Tactical Agility for AI-Enabled Multi-Domain Operations

Eric M. Sturzinger, Shilpa A. George, and Mahadev Satyanarayanan

Carnegie Mellon University, Pittsburgh, Pennsylvania, United States

ABSTRACT

Commanders must remain agile and adaptive in the future Artificial Intelligence (AI)-enabled multi-domain battlespace, where critical decisions are made at the tactical edge. Over-reliance on static, cloud-centric approaches to Machine Learning Operations (MLOps) compromises such agility and adaptability. These systems must operate in a dynamic threat environment, and learn to detect novel threats during operation. They must be able to perform this learning through the execution of tactical MLOps under austere and degraded conditions, especially limited wireless network bandwidth. In response to these requirements, this paper describes *Hawk*, a system that leverages edge proximity for rapid and iterative execution of the Observe stage of the Observe-Orient-Decide-Act loop. Central to this architecture is the use of tactical *cloudlets.* These mini data centers provide cloud-like computing resources without the communication latency to exascale data centers. Hawk enables a human to guide MLOps at low cognitive load, thus enabling an operational objective to be achieved at speed and scale while remaining usable and explainable.

Keywords: Sensing, Machine Learning, Video Analytics, Cloudlet, Edge Computing, Cloud Computing, Low Bandwidth, High Latency, Austere Environments, Hawk

1. INTRODUCTION: TACTICAL AGILITY

The US Army's recent inclusion in formal doctrine of its new concept known as Multi-Domain Operations (MDO)[1] necessitates a wide variety of AI-enabled and autonomous systems to augment existing Soldiers and Units. Much of the recent literature describes different applications of powerful AI systems across the air, land, sea, space, and cyber domains to increase the lethality of the joint force. The migration toward Joint All Domain Command and Control (JADC2) and the desire to achieve information dominance require a vastly greater scale of sensing platforms across all domains. Adaptability, robustness, and trust are frequently identified as high priorities for AI and autonomous systems that are deployed in future combat environments.

A key goal of AI-enabled systems is to accelerate the OODA (Observe-Orient-Decide-Act) loop. To achieve this, operational decisions with respect to mission changes must be made at the lowest levels (tactical edge) in the shortest amount of time, with the most accurate and fresh information available. Although more aggregated information about a threat may be available at higher echelons, escalating decisions to them from the tactical edge may suffer unacceptable communication latency and reliability. At the speed and scale of future MDO, making decisions at the lowest possible echelon is ideal. *Tactical agility* thus emerges as a key attribute. We define tactical agility as the ability of a unit employing such a system to rapidly modify and adapt existing mission parameters in order to preserve or enhance its immediate operational effectiveness.

Today's cloud-based AI/ML systems have poor tactical agility. They take a phased approach to the end-to-end processing pipeline, with rigid decoupling of phases. The phases include (a) data collection and creation of a training set (including labeling), (b) training a deep neural network (DNN) model, (c) deployment of the model, and (d) inference of operational data with that model. This decoupled and cloud-centric approach was adequate for the civilian context in which it was conceived, but lacks the tactical agility that is crucial for MDO.

The importance of agility and adaptation for mobile computing systems that embody AI was first identified over a quarter of a century ago.[2] What is new today is the realization that AI/ML processing needs to be rethought from first principles to achieve these attributes. This paper introduces a new distributed AI architecture whose functions are tightly-coupled and pipelined in order to maximize tactical agility. Central to this

Send correspondence to Eric M. Sturzinger, e-mail: esturzin@andrew.cmu.edu.

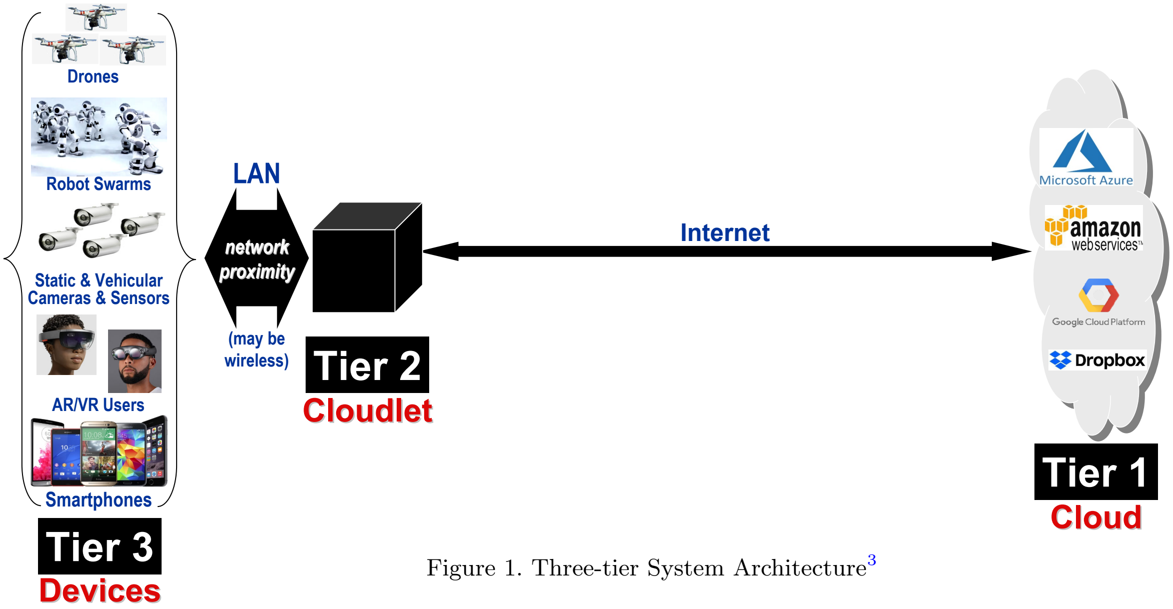

Figure 1. Three-tier System Architecture[3]

architecture is a robust set of tactical *cloudlets,* which are edge computing nodes with various levels of capability that support all required local workloads. This architecture is best understood relative to the tiered model of computing shown in Figure 1. Tier-1 is the cloud, providing strategic assets, information aggregation and archiving, and failure-resilience that are not available at other tiers. Tier-3 consists of sensor-rich mobile or static devices that are embedded in the battlespace. These provide raw data capture, preprocessing, and actuation (for cyber-physical systems) or interaction (for cyber-human systems). Tier-2 can be viewed as forward deployed infrastructure that "brings the cloud closer." The *network proximity* of Tier-2 to Tier-3, and the cloud-like computing resources available at Tier-2, together enable new tactically agile AI-enabled applications that are simultaneously bandwidth-hungry, latency-sensitive, and compute-intensive.

This paper explores how to ensure tactical agility of AI and autonomous systems in dynamic operational environments as they continue to augment units in all domains across the joint force. Appendix A provides contextual background on the operational scenario and its critical components. Section 2 describes *Hawk,* an AI/ML system that achieves tactical agility by seamlessly integrating data capture, inferencing, transmission, labeling, and training. Generalizing from Hawk, Section 3 details the system architecture and processes that are essential to achieving tactical agility in AI systems. Section 4 describes related work, and Section 5 summarizes our conclusions and vision for future work.

2. HAWK: A PIPELINED ARCHITECTURE FOR MLOPS

Hawk is an AI/ML system that tightly couples core internal functions that traditional cloud-based systems prefer to strongly decouple for purposes of economies of scale and standardization. The goal of Hawk in a tactical environment is to reliably detect an object for which we have very few existing samples prior to a reconnaissance mission. It achieves this goal through the tight coupling and interdependency of core system processes. The end to end workflow involves sensors and two kinds of cloudlets, as depicted in Figure 2. We describe these components below.

Sensor: Converts environmental data into digital format for processing. It is the single platform that enables data collection, without which AI systems could not sufficiently perform their tasks. To be more specific, a sensing node senses "raw" data from the environment that has to be cleaned, refined, compressed, or otherwise modified for further analysis. Although human-understandable data could be derived from this raw data at a different node, we specify the sensing node to be the first node where new, raw data enters the system.

Home-Cloudlet: This node aggregates data samples collected from many different sensors in a given area/system

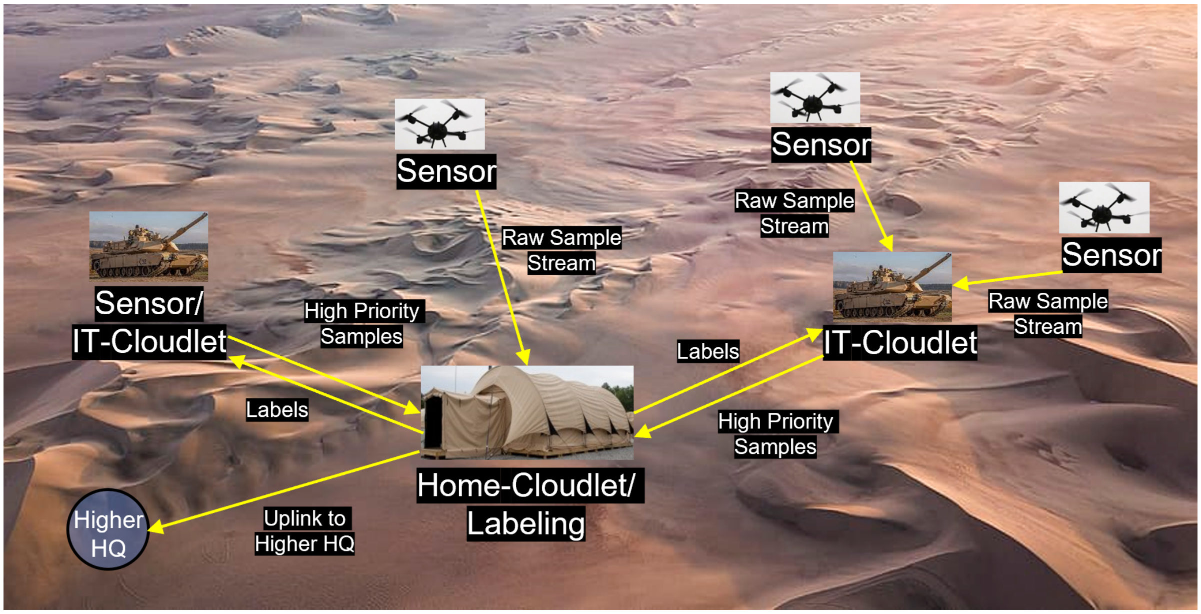

Figure 2. Operational Employment of Hawk

hierarchy. The node is typically where the mission is launched from and is the primary Command and Control (C2) node for data fusion, mission progress, and other system statuses. Although humans may be able to access such information from other nodes, as described below, this node is meant to perform core mission functions that enable all other functions and processes of the system work seamlessly together in a tightly-coupled manner.

Inference & Training Cloudlets (IT-Cloudlets): As its name implies, an IT-Cloudlet performs two distinct functions. First, it continuously receives a steady flow of samples from one or more sensors and and performs inferencing on them. Second, it performs periodic self-improvement of the model used for inferencing by retraining it with an augmented training set that includes freshly acquired and labeled data. At each IT-Cloudlet, prioritization and transmission of data to the Home-Cloudlet for labeling happens continuously. From time to time, when appropriate conditions have been met in terms of training set augmentation, training of a new model is triggered on an IT-Cloudlet. This new and improved model is locally deployed as soon as it is available.

Workflow: As we step through an entire Hawk mission, we will discuss several key functions and their role in the overall system. As depicted in Figure 2, IT-Cloudlets (whether in a fixed location or in a ground vehicle) and various sensors (within ground vehicles or as separate drones) are deployed to the operating locations from the Home-Cloudlet location containing some Soldier presence. This figure gives an example of the coupling or decoupling of Tier 2 and Tier 3 systems shown in Figure 1. The sensors, drone or ground vehicle, are considered Tier-3 while the IT-Cloudlet that runs the inference process is considered Tier-2. Note that it is common for Tier-2 and Tier-3 to be contained within the same physical platform. The distribution and scale of the IT-cloudlets and different types of sensors are almost entirely mission dependent. However, there are some obvious constraints, such as the wireless network range from a drone to a ground vehicle or from ground vehicle to the Home-Cloudlet. In more detail, hardware, data link protocols, on board power, and other technical characteristics of the deployed system play a significant role in the operational limitations of the system that Soldiers would need to thoroughly understand. We show in the diagram that the Home-Cloudlet is in relatively close proximity to the IT-Cloudlets where network bandwidth is likely not a limiting factor of the system, however, in the extreme worst case such as an IT-Cloudlet being deployed in geo-stationary orbit, network bandwidth is by far the limiting factor of the system's performance.

Based on Mission, Enemy, Terrain & Weather, Troops, Time Available, and Civilian Considerations (METT-TC), a small unit of Soldiers would first determine the intended target object(s) that they need Hawk to reliably detect. We will assume they have been provided with a small set of images (approximately 10-20) of a T-101,

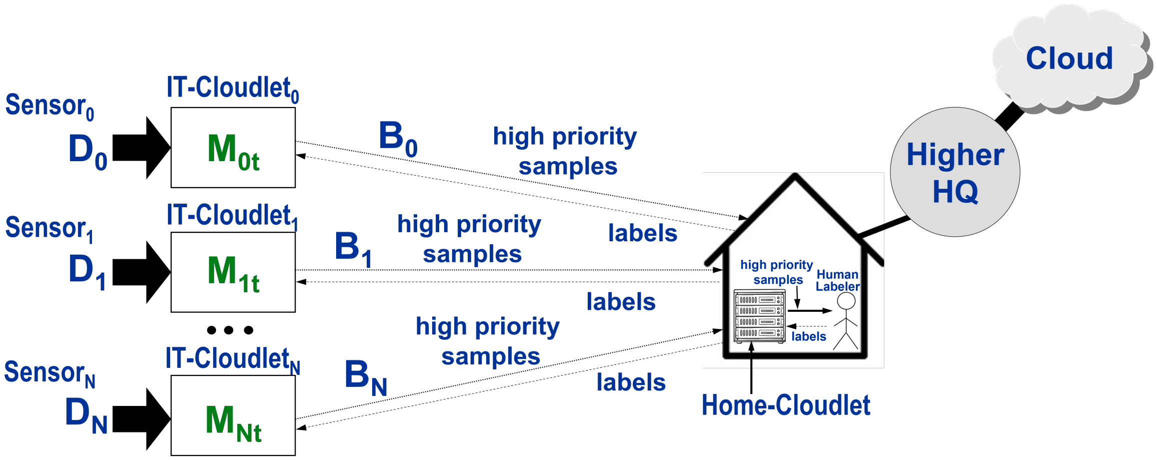

Figure 3. Hawk Logical Diagram

a new enemy tank, for example. The unit leverages an ML model that has been pre-trained on data of many different object classes to take advantage of its trained "feature detector", a neural network architecture that allows for the detection of various general features of many different types of objects. It performs the initial "retraining" of this model on the given small dataset. Because of the small size of this dataset, the retraining should only take a few minutes. Once trained, the Soldiers would test the new model on a hold out, or test set to ensure the model achieves some reasonable level of performance, keeping in mind that they've only trained it on a very small dataset (which implies a significantly reduced expectation of accuracy). If the initial model, M_0, is suitable, then it is deployed to all IT-Cloudlets planned for the upcoming mission. If such cloudlets have already been staged in their respective locations, the model would of course need to be transmitted wirelessly to them. Thus, in severely limited network bandwidth environments, this initial model *must* be deployed on the IT-Cloudlets prior to departure from the Home-Cloudlet location.

Figure 3 shows the logical structure of Hawk and the flow of information across the entire system. Sensor data is processed by the models running on the IT-Cloudlets, where a small number of high priority samples are then transmitted to the Home-Cloudlet for inspection and labeling by Soldiers. The labels are sent back to the IT-Cloudlets to store in preparation for future model retraining iterations. D_0 represents the data rate that Sensor$_0$ transmits raw data to IT-Cloudlet$_0$ while M_{0t} represents the current model running inference on IT-Cloudlet$_0$ at time t. B_0 represents the rate at which high priority samples are transmitted to the Home-Cloudlet, which can vary greatly depending on mission configuration as well as availability and reliability of network bandwidth. Notice in the top right of Figure 3 that there is a Higher Headquarters (HQ) node connected to the Home-Cloudlet. One can imagine many physical and logical network hops lie between the Higher HQ node and the cloud itself. This diagram reinforces the point that the tight coupling of AI/ML processes for tactical agility cannot be realized with dependencies on cloud resources.

The mission is now ready to execute as all IT-Cloudlets are running the deployed initial model and all sensors are prepared to monitor their geographic areas. Once the commander so determines, the mission begins and the sensors begin transmitting data to the IT-Cloudlets running inference processes. Refer to Figure 4 to follow the sequence of events of a toy mission example as a function of time where time flows from top to bottom and the processes that execute on each cloudlet from left to right. In this mission, we have three total sensors and two IT-Cloudlets. The first events to occur are Sensor 1 and Sensor 2 collecting Sample A and Sample B, respectively. We call Infer. A and Infer. B the inferencing of the first and second samples across the set of IT-Cloudlets. Because these samples are transmitted to the same IT-Cloudlet (1), this cloudlet must perform local "Sample Prioritization", which is a form of active learning. Upon inference, each sample is assigned a score (or pseudo label) representing the probability that the given sample contains the threat object identified prior to mission start. The set of samples that have been inferenced at any point in time will be prioritized according to this score and only the highest scoring samples are then transmitted to the Home-Cloudlet. The rate at which the prioritized samples are sent to the Home-Cloudlet depends on whether network bandwidth is abundant and

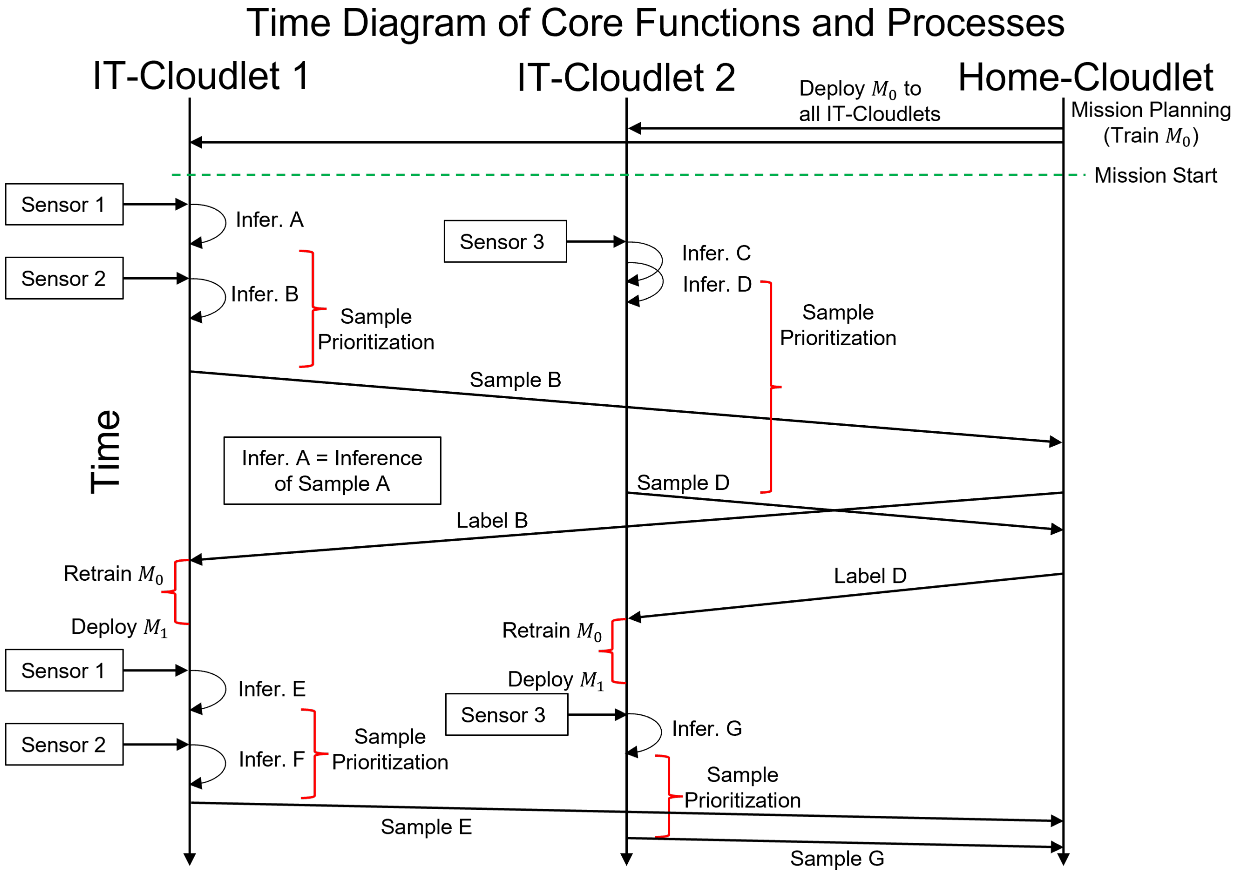

Figure 4. Time Diagram of a Toy Hawk Mission

reliable as well as the rate at which the Soldier(s) can label. If network bandwidth is not a concern, then the IT-Cloudlets will only transmit samples to the Home-Cloudlet at the same rate that the Soldiers can label. However, if the network is limited or unreliable, the IT-Cloudlets will transmit as many samples as the link data rate allows. We see from the diagram that Sample B had been assigned a greater score, and is thus transmitted to the Home Cloudlet. Sample B is labeled at the Home-Cloudlet, which is immediately transmitted back to IT-Cloudlet 1. Let us assume for simplicity that we would like to retrain our inference models after every single "positive" label received (this is tunable in reality). This means that after receiving the label for Sample B, IT-Cloudlet 1 retrains model M_0, which requires some period of time, after which M_1 is produced. The key here is that while model M_1 is being trained on IT-Cloudlet 1, model M_0 is still inferencing incoming samples from Sensor 1 and Sensor 2 (although it is not depicted in this diagram for visual clarity). Samples E and F are transmitted to IT-Cloudlet 1 to be inferenced with Sample E being selected from the prioritization process and sent to the Home-Cloudlet for labeling, repeating the process. Also critical to note here is that similar events occur in parallel with Sensor 3 and Samples C, D, and G via IT-Cloudlet 2. Notice that prioritized samples are transmitted from multiple IT-Cloudlets to the Home-Cloudlet concurrently while also receiving respective labels for previous samples from the Home-Cloudlet. Once conditions are met, concurrent training and inference occur on the same IT-Cloudlet until the new model has completed training. Each subsequent new model is deployed with the assumption that it will perform in a superior manner than its predecessor because it will have been trained on an augmented training set containing many more examples of the intended threat object. Data collection off the sensor platforms are also tightly integrated into this iterative process. The final key aspect of the system is the human labeling function. Hawk only adds positive examples to a training set if those examples have been confirmed as positive by the human labeler. The labeling rate of the human directly influences training set growth, hence the triggering of new model training, and thereby the rate of self-improvement.

Extensive experimental evaluation of Hawk has shown impressive results in terms of the number of positive samples detected even under extremely austere network conditions. Even at 12 kbps, Hawk is effective on data from drone surveillance, planetary exploration, and underwater sensing. Hawk is able to discover up to 87% of the positives that would have been created by a brute force model. Such a model would have been created by fully supervised learning, which involves transmitting all data to the cloud and labeling it there.

We have now described the components, functions, and sequential processes of Hawk. Why can this system not be designed leveraging the scale and resources of the cloud? Classic ML pipelines strongly decouple data collection, curation, model training, and inference. In our context, because the data we must inference is at the tactical edge, we would need to ensure a stable internet connection from sensors all the way to the cloud where our inference and training would occur. Not only would the inference latency be extremely high from the perspective of the Soldiers positioned locally at the Home-Cloudlet, the number of dependencies on network paths well beyond the unit's control would greatly impact success of the mission. Similarly, the cloud is optimally designed for training and inferencing of very large datasets, which is the opposite of what the Hawk system needs to perform at a suitable level. While traditional ML approaches clearly do not apply to this scenario, a tightly-coupled, interdependent, and parallel AI/ML system enabled by edge computing best achieves the desire for tactical agility in future MDO.

3. TACTICAL AGILITY FOR AI-ENABLED OPERATIONS

In Section 2 we introduced Hawk and described its components, functions, and sequential and parallel processes. Hawk is designed as a specific architecture for a specific purpose and mission set. In this section, we discuss unique characteristics of a system such as Hawk and explore the broader implications of the design on tactical agility. We first describe the simultaneous and parallel processing that are critical to a tactically agile system. Second, we discuss the tightly-coupled interdependencies that were briefly mentioned in the previous section. Finally, we summarize the advantage edge computing has over the traditional cloud paradigm in the context of tactical agility for MDO.

Traditional ML model development separates training and inference as sequential steps in the ML Ops process. Technically, Hawk follows the same paradigm. However, the difference is while the IT-Cloudlet uses model M_0 for inference, it may be currently training model M_1. This ability to train a new model while simultaneously performing inference of incoming samples from sensors is a critical design decision that saves time, which allows for the system to achieve a much greater level of performance in a shorter period of time overall. Further, sample prioritization runs concurrently with inferencing, thus constantly reprioritizing samples according to their score. With respect to network transmission, high priority samples are sent to the Home-Cloudlet from the IT-Cloudlets while text-based labels of previous samples are sent from the Home-Cloudlet back to the IT-Cloudlets. Further, model and network functions occur concurrently, unless the training process happens to be waiting to receive its n^{th} sample for retraining. Human labeling effort takes place at Soldiers' discretion, concurrently with all of the above processes. Execution of all these tightly-coupled processes in a tactical area without requiring cloud access greatly enhances tactical agility. The loosely-coupled and serialized phasing of traditional cloud approaches combined with long latency/low bandwidth simply cannot achieve such tactical agility.

We now discuss the impact of the tight coupling and interdependence of the subsystems of an AI/ML system such as Hawk. We explained how several primary system functions execute in parallel to avoid not only network latency but also unnecessary process serialization (inference and training). Figure 5 depicts the interdependencies of an iterative learning system such as Hawk. The clearest dependence in a system learning in real time is that of the performance of the currently inferencing model version on the data it had access to for training. If a given model never gets retrained, it will not improve, but will just perform at the same level of accuracy for the duration of the mission. The inference accuracy is partially dependent on the ability of individual sensors to generate a representative sample distribution. If the sensors are quite far from each other with almost no chance of scanning the same terrain, then this is not an issue. However, in concentrated sensor clusters, such as a drone swarm, it is very likely many sensors produce duplicate samples, wasting processing time on the IT-Cloudlets. To meet the proper conditions for model retraining, an IT-Cloudlet must receive some number of labeled samples from the Home-Cloudlet. This is directly dependent on the reliability of the network connections between the two as well as the rate at which the humans can label. Moreover, the number of labeled samples (containing

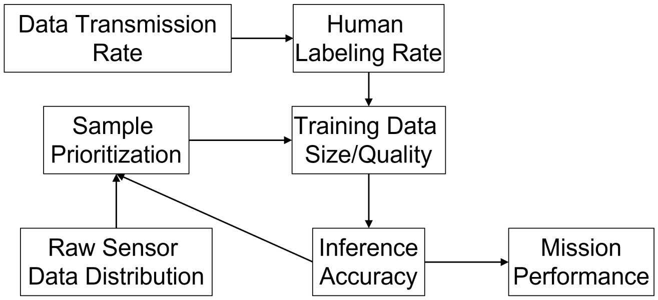

Figure 5. Dependency Graph of Hawk Core Functions

the threat target) is heavily dependent on the accuracy of the currently inferencing model and the ability of the sample prioritization algorithm to transmit the samples most likely to contain the target object. A cloud-based approach would not be able to support a tactically-agile AI/ML system so tightly-coupled as this.

The critical feedback loops present in a system like Hawk are between inferencing, sample prioritization, transmission, labeling, and training. Without the fast, effective, and efficient function of this cycle, the proper feedback will not be recirculated into the system. This is critical as a system that learns in real time must be able to leverage relevant, fresh, and quality data as quickly as possible in order to achieve its mission objectives. While effectively closing the feedback loop is difficult, it is even more challenging to produce a feedback loop that provides quality information and actually *improves* the performance of the system. This is why all of the tightly-coupled and interdependent system functions must be designed and engineered appropriately to the respective mission set. Tight feedback loops for systems that learn in real time is largely a foreign concept to how the joint force currently operates. To build and maintain trust in AI/ML systems in general, Soldiers must experiment with such new systems to develop familiarity and comfort as they would with a new rifle or tank.

Designing and engineering tactical agility into an AI/ML system will not be easy, however, it can be done with the right vision. Leveraging the scale, low cost, and standardization of cloud resources is in many cases the correct answer, especially for business applications or offline data processing. However, with respect to time-sensitive reconnaissance operations with limited knowledge and data of potential enemy targets (e.g. tanks, missile launchers, etc.), an edge computing approach is optimal. In such missions, there is often a severe mismatch between the incoming sensor data rate and the much lower backhaul bandwidth to higher HQ and the Internet. Degraded, Intermittent, and Limited (DIL) conditions may also constrain bandwidth to the point of rendering cloud-centric systems ineffective. Edge computing employs the power and flexibility of tactical cloudlets that enable the effective and efficient operation of an AI/ML system that can learn in real time (with minimal human cognitive load) through a set of tightly-coupled and interdependent core functions. The nature of such a system renders it tactically agile.

4. RELATED WORK

The role of cloudlets in hostile environments was first explored a decade ago by identifying the inherent limitations of the cloud for applicable domains.[4] The literature concerning the need for edge computing in a tactical environment has exploded in the past several years due in part to the recognition that AI/ML systems will be force multipliers at the lowest echelons for MDO. Relative to that large body of literature, this paper shows through a specific in-depth case study exactly how edge computing enables tactical agility to be achieved. We are not aware of any other work that achieves the tactical agility of a Hawk-like system.

Although these systems have significant upside potential in their ability to enhance information dominance and lethality, they do require additional computing, storage, network, etc. resources to function effectively and reliably while being trusted by Soldiers. In Task Objective three of his commentary, de Czege discussed broader implications of current strategy and the need for the US to modernize our obsolete way of war, by adapting to the revolutionizing impact of the technology of war."[5] More technically focused related work consists predominantly within four main areas: network, edge computing generally, edge learning approaches, and human-machine teaming, all critical aspects of ensuring tactical agility.

4.1 Network

To realize the potential that a growing number of AI/ML systems can bring to the edge, a robust, resilient tactical network architecture must be able to support their data needs. Kiser et al. discussed how the Air Force's Combat Cloud (CC) would enable data distribution and information sharing within a battlespace. The CC must be self-forming, self-healing, gratefully degradable and redundant, which would enhance Command and Control (C2) and operational agility for joint force and allies.[6] Perry et al. described a tree-search approach to compute Value of Information (VoI) of individual personnel in a tactical environment, which determines whether a given piece of information is relevant to an operator.[7] This calculation resulted in an efficient mechanism to minimize excess network traffic in austere environments. Barz et al. explored the characteristics of adaptive communications at the tactical edge and focused partly on "joint orchestration of network, processing, and AI" to improve decision-making and other actions critical to this domain.[8] Inherent in this work was the interdependency between networks and AI systems that we discussed earlier. Mohlenhof et al. proposed a reinforcement learning-based approach to improve the use of network resources in DIL networks. RL agents were trained to accomplish the goal, which was to optimally utilize tactical network resources.[9] Judd et al. described the ongoing development of SMARTnet, a system designed to transform, prioritize, and control the flow of information across tactical DIL networks by understanding the current military context through the use of AI/ML approaches.[10]

4.2 Edge Computing

Edge computing is undoubtedly the key enabler for tactical agility via AI/ML operations. Satyanarayanan provided a contemporary assessment of where edge computing fits in the long-term evolving computing paradigm and described a vision of its most applicable use cases moving forward.[11] Ruvinsky et al. defined an AI system design and framework that leverages mobile High Performance Computing (HPC) hardware, ML, and Internet of Things (IoT) to generate near-real-time analytics in a tactical operation environment.[12] Park et al. applied tactical HPC systems to the compute-intensive urban reconnaissance use case to enhance situational awareness and tactical intelligence.[13] Alshabana, et al. measured the performance tradeoffs across different GPU models and deep learning workloads.[14] They showed the nuanced implications of using GPUs suitable for the tactical edge compared to those more commonly used in the large-scale datacenters. Perazzone et al. studied adaptive resource allocation for resource-constrained environments by determining where to place inference task execution which ultimately cause tradeoffs between accuracy and latency.[15] Lewis et al. introduced the concept of provisioning tactical cloudlets in austere environments where resource-constrained edge devices could not provide adequate computing resources for the given tasks and cloud resources would introduce unacceptable latency.[16] Chin et al. introduced the TAK-ML (Tactical Assault Kit-Machine Learning) framework that combines Android TAK (ATAK) handheld devices as well as TAK servers to offload the cognitive burden on Soldiers and commanders of the myriad data collection and analysis tasks in tactical environments.[17]

4.3 Edge Learning approaches

While the previous subsection described the critical need for edge computing in general, we now focus on specific applications to AI/ML systems of edge computing. Langleite et al. conducted a study on explainable AI and addressed the need for big data solutions to support AI on tactical infrastructure.[18] The authors designed their experiments around the foundational human cognition concepts of fast and slow thinking to describe how humans can use such powerful computation. Dasari et al. discussed deep neural network optimization approaches for tactical unmanned ground vehicles.[19] This type of analysis will continue to become more critical as more AI-enabled systems are deployed to the tactical edge. Misra et al. introduced an approach to dependable ML-based inference on resource-constrained edge devices.[20] This method used edge devices to identify and leverage suitable peer IoT nodes for state sharing and improvement of intelligence tasks. Geerhart et al. explored model normalization in resource-constrained environments through reducing the number of parameters and choosing a model with a small computational cost.[21] Busart studied how federated learning could be conducted across edge computing devices connected with limited bandwidth networks.[22] His results showed that the specific federated learning architecture allowed for continuous learning upon initial deployment, which improves performance over time. Cirincione and Verma described the applicability of federated learning to MDO, specifically at the tactical edge.[23] They also defined the operational factors impacting the deployment of ML systems along with proposing strategies for AI/ML deployment techniques. Blasch, et al. studied the merging of AI/ML advancements with sensor data fusion (SDF), in part documenting the impacts of such merging to the success of future multi-domain operations.[24] Toth and Hughes recounted initial success with sensor and data interoperability between several countries' systems, however, it led to an overwhelming volume of data to humans on the ground. They describe CATE (Collaborative AI at the tactical edge), which "is an effort to develop a prototype AI/ML architecture that enables simple, rapid integration multi-agent AI technology into the processing, exploitation, and dissemination chain" at the edge.[25] The authors in Ref. 26 defined a hybrid-cloud architecture that would enable iterative model retraining in tactical environments.

4.4 Human-machine teaming

During operation AI systems requires both automated processes and humans interact to some extent, thus we provide recent related work on human-machine teaming. Rawat presented challenges and perspectives on AI-enabled tactical autonomy that analyzed in part human-machine interfaces to improve trust, efficiency, and efficacy of autonomous systems.[27] Clarke and Knudson "developed a framework for analyzing task-to-technology matches and team design for military human-machine teams". Situational awareness, decision making, and team dynamics were the primary topics around which the framework was based.[28] Warren and Hillas examined the concept of trust and other related aspects of human-machine teaming along with the inherent nature of AI technologies being dual-use, both directly having significant overlap with our work.[29] Schaefer et al. from Army Research Labs (ARL) described in detail four areas of critical research relating to human-autonomy teaming at the tactical edge: enabling Soldiers to predict AI, quantifying Soldier understanding of AI, Soldier-guided AI adaptation, and characterizing Soldier-AI performance.[30] These areas directly relate to Hawk as described and intersect with the implications of an AI/ML system that can learn in real time.

5. CONCLUSION AND FUTURE WORK

In this paper we have described Hawk, an AI/ML system that continuously improves its ability to accurately detect targets determined by Soldiers given limited data, and discussed its broader implications for tactical agility in MDO. It is a distributed AI/ML architecture that leverages a tightly-coupled set of functions such as labeling, sensing and data collection, model training, model inference, and data transmission along with the critical system-level parallelism due to the system's interdependent nature. We also addressed the key enabler for tactical agility that is robust edge computing through the employment of tactical cloudlets. These cloudlets host a diverse set of computing workloads that allow for continuous, real-time, and online learning critical for a complex and dynamic operational environment. While leveraging the scale and resources of the cloud for static environments with high bandwidth network connections is ideal, austere environments at the tactical edge require a uniquely designed and deployed edge computing architecture to maintain an advantage over an adversary. While fielding and deploying AI systems at scale may seem complex, it has the potential to become a revolutionary change in military doctrine if designed and engineered in a manner that minimizes additional human cognitive effort and maximizes hybrid human-machine tactical agility in MDO.

In future work, we will expand on the application of Hawk to specific scenarios with potential derivative systems that may perform quantitatively better with less human supervision. We will also study the tradeoffs between more or less human involvement both in the mission planning and monitoring phases to ensure we do not remove the human from functions requiring ethical evaluation as well as where expert knowledge and experience may be more useful than automation.

ACKNOWLEDGMENTS

This material is based upon work supported by the U.S. Army Research Office and the U.S. Army Futures Command under Contract No. W519TC-23-C-0003. The development of Hawk was supported by the Defense Advanced Research Projects Agency (DARPA) under Contract No. HR001117C0051, and by Lockheed Martin under Award MRA19001RPS006. This work was done in the CMU Living Edge Lab, which is supported by Intel, ARM, Vodafone, Deutsche Telekom, CableLabs, Crown Castle, InterDigital, Seagate, Microsoft, the VMware University Research Fund, and the Conklin Kistler family fund. Any opinions, findings, conclusions or recommendations expressed in this document are those of the authors and do not necessarily reflect the view(s) of their employers or the above funding sources.

APPENDIX A. KEY COMPONENTS OF AN AI-ENABLED ENVIRONMENT

For the foreseeable future, the vast majority of AI and autonomous systems will be used for intelligence, surveillance, and reconnaissance (ISR) mission sets. This is because the technologies underlying such mission sets, especially with respect to computer vision, have matured rapidly over the past decade. A joint force can leverage powerful algorithms to detect and track relevant targets for long periods of time, possibly traveling at high speeds over long distances. Moreover, ISR systems are ideally suited to leverage AI systems to conduct tasks that are otherwise dangerous, monotonous, or boring, especially on longer timespans. In the context of ISR missions, the terms below characterize key components of the operational environment.

Sensor: A sensor is any physical device that converts environmental stimuli such as electromagnetic energy in given frequency bands to electronic, and more specifically, digital data. Sensors can be as complex as a reconnaissance satellite or as simple as a microphone or tiny camera. They are the primary mechanism through which US forces collect tactical, operational, and strategic information separate from human intelligence or other information collection. Sensors can be used to provide general information across both large geographic areas as well as specific information within a small area. Overall, sensors are heterogeneous across many variables, including the domain in which they operate, and can be tailored to provide value to any anticipated mission set. The number of sensors in future MDO is expected to scale greatly, especially with respect to (semi-) autonomous systems and those that are generally expendable/attritable. The primary objective of a sensor is to generate the highest quality of data used for analysis and a decision whether to act on it. This paper primarily addresses AI systems leveraging multiple sensors.

Shooter: A shooter is a platform that most associate with military operations: a tank, aircraft, ship, mortar system, artillery, even an individual Soldier with a rifle. As expected, shooters are heterogeneous as well in their scale, speed, lethality, and domain. In every case, in some form or another, a shooter requires information derived from a sensor, whether from a human or a machine. Although in practical terms shooters typically have many sensors co-located on platform, we note the “shooter” is simply the mechanism which applies some kinetic or non-kinetic effects on a target, whether it leveraged information from an onboard or offboard sensor.

C2 Node: This node is usually a separate intermediate point where, based on information provide by a sensor, a determination is made by a human as to whether to engage a target. This is where the nuanced difference between “human-in-the-loop”and “human-on-the-loop” matters. With the former, a shooter will not engage a target without first receiving an explicit order from the C2 node while with the latter, the human at the C2 node is simply merely aware or notified that a sensor has received information that is being passed to the shooter for engagement. From an ethical perspective, the latter method will likely only be used in scenarios where risk of collateral damage is minimum.

Intermediate Processing Node: This node performs high-level computation on data such as format conversion, compression, or otherwise change the data such that it’s compatible with what C2 or shooter node requires, or even other sensing nodes. We include it in our list here because it may be the case that the sensor-C2-shooter chain may depend on this node given its computational capabilities, security features, physical location, network capacity, echelon placement, etc. These nodes are often co-located with a sensor, C2, or shooter node but can also be a stand-alone node as previously described. Processing nodes will only become more ubiquitous on the battlefield as edge computing systems are developed and fielded to units at higher echelons and processing-heavy AI systems require their support to generate the greatest operational advantage.

REFERENCES

[1] US Army Training and Doctrine Command (TRADOC), [*The US Army in Multi-Domain Operations 2028*], United States Army, Fort Eustis, VA (2018).

[2] Noble, B., Satyanarayanan, M., Narayanan, D., Tilton, J., Flinn, J., and Walker, K., "Agile Application-Aware Adaptation for Mobility," in [*Proceedings of the 16th ACM Symposium on Operating Systems Principles*], (October 1997).

[3] Satyanarayanan, M., Gao, W., and Lucia, B., "The Computing Landscape of the 21st Century," in [*Proceedings of the 20th International Workshop on Mobile Computing Systems and Applications (HotMobile '19)*], (2019).

[4] Satyanarayanan, M., Lewis, G., Morris, E., Simanta, S., Boleng, J., and Ha, K., "The role of cloudlets in hostile environments," *IEEE Pervasive Computing* **12**(4), 40–49 (2013).

[5] De Czege, H. W., [*Commentary on" The US Army in Multi-domain Operations 2028"*], JSTOR (2020).

[6] Kiser, A., Hess, J., Bouhafa, E. M., and Williams, S., "The combat cloud: Enabling multi-domain command and control across the range of military operations," tech. rep., AIR COMMAND AND STAFF COLL MAXWELL AFB AL MAXWELL AFB United States (2017).

[7] Perry, J. M., Galliera, R., and Suri, N., "A machine learning approach to the determination of value of information to operators and applications on the tactical edge," *Procedia Computer Science* **205**, 137–146 (2022).

[8] Barz, C., Cramer, E., Fronteddu, R., Hauge, M., Marcus, K., Nilsson, J., Poltronieri, F., Tortonesi, M., Suri, N., and Zaccarini, M., "Enabling adaptive communications at the tactical edge," in [*MILCOM 2022-2022 IEEE Military Communications Conference (MILCOM)*], 1038–1044, IEEE (2022).

[9] Möhlenhof, T., Jansen, N., and Rachid, W., "Reinforcement learning environment for tactical networks," in [*2021 International Conference on Military Communication and Information Systems (ICMCIS)*], 1–8, IEEE (2021).

[10] Judd, G. B., Szabo, C. M., Chan, K. S., Radenovic, V., Boyd, P., Marcus, K., and Ward, D., "Representing and reasoning over military context information in complex multi domain battlespaces using artificial intelligence and machine learning," in [*Artificial Intelligence and Machine Learning for Multi-Domain Operations Applications*], **11006**, 49–63, SPIE (2019).

[11] Satyanarayanan, M., "The emergence of edge computing," *Computer* **50**(1), 30–39 (2017).

[12] Ruvinsky, A. I., Garton, T. W., Chausse, D. P., Agrawal, R. K., Yu, H. F., and Miller, E. L., "Accelerating the tactical decision process with high-performance computing (hpc) on the edge: motivation, framework, and use cases," (2021).

[13] Park, S. J., Allen, S., and Shires, D., "Urban reconnaissance planning: Discovering new applications with tactical high performance computing," in [*MILCOM 2016-2016 IEEE Military Communications Conference*], 1121–1124, IEEE (2016).

[14] Alshabanah, A., Balasubramanian, K., Krishnamachari, B., and Annavaram, M., "Characterizing ml training performance at the tactical edge," in [*Artificial Intelligence and Machine Learning for Multi-Domain Operations Applications IV*], **12113**, 500–513, SPIE (2022).

[15] Perazzone, J., Dwyer, M., Chan, K., Anderson, C., and Brown, S., "Enabling machine learning on resource-constrained tactical networks," in [*MILCOM 2022-2022 IEEE Military Communications Conference (MILCOM)*], 932–937, IEEE (2022).

[16] Lewis, G., Echeverría, S., Simanta, S., Bradshaw, B., and Root, J., "Tactical cloudlets: Moving cloud computing to the edge," in [*2014 IEEE Military Communications Conference*], 1440–1446, IEEE (2014).

[17] Chin, P., Do, E., Doucette, C., Kalashian, B., Last, D., Lenz, N., Lu, E., Minor, D., Noyes, E., Rock, C., et al., "Tak-ml: Applying machine learning at the tactical edge," in [*MILCOM 2021-2021 IEEE Military Communications Conference (MILCOM)*], 108–114, IEEE (2021).

[18] Langleite, R. S., Opland, E. A. F., and Johnsen, F. T., "Big data solutions on tactical infrastructure," (2022).

[19] Dasari, V. R., Geerhart, B. E., Wang, P., and Alexander, D. M., "Deep neural network model optimizations for resource constrained tactical edge computing platforms," in [*Disruptive Technologies in Information Sciences V*], **11751**, 47–52, SPIE (2021).

[20] Misra, A., Jayarajah, K., Weerakoon, D., Tandriansyah, R., Yao, S., and Abdelzaher, T., "Dependable machine intelligence at the tactical edge," in [*Artificial Intelligence and Machine Learning for Multi-Domain Operations Applications*], **11006**, 64–77, SPIE (2019).

[21] Geerhart III, B. E., Dasari, V. R., Wang, P., and Alexander, D. M., "Efficient normalization techniques to optimize ai models for deployment in tactical edge," in [*Disruptive Technologies in Information Sciences V*], **11751**, 53–57, SPIE (2021).

[22] Busart III, C. E., *Federated learning architecture to enable continuous learning at the tactical edge for situational awareness*, PhD thesis, The George Washington University (2020).

[23] Cirincione, G. and Verma, D., "Federated machine learning for multi-domain operations at the tactical edge," in [*Artificial Intelligence and Machine Learning for Multi-Domain Operations Applications*], **11006**, 29–48, SPIE (2019).

[24] Blasch, E., Pham, T., Chong, C.-Y., Koch, W., Leung, H., Braines, D., and Abdelzaher, T., "Machine learning/artificial intelligence for sensor data fusion–opportunities and challenges," *IEEE Aerospace and Electronic Systems Magazine* **36**(7), 80–93 (2021).

[25] Toth, S. and Hughes, W., "The journey to collaborative ai at the tactical edge (cate)," in [*Artificial Intelligence and Machine Learning for Multi-Domain Operations Applications III*], **11746**, 144–163, SPIE (2021).

[26] Sturzinger, E. M., Lowrance, C. J., Faber, I. J., Choi, J. J., and MacCalman, A. D., "Improving the performance of ai models in tactical environments using a hybrid cloud architecture," in [*Artificial Intelligence and Machine Learning for Multi-Domain Operations Applications III*], **11746**, 18–32, SPIE (2021).

[27] Rawat, D. B., "Artificial intelligence meets tactical autonomy: Challenges and perspectives," in [*2022 IEEE 4th International Conference on Cognitive Machine Intelligence (CogMI)*], 49–51, IEEE (2022).

[28] Clarke, A. J. and Knudson, D. I., "Examination of cognitive load in the human machine teaming context," tech. rep., Naval Postgraduate School Monterey United States (2018).

[29] Warren, A. and Hillas, A., "Friend or frenemy? the role of trust in human-machine teaming and lethal autonomous weapons systems," *Small Wars & Insurgencies* **31**(4), 822–850 (2020).

[30] Schaefer, K. E., Perelman, B., Rexwinkle, J., Canady, J., Neubauer, C., Waytowich, N., Larkin, G., Cox, K., Geuss, M., Gremillion, G., et al., "Human-autonomy teaming for the tactical edge: The importance of humans in artificial intelligence research and development," in [*Systems Engineering and Artificial Intelligence*], 115–148, Springer (2021).