAR target classification with limited labelled data," *Remote Sensing 2017,* vol. 9, no. 9, p. 907, 2017.

[176] W. Zhu, M. Li, W. Chen and X. Ran, "Radar emitter recognition based on transfer learning," *DEStech Transactions on Computer Science and Engineering, (csae),* pp. 838-844, 2017.

[177] T. Hospedales, A. Antoniou, P. Micaelli and A. Storkey, "Meta-learning in neural networks: a survey," *arXiv Preprint, arXiv:2004.05439,* 2020.

[178] E. Real, "Using evolutionary autoML to discover neural network architectures," Google AI Blog, [Online]. Available: https://ai.googleblog.com/2018/03/using-evolutionary-automl-to-discover.html. [Accessed 21 August 2020].

[179] Y. Lucun, L. Bottou, Y. Bengio and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE,* vol. 86, no. 11, pp. 2278 - 2324, 1998.

[180] A. Krizhevsky, A. Sutskever and G. Hinton, "ImageNet classification with deep convolutional neural networks," *Advances in neural information processing systems 25 (2012),* pp. 1097-1105, 2012.

[181] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," in *arXiv preprint arXiv:1409.1556,* 2014.

[182] M. Lin, Q. Chen and S. Yan, "Network In Network," in *arXiv:1312.4400v3,* 2013.

[183] S. Christian, L. Wei, J. Yangqing, S. Pierre, R. Scott, A. Dragomir, E. Dumitru, V. Vincent and R. Andrew, "Going Deeper With Convolutions," in *In Peer Reviewed Proc. of the IEEE conference on computer vision and pattern recognition,* 2015.

[184] H. Kaiming, Z. Xiangyu, R. Shaoqing and S. Jian, "Deep Residual Learning for Image Recognition," in *Peer Reviewed Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR),* 2016.

[185] H. Gao, D. Chen, T. Li, L. Van Der Maaten and K. Q. Weinberger, "Multi-scale dense convolutional networks for efficient prediction," in *arXiv preprint arXiv:1703.09844,* 2017.

[186] J. Hu, L. Shen and G. Sun, "Squeeze-and-excitation networks," in *Peer Reviewed Proc.of the IEEE conference on computer vision and pattern recognition*, 2018.

[187] X. Zhang, X. Zhou, M. Lin and J. Sun, "ShuffleNet: An Extremely Efficient Convolutional Neural Network for Mobile Devices," in *Peer Reviewed Proc.of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[188] D. Sinha and E.-S. Mohamed, "Thin MobileNet: An Enhanced MobileNet Architecture," in *Peer Reviewed Proc. of IEEE 10th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON)*, 2019.

[189] D. Soni, "Introduction to evolutionary algorithms," Medium, [Online]. Available: https://towardsdatascience.com/introduction-to-evolutionary-algorithms-a8594b484ac. [Accessed 1 August 2020].

[190] Z. Michalewicz, "Evolutionary programming and genetic programming," in *Genetic Algorithms + Data Structures = Evolution Programs*, 3rd ed., New York, USA, Springer-Verlag Berlin Heidelberg GmbH, 1996, pp. 283-287.

[191] D. Camara, "Evolution and evolutionary algorithms," in *Bio-Inspired Networking*, London, UK, ISTE Press Ltd, 2015, pp. 1-30.

[192] K. Sörensen and F. Glover, "Metaheuristics," in *Encyclopaedia of Operations Research and Management Science 62*, 2013, pp. 960-970.

[193] C. R. Reeves, "Fitness landscapes and evolutionary algorithms," in *Peer Reviewed Proc. of European Conference on Artificial Evolution*, 1999.

[194] M. Boulif, "Heterogeneous parallel genetic algorithm paradigm," *arXiv Preprint, arXiv:1905.06636,* 2019.

[195] P. Ozturk and A. Tidemann, "A review of case based reasoning in cognition-action continuum: a step towards bridging symbolic and non-symbolic artificial intelligence. in press.," *The Knowledge Engineering Review,* vol. 20, no. 1, pp. 51-77, 2014.

[196] P. Oztürk and A. Tidemann, "A review of case-based reasoning in cognition–action continuum: a step toward bridging symbolic and non-symbolic artificial intelligence," *The Knowledge Engineering Review,* vol. 29, no. 1, p. 51–77, 2014.

[197] A. Aamodt and E. Plaza, "Case-based reasoning: foundational issues, methodological variations, and system approaches," *AI Communications,* vol. 7, no. 1, pp. 39-59, 1994.

[198] B. Smyth and M. T. Keane, "Experiments on adaptation-guided retrieval in case-based design," in *Peer Reviewed Proc. of the International Conference on Case-Based Reasoning ICCBR*, 1995.

[199] N. Löw, J. Hesser and M. Blessing, "Multiple retrieval case-based reasoning for incomplete datasets," *Journal of Biomedical Informatics,* vol. 92, pp. 103-127, 2019.

[200] D. L. Poole and A. K. Mackworth, "7 Learning: overview and supervised learning," in *Artificial Intelligence: Foundations of Computational Agents*, 2nd ed., New York, USA, Cambridge Univ. Press, 2019, pp. 288-287.

[201] S. T. Li and H. F. Ho, "Predicting financial activity with evolutionary fuzzy case-based reasoning," *Expert systems with Applications,* vol. 36, no. 1, pp. 411-422, 2009.

[202] R. Adler, M. N. Akram, P. Bauer, P. Feth, P. Gerber, A. Jedlitschka, L. Jöckel, M. Kläs and D. Schneider, "Hardening of Artificial Neural Networks for Use in Safety-Critical Applications -- A Mapping Study," *arXiv:1909.03036v1,* 2019.

[203] A. Bryman, Social research methods, 4th ed., Oxford, UK: Oxford Univ. Press, 2012.

[204] J. Cushing, “Aristotle and Francis Bacon,” in *Philosophical concepts in physics: the historical relation between philosophy and scientific theories*, Cambridge, UK, Cambridge Univ. Press, 1998, pp. 15-28.

[205] J. Gill and P. Johnson, Research methods for managers, 4th ed., London, UK: Sage Publication Ltd, 2011.

[206] N. Colin and M. Moruzzis, “Radar target recognition by fuzzy logic,” in *Peer Reviewed Proc. of the 1997 IEEE National Radar Conference*, Syracuse, NY, USA, 1997.

[207] N. Egger, J. E. Ball and J. Rogers, “Radar angle of arrival system design optimization using a genetic algorithm,” *Electronics,* vol. 6, no. 1, p. 24, 2017.

[208] S. Unnithan, “S-400: the geopolitical missile,” India Today, 6 August 2018. [Online]. Available: https://www.indiatoday.in/magazine/the-big-story/story/20180813-s-400-the-geopolitical-missile-1303340-2018-08-06. [Accessed 26 March 2022].

[209] M. Sugars, “What is an instrument flight rules (IFR) airport?,” Avlite, 17 February 2021. [Online]. Available: https://www.avlite.com/blog/2021/02/17/what-is-an-instrument-flight-rules-ifr-airport/#:~:text=Instrument%20Flight%20Rules%20(IFR)%20are,piloting%20and%20non%2Dvisual%20runways.. [Accessed 30 July 2022].

[210] M. Sugars, “What is a visual flight rules (VFR) airfiekd?,” Avlite, 10 February 2021. [Online]. Available: https://www.avlite.com/blog/2021/02/10/what-is-a-visual-flight-rules-vfr-airfield/. [Accessed 30 July 2022].

[211] P. Mukharjee, “NAVTEX On Ships: Working, Types Of Messages And Advantages,” Marine insight, 9 July 2021. [Online]. Available: https://www.marineinsight.com/marine-navigation/navtex-on-ships/. [Accessed 2021 July 30].

[212] MI News Network;, “What are Vessel Traffic Services?,” Marine insight, 6 January 2021. [Online]. Available: https://www.marineinsight.com/marine-navigation/what-are-vessel-traffic-services/. [Accessed 2022 July 30].

[213] S. Dickerson and D. N. Mavris, Architecture and principles of systems engineering, London, UK: CRC Press, 2009, p. 179.

[214] Wikipedia, “The Egyptian SA-2 surface-to-air Fan Song E fire control radar of the SA-2 system, in operation during the multinational joint service exercise bright star '85,” Wikipedia, [Online]. Available: https://en.wikipedia.org/wiki/Fan_Song#/media/File:Fan_Song_fire_control_radar_of_the_SA-2_SAM-system.JPEG. [Accessed 17 10 2018].

[215] GlobalSecurity, “Fan Song,” GlobalSecurity, [Online]. Available: https://www.globalsecurity.org/military/world/russia/fan-song.htm. [Accessed 1 10 2018].

[216] Wikipedia, “Fan Song,” Wikipedia, [Online]. Available: en.wikipedia.org/wiki/Fan_Song . [Accessed 1 10 2018].

[217] Wikipedia, “S-75 devina,” Wikipedia, [Online]. Available: en.wikipedia.org/wiki/S-75_Dvina. [Accessed 1 10 2018].

[218] FAS, “V-75 SA-2 guideline,” FAS, [Online]. Available: https://nuke.fas.org/guide/russia/airdef/v-75.htm. [Accessed 1 10 2018].

[219] ArmyRecognition, “Sayyad-1 ground-to-air missile system SAM technical data,”

ArmyRecognition, 23 11 2011. [Online]. Available: https://www.armyrecognition.com/iran_iranian_army_missile_systems_vehicles_uk/sayyad-1_ground-to-air_missile_system_sam_technical_data_sheet_specifications_description_pictures.html. [Accessed 1 10 2018].

[220] C. Kopp, “Engagement and fire control radars,” Air Power Ausralia, 27 1 2014. [Online]. Available: http://www.ausairpower.net/APA-Engagement-Fire-Control.html. [Accessed 1 10 2018].

[221] C. Kopp, “Almaz S-75 dvina/desna/volkhov,” Air Power Australia, 05 2012. [Online]. Available: http://www.ausairpower.net/APA-S-75-Volkhov.html. [Accessed 1 10 2018].

[222] C. Kopp and M. Gyűrösi, “SNR-75M3 Fan Song E engagement radar,” Air Power Australia, 05 2021. [Online]. Available: http://www.ausairpower.net/APA-SNR-75-Fan-Song.html. [Accessed 1 10 2018].

[223] GlobalSecurity, “Fan Song,” GlobalSecurity, [Online]. Available: https://www.globalsecurity.org/military/world/russia/fan-song.htm. [Accessed 1 10 2018].

[224] MilitaryFactory, “SA-2 (guideline) / S-75 dvina,” MilitaryFactory, 14 11 2017. [Online]. Available: https://www.militaryfactory.com/armor/detail.php?armor_id=133. [Accessed 1 10 2018].

[225] A. Gammerman, V. Vovk and V. Vapnik, “Learning by transduction,” *arXiv Preprint arXiv:1301.7375,* 2013.

[226] M. Abdella and T. Marwala, “The use of genetic algorithms and neural networks to approximate missing data in database,” in *Pear Reviewed Proc. of the IEEE 3rd International Conference on Computational Cybernetics ICCC*, Mauritius,, 2005.

[227] W. Wiegerinck, B. Kappen and W. Burgers, “Interactive collaborative information systems,” in *Bayesian networks for expert systems: Theory and practical applications*, Heidelberg, Germany, Springer, Berlin, 2010, pp. 547-578.

[228] A. Khalak and E. Wemhoff, “A multi-hypothesis estimation approach to diagnosis and prognosis of degrading systems,” in *Peer Reviewed Proc. of the 2005 IEEE Aerospace Conference,* Big Sky, USA, 2005.

[229] S. J. Roberts, “Intelligence Concepts—The Intelligence Cycle,” @srobers, 16 December 2015. [Online]. Available: https://sroberts.io/posts/intelligence-concepts-the-intelligence-cycle/. [Accessed 2022 July 30].

[230] DSIAC, “Radio frequency, directed energy weapon design tool,” Defence Systems Information Analysis Centre, 2021. [Online]. Available: https://dsiac.org/articles/radio-frequency-directed-energy-weapon-design-tool/. [Accessed 22 11 2021].

[231] F. Tamburini, E. Mari, A. Sponselli, B. Thidé, A. Bianchini and F. Romanato, “Encoding many channels on the same frequency through radio vorticity: first experimental test,” *New Journal of Physics,* vol. 14, no. March, p. 033001, 2021.

[232] A. Greve, C. Kramer and W. Wild, “The beam pattern of the IRAM 30–m telescope,” *Astronomy & Astrophysics Supplement Series,* vol. 133, p. 2710284, 1998.

[233] J. Donlop and D. G. Smith, “Antennas,” in *Telecommunications Engineering,* 3rd, Ed., Boca Raton, US, CRC Press, 1997, p. Equation 7.26.

[234] S. Mussmann and S. Ermon, “Learning and inference via maximum inner product search,”

in *Preer Reviewed Proc. of the International Conference on Machine Learning*, 2016.

[235] G. Karunaratne, M. Schmuck, M. Le Gallo, G. Cherubini, L. Benini, A. Sebastian and A. Rahimi, "Robust high-dimensional memory-augmented neural networks," *Nature Communications,* vol. 12, no. 2468, 2021.

[236] M. M. Arat, "Weight initialization schemes - Xavier (Glorot) and He," Mustafa Murat ARAT, 25 February 2019. [Online]. Available: https://mmuratarat.github.io/2019-02-25/xavier-glorot-he-weight-init. [Accessed 23 December 2021].

[237] Q. Wang and Z. Shen, "Control comparison of inverted pendulum system based on PID," in *Peer Reviewed Proc. of the IOP Conference Series: Materials Science and Engineering*, 2018.

[238] P. J. Navarathna and V. P. Malagi, "Artificial intelligence in smart city analysis," in *Peer Reviewed Proc. of the International Conference on Smart Systems and Inventive Technology (ICSSIT)s*, Tirunelveli, India, 2018.

[239] J. T. De Souza, A. C. De Francisco, C. M. Piekarski and G. F. Do Prado, "Data mining and machine learning to promote smart cities: a systematic review from 2000 to 2018," *Sustainability,* vol. 11, no. 4, p. 1077, 2019.

[240] S. Srivastava, A. Bisht and N. Narayan, "Safety and security in smart cities using artificial intelligence — a review," in *Peer Reviewed Proc. of the 2017 7th International Conference on Cloud Computing, Data Science & Engineering - Confluence*, Noida, India, 2017.

[241] J. C. Knight, "Safety critical systems: challenges and directions," in *Peer Reviewed Proc. of the ICSE '02: Proceedings of the 24th International Conference on Software Engineering*, Orlando Florida, USA, 2002.

[242] A. Serban, "Designing safety critical software systems to manage inherent uncertainty," in *Peer Reviewed Proc. of the IEEE International Conference on Software Architecture Companion (ICSA-C)*, Hamburg, Germany, 2019.

[243] P. Carpenter, "Verification of requirements for safety-critical software," in *Peer Reviewed Proc. of the SIGAda`99 10/99*, Redondo Beach, CA, USA, 1999.

[244] M. Borg, C. Englund, W. Krzysztof, B. Duran, C. Levandowski, S. Gao, Y. Tan, H. Kaijser, H. Lönn and J. Törnqvist, "Safely Entering the Deep: A Review of Verification and Validation for Machine Learning and a Challenge Elicitation in the Automotive Industry," arXiv Preprint, arXiv:1812.05389, 2018.

[245] J. Ding, X.-H. Hu and V. Gudivada, "A machine learning based framework for verification and validation of massive scale image data," *IEEE Transactions on Big Data,* vol. 7, no. 2, pp. 451-467, 2017.

[246] I. J. Rudas and L. Horvath, "Modeling man-machine processes in CAD/CAM and flexible manufacturing systems," in *Peer Reviewed Proc. of the International Conference on Industrial Electronics, Control and Instrumentation*, Taipei, Taiwan, 1996.

[247] H. Chan, E. Rice, P. Vayanos, M. Tambe and M. Morton, "Evidence from the past: AI decision aids to improve housing systems for homeless youth," in *Peer Reviewed Proc. of the AAAI Fall Symposium Series*, Westin Arlington Gateway, Virginia, USA, 2017.

[248] S. D. Zetumer and H. Harris, "Identifying strangulated small bowel obstruction with machine learning," *Journal of Clinical and Translational Science*, vol. 1, no. S1:2476, pp. 19-19, 2018.

[249] S. M. Siti and S. B. Zaibon, “Decisional guidance for computerised personal decision aid (ComPDA),” in *Peer Reviewed Proc. of the Knowledge Management International Conference (KMICe) 2012*, Johor Bahru, Malaysia, 2012.

[250] M. M. Singh and P. Geyer, “Statistical decision assistance for determining energy-efficient options in building design under uncertainty,” in *Peer Reviewed Proc. of the 26th International Workshop on Intelligent Computing in Engineering*, Leuven, Belgium, 2019.

[251] S. Nosratabadi, A. Mosavi, R. Keivani, S. Ardabili and F. Aram, “State of the art survey of deep learning and machine learning models for smart cities and urban sustainability,” in *Peer Reviewed Proc. of the International Conference on Global Research and Education*, Budapest and Balatonfüred, Hungary, 2019.

[252] F. Fan and G. E. WANG, “Learning from pseudo-randomness with an artificial neural network—does god play pseudo-dice?,” *IEEE Access*, vol. 6, pp. 22987-22992, 2018.

[253] W. Duch, R. Adamczak and N. Jankowski, “Initialization and optimization of multilayered perceptrons,” in *Peer Reviewed Proc. of the 3rd Conf. Neural Networks Applications.*, 1997.

[254] V. Kakaraparthi, “Xavier and He Normal (He-et-al) initialization,” Medium, [Online]. Available: https://medium.com/@prateekvishnu/xavier-and-he-normal-he-et-al-initialization-8e3d7a087528. [Accessed 5 October 2019].

[255] A. Ananthram, “Random initialization for neural networks: a thing of the past,” towards data science, 25 February 2018. [Online]. Available: https://towardsdatascience.com/random-initialization-for-neural-networks-a-thing-of-the-past-bfcdd806bf9e. [Accessed 5 October 2019].

[256] B. Niederberger, “Set process priority in windows (python recipe),” ActiveState, 2 June 2006. [Online]. Available: https://code.activestate.com/recipes/496767-set-process-priority-in-windows/. [Accessed 2 June 2019].

[257] Shimao, “What process controls the CPU affinity of new python processes,” superuser, 2 Dec 2017. [Online]. Available: https://superuser.com/questions/1273705/what-process-controls-the-cpu-affinity-of-new-python-processes. [Accessed 2 June 2019].

[258] M. Herf, “Know your FPU,” stereopsis, 5 April 2000. [Online]. Available: http://stereopsis.com/FPU.html. [Accessed 29 January 2019].

[259] Y. Blumenfeld, D. Gilboa and D. Soudry, “Beyond signal propagation: is feature diversity necessary in deep neural network initialization?,” in *Peer Revived Proc. of the 37th International Conference on Machine Learning*, 2020.

[260] J. Knight, “Safety critical systems: challenges and directions,” in *Peer Reviewed Proc. of the International Conference on Software Engineering*, 2002.

[261] D. Połap, M. Włodarczyk-Sielicka and N. Wawrzyniak, “Automatic ship classification for a riverside monitoring system using a cascade of artificial intelligence techniques including penalties and rewards,” *ISA Transactions,* 2021.

[262] F. Ali, A. Ali, M. Imran, R. A. Naqvi, M. H. Siddiq and K. S. Kwak, “Traffic accident detection and condition analysis based on social networking data, Accident Analysis & Prevention,” *Accident Analysis & Prevention,* vol. 151, no. 105973, 2021.

[263] M. Holen, R. Saha, M. Goodwin, C. W. Omlin and K. E. Sandsmark, “Road detection for reinforcement learning based autonomous car,” in *Peer Reviewed Proc. of the ICISS 2020 The 3rd International Conference on Information Science and System*, Cambridge, UK,

2020.

[264] D. J. Fremont, J. Chiu, D. D. Margineantu, D. Osipychev and S. A. Seshia, "Formal analysis and redesign of a neural network-based aircraft taxiing system with VerifAI," *Computer Aided Verification. CAV 2020. Lecture Notes in Computer,* vol. 12224, 2020.

[265] S. Thombre, Z. Zhao, H. Ramm-Schmidt, J. M. Vallet García, T. Malkamäki, S. Nikolskiy, T. Hammarberg, H. Nuortie, M. Zahidul, H. Bhuiyan, S. Särkkä and V. V. Lehtola, "Sensors and AI techniques for situational awareness in autonomous ships: a review," *IEEE Transactions on Intelligent Transportation Systems,* pp. 1-20, 2020.

[266] M. S. Seyfioğlu and S. Z. Gürbüz, "Deep neural network initialization methods for micro-Doppler classification with low training sample support," *IEEE Geoscience and Remote Sensing Letters,* vol. 14, no. 12, pp. 2462-2466, 2017.

[267] M. Seuret, M. Alberti, M. Liwicki and R. Ingold, "PCA-initialized deep neural networks applied to document image analysis," in *Peer Reviewed Proc of the 14th IAPR International Conference on Document Analysis and Recognition (ICDAR),* 2017.

[268] H. Zhang, Y. N. Dauphin and T. Ma, "Fixup initialization: residual learning without normalization," *arXiv Preprint, arXiv:1901.09321,* 2019.

[269] K. D. Humbird, J. L. Peterson and R. G. Mcclarren, "Deep neural network initialization with decision trees," *IEEE Transactions on Neural Networks and Learning Systems,* vol. 30, no. 5, pp. 1286-1295, 2019.

[270] M. F. Ferreira, R. Camacho and L. F. Teixeira, "Autoencoders as weight initialization of deep classification networks for cancer," *BMC medical informatics and decision making,* Vols. 20, sup. 5:141, 2019.

[271] Y. Wang, Y. Rong, H. Pan, K. Liu, F. Hu, F. Wu, W. Peng, X. Xue and J. Chen, "PCA based kernel initialization for convolutional neural networks," *Data Mining and Big Data. DMBD 2020. Communications in Computer and Information,* vol. 1234, 2020.

[272] W. Ding, Y. Sun, L. Ren, H. Ju, Z. Feng and M. Li, "Multiple lesions detection of fundus images based on convolution neural network algorithm with improved SFLA," *Special Section on Deep Learning Algorithms for Internet of Medical Things,* vol. 8, pp. 97618-97631, 2020.

[273] Z. Lyu, A. A. ElSaid, J. Karns, M. W. Mkaouer and T. Desell, "An experimental study of weight initialization and Lamarckian inheritance on neuroevolution," *EvoApplications,* pp. 584-600, 2021.

[274] D. H. Hubel and T. N. Wiesel, "Receptive fields and functional architecture of monkey striate cortex," *The Journal of Physiology,* vol. 195, no. 1, pp. 215-243, 1968.

[275] D. H. Hubel and T. N. Wiesel, "Shape and arrangement of columns in cat's striate context," *The Journal of Physiology,* vol. 165, pp. 559-568, 1963.

[276] Y. LeCun, K. Kavukcuoglu and C. Farabet, "Convolutional networks and applications in vision," in *Peer Reviewed Proc. of the IEEE International Symposium on Circuits and Systems,* Paris, France, 2010.

[277] J. Masci, U. Meier, D. Cireşan and J. Schmidhuber, "Stacked convolutional auto-encoders for hierarchical feature extraction," in *Peer Reviewed Proc. of the International conference on artificial neural networks ICANN,* Espoo, Finland, 2011.

[278] J. Torres, "Convolutional neural networks for beginners using keras & tensorflow 2," towardsdatascience, 21 April 2020. [Online]. Available:

https://towardsdatascience.com/convolutional-neural-networks-for-beginners-using-keras-and-tensorflow-2-c578f7b3bf25. [Accessed 23 December 2021].

[279] Kassem, "MNIST: simple CNN keras (accuracy : 0.99)=>top 1%," Kaggle.com, 27 June 2019. [Online]. Available: https://www.kaggle.com/elcaiseri/mnist-simple-cnn-keras-accuracy-0-99-top-1. [Accessed 23 December 2021].

[280] A. Krizhevsky, I. Sutskever and G. Hinton, "ImageNet classification with deep convolutional neural networks," *NIPS,* 2012.

[281] I. Goodfellow, P. McDaniel and N. Papernot, "Making machine learning robust against adversarial inputs," *Communications of the ACM,* vol. 61, no. 7, pp. 56-66, 2018.

[282] C. Molnar, "10.4 Adversarial examples," Christophm.github.io, 19 December 2021. [Online]. Available: https://christophm.github.io/interpretable-ml-book/adversarial.html. [Accessed 24 December 2021].

[283] S. Theiler, "Implementing adversarial attacks and defenses in keras & tensorflow 2.0,," medium, 23 September 2019. [Online]. Available: https://medium.com/analytics-vidhya/implementing-adversarial-attacks-and-defenses-in-keras-tensorflow-2-0-cab6120c5715. [Accessed 2021 December 24].

[284] L. Schwinn, R. Raab and B. Eskofier, "Towards rapid and robust adversarial training with one-step attacks," *arXiv Preprint, arXiv:2002.10097,* 2020.

[285] J. E. Hoag and C. W. Thompson, "A parallel general-purpose synthetic data generator," *ACM SIGMOD Record,* vol. 36, no. 1, pp. 19-24, 2007.

[286] S. Popic, B. Pavkovic, I. Velikic and N. Teslic, "Data generators: a short survey of techniques and use cases with focus on testing," in *Peer Reviewed Proc. of the IEEE 9th International Conference on Consumer Electronics - Berlin (ICCE-Berlin),* Berlin, Germany, 2019.

[287] D. W. Greig, S. F. Yarker and C. McComb, "ARES: radar data generator for systems design and development," in *Peer Reviewed Proc. of the 2002 International Radar Conference,* Edinburgh, UK, 2002.

[288] J. Liu, H. Meng and X. Wang, "A new pulse deinterleaving algorithm based on multiple hypothesis tracking," in *Peer Reviewed Proc. of the 2009 International Radar Conference "Surveillance for a Safer World" (RADAR 2009),* Bordeaux, France, 2009.

[289] A. Erdogan, J. Lin, J. Juliano and K. George, "Pulse on pulse deinterleaving radar algorithm," in *Peer Reviewed Proc. of the IEEE Annual Computing and Communication Workshop and Conference (CCWC),* Las Vegas, NV, USA, 2020.

[290] D. Ciuonzo, G. Romano and R. Solimene, "Performance analysis of time-reversal MUSIC," *IEEE Transactions on Signal Processing,* vol. 63, no. 10, pp. 2650 - 2662, 2015.

[291] ITU, "Recommendation ITU-R M.1313 technical characteristics of marine radionavivation radars," ITU, 1997. [Online]. Available: https://www.itu.int/dms_pubrec/itu-r/rec/m/R-REC-M.1313-0-199710-S!!PDF-E.pdf. [Accessed 28 December 2021].

[292] NAVAIR, "Electronic warfare and radar systems," NAVAL AIR SYSTEMS COMMAND, 1 April 1997. [Online]. Available: https://www.isibang.ac.in/~library/onlinerz/resources/EW_Radar_Handbook.pdf. [Accessed 20 April 2020].

[293] G. E. Box and M. E. Muller, "A note on the generation of random normal deviates," *Ann.*

Math. Statist., vol. 29, no. 2, pp. 610-611, 1958.

[294] B. R. Mahafza, Introduction to radar analysis, London, UK: Boca Raton: CRC Press, 1998, pp. 251-254.

[295] M. Ben-Ari, "A tutorial on Euler angles and quaternions," Department of Science Teaching Weizmann Institute of Science Version 2.0.1, 2014. [Online]. Available: https://www.weizmann.ac.il/sci-tea/benari/sites/sci-tea.benari/files/uploads/softwareAndLearningMaterials/quaternion-tutorial-2-0-1.pdf. [Accessed 1 May 2020].

[296] P. Tait, Introduction to radar target recognition, London, UK: IET, 2009, pp. 151-155.

[297] B. Mahafza, Introduction to radar analysis, Boca Raton: CRC Press, 1998, pp. 102-103.

[298] NAWCWD Avionics Department, Electronic warfare and radar systems engineering handbook, NACAIR, 2013.

[299] M. Budge, "Radar range equation," www.uah.edu, 2011. [Online]. Available: http://www.ece.uah.edu/courses/material/EE619-2011/RadarRangeEquation(2)2011.pdf. [Accessed 11 May 2020].

[300] N. Abdellaoui, P. Hubbard and P. Duncan, "An enhanced synthetic environment for maritime surveillance," Apps.dtic.mil, [Online]. Available: https://apps.dtic.mil/dtic/tr/fulltext/u2/a640449.pdf. [Accessed 1 May 2020].

Appendix A

FURTHER EXPLANATION OF LANGUAGE SPECIFICATION MODIFICATION

The C4L language specification builds upon the C3L baseline and refers to the Master's degree thesis [69], which explains the execution and numerical concepts of C3L as they differ from most other languages. The extension of C3L was to include both emission and receiver behaviour specifications. Although only used in this work for emitters, it is also applicable to radar receivers for processing. The language elements for an emitter specification are scheduled within C4L and needed to represent different modulations of frequency, pulses, PRIs, antenna beams, and scans defined with different modulations, either concurrent sequences or as part of a state change selection or iteration.

Since the Master's degree thesis [69], there has been an additional parameter feature within C3L, and it can now define a parameter as Static, Semi-Dynamic, or Dynamic:

- **Static**: parameters do not change when executing but provide a standard script to be re-configured with specific values. Static parameters permit a countermeasure or emitter technique to be generalized and configured for specific platform equipment features or limitations.
- **Semi-Dynamic**: parameters are use-case-specific configurations. They provide adaptability to a specification for a specific engagement circumstance. Semi-Dynamic parameters permit a countermeasure or emitter technique to re-configure and match a specific platform and threat engagement.
- **Dynamic**: parameters are updated constantly and make countermeasures reactive. C3L compiles to a single execution thread for data integrity for dynamic updates. Dynamic parameters permit a countermeasure or emitter technique to adapt during the engagement reactively.

The following updates to the C3L language were made during this research to support the synthetic dataset generator for radar emitters:

11.1 Scheduling Specification

The BNF definition extends to having a *SensorElement* within the definition of the *Elements*. See Figure 50 for the *SensorElement* inclusion shown in red and BNF language chaining shown with arrows.

<Behaviour> ::= “Schedule” [<Identifier>] <ScheduleType> [<SuccessCriteria>] [<FailureCriteria>] **<RequirePrescript>** [<Behaviour>]

<RequirePrescript> ::= **<BlockStatement>** | <Behaviour>

<BlockStatement> ::= “{”**<ElementaryProg>** “}” | <Element> “;”

<ElementaryProg> ::= [<Identifier>] **<Element>**“;” | [<ElementaryProg>] |[<Behaviour>] [<ElementaryProg>]

<Element> ::= <ChaffElement> | <FlareElement> | <DecoyElement> | <EcmElement> | <ManoeuvreElement> | <InActionElement> | **<SensorElement>**

FIGURE 50 SCHEDULING LANGUAGE BNF

11.2 Sub-Scheduling Specification

The *SensorElement* can sub-schedule an *Element* with or without countermeasures. See Figure 51 for the *SensorElement*, allowing the sub-scheduling of a *SensorProgram* command.

<SensorElement> ::= "Emitter" [<Identifier>] ["On" | "Off"] {**<EmitterTechnique>**} ";"

<EmitterTechnique> ::= "SensorProgram" [<Identifier>] <ScheduleType> [<SuccessCriteria>] [<FailureCriteria>] **<SensorPrescript>** [<EmitterTechnique>] ;

<SensorPrescript> ::= <SensorBlockState> | <EmitterTechnique>

<SensorBlockState> ::= "{"**<SensorProgram>**"}" | <SensorTechnique>

<SensorProgram> ::= <Identifier> **<SensorTechnique>** ";" [<SensorProgram>] | [<EmitterTechnique>] [<SensorProgram>]

<SensorTechnique> ::= "Sensor" "ChanNum" <Integer> **[<TxScanConfig>] [<RxScanConfig>]** [<TxModConfig>] [<RxModConfig>] [<InActionElement>]

FIGURE 51 SUB-SCHEDULING LANGUAGE BNF

The sub-scheduling of a *SensorProgram* allows definitions to be partial in the Transmit (Tx) and Receive (Rx) configurations of scans and modulations, which means that the configurations can be at different hierarchical levels and differ in transmit and receive, shown in red within Figure 51, such that the Scan-on-Receive-only modes are isolated from the transmit scan and PRI modulation definitions.

11.3 The Scan and Beam Configurations

The Scan Configurations on either transmit or receive can define the scan and beams as in Figure 52 shown in red, with the parameters to modulate the beamwidth and scan, and as such, an electronic-scan antenna's beam broadening effect can modulate in the same statement for convenience.

<TxScanConfig> **::= "TxScanConfig" <ScanParameters>**

<RxScanConfig> **::= "RxScanConfig" <ScanParameters>**

<ScanParameters> **::= <RepeatType> "WFNum"<Integer> ["PointAngles" <DirectionVar> ["RollAxis"<TermParam>]] [<Location Var>] ["ScanCentre" <ScanCentre>] [<ScanMounting>] [<ScanStabilisation>] ["BeamWidth" <DirectionVarBW> ["RollAxis" <TermParam>]] ["Suppression" <Suppression>] ["Pol"<PolType>] ["Purity" <TermParams>]**

<DirectionVar> ::= ["Azimuth"<TermParam>] ["Elevation"<TermParam>]

<DirectionVarBW> ::= ["Azimuth"<TermParam>["CoSec"]] ["Elevation" <TermParam> ["CoSec"]]

<LocationVar> ::= ["x"<TermParam>] ["y"<TermParam>] ["z"<TermParam>]

<ScanCentre> ::= "RelativeTo" | "Horizon" | "Target" <Integer>

<ScanMounting ::= "AzOnElMount" | "ElOnAzMount"

<ScanStabilisation> ::= "HorizonStabilised" | "PlatformStabilised"

<Suppression> ::= <TermParams> <TermParams>

<PolType> ::= "Linear" <TermParams> | "Clockwise" | "AntiClockwise"

FIGURE 52 SCAN CONFIGURATION LANGUAGE BNF

Each scan data element parameter relates to a definition in the Threat Analysis method in Chapter 4, and as such, values for these parameters become known. Missing parameters can use the modified expert system method in Chapter 5 to estimate better values. Also, the *certainty* value concerning the *perfected value* in the modified expert system method may assist in establishing value ranges for Monti-Carlo and stochastic modelling.

11.4 The Modulation Configurations

The Modulation Configurations on either transmit or receive can define the Power, Gain, Pulse, PRI, Frequency, and timings, as in Figure 53 shown in red.

<TxModConfig>::= "TxModConfig" <ModParameters>

<RxModConfig>::= "RxModConfig" <ModParameters>

<ModParameters>::= <RepeatType> "WFNum" <Integer> ["Carrier"<TermParam>] ["GapDuration"<TermParam>] ["Duration"<TermParam>] ["StartRF"<TermParam>] ["EndRF"<TermParam>] ["Phase"<TermParam>] ["Gain"<TermParam>] ["RepetitionInterval"<TermParam>] ["NoiseFigure"<TermParam>] ["Temperature"<TermParam>] [<PxTags>]

<PxTags>::= ["Probe"] ["DataLink"] ["Interrogator"] ["Transponder"] ["Illuminator"] ["ECMListen"] ["Calibration"] ["TargetTrack"] ["TargetAcquire"] ["EWSearch"] ["AirSearch"] ["IDRecognise"] ["HeightFind"] ["Effect"] ["Engage"] ["Prosecute"] ["CSD"] ["RMI"]

FIGURE 53 MODULATION CONFIGURATION LANGUAGE BNF

The *GapDuration*, *Duration*, and *RepetitionInterval* define the Transmitter's pulse delay, the Pulse width, and the PRI. The receiver defines the Dead Time, Listen Time, and PRI, allowing modelling Pulsed and CW functions, and this allows the Listen Time to be swathe over the target for Target Tracking Radars. Also, the phase values allow coherent, non-coherent, or coherent on receive configurations. The *StartRF*, *EndRF*, and Carrier definitions allow agility and FM modulation on the transmit side and allow the receiver bandwidth and Local Oscillator (LO) offsets on the receiver side, supporting; Bi-polar or Uni-polar receivers. The processing tags (*PxTags*) can define the ground truth of the emitter function code towards a kill chain stage match. Again each modulation data element parameter relates to the Threat Analysis method in Chapter 4 and missing parameters to the modified expert system method in Chapter 5 to estimate better values and stochastic ranges.

11.5 Channel and Waveform Configurations

In Figure 51, Figure 52, and Figure 53, the parameters can be combined or separated using waveform number *(WFNo)* and Channel Number *(ChanNum)*. Using different *ChanNum* values declares that many *WFNo* configurations are separated to define beams for *mono-pulse*. Using the same *ChanNum* value with a different *WFNo* allows them to be combined to define *contiguous pulses*, *phase modulation on the pulse*, and complex beam shapes; and this can define several discrete transmitters or receivers, to support many isotropic beams with varying phase offsets applied for beam-forming as in an Electronically Scanned (E-Scan) array. Figure 54 is a result from a Matlab model developed in this research that illustrates the transmit pulse propagation at a moment in time and shows the effect of combining emitters with phase offset steps in those emitters.

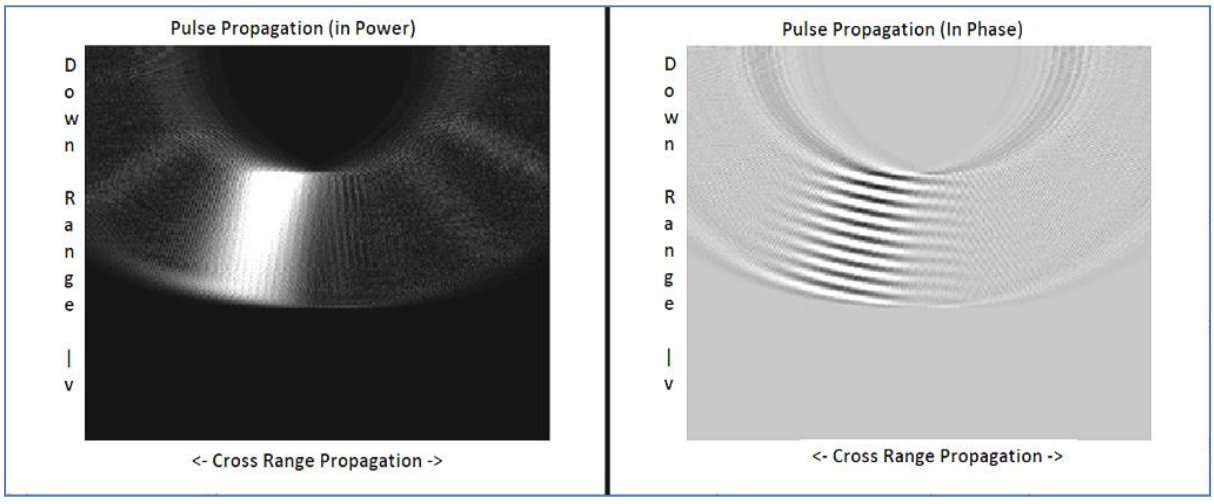


FIGURE 54 E-SCAN BEAM STEERING ILLUSTRATION

Figure 54 left is in power and right is in phase, but the sample rate needs to be above the Nyquist of the carrier frequency, which is too much memory usage in this application, but the same emitter DLLs and architecture are modular and reusable in other applications.

11.6 Variable Step Simulation Support

The numerical representation allows value calculation for any time step, even variable steps, and defines integral functions rather than discrete values, and these are called *TermParams*. *TermParams* define start, stop, rate, and step as a slope over time and include ten numerical derivatives in Figure 55 in the BNF language syntax.

<TermParams>	**::= [<Position> [<Velocity> [<Acceleration> [<Jerk> [<Snap> [<Crackle> [<Pop> [<Lock> [<Drop> [<Shot> [<Put>]]]]]]]]]]**
<Position>	::= "Position" <ParamSet>
<Velocity>	::= "Velocity" <ParamSet>
<Acceleration>	::= "Accel" <ParamSet>
<Jerk>	::= <JerkAmbiguity> <ParamSet>
<JerkAmbiguity>	::= "Jerk" | "Jolt"
<Snap>	::= <SnapAmbiguity> <ParamSet>
<SnapAmbiguity>	::= "Snap" | "Spasm" | "Jounce" | "Surge" | "Sprite"
<Crackle>	::= "Crackle" <ParamSet>
<Pop>	::= "Pop" <ParamSet>
<Lock>	::= "Lock" <ParamSet>
<Drop>	::= "Drop" <ParamSet>
<Shot>	::= "Shot" <ParamSet>
<Put>	::= "Put" <ParamSet>
<ParamSet>	::= <StartVal> [<End> <Rate> [<Step>]]
<StartVal>	::= "Current"| <Start>
<StepPrams>	::= <Start> [<End> <Rate>]
<Start>	::= "Start" <RegularExpression>
<End>	::= "End" <RegularExpression>
<Rate>	::= "Rate" <RegularExpression>
<Step>	::= "Step" <StepPrams>

FIGURE 55 NUMERICAL REPRESENTATION FOR VARIABLE STEP SIMULATION

11.7 Emitter Physics & Propagation Library

The Data Generator uses the DLL with an Emitter Physics Propagation Library developed in this research. The emitter mark-up language defines radar behaviours, although in this example the receiver is an ELINT collector rather than a radar receiver. The ELINT collector parameters are defined as parameters, so the same DLL can also be used for radar modelling.

11.7.1 Noise Floor

The noise figure (f_n) can be defined on both transmit and receive for receiver Noise and support Noise Radar. The definition of the noise level is as per [292] and uses the Box-Muller Transform [293]. Equations (105), (106), (107), (108), (109), (110), (111), and (112) define the noise floor in both I and Q components.

$$t = < TempInC > + 273.15 \quad , \tag{105}$$

$$f_n = 10.0^{\left(\frac{noise\ Figure}{10}\right)} \quad , \tag{106}$$

$$\beta = abs(Bandwidth_{End} - Bandwidth_{Start}) \quad , \tag{107}$$

$$K = 1.380649x10^{-23} \quad , \tag{108}$$

$$Level_{Noise} = K \cdot \beta \cdot t \cdot f_n \cdot 1000mw \quad , \tag{109}$$

$$nf = \sqrt{Level_{Noise}} \cdot \sqrt{\left(\frac{2}{\pi}\right)} \quad , \tag{110}$$

$$I_{Waveform}[1..Length_{swathe}] = BoxMuller() \cdot nf + wfChan(\dots) \quad , \tag{111}$$

$$Q_{Waveform}[1..Length_{swathe}] = BoxMuller() \cdot nf + wfChan(\dots) \quad . \tag{112}$$

11.7.2 Delta Phase Offset

Other distortions may occur for off-bore-sight squint angles as a delta phase ($\Delta\phi$) offset and support phase comparison mono-pulse Interferometry. The positional displacement of beams sets a $\Delta\phi$ offset between different channels. The positional displacement is in metres in the *x*, *y*, and *z*. The $\Delta\phi$ value calculations are as per [294] and in Equations (113), (114), (115), (116), and (117).

$$\phi = sin(El - \text{atan2}\,(z, y)) \cdot \frac{y}{\lambda} \quad , \tag{113}$$

$$\theta = sin(Az - \text{atan2}\,(z, x)) \cdot \frac{x}{\lambda} \quad , \tag{114}$$

$$\pi wrap(a) = a - 2\pi \cdot \left\lfloor \frac{a+\pi}{2\pi} \right\rfloor \quad , \tag{115}$$

$$\lambda = \frac{c}{f} \quad , \tag{116}$$

$$\Delta\theta = \pi wrap\big((\phi - \lfloor\phi\rfloor) \cdot 2\pi\big) + \pi wrap\big((\theta - \lfloor\theta\rfloor) \cdot 2\pi\big) \quad . \tag{117}$$

When plotted progressively on off-bore-sight angles, the physics library test module developed in this research, the *cos(Δθ)* shown in Figure 56 left shows the $\Delta\phi$ phase effect. The *x* or *y* value can be inverted to invert the $\Delta\phi$ pattern in Figure 56 right, supporting phase comparison mono pulse.

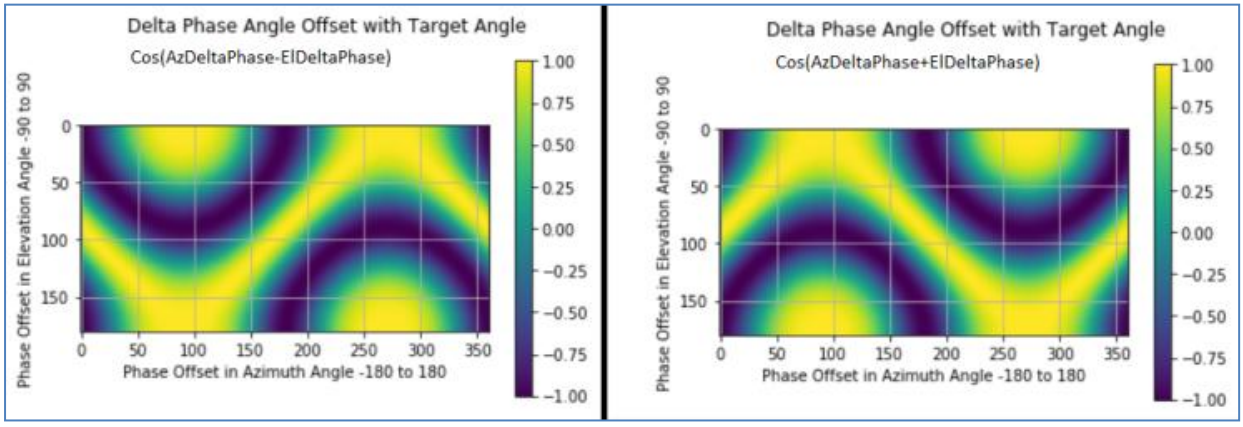


FIGURE 56 DELTA PHASE (ΔΦ) OFFSET FOR OFF BORE-SIGHT ANGLES (TEST MODULE)

11.7.3 Scan & Point Angles

The scan configurations define the scan point angles [295] in the scan configurations; they include a *Scan Centre* to define if the angle concerns a target, the horizon, or the relative point, making the definition reusable in a different scenario. The *Scan Orientation* defines if the scan angle is space or platform stabilized and also if the elevation gimbals are mounted on the azimuth gimbals or vice versa to support, definitions for Air Intercept (AI) radars and many Anti-Aircraft Artillery (AAA) or Surface to Air Missile (SAM) systems. The trigonometrically transformed angle starts with a comparison of positions as in Equation (118):

$$[\Delta_x, \Delta_y, \Delta_z] = [x_{target} - x_{origin}, y_{target} - y_{origin}, z_{target} - z_{origin}] \quad . \tag{118}$$

When the emitter or receiver scan stabilization is: azimuth (Θ) gimbals mounted on the elevation (Φ) mount, then less gain is required in the azimuth motor, and the transform is as Equation (119). That gimbals mounting is consistent with Air Intercept (AI) radars that scan the horizon in search.

$$\left[\Theta_{point}, \Phi_{point}\right] = axisAI(x_{origin}, y_{origin}, z_{origin},\ x_{target}, y_{target}, z_{target},\ \Theta_{origin}, \Phi_{origin},\ \Theta_{Scan}, \Phi_{Scan}, \Psi_{roll}) \ . \tag{119}$$

Where *axisAI* is defined by (120) to (135):

$$\Theta_{point} = \ arctan\left(\frac{Y_{piont}}{X_{point}}\right) \ , \tag{120}$$

$$\Phi_{point} = \ arctan\left(\frac{Z_{point}}{\sqrt{{X_{point}}^2 + {Y_{point}}^2}}\right) \ , \tag{121}$$

$$X_{point} = \ cos(\Phi_{LongFlightDatum}) \ * \ X_{plat} + \ \sin(\Phi_{LongFlightDatum}) \cdot Z_{plat} \ , \tag{122}$$

$$Y_{point} = Y_{plat} \ , \tag{123}$$

$$Z_{point} = \ -sin(\Phi_{LongFlightDatum}) \cdot X_{plat} \ + \ \cos\left(\Phi_{LongFlightDatum}\right) \cdot Z_{plat} \ , \tag{124}$$

$$X_{plat} = \cos\left(\Phi_{origin}\right) \cdot cos\left(\Theta_{origin}\right) \cdot X_{space} + \cdot cos\left(\Phi_{origin}\right) \cdot sin\left(\Theta_{origin}\right) \cdot Y_{space} + \sin\left(\Phi_{origin}\right) \cdot Z_{space} \ , \tag{125}$$

$$Y_{plat} = -(sin(roll + \pi) \cdot sin(\Phi_{origin}) \cdot \cos\left(\Theta_{origin}\right) \cdot \ cos(\Psi_{roll} + \pi) \cdot \sin\left(\Theta_{origin}\right)) \cdot X_{space} \ + \ (-sin(\Psi_{roll} + \pi) \cdot \sin\left(\Phi_{origin}\right) \cdot \sin\left(\Theta_{origin}\right) + cos(\Psi_{roll} + \pi) \cdot \cos\left(\Theta_{origin}\right)) \ \cdot Y_{space} \ + \ sin(\Psi_{roll} + \pi) \cdot \cos\left(\Phi_{origin}\right) \cdot Z_{space} \ , \tag{126}$$

$$Z_{plat} = (-cos(roll + \pi) \cdot \sin\left(\Phi_{origin}\right) \cdot \cos\left(\Theta_{origin}\right) + sin(\Psi_{roll} + \pi) \cdot \sin\left(\Theta_{origin}\right)) \cdot X_{space} \ - \ (cos(\Psi_{roll} + \pi) \cdot \sin\left(\Phi_{origin}\right) \cdot \sin\left(\Theta_{origin}\right) + sin(\Psi_{roll} + \pi) \cdot \cos\left(\Theta_{origin}\right)) \cdot Y_{space} \ + \ cos(\Psi_{roll} + \pi) \cdot \cos\left(\Phi_{origin}\right) \cdot Z_{space} \ , \tag{127}$$

$$x_{space} \ = \ r_{slant} \cdot cos(\Phi_{pitch}) \cdot cos(\Theta_{yaw}) \ , \tag{128}$$

$$Y_{space} \ = \ r_{slant} \cdot \ cos(\Phi_{pitch}) \cdot \sin\left(\Theta_{yaw}\right) \ , \tag{129}$$

$$Z_{space} = \ r_{slant} \cdot \ sin(\Phi_{pitch}) \ , \tag{130}$$

$$\Theta_{yaw} = \ PiWrap\left(arctan\left(\frac{\Delta_x}{\Delta_y}\right) - \Theta_{Scan}\right) \ , \tag{131}$$

$$\Phi_{pitch} = \ PiWrap\left(El_{Scan} - arctan\left(\frac{\Delta_z}{r_{ground}}\right)\right) \ \ , \tag{132}$$

$$r_{ground} = \sqrt{{\Delta_y}^2 + {\Delta_x}^2} \ , \tag{133}$$

$$r_{slant} = \sqrt{{r_{ground}}^2 + {\Delta_z}^2} \ , \tag{134}$$

$$PiWrap(v) = v \ - \ 2\pi \cdot \left\lfloor\frac{(v + \pi)}{2\pi}\right\rfloor \ . \tag{135}$$

When the scan's stabilization is: the configuration: Elevation gimbals mounted on the Azimuth mount, then the extra gain will be required in the azimuth motor, and this configuration is consistent with many Anti-Aircraft Artillery (AAA) or Surface to Air Missile (SAM) systems and the coordinate transform is as in Equation (136) instead.

$$axisSAM(x_{origin}, y_{origin}, z_{origin}, x_{target}, y_{target}, z_{target}, \Theta_{origin}, \Phi_{origin}, \Theta_{Scan}, \Phi_{Scan}, \Psi_{roll}) = axisAI(x_{origin}, y_{origin}, z_{origin}, x_{target}, y_{target}, z_{target}, Az_{origin} \cdot cos(\Phi_{origin}), \Phi_{origin}, \Theta_{Scan} \cdot cos(\Phi_{Scan}), \Phi_{Scan}, \Psi_{roll}) \quad (136)$$

The scanner system will point to the beam of the emitter or receiver, but the beam itself has a shaping that will modify the power in that beam.

11.7.4 Beam Shaping

Beam shapes are defined with the beamwidth as the dimension across the beam at the 3dB point (half power) using *sin(x)/x* form as in [296], [297] see Equation (137). This approximation typically requires a *numerical method* for the 3dB power-point for every defined beamwidth. However, the method presented in this research dissertation results in that point without the need for a *numerical method*. Figure 57 shows the B_{Loss1} beam shape shown in a Redline, the B_{Loss2} in Green, and the average in the Blue-line. That Blue-line crosses at the 3dB point without a *numerical method*.

$$BeamLoss = BeamShapeLossSinX(bore, bw, SideLobeSpression) \quad (137)$$

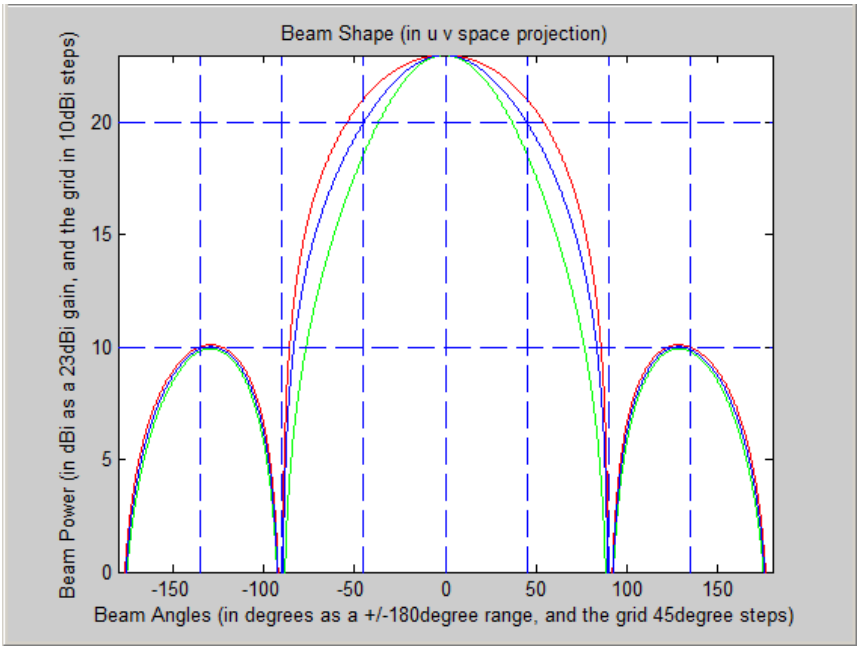


FIGURE 57 BEAM SHAPING FOR ACCURATE 3DB POINT LOCATION TO ANGLE (TEST MODULE)

See Figure 57. There are no losses when the off bore-sight angle (bore) is precisely on bore-sight, and the loss factor is 1.0. However, in the case of the off-bore-sight angle *(bore)* is within a beamwidth *(bw)* but not precisely on bore-sight then Equations (138), (139), (140), and (141) are used:

$$bore_{ratio} = \frac{bore \cdot \pi}{bw} \quad , \quad (138)$$

$$beam = \left[\frac{sin(bore_{ratio})}{bore_{ratio}}\right] \quad , \quad (139)$$

$$\{B_{Loss1} = beam | 0 < [bore] \leq bw\} \quad , \quad (140)$$

$$\left\{B_{Loss2} = \frac{(11^{beam})-1}{10} \middle| 0 < [bore] \leq bw\right\} \quad . \quad (141)$$

In the case that the off bore-sight angle *(bore)* is out with the beamwidth *(bw)*, then the different Equations (142), (143), (144), (145), and (146) are used:

$$SL_{ref1} = \left[\frac{sin(1.5\pi)}{1.5\pi}\right] \quad , \quad (142)$$

$$SL_{ref2} = \frac{\left(12^{SL_{ref1}}\right)-1}{10} \ , \tag{143}$$

$$Sl_{Sup} = 10^{\left(\frac{-SideLobeSpression}{10}\right)} \ , \tag{144}$$

$$\left\{B_{Loss1} = beam \cdot \frac{Sl_{Sup}}{SL_{ref1}} \middle| bw < [bore] \leq \pi\right\} \ , \tag{145}$$

$$\left\{B_{Loss2} = \frac{(11^{beam})-1}{10} \cdot \frac{Sl_{Sup}}{SL_{ref2}} \middle| bw < [bore] \leq \pi\right\} \ . \tag{146}$$

Nevertheless, there is more than one classification of beam shape when considered in two dimensions for Rectangular and Concentric beam shapes of side lobes. Concentric beam shapes are for illuminators or conical scanners, and rectangular beam shapes are for surveillance and some airborne radars as a rule of thumb.

11.7.4.1 Rectangular Beam-shape

Azimuth and Elevation losses combine as a cumulative beam loss in Equation (147):

$$BeamLoss = \left(\frac{(AzB_{Loss1} + AzB_{Loss2})}{2}\right) \cdot \left(\frac{(ElB_{Loss1} + ElB_{Loss2})}{2}\right) \ . \tag{147}$$

The rectangular beam shape in Figure 58 left can rotate to support diagonal scans shown in Figure 58 right, such as the SNR-125 radar (SA-3 Low Blow). Figure 58 left and right are examples from the physics library test module.

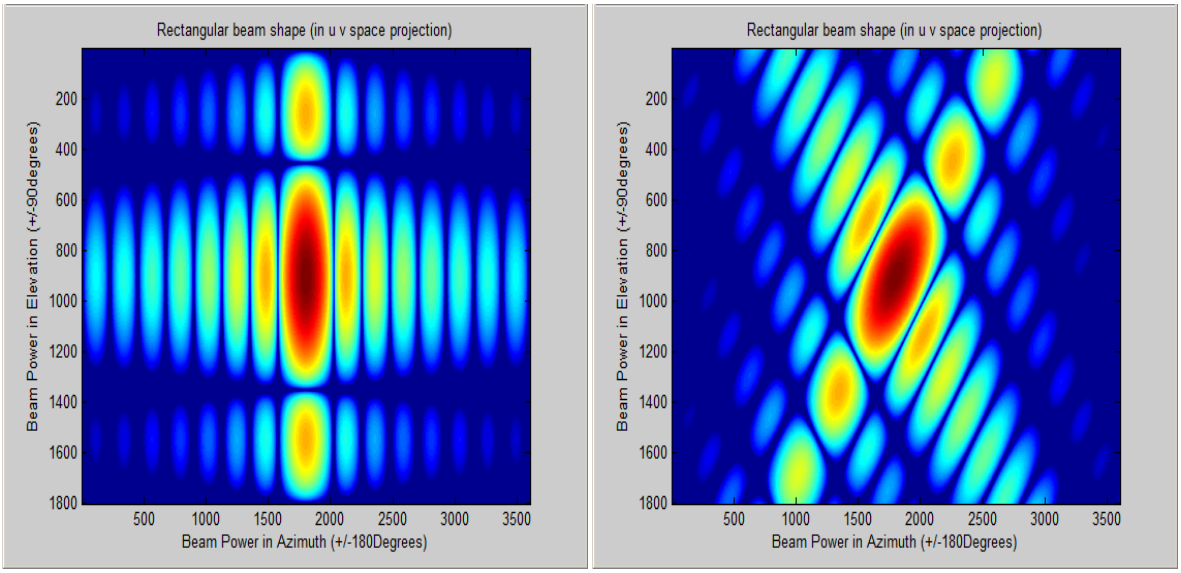


FIGURE 58 BEAM SHAPE OF A RECTANGULAR BEAM IN ORIENTATIONS (TEST MODULE)

$$[\Theta_{rot}, \Phi_{rot}] = rotatebeam(\Theta, \Phi, \Psi) \ . \tag{148}$$

Where the angles rotate by ψ in azimuth Θ and elevation (Φ) in Equations: (149), (150), (151), (152), and (153), note that this angle transform is in *u/v space*, and the coordinates system is in the *axisAI*, and *axisSAM* transforms.

$$\Phi_{rot} = sin(\varphi_\Lambda) \cdot \sigma \ , \tag{149}$$

$$\Theta_{rot} = cos(\varphi_\Lambda) \cdot \sigma \ , \tag{150}$$

$$\sigma = \sqrt{\Phi^2 + \Theta^2} \ , \tag{151}$$

$$\varphi_\Lambda = PiWrap(\varphi + \Psi) \ , \tag{152}$$

$$\varphi = arctan\left(\frac{\Phi}{\Theta}\right) \ . \tag{153}$$

Thus the rectangular beam loss for a point angle can be calculated in Equations: (154), (155), and (156).

$$beamloss_{rect} = recBeamLoss(\Theta, \Phi, \Theta_{bw}, \Phi_{bw}, \Theta_{slsup}, \Phi_{slsup}, \Psi) \ , \quad (154)$$

$$recBeamLoss(\Theta, \Phi, \Theta_{bw}, \Phi_{bw}, \Theta_{slsup}, \Phi_{slsup}, \Psi) = \quad (155)$$
$$(beamLossSinX(\Theta_{rot}, \Theta_{bw}, \Theta_{slsup}) \cdot beamLossSinX(\Phi_{rot}, \Phi_{bw}, \Phi_{slsup})) \ ,$$

$$[\Theta_{rot}, \Phi_{rot}] = RotateBeam(\Theta, \Phi, \Psi) \ . \quad (156)$$

11.7.4.2 Concentric Beam-shape

The indication of a concentric beam shape is by one of the beam shape values being set to zero in the language, in either azimuth or elevation, as in the physics library test module example shown in Figure 59:

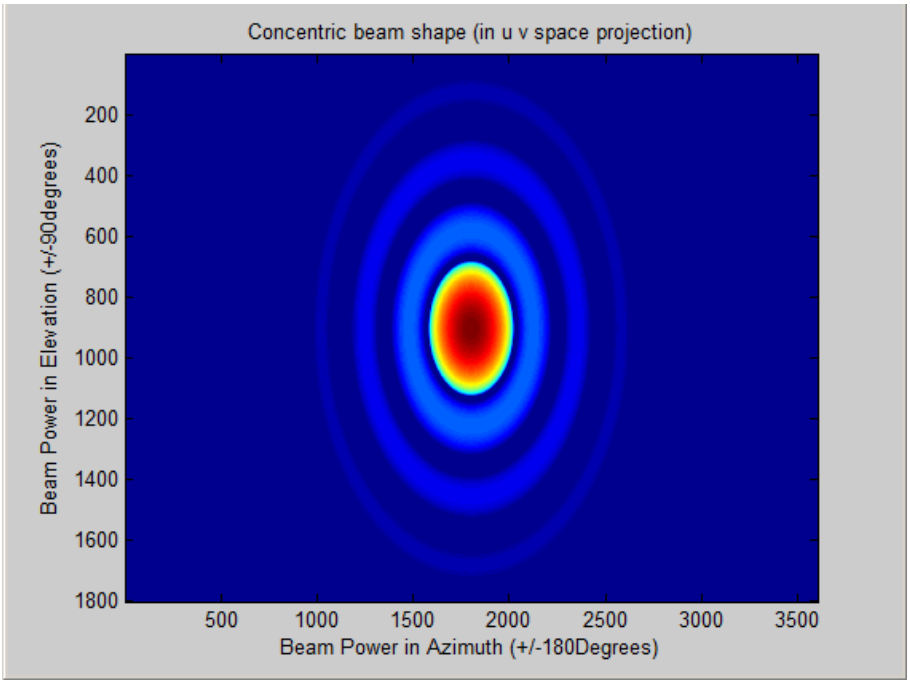


FIGURE 59 BEAM SHAPE OF A CONCENTRIC BEAM (TEST MODULE)

Again, when the off bore-sight angle (bore) is precisely on bore-sight, there are no losses, and the loss factor is 1.0.

In the case of the off bore-sight angle *(bore)* is within a beamwidth *(bw)* but not precisely on bore-sight, then the Equations are: (157), (158), (159), (160), (161), and (162):

$$beamloss_{circ} = circBeamLoss(\Theta, \Phi, \Theta_{bw}, \Phi_{bw}, \Theta_{slsup}, \Phi_{slsup}, \Psi) \ , \quad (157)$$

$$circBeamLoss(\Theta, \Phi, \Theta_{bw}, \Phi_{bw}, \Theta_{slsup}, \Phi_{slsup}, \Psi) = beamLossSinX(bore, bw, sup) \ , \quad (158)$$

$$bore = \sqrt{\Theta_{rot}{}^2 + \Phi_{rot}{}^2} \ , \quad (159)$$

$$[\Theta_{rot}, \Phi_{rot}] = RotateBeam(\Theta, \Phi, \Psi) \ , \quad (160)$$

$$bw = \sqrt{\Theta_{bw}{}^2 + \Phi_{bw}{}^2} \ , \quad (161)$$

$$sup = min(\Theta_{slsup}, \Phi_{slsup}) \ . \quad (162)$$

11.7.5 Polarisation Miss-Matches

The polarisation type and *polarisation angles* are defined for both transmit and receive beams, as miss-matches in polarisation types will cause losses [298]. Some radars use different polarisations on

transmit and receive or multiplexed. Cross-polarisation influences are in support of *ChanNum* and *WFNo*.

If both the transmitter and receiver are linear, then Equations (163), (164), (165), (166), and (167) are used:

$$Factor_{Tx} = \left(1 - \frac{Purity_{Tx}}{100}\right) + 1 \ , \tag{163}$$

$$Factor_{Rx} = \left(1 - \frac{Purity_{Rx}}{100}\right) + 1 \ , \tag{164}$$

$$\Delta_{Angle} = (cos([piwrap(Angle_{Tx} - Angle_{Tx})]))^2 \ , \tag{165}$$

$$\Delta_{Purity} = min(min(Factor_{Tx} \cdot Factor_{Rx}, Factor_{Tx}), Factor_{Rx}) \ , \tag{166}$$

$$PolarLoss = max(max(\Delta_{Purity} \cdot \Delta_{Angle}, Factor_{Tx} - 1), Factor_{Rx} - 1) \ . \tag{167}$$

If both the transmitter and receiver are circularly polarised but are miss-matched in handedness, then Equations (168), (169), and (170) are used:

$$Factor_{Tx} = \left(1 - \frac{Purity_{Tx}}{100}\right) \ , \tag{168}$$

$$Factor_{Rx} = \left(1 - \frac{Purity_{Rx}}{100}\right) \ , \tag{169}$$

$$PolarLoss = Factor_{Tx} + Factor_{Rx} \ . \tag{170}$$

If there is a mix in the transmitter and receiver polarisation between linear and circular, then Equations (171), (172), and (173) are used:

$$Factor_{Tx} = \frac{\left(1 - \frac{Purity_{Tx}}{200}\right)}{2} \ , \tag{171}$$

$$Factor_{Rx} = \frac{\left(1 - \frac{Purity_{Rx}}{200}\right)}{2} \ , \tag{172}$$

$$PolarLoss = Factor_{Tx} + Factor_{Rx} \ . \tag{173}$$

If the transmitter and receiver polarisations are a match in the circular type, then Equations (174), (175), and (176) are used:

$$Factor_{Tx} = \left(1 - \frac{Purity_{Tx}}{100}\right) + 1 \ , \tag{174}$$

$$Factor_{Rx} = \left(1 - \frac{Purity_{Rx}}{100}\right) + 1 \ , \tag{175}$$

$$PolarLoss = min(min(Factor_{Tx} \cdot Factor_{Rx}, Factor_{Tx}), Factor_{Rx}) \ . \tag{176}$$

11.7.6 Emission Waveforms

The *Tx Configuration* allows Frequency Modulation (FM), and the *Rx Configuration* uses these for a receiver bandwidth.

With the different *WFNo* with the same *ChanNum*, more than one definition can combine to support more complex waveforms that are nonlinear, are phase modulated, or have a contiguous pulse.

Figure 60, left and right, shows the Phase and Amplitude results for a linear chirp from the test module of the physics library developed as part of this research.

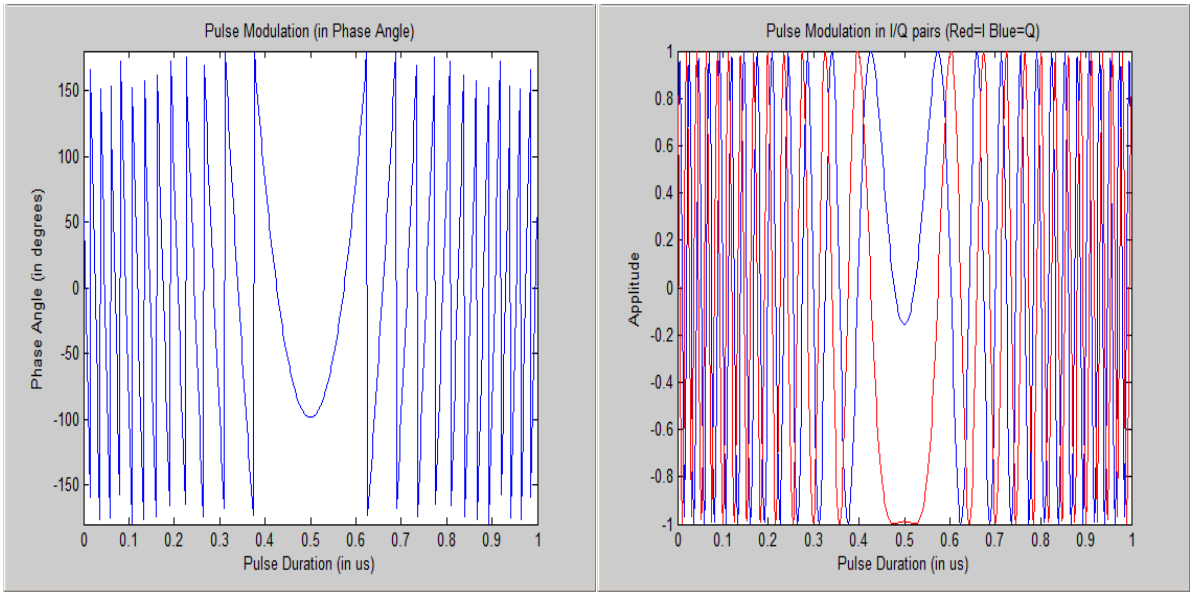


FIGURE 60 WAVEFORM MODULATION LINEAR CHIRP EXAMPLE (TEST MODULE)

The waveform phase (WF_θ) in Figure 60 is a test module example from the physics library generated from the Equations (177), (178), (179), (180), (181), and (182):

$$\delta = Sample_{rate} \ , \tag{177}$$

$$Sample_{Period} = \frac{1}{\delta} \ , \tag{178}$$

$$\alpha = StartFreq \ , \tag{179}$$

$$n_{wf} = min\left(max\left(\left\lceil\left(\frac{Duration}{Sample_{Period}}\right)\right\rceil, 1\right), \left\lfloor\frac{RepetionRate}{Sample_{Period}}\right\rfloor\right) \ , \tag{180}$$

$$\Delta = \frac{(EndFreq - \alpha)}{(n_{wf} \cdot 2)} \ , \tag{181}$$

$$WF_\theta = \bigcup_{i=0}^{i=n_{wf}-1} \begin{cases} mod\left(\frac{2\pi}{\left(\frac{1}{(i\cdot\Delta+\alpha)}\cdot\delta\right)} \cdot (i+1) + Phase_{Offset}, 2\pi\right) - \pi & \text{if } (i\cdot\Delta+\alpha) \neq 0 \\ piWrap(Phase_{Offset}) & otherwise \end{cases} . \tag{182}$$

After the waveform is applied, a frequency is mixed for the receiver's bandwidth relative to the *Tx Carrier*, so the waveform appears at the right frequency concerning the ELINT collector receiver. The *TxGain* represents the *ERP* and includes the transmitter power and antenna gain. The *RxGain* is the receive antenna gain and amplification stages, and both are in the radar-range-equation [299]. A phase rate is applied along the pulse to support FMCW radar processing and between the pulses to support Pulse-Doppler processing radar configurations.

11.7.7 Spreading Losses

Set in the language, the emitter power and receiver gain with f as the carrier frequency, and *TxGain* represents the *ERP*, which includes the transmitter antenna gain and transmitter power. The *RxGain* is the receive antenna gain and any amplification stages combined.

$$Gain = (10^{\left(\frac{ERP}{10}\right)} \cdot BeamLoss_{Tx} \cdot Polar_{Loss} \cdot 10^{\left(\frac{RCS}{10}\right)} \cdot 10^{\left(\frac{RxGain}{10}\right)} \cdot BeamLoss_{Rx} \cdot \left(\frac{c}{f}\right)^2 \cdot 1000 \ , \tag{183}$$

$$Power_{RxOut} = \begin{cases} \left(\frac{Gain}{4\pi\cdot r_{slantOut}{}^2}\right) & r_{slantOut} > 0.0 \\ Gain_{RxPower} & otheise \end{cases} \ , \tag{184}$$

$$Power_{RxIn} = \begin{cases} \left(\frac{Power_{RxOut}}{(4\pi)^2 \cdot {r_{slantIn}}^2}\right) & r_{slantIn} > 0.0 \\ Power_{RxOut} & otherwise \end{cases}, \quad (185)$$

$$Signal_{Rx} = \sqrt{Power_{RxIn}} \ . \quad (186)$$

Equations (183), (184), (185), and (186) define the spreading loss, and the spreading loss for a single direction of propagation (as in this case) and is when $r_{slantIn}$ is 0 metres and an *RCS* of 0dBsm.

11.7.8 Phase Offset

The emitter 'phase offset' and receiver 'phase offset' can support the following radar configurations: coherent, non-coherent, and coherency-on-receive.

11.7.9 Doppler Effect

The calculations of radial-velocity of targets, emitter, and receiver are in Equations: (187), (188), (189), (190), and (191).

$$v_{radial} = \frac{\Delta_X}{\Delta_{Mag}} \cdot \Delta v_{X_{In}} + \frac{\Delta_y}{\Delta_{Mag}} \cdot \Delta v_{Y_{In}} + \frac{\Delta_z}{\Delta_{Mag}} \cdot \Delta v_{Z_{In}} + \frac{\Delta_X}{\Delta_{Mag}} \cdot \Delta v_{X_{Out}} + \frac{\Delta_y}{\Delta_{Mag}} \cdot \Delta v_{Y_{Out}} \frac{\Delta_z}{\Delta_{Mag}} \cdot \Delta v_{Z_{Out}} \ , \quad (187)$$

$$\Delta_{mag} = \sqrt{{\Delta_X}^2 + {\Delta_y}^2 + z^2} \ , \quad (188)$$

$$[\Delta_X, \ \Delta_Y, \ \Delta_Z] = \left[\left(X_{TargetPos} - X_{RxPos}\right), \ \left(Y_{TargetPos} - Y_{RxPos}\right), \ \left(Z_{TargetPos} - Z_{RxPos}\right)\right] \ , \quad (189)$$

$$[\Delta v_{X\_Out}, \ \Delta v_{Y\_Out}, \ \Delta v_{Z\_Out}] = \left[\left(Xv_{Emitter} - Xv_{Target}\right), \ \left(Yv_{Emitter} - Yv_{Target}\right), \ \left(Zv_{Emitter} - Zv_{Target}\right)\right] , \quad (190)$$

$$[\Delta v_{X\_In}, \ \Delta v_{Y\_In}, \ \Delta v_{Z\_In}] = \left[\left(Xv_{Target} - Xv_{Rx}\right), \ \left(Yv_{Target} - Yv_{Rx}\right), \ \left(Zv_{Target} - Zv_{Rx}\right)\right] \ . \quad (191)$$

When confined to just the emitter and receiver (as in this case of an ELINT collector), the radial-velocity value is with the emitter's position and velocity set to the targets.

The calculation of *Phase Increment* per pulse and sample is in Equations (192) and (193).

$$DopPhaseInc_{PRI} = WrapValue\left(\frac{-v_{radial} \cdot f}{c} \ , \frac{1}{PRI}\right) \cdot 2\pi \ , \quad (192)$$

$$DopPhaseInc_{Sample} = 2\pi \cdot WrapValue\left(\frac{-v_{radial} \cdot f}{c}, Sample_{Period}\right) \cdot Sample_{Period} \ . \quad (193)$$

11.7.10 Sampling Rate

The sample rate would naturally be relatively high to capture all phases and modulations. However, the aliasing of that bandwidth is by the collector sample rate, and the waveform and modulations are aliasing at the collector's sample rate (50KHz) for this LPRF case. The aliasing is modifying the waveform's generation to apply an Intermediate Frequency (IF) conversion. That lowers the emitter signal frequency relative to the sample rate and generates the signal aliased to the collector sample rate using the waveform's start and end frequencies in Equations (194) and (195).

$$Start_{f\_WF} = \frac{\left(\left(f_{WF\_start\_freq} + f_{emit}\right) \cdot \frac{SampleRate_{Collect}}{f_{Ccntre\_collect}}\right) - \left(f_{centre\_collect} \cdot \frac{SampleRate_{Collect}}{f_{Centre\_collect}}\right)}{2} \ , \quad (194)$$

$$End_{f\_WF} = \frac{\left(\left(f_{WF\_end\_freq} + f_{emit}\right) \cdot \frac{SampleRate_{Collect}}{f_{centre\_collect}}\right) - \left(f_{centre\_collect} \cdot \frac{SampleRate_{Collect}}{f_{Centre\_collect}}\right)}{2} \ . \quad (195)$$

11.8 Signal Generation and Image

The In-phase (I cos component) and Quadrature (Q sin component) generate with the noise for a collector, and the emitter PRI waveforms are added at the received power at the emitter PRI timing forming a matrix of complex numbers. That matrix width is *10ms* wide by several rows in height

equivalent to *1 second (100)*. The red channel is a *Hanning* window Fourier transform as a *spectrogram*, and the green and blue channels are In-phase and Quadrature phase values multiplied by their magnitude. The images are created in this form in a sliding window of PRI as a pulsed synchronized image set forming the (SD_CMRM_Iv1) [54] dataset. In Figure 61 are three images from the dataset generator in their original pixel aspect ratio.

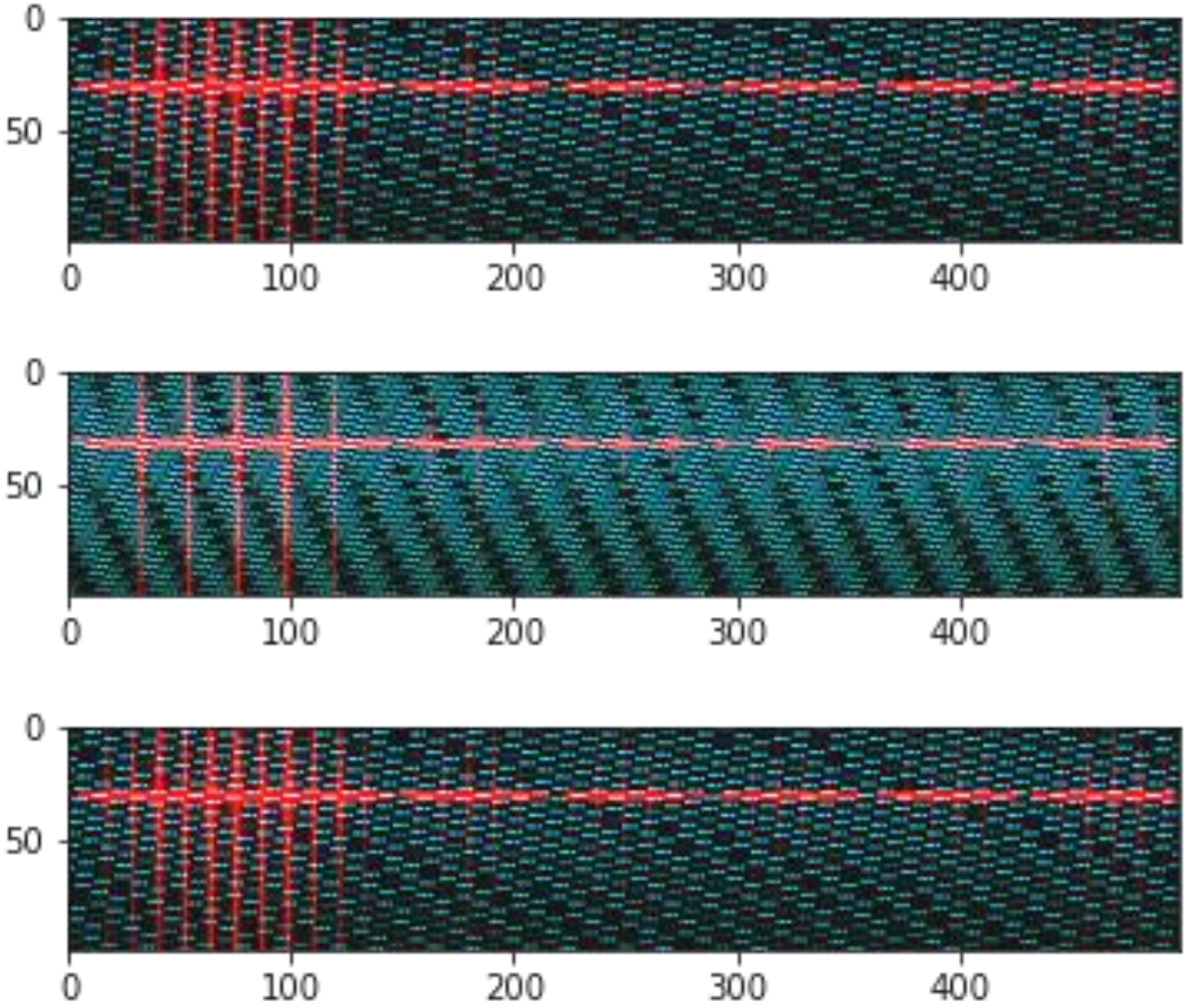


FIGURE 61 THREE EXAMPLE IMAGES FROM THE DATASET GENERATOR

This page is intentionally blank.