OUTSTANDING TECHNICAL ISSUES

Outstanding Technical Issues as interacting with each phase of the following frameworks; The Lifecycle of a weapon system and NATO's Joint Targeting Cycle.

In 2018 the UK presented a paper to the United Nations' Convention on Certain Conventional Weapons in which it suggested that "A compendium of good practice mapped against a weapon lifecycle would provide a clear framework for the operationalisation of the guiding principles by states." The paper refers both to the "Lifecyle of a Weapon System", which is illustrated in phases by the Chair of the CCW's sunrise diagram, and NATO's Allied Joint Targeting Cycle. In its 2020 commentary, the UK equally states that "a compendium would require input from multiple stakeholders across disciplines, including governments, industry and civil society".

As civil society experts, the tech stream of the UK Campaign to Stop Killer Robots has responded to this call. Supplementing their 2021 research paper on problematic AWS technology, the group has mapped those issues which make operationalisation and reliable development of this technology fundamentally problematic in technical and practicable terms. The following table is a visual tool which indicates our concern with AWS' inadequacy at every level. The ubiquity of this concern is evident in the densely concentrated interactions between tech issues and targeting phases and the telling overlap all the way from pre-development (political and international) to post use assessment in its macro-narrative.

The table shows us that almost every process interacts with command and control, targeting, accuracy and the ability to comply with law. The modular nature of this technology creates additional concerns with the 'dual-use' problems involved in its development, application and potential rogue use. This reiterates the fact that oversight and the need for military control (political, legal and strategic) is required due to the nature of computers. Autonomy is evidently problematic without meaningful human control.

AWS' MACHINE LEARNING (ML) SPINE

Weapon architecture based upon neural networks **involves intractable complexity**. AWS' data input requires three characteristics (a weighting or synaptic value, a summing function and a threshold-based output function) but the model's veracity **requires** that all such **inputs are received free from corruption, noise and arrive in sufficiently full detail.**

The weapon's picture-building requires that **sensed signals are labelled, matched with training sets and then weighted before intermediate output can be calculated** in order to establish **patterns**. Model performance dramatically deteriorates in lockstep with the **polling frequency** of the weapon's primary sensors.

The AWS' picture-building **requires compensation routines** to manage for variable intensity in this primary sensed data.

Picture-building is compromised by AI's inherent 'Exclusive-Or Problem' (when **no combination of weighting values triggers thresholds** that have been set during AWS configuration).

AWS' neural networks comprise signal paths traversing from front to back but such ML architecture empirically learns at different, unbalanced and unknowable rates. An ancillary constraint here is that **the unsupervised weapon must minimally have its architecture fixed before training starts thereby constraining the degree of improvement possible** through subsequent training.

Routines will be required to manage the AI characteristic of 'data discounting' whereby **battlefield features with only a small number of data examples may be smoothed to the extent of formlessness** despite those data-points' possibly critical importance (the notion of war being 9/10th inactivity and 1/10th chaotic activity).

Data discounting may also be caused by an overwhelming number of **learning examples in one set** of the weapon's data planes **undoing the training effect on the learning examples in a different dataset.**

Data discounting may be caused by erroneous model sensitivity (the AWS' detection rate) **or an incorrectly set model specificity** (here, the AWS' false alarm rate). Neither of these characteristics can be recognised or tested upon deployment.

The **weapon's network connections will require management** to prevent interaction in non-linear and complex fashions, the forming of new connections and even new neural units while disabling others. **Current ML models restrict learning to the network's top layer** while lower layers remain random transformations that do not exhibit much input capture.

Descent gradients (the weapon's first-order optimization algorithm for finding a local minimum of a differentiable function in the data returned by its sensors) are prone to inherent dilution in these lower layers **providing unpredictable and weak guidance to the overall learning process of the weapon.**

Descent gradients also **demonstrate plateauing of performance** as the gradient reduces.

Management of ML connections remains an intractable challenge. **Configuration routines must balance freezing connections once routines are learnt** (resulting in the AWS being a 'one trick pony') or having them remain open in a state of perpetual learning (resulting then in a unstable ever-learning weapon that the local commander cannot understand).

ML decision-making's 'satisfying' approach is **systemically inappropriate** whereby, for instance, a 90% threshold (here, a data match of 90%) is the defined **trigger point for AWS engagement.**

DATA POLLING AND MANAGEMENT

Data receipt is critical to this AWS model but, as **noise increases in datasets**, class boundaries that separate different class examples become **impossible for the weapon to define** and then separate for ongoing statistical analysis.

Repeatedly sensed data may also push the weapon's neurons into saturation which then **desensitises neurons to all inputs.**

It is also systemically challenging for the **AWS' sensors, the likely sole source of inputs** for its decision processes, to garner consistent information. **Smoke, reflectance, image echo** as well as issues around data intensity intractably complicate processing. Here, class boundaries separating **different data examples resist definition where that data is partial, noisy or indistinct, requiring human intervention** if the weapon is to designate data strings for further statistical analysis.

Data mismatch against its training set, anything that is statistically out of the ordinary (whether the result of feint, by enemy surprise or by inadequate data separation) **will compromise on-board data analysis.**

Error in the weapon's sensing of its current state **must carry forward** in the machine's future learning and future battlefield actions.

Similarly, much of what the weapon has **recently learnt may be invalid** if its environment or its **combat task changes,** a trade-off between a weapon that is 'constantly learning' as opposed to one that is using what is already known to work at the cost of missing out on further improvement.

Inappropriate variability arises from ML's systemic characteristic of 'dropout' whereby learning routines must **omit randomly selected neurones** in order to reduce over-fitting and false correlation.

It is **unknowable from the outset** if a weapon's training data is both **sufficiently relevant** to its y-function (the task that the commander has for each weapon) or of **sufficient size** appropriately to train that weapon network.

The efficacy of ML is also inescapably dependent upon the fit between image classification and the weapon's training: a marginally different set-up or a **marginally different training dataset** to the AWS' sensed data empirically leads to **substantial discrepancies in output.**

Relevant learning data sets with which to 'teach' an unsupervised weapon are **rare;** existing military datasets are either **restricted**, particular to a setting (and therefore **irrelevant**) or **very narrow** task.

A key weakness for the ML model is captured in the acronym **CACE ('change anything, change everything')**; increasing the breadth of training parameters empirically leads to inappropriately random outputs (eg. **manipulating a fraction of an image's pixels** in order to defeat current recognition routines).

AWS efficacy will depend upon **management of rapid but varied obsolescence** within its sensed data. Data obsolescence **leads to systemic instability.** For example, AWS' movement and all relevant navigable space must be **identified, processed and made 'map-ready'** for each of the weapon's representations. The resulting dataset must dynamically be searched **in real-time** to evaluate every available path, chose the optimal path and, finally, goals, values and action selection **must all be revised** to account for that newly selected path.

Polling frequency (the rate of recurrence that the AWS polls new data from its sensors) will determine the **weapon's ability to handle data**, its memory management and processing efficacy. Polling frequency also governs rates of data decay and is complicated by itself being a dynamic and changing function.

Processes are required to **manage the issue of data saturation** in order to prevent the model's **desensitising**. Arbitration contributes to an appropriate model for determining how one sensor input is preferred over others; the configuration issue of settling input intensities. Two variables that may be useless by themselves can be useful together. Similarly, a single variable that is useless by itself can be instrumental with others.

Processes will be required to manage AWS' **data processing order** (and the management of different outcomes according to which data string is processed in which order), managing both the **conundrum of 'signal intensity' as well as 'data habituation'** (the decreased response of ML to repeated stimuli).

Processes will be required to manage **overfitting of sensed data** to that agent's training data or its initial representation upon deployment (the machine's Day One state following configuration).

Processes are required to deal with AWS' **stale data.**

Routines will be required to identify and then manage loose data dependencies (the creation of **inappropriate associations based on mistaken correlations** during propagation forwards and backwards in the AWS' neural network).

Integration of otherwise **separate proprietary coding routines** (that together will comprise AWS' operation) will require routines to manage resulting glue code, pipeline jungles and dead code.

It will be necessary to manage and **arbitrate 'un-learning routines'** in AWS data processes.

Processes will also be required to **manage 'partially observed states'** in the AWS' sensed data and how the weapon backfills appropriately for missing, broken or unexpected data in its matching and decision processes.

ADDITIONAL CODING CHALLENGES

The **'smoothing routines'** required to prepare and then manage data handling necessitates that the coding basis for AWS' operation is that of an **inappropriate statistical 'approximator'.**

Autonomous componentry must recognise and manage coding errors, the challenge of **re-factoring code** in an AWS that is likely **out of communication.**

AWS' coding requires the **artificial** imposition of start and end-states to avoid AWS being in an inappropriate state of 'perpetual intermediacy'. Such coding choices must invariably be constrained by earlier choices. Unless AWS **tasking is very restricted**, critical pathways will remain undiscovered, ignored or misunderstood.

Coding is systemically **poor to deal with ambiguity, context or situational awareness,** like the 1,000 page Norwegian rules of engagement. The information contained within a command **must be coupled to previously given information as well as to information that is to follow.** AWS must factor for **nested structures and conditionals which regularly characterize complex instructions**. Computers cannot build the same hierarchical complexity of human language. The challenge is also that such **categorizations are volatile and change unpredictably** according to new intelligence, new feedback and local input from the weapon's sensors.

Processes are required to **manage AWS' goal-setting and value-setting**. Goals are associated with AWS independent plan of action, while values assess the viability of these plans.

Errors in goal and value setting may have quite unforeseen battlefield consequences that include 'infrastructure profusion' where an **independent weapon** might unexpectedly allocate disproportionately large parts of its **reachable resources into the service of some inappropriately set goal.**

Routines and bias filters will be required to manage the programming issue of 'value-loading' in the AWS, the means of directing actions in an AI agent. The current absence of established mechanism to manage this foundational problem (explicit representation, evolution by selection, associative value accretion, use of motivational scaffolds or reinforcement learning) demonstrates the degree of **invention still required** if ML is to provide an **appropriate deployment spine** for AWS. **Human supervision.**

AWS data-points (for example, in targeting sequences) are invariably revealed incrementally; the systemic issue is therefore when (in that sequence) the AWS can make **an engagement decision** that is efficient and compliant.

Dampening protocols will be required to avoid the AWS oscillating around a desired state and becoming inappropriately **paralysed in its decision-making.**

Human intervention is also required to mediate between locking-in AWS' deployment **assumptions versus modifying** those assumptions in light of **new** instructions, **new** priorities or **newly** sensed data.

Human intervention is currently required to manage and validate download and enactment of software patches in what is otherwise an independent (and thus **incommunicado**) weapon system.

Supervision is currently required to establish **attribution** in weapon performance and behaviour.

Processes will be required to **manage 'anchoring'**, [the degree by which a weapon's initial representation is amended in light of newly sensed information from the platform's sensors.]

Computers continue to **struggle to interpret context**: vision software may identify a soldier walking but is **unable to determine why** the soldier is walking. This also renders autonomous systems particularly **vulnerable to trickery**.

ACTION SELECTION IN AWS DEPLOYMENT

Processes are required to manage challenges around autonomous weapon **'attention'**, prioritising one data string over other sensed information (the **'cocktail party effect' of seemingly irrelevant information that may be key** to that weapon's compliant and useful operation).

Weapon actions must comprise the **appropriate reaction to every relevant sensed stimulus**. AWS cannot offer **intermittent or erratic performance where only specific sensed inputs lead to weapon outputs**.

Routines are required to **manage the firing sequence** for all weapon instructions. Each routine may result in quite **different outputs being triggered depending upon the order** in which command instructions are processed by the unsupervised weapon.

Processes are required to manage the characteristic of the **'undeclared consumer'** whereby interim data decisions are propagated forward to calculate next values in a decision routine. The systemic issue arises from **that next value becoming the undeclared consumer of a prior data decision** which, **not itself a primary datapoint directly taken from sensed data,** may or **may not have been correct.**

Protocols are required to deal with the **AWS' temporal framing,** the time element of an engagement routine (for instance, a sequential engagement decision).

Protocols are required to manage AWS' **fatigue processes** allowing, for instance, **change of decision policy** within that weapon.

Processes are required to **manage the complexity around a veto** including the management in the AWS of a **partial or delayed response, hand-off and withdrawal routines** and the onward communication of states to the local command

Protocols are required to ensure **post-engagement** damage **assessment.**

Routines are required to manage weapon **verification, validation and testing** in a **communications denied** environment.

Extensive Red-Teaming **(attack simulation)** is required for AWS processes.

LEADERSHIP CHALLENGES TO AWS DEPLOYMENT

The difference between a weapon's current state and its desired states is the weapon's **observed** error, the object of the AWS' action selection being to minimize that error. Two technical issues arise. The **pace of this error correction** is not obvious. It depends, for instance, on **how often the error is computed and how much correction is then made on each feedback loop**. Feedback loops, are significantly less effective at modifying a weapon's higher level action selection such as prioritisation, coordination and collaboration.

The AWS must also be **appropriately front-facing** and determine its system state ahead of time.

An unsupervised weapon must not generally forget acquired skills (the notion of **'catastrophic forgetting'**), a recognised limitation of neural network models.

'Hybrid autonomy' suggests the toggling of command between human and machine. Research points to **erratic performance when humans are required to intervene in moments of high stress** or in situations of limited information.

As weapon autonomy is introduced, slack time in battle processes is reduced, scope for rule-bending and initiative is removed and, in the case of weapons based upon ML, the **leader's ability to predict and influence outcomes is lessened.**

Factors arising from International Humanitarian Law (IHL) involve evaluative and contextual judgement for which the **local commander must remain responsible and accountable.** Sufficient control must therefore be retained to enable **situation-specific judgement to comply** with those rules.

Routines will be required to **manage the AWS' failure** modes (outright veto, 'fail too safe', 'fail dangerously' and 'fail deadly') as well as the integration of otherwise independent assets into the commander's wider portfolio of battle plan assets.

Taniel Yusef, Paddy Walker; on behalf of the Tech Developers Working Group, UK Campaign to Stop Killer Robots.