Planning for AI Sustainment: A Methodology for Maintenance and Cost Management

MAJ Iain Cruickshank and MAJ Shane Kohtz

Army Cyber Institute, United States Military Academy

iain.cruickshank@westpoint.edu and (845) 938-7566

shane.kohtz@westpoint.edu and (845) 938-9657

Major Iain Cruickshank – is a Functional Area 49 (Operations Research/Systems Analysis) officer in the U.S. Army. He is currently a senior research scientist at the Army Cyber Institute. He has previous assignments with the Army's Artificial Intelligence Integration Center, the 780th Military Intelligence Brigade, and the 101st Airborne Division. He holds a Ph.D. in Societal Computing from Carnegie Mellon University, which was obtained as a National Science Foundation Graduate Research Fellow, and an MS in Operations Research from the University of Edinburgh, which was obtained as a Rotary Ambassadorial Scholar.

MAJ Shane Kohtz – is a Functional Area 51 (Acquisition) officer in the U.S. Army. He is currently a cyber research manager at the Army Cyber Institute. He has previous assignments with the Missile Defense Agency, the Army's Program Executive Office Intelligence Electronic Warfare & Sensors, the 1st Infantry Division, and the 101st Airborne Division. He holds a MBA from the Naval Postgraduate School with a focus in Systems Acquisition Management. He is a member of the Army Acquisition Corps and holds a DAWIA Advanced certification in program management.

Abstract

The sustainment requirements of Artificial Intelligence (AI)-enabled systems are largely unexplored within the Department of Defense's Programs of Record (POR). Many programs often overlook maintenance needs for AI systems, extending beyond base hardware or software upkeep. However, prior research indicates a distinctive maintenance requirement for the machine learning models that power AI-enabled systems, and outlines strategies for planning and integrating AI maintenance into product support (Cruickshank and Kohtz, 2023). Notably, the adoption of industry best practices, program maintenance considerations, and Machine Learning Operations (MLOps) are crucial for crafting an AI system's sustainment strategy. This research builds upon the existing framework to further comprehend the extent of preventative and routine maintenance required by an AI-enabled system. We specifically investigate the degree of maintenance or "touch-time" needed to sustain a system's machine learning model(s). By examining a typical year of operations and sustainment for an AI-enabled computer vision system, we highlight primary maintenance considerations (i.e., maintenance tasks, task difficulty, and task frequency) and propose a method to estimate these factors. We then apply varying levels of maintenance based on organic, hybrid, or contractor logistics support to fully comprehend the sustainment costs. Our research offers a robust framework for program offices to more accurately predict initial and ongoing operation and sustainment costs when conducting a business case analysis. This will enable the selection of the most cost-effective sustainment strategy for a POR that intends to use any AI enabled system.

Introduction

As technology advances on the battlefield, the task of achieving superiority over adversaries becomes increasingly challenging. The introduction of Artificial Intelligence (AI)-enabled systems with Machine Learning (ML) models holds promise in offering a competitive edge. However, this technology is still in its early stages of development and deployment across the armed forces. Understanding the full scope and sustainment requirements of integrating this technology into both existing and new weapon systems presents a significant challenge. This research aims to bridge the gap in understanding and planning for future product support strategies. An AI-enabled system demands additional maintenance beyond the typical hardware and software upkeep observed in existing systems. It is imperative to recognize the necessity of treating ML models as systems within systems, marking a paradigm shift essential for determining appropriate sustainment strategies. The background of this paper delves into previous research inputs that inform a product support strategy for an AI-enabled system.

Our research has analyzed the tasks necessary for ML maintenance and identified those tasks that service members are capable of performing and identified those maintenance tasks that are best executed by contractors due to their advanced technical requirements. As a result, we recommend a hybrid approach to product support strategy, combining contractor and service support, for any AI-enabled system. This research implements the aforementioned approach to develop labor hour requirements for specific maintenance costs and a cost estimating model for two different maintenance strategy options: contractor-only and a hybrid of contractor and service support. These approaches are essential for enabling decision-makers to discern the most cost-effective approach in the early planning stages of a product support strategy over a typical 20-year service life cycle for a program of record (POR)

Background

This section provides crucial background information pertinent to the proposed maintenance cost models presented in this paper. Specifically, it covers the maintenance requirements of AI-enabled systems, the MLOps paradigm concerning the utilization of ML models in real-world systems, and previous research regarding the maintenance of ML models in military AI-enabled systems.

AI-enabled systems and their maintenance. AI-enabled systems, akin to any technological apparatus, require maintenance. Such systems comprise traditional software and, potentially, hardware, contingent upon the system's purpose, in addition to AI components. The AI components often rely on various hardware and software dependencies, commonly referred to as a ‘stack’ (Moore, 2018). Figure 3 illustrates the AI stack. The critical elements of AI components, which render the entire system AI-enabled, are the ML models. These models empower the system to execute automated behaviors and tasks that typically demand human-level perception or reasoning, serving as the 'brain' of the AI-enabled system. Just like every other component of an AI-enabled system, these ML models also require maintenance.

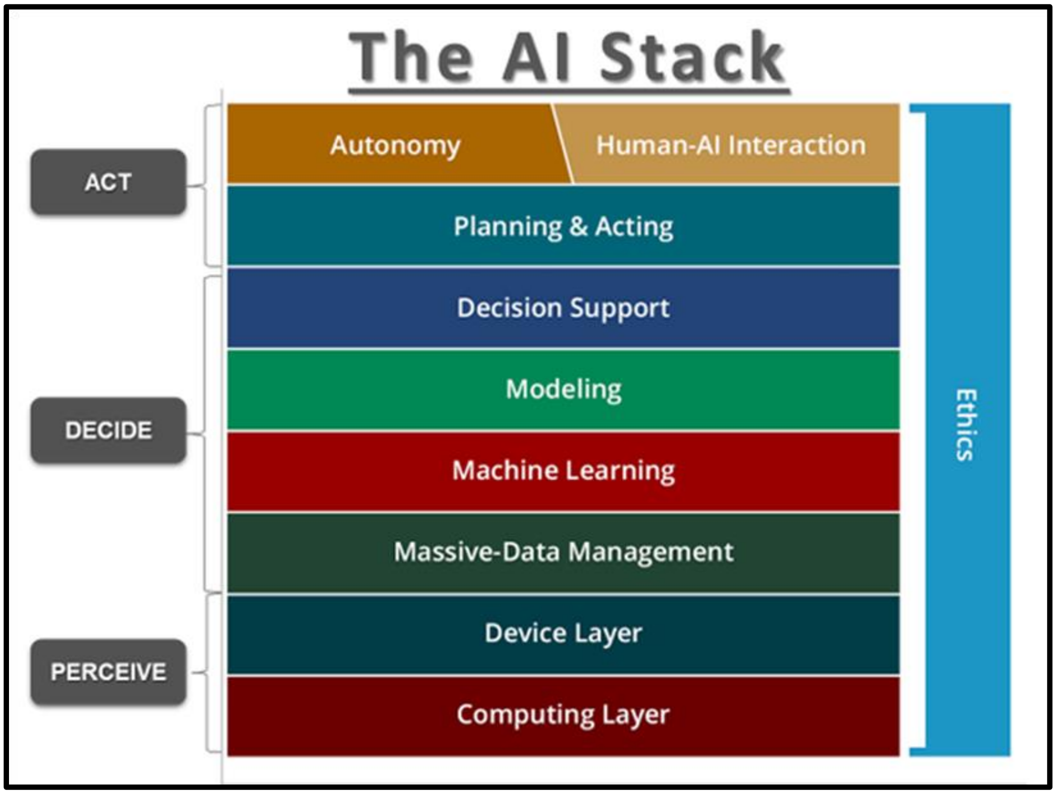

Figure 1. Carnegie Mellon University's AI stack, depicting the necessary components of an AI-enabled system. Source: Moore (2018).

Despite their potential, ML models still encounter several issues that necessitate frequent maintenance. ML models inherently learn correlations useful for specific tasks from the data presented to them. Consequently, performance issues may arise if the data during utilization differs from the training data (i.e., Out-of-Domain Data problem) (Patruno, 2019). For instance, a computer vision ML model designed to detect specific vehicles from a ground perspective may fail when confronted with differences in background or biome between its training data and operational environment (e.g., urban versus rural settings). Additionally, ML models can suffer from issues like model drift (Talby, 2018), data drift (Evidently AI, 2021), concept drift (Patruno, 2019), or changes in hardware such as sensors. All of these changes, which generally would not perturb a change in a human's task performance, significantly impact ML model performance. Furthermore, ML models can be directly targeted through Adversarial ML, resulting in substantial degradation of model performance (Talby, 2018). Notably, many of these issues are unique to ML and ML-enabled systems; changes such as alterations to image backgrounds do not affect the hardware or software of traditional digital systems. Hence, ML models entail their own inherent issues necessitating maintenance beyond that required for traditional hardware and software systems.

While ML models face several issues that can greatly affect their performance, addressing these issues often demands fewer resources and expertise compared to the initial development of the ML model. Maintaining ML models deployed in real-world settings—referred to as model deployment—can typically be managed through a suite of updating and monitoring processes, collectively forming part of the industrial ML paradigm known as MLOps (Treveil et al., 2020). At its core, MLOps encompasses a set of practices aimed at operationalizing ML systems (Treveil et al., 2020). Figure 4 illustrates the core components and relationships of MLOps. Although the principles and practices of MLOps remain an active area of research, three practices integral to MLOps include data and model monitoring in production, continuous model updates in response to changes, and integrating model maintenance into model operation (Treveil et al., 2020). These practices are essential for organizations and businesses to utilize ML models despite their inherent issues. Thus, effective employment of ML models in real-world and production systems within the MLOps paradigm requires the implementation of appropriate tools and practices for monitoring ML models and their data, as well as procedures for updating ML models as close to operational use as possible.

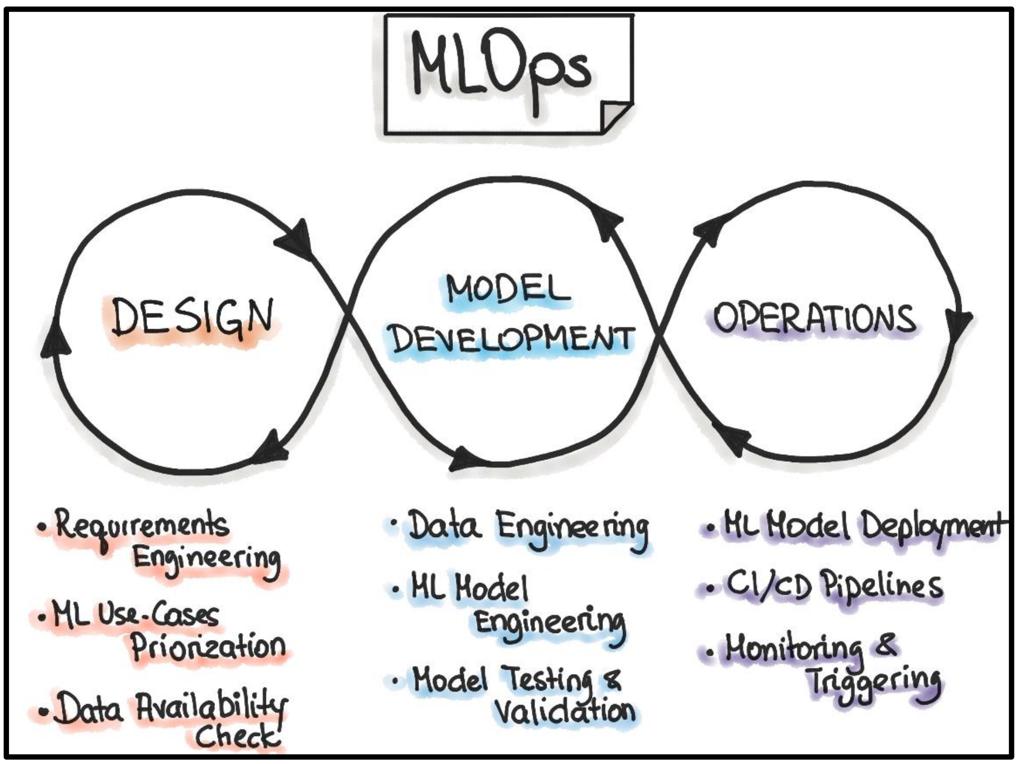

Figure 2. Core components of MLOps and their relationships. Source: Visengeriyeva (2023, March 30).

Of particular significance within the MLOps paradigm is *model retraining*. Ideally, model retraining involves rerunning all steps required to train an ML model with a new dataset, necessitating changes only to the model's weights, not its code (Patruno, 2019). This form of maintenance typically occurs whenever the data changes and an updated training dataset becomes available (Evidently AI, 2021). Hence, this maintenance generally takes two forms: periodic and dynamic (Evidently AI, 2021). Periodic retraining involves anticipated changes in data, such as quarterly or yearly shifts in business practices, while dynamic retraining occurs whenever there are changes in the data generation process, such as collecting data in an adversarial environment (e.g., detecting credit fraud) or in a naturally dynamic process (e.g., labeling objects in imagery). The frequency of dynamic retraining can vary considerably depending on the ML application; some ML models require daily updates, while others may only need monthly or yearly updates (Evidently AI, 2021). Furthermore, depending on the system in which the ML model is used, retraining may also involve updating models across several devices, wherein the new model is pushed or flashed onto those devices; once retrained, the model must be reintegrated back into the AI-enabled system. Regardless of the frequency of ML model retraining, all experts agree that this process is essential for any ML-enabled system. Thus, model retraining is a necessary component of any ML model and may need to occur as frequently as daily.

Maintenance Considerations for Military AI-Enabled Systems. In contemplating maintenance strategies for AI-enabled systems, several key considerations emerge. These considerations are pivotal in guiding program offices during the Product Support Business Case Analysis (PS BCA), which informs both the Product Support Strategy (PSS) and Life Cycle Sustainment Plan (LCSP) (Department of Defense [DOD], 2022). The PS BCA assesses alternative sustainment options, including organic, contractor, or a hybrid mix of support, thereby informing the program's sustainment strategy (Department of Defense [DOD], 2014). Notably, approximately 85% of sustainment costs are established during the requirement setting phase (Schinasi, 2003). Therefore, understanding the requirements and maintenance demands of AI/ML systems is crucial during the strategy development phase to facilitate effective planning and budgeting for sustainment. This encompasses the maintenance of ML models in addition to the hardware and software components underlying the AI stack, which are essential for ML model operation within the system.

Various paradigms exist for approaching maintenance of ML models, akin to sustaining other components of a system, offering both contract and organic service support alternatives. At one end of the spectrum lies exclusive contract-based maintenance for ML models. Under this arrangement, contractors assume full responsibility for all aspects of model maintenance, encompassing data and model monitoring, development of test and evaluation metrics, creation of model retraining procedures, actual model updating, retirement and replacement, and model governance to ensure compliance with necessary guidelines and regulations. A specific iteration of this approach is the ML-as-a-Service (MaaS) model, often implemented via application programming interfaces (APIs). Here, contractors oversee the model's entire lifecycle, including initial development and ongoing maintenance, while users interact with the model through APIs, typically operating on a pay-per-usage pricing model. This type of model is currently used by companies like OpenAI and by organizations like the XVIIIth Airborne Corps and often works on a pay-per-usage type of pricing scheme.

While the contractor-only approaches present the simplest approach to maintenance planning, they have serious pitfalls that must be considered. For the MaaS model, despite the simplicity of this model, much like any other pay-per-use pricing scheme (e.g. cloud services, SaaS), it can quickly become exorbitantly expensive if there is a lot of use of the service. Additionally, it requires connectivity back to the API to work. So, if the AI-enabled system is meant to work in austere environment or have a lot of usage on the ML-models, going through a

MaaS model may be overly costly. Additionally, having contractors perform all the functions of ML maintenance ignores the hard-learned lessons behind the MLOps paradigm; namely the operation of the ML model has been separated from its maintenance and development. A primary reason why MLOps places the development and maintenance of ML models so close to the running of ML models is that these models require constant monitoring and frequent updating (Treveil, et al., 2020). In fact, one form of updating, model retraining, can occur as frequently as daily for an ML model in production in an adversarial and dynamic environment. As with our previous computer vision example of detecting objects from a ground perspective, the ML model would need to be, at a minimum, retrained every time the biome changes (e.g. moving from rural to urban) and every time an organization wants to detect a new or different set of objects. Conceivably, such a change in an ML model's operating environment could occur several times over the course of a single operation for a military unit. Thus, given the frequent nature of ML model maintenance, having contractors provide all this maintenance could be cost prohibitive.

At the other end of the spectrum is a service only solution, where servicemembers and DOD civilians are responsible for all of the aforementioned ML model maintenance tasks. While this certainly presents some potential for cost savings in terms of maintenance, the Army and DOD may lack the skill sets in house, in sufficient numbers, to perform some maintenance functions. This is especially true for maintenance functions like designing a test and evaluation scheme for both the ML model and its data as well as determining the right model retraining procedures (e.g. active learning, fine-tuning, using adapters, prompt engineering, etc.). These types of maintenance tasks often take a seasoned data scientist with domain area expertise and, often, advanced education. That said, some of the maintenance tasks actually require very little education and can be learned with suitable training. For example, actually performing model updates, given a guide to the model's retraining procedures, is a trainable task that does not require an advanced educational background. Thus, planning to do the full spectrum of model maintenance in house may be infeasible, given constraints on in house ML expertise.

Recognizing the limitations of both contractor-provided and service-provided maintenance, recent research has advocated for a hybrid approach. This approach leverages contractor expertise while entrusting servicemembers with maintenance execution of the most frequent maintenance tasks (Cruickshank & Kohtz, 2023). Essentially, tasks such as model monitoring and retraining can be trainable for in-house execution within a properly implemented AI-enabled system, allowing servicemembers to handle these responsibilities. Conversely, less frequent, higher-expertise tasks can be outsourced to contractors. The proposed hybrid approach is detailed in Figure 5, illustrating the division of sustainment tasks among the involved components.

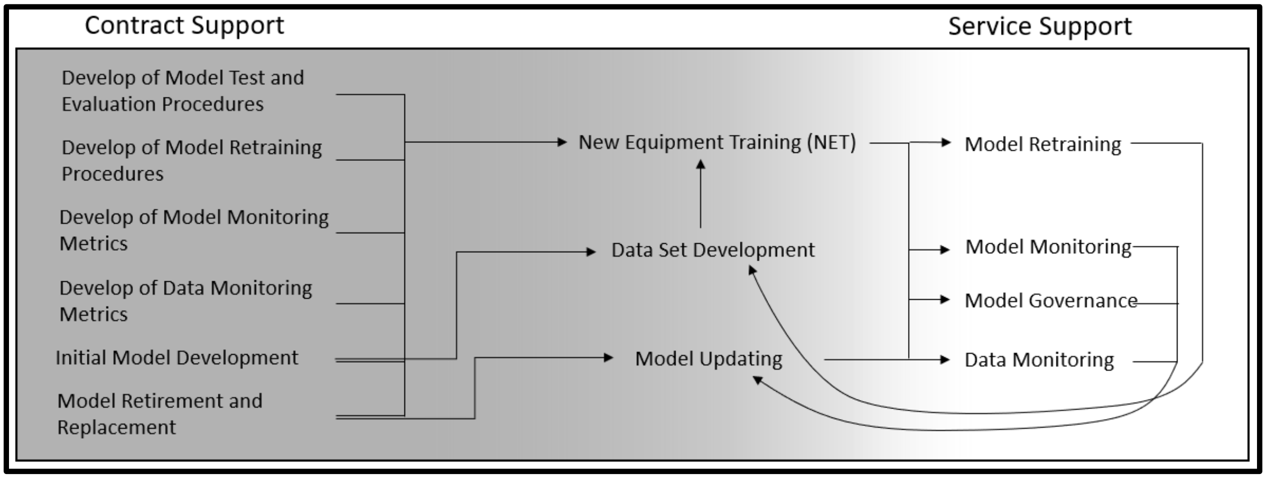

Figure 3. ML model sustainment tasks in a hybrid maintenance plan with associated dependencies between contractor and service maintenance tasks.

Moreover, the hybrid model for ML model maintenance encompasses additional considerations:

- *Data Rights*: program offices, looking to have ML models in their programs, may negotiate limited rights for implementation of the ML models since government operators would be doing the model retraining and monitoring. However, since the deliverables will most likely come from mixed funding, the program offices should, at a minimum, negotiate for government purpose rights of the technical data and deliverables. This approach will give the program office flexibility in the future if they decide to change the sustainment strategy.
- *ML Model Touch-Time Analysis*: As has been mentioned within this paper, ML models, the brain of any AI-enabled system, require model retraining for various reasons. The amount of model retraining for any given ML model is highly context dependent; it can vary from daily retraining up to monthly or even yearly or more (Evidently AI, 2021). Thus, as part of the PS BCA, there needs to be a retraining requirements analysis. This analysis should, at a minimum, consider how often the data environment for the AI-enabled system predictably changes, whether it will be used in an adversarial environment (i.e. data environment where people generating the data attempt to change data generation patterns to fool the system), and how often the data generation process changes physical locations (i.e. a sensor moves from one geographic region to another). With the information from this analysis, a program office can have a much better estimation of the maintenance cost requirements.

A Framework for Estimating Machine Learning Component Maintenance Cost

Maintaining a machine learning model encompasses several task categories, including system monitoring, model updating, data curation, and model curation (Cruickshank and Kohtz, 2023). Although maintaining an ML model involves numerous tasks, these tasks vary in required expertise and frequency of execution (Cruickshank, Zenkevich, Evidently AI). For instance, retraining an ML model demands more expertise compared to monitoring its usage, and the frequency of model retraining is typically less than that of model monitoring, which should occur whenever the model is in use. Thus, the scope and scale of maintenance for a specific ML-enabled system are determined by variables such as the frequency of maintenance tasks, the expertise needed, and the number of unique ML models requiring maintenance.

Based on these determinants, one can estimate the maintenance costs for ML models within a system using a multiplicative model. Specifically, the maintenance cost estimate for ML models is calculated by multiplying the number of unique ML models in the system by the skill premium per time for each task required to maintain each unique model, further multiplied by the time each skill needs to be performed within a given period. This maintenance cost estimate can be expressed by the following function:

$$\sum_{m=1}^{num\_models} \sum_{i=1}^{num\_tasks} skill\_premium\_per\_time_{m,i} \times \left(\sum_{t=1}^{num\_events} maintenance\_time_{m,i,t} \right)$$

Within this function, the main items to track are the skill premiums and the maintenance times. The skill premiums are the cost (per person, per unit, etc.) to execute a particular task, *i*, for a particular model, *m*. For example, this would be something like the cost per hour to have a maintenance technician retrain an object detection computer vision model. The second major item to track for the cost estimate is the amount of time a particular task is performed for a particular model. Building on the previous example, this would be how many hours it takes that technician to do a retraining of the object detection computer vision model by how many times they do the retraining in a given time period (e.g., per year). With these items, the cost estimate becomes a matter of multiplying the time by the skill premium and summing it up across all time, tasks and models for an ML-enabled system.

This model does have some important caveats. This model is meant to produce cost estimates at a program-level and for the ML models specifically. Thus, there may be additional costs for specific units as well as for other components of the system, most notably the hardware. Additionally, fixed costs like the training for personnel to execute certain maintenance tasks are not explicitly included in the model but could be easily included in the skill premiums through amortization of the training cost.

A Worked Example

Having now established a cost estimate model for the ML models of a ML-enabled system, we will now work a simple, yet common across the services example, of Automated Threat Recognition (Ferraris, 2021). In this particular case, we will consider an Automated Threat Recognition (ATR) system that is part of a sensor system for a vehicle like a tank or drone. This ATR system will be a simple one consisting of a single, supervised object detection algorithm for one spectra of imagery (e.g., visible, EO, IR, etc.). For this example, we will consider that the ATR system is used at a Brigade Level, within a standard U.S. Army Division. This Brigade has 1 major training center rotation and 1-2 brigade level exercises that utilize the system per year. Its likely the ATR system would be used in other lower-level training exercises (e.g., Battalion-level training events), but those will most likely have a negligible impact to the ML model maintenance.

Based on the given scenario, there are several model maintenance tasks that would need to be performed each year.

- Once per year, there would need to be a model and software update to keep the ML model on pace with state-of-the-art and address any architectural flaws that may have come out during the previous year. Furthermore, since we are using a supervised ML model, this would also be the chance to revise what classes the model can detect objects of. All of this maintenance will likely need to come with a certification of model safety and ethical usage, given current AI policy guidelines.
- For the training center rotation, there would need to be model retraining for all models and data collection and labeling. This is needed to address the change in scenery that will occur between the host station environment and the training center environment. During the training event, there will also need to be data monitoring and model monitoring, with a possible need for additional model retraining, and data collection and labeling.
- For Brigade level exercises there would need to be model monitoring and data monitoring. Depending on the nature of the training exercises, there may also be a need for model retraining, and data collection and labeling.

During these training events, the service-provided maintenance would require, at a minimum, two trained officers or NCOs to support operations. In the data management cell, we project an O-3 (Captain) and E-6 (Staff Sergeant) to complete certain maintenance tasks. If a contractor provided this support, two Field Service Representatives (FSR) would provide this support. This maintenance cell would be located at the Brigade Tactical Operations Cell (TOC) to execute the consolidated maintenance of the models across multiple platforms. The maintenance cell would retrieve the periodic inputs of data from Battalion level TOCs and down to the platform (i.e., tank) crew level. The machine learning model data would arrive at the Brigade TOC through the Army provided Command and Control (C2) networks or from removable hardware through Logistic Package Operations (LOGPAC). Through either method, the transfer of data would occur at least once every 24 hours. If there are some functions that require reach back or depot level support (i.e. model renewal, certification, or data collecting and labeling), the maintenance cell would then transfer the data through similar means to higher echelon support level. Below is a depiction of the typical machine learning maintenance workflow during Brigade level collective training or deployments (Figure 4).

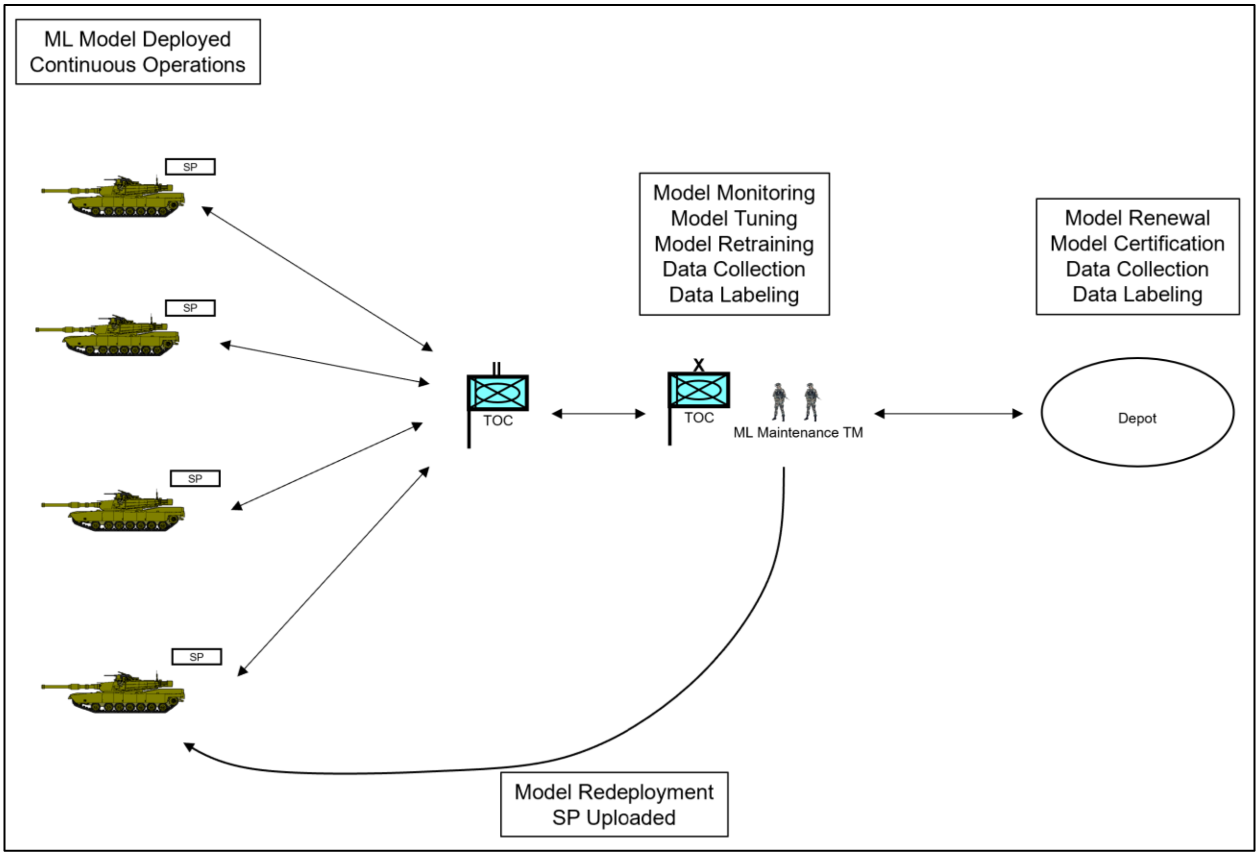

Figure 4: ML Maintenance Operations Workflow

When completing the cost estimate for ML maintenance operations, we used estimates for the skill premium rates. We based the hourly rates for contractor rates on industry median rates. We also created two categories of rates, depending on whether the activity is taking place at home station or in a deployed (i.e., CTC) environment. For the tasks of model retraining, model monitoring, and data monitoring we used the median hourly rate for a data scientist at $60.00 for home station and $80.00 for forward deployed. For the data collection and labeling task we used an industry estimate from labeling services of $40.00 per hour for home station and $75.00 for forward deployed. And, for the model renewal and certification, which includes all of the tasks of updating the model to a new model, instruction in that model's monitoring metrics and retraining procedures, and implementation testing of the new model we used a senior or chief data scientist hourly rate of $120.00 an hour. For the service-provided rates, we used the median hourly salary for an O-3 (Captain) for model retraining and the median hourly salary for an E-6 (Staff Sergeant) for model monitoring, data monitoring, and data collection and labeling, based on who is most likely to perform the particular skills. We also included in an additional $5.00 per hour and $4.00 per hour, respectively, for amortized training that these service-provided personnel would need to execute the maintenance tasks. We also observe that contractor rates will almost certainly be higher than service-provided rates for these tasks and that some tasks, due to their high skill requirements may only be able to be done by contractors (i.e., model renewal and certification).

Similar to the skill premium rates, we also used estimates for the number of hours per skill execution. We allowed for differences in hours depending on the event. For example, a CTC rotation will require more hours for some skills than home station training or periodic maintenance. For model retraining, we estimate an average of 32 hours per retraining, based on two people doing, on average 16 hours of work. We also include the time it takes to put the model back into operation following retraining as part of this estimate. For data collection and labeling, we estimated around 3 people doing 40 hours of work for an average of 120 hours. For both data and model monitoring we estimated around 40 hours of work for two people for an average of 80 hours per event. Finally, for the Model certification and renewal we estimated around 180 hours to accomplish this, but note this estimate could vary wildly as the tasks that comprise this task have not been done all together around an ML-enabled system yet. Each of these estimates were increased for a CTC rotation, with mean hours for model retraining, data collection and labeling, data monitoring, and model monitoring going to 40, 320, 160, and 160 hours respectively. Finally, its important to note that these estimates are not meant to be definitive, but rather meant to illustrate how there is a cost to ML-maintenance and that this cost can vary significantly between different maintenance strategies.

Based on the scenario previously outlined and the estimates for the drivers of ML maintenance, we propose two different maintenance strategies. The first strategy is to use contractors for all ML model maintenance. The second is to use service provided maintenance for all tasks except for those tasks under model renewal and certification and allow for some additional contractor support on data collection and labeling.

Once we estimated the touch time requirements for maintenance of ML models, the cost estimates were expanded to encompass a typical 20-year service life for a ground platform (Department of Defense [DOD], 2020). The cost estimates we developed for the potential maintenance strategies rely on the current year (2024) to normalize costs. As a result, the approach utilizes the current-year cost approach because current-year estimates are necessary when conducting comparisons (Department of Defense [DOD], 2021). The current-year costs, with 2024 as the base year, utilizes the Army/Navy Joint Inflation Calculator Indices (JIC) to support the comparison of product support strategies (Army Financial Management & Comptroller, 2023). The first strategy with contractor only support utilizes the Operation and Maintenance Army (OMA) appropriation weighted index because program offices must obligate funds in the fiscal year; however, disbursements from the treasury could happen after the initial obligation year (DOD, 2021). The second strategy that utilizes a hybrid approach for product support incorporated a composite index for Military Pay Army (MPA) appropriation to capture a blend of inflation and escalation cost increases for military pay since this option utilizes service members (DOD, 2021). The weighted composite index for this option is applied to cumulative costs for service members and contractor estimates because there could be OMA fund disbursements after the obligation year for contractors and our model includes the assumption that the pay escalation will be similar between contractor and military personnel. These cost estimate inputs are necessary to support a fair and realistic comparison between the two options, and the model in current-year dollars provides the best picture for decision makers to understand the most cost-effective option.

A system with a 20-year service life will have a phase-in, steady-state, and phase-out periods (DOD, 2020). The base year for our notional ATR model starts in 2024 and ends after a 20-year service life in 2044. The following graph (Figure 5) displays the cumulative cost estimates of these two strategies for just one armored brigade over the typical 20-year service life period.

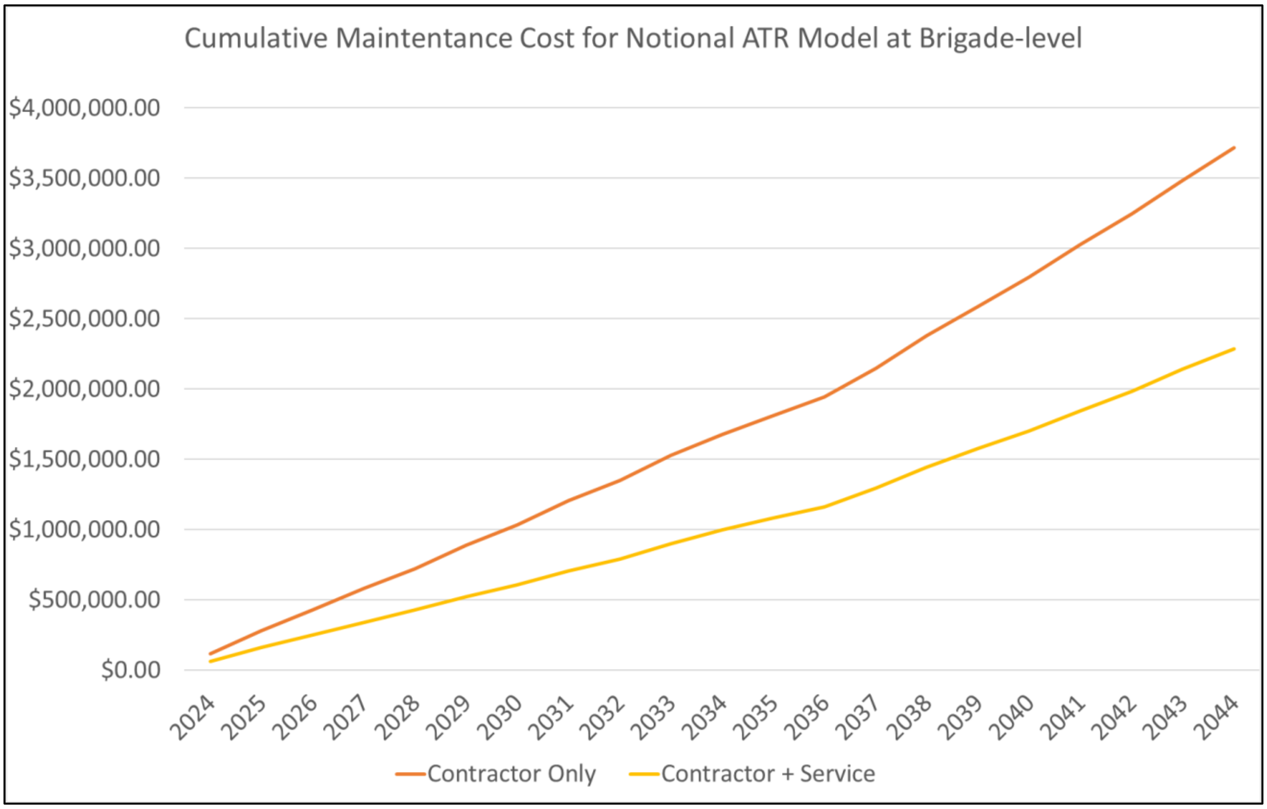

Figure 5: Cumulative Maintenance Cost for Notional ATR Model at Brigade-Level

From this plot we can observe that there is a significant cost per year for each strategy, and that a contractor only strategy grows in cost more quickly over time than contractor + service provided maintenance strategy. The graph shows the cost implications of adding one ATR system to vehicles for just one armor brigade in the entire Army. The difference in ML model maintenance between both options exceeds $1,000,000 across the service life for one brigade. As a result, the costs for the ATR Notional system across all armor brigades will have a larger impact. The plot below (Figure 6) depicts the costs associated with implementing the ATR Notional system across all nine armor brigades in the Army.

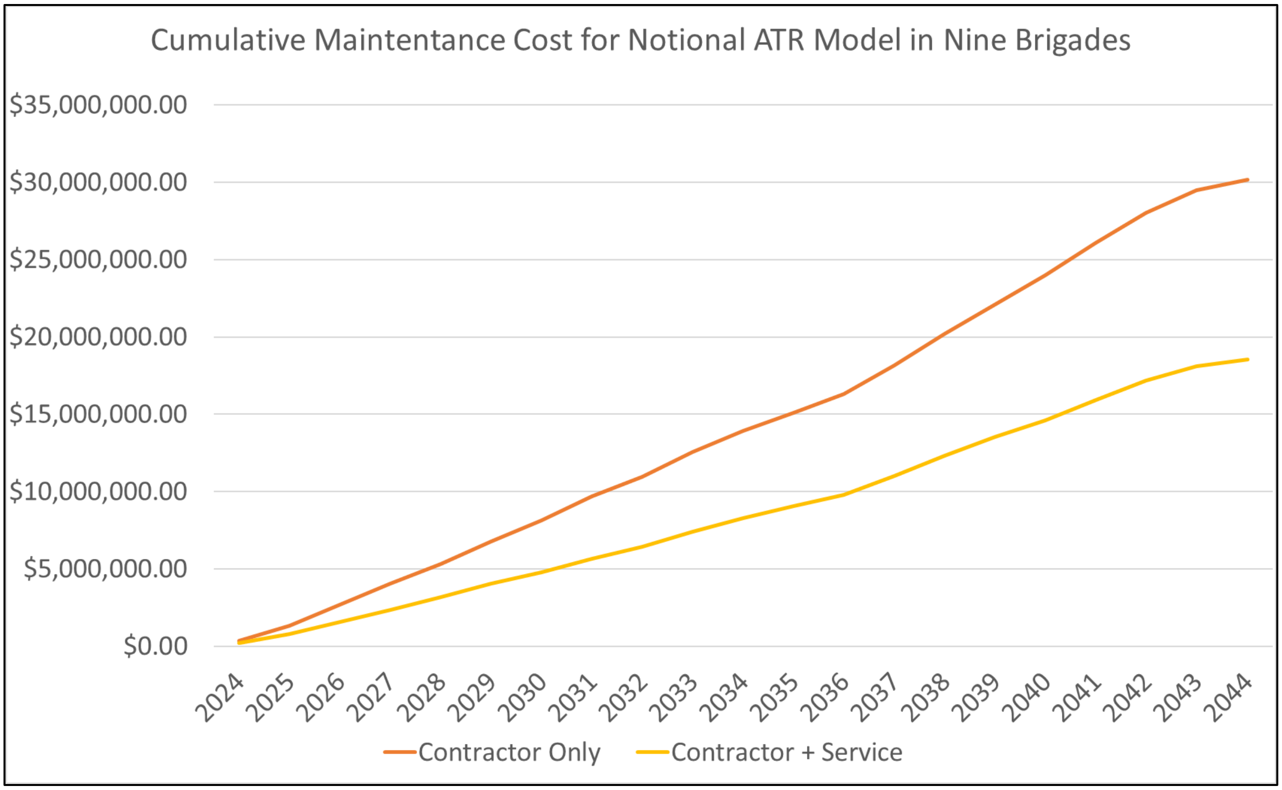

Figure 6: Cumulative Maintenance Cost for Notional ATR Model in Nine Brigades

The cost estimation of a program of record (POR) for the Notional ATR Model in a Sensor Payload highlights the importance of a hybrid machine learning maintenance strategy. The indices and use of 2024 as the current-year funds in the ATR Notional System cost estimate for one brigade is the same for a POR supporting nine brigades and across all of this research's cost estimates. The POR model incorporated typical planning considerations for the Operations and Sustainment phase of a program of record. The cost estimation model has a phase-in period of two years (2024 and 2025) with three and then six brigades utilizing the system for the first two years. Steady-state operations last 16 years from 2026-2042. The phase-out period reduces the amount of systems for six and then three brigades at the end of a 20-year service life in 2044. Once a system is at scale and fielded across nine armor brigades, the difference in the cost effectiveness of maintenance strategies become more apparent. For example, the potential cost savings with a hybrid approach implementation is nearly $11,595,000 with the contractor only strategy estimate close to $30,158,000. The difference in strategy maintenance costs demonstrates how imperative it is for the services to support with an appropriate level of organic support for AI-enabled systems.

The next graph depicts how maintenance of a AI-enabled system will be different than typical hardware and software maintenance of a system. The amount of touch time required for ML systems could potentially surpass the costs of a hardware and software stack. In order to understand how ML model maintenance may compare to a system maintenance plan, we utilized an analogous estimating approach from two different sources to establish a notional hardware and software stack for the Sensor Payload with the ATR model. First, for software, the cost model utilized a similar system (Common Sensor Payload) cost estimate from a previous business case analysis for software sustainment. The previous analysis evaluated multiple support strategies, and we utilized the estimate with a 50/50 maintenance ratio of contractor and government support (Software Engineering Center [SEC], 2022). This software sustainment strategy incorporates software, firmware, and obsolescence updates to the fielded systems. Second, for the hardware maintenance, the cost model incorporated the example component level cost estimate for a avionics or electronic subsystem from the Operating and Support Cost-Estimating Guide (DOD, 2020). This example of an O&S hardware cost estimate is for a notional system like the common sensor payload that incorporates spares, parts, labor, and recurring training to support hardware maintenance for 10 units with host platforms (DOD, 2020). We then scaled the software and hardware cost estimates to model costs across nine brigades for a 20-year service life in current-year dollars (2024). The cost model approach illustrates how ML model maintenance occurs in addition to the hardware and software stack maintenance of a system. As a result, given the potential for extensive touch time, the ML model maintenance will increase existing maintenance and could potentially be higher than the current paradigm of system software maintenance while less than hardware maintenance. Figure 7 below shows how machine learning model maintenance may compare to existing system maintenance.

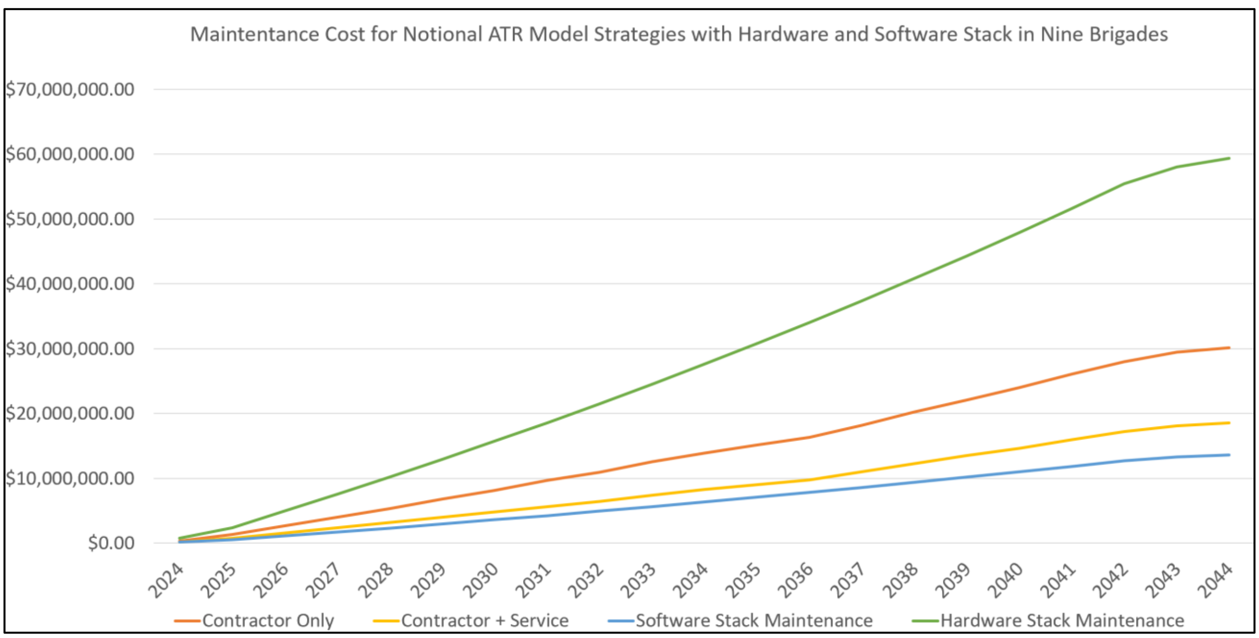

Figure 7: Cumulative Maintenance Cost for Notional ATR Model Strategies with Hardware and Software Maintenance in Nine Brigades

Conclusion

Implementation of AI and ML models to provide fighting forces at the tactical edge remains a priority to gain the advantage on the battlefield. The services and program offices chartered with delivering capability to the warfighter must understand the nuances of ML model maintenance when developing product support strategies. A mindset of treating AI and ML capability as a system within a system is imperative when developing strategies and estimating costs. Machine learning models are an added layer of maintenance on top of a system's typical hardware and software maintenance, and program offices must treat it differently. The graph (Figure 8) below highlights how adding or implementing ML capability in a program of record will increase the O&S costs of a program of record. This is why relying on organic support for some maintenance support is a must do when planning sustainment strategies.

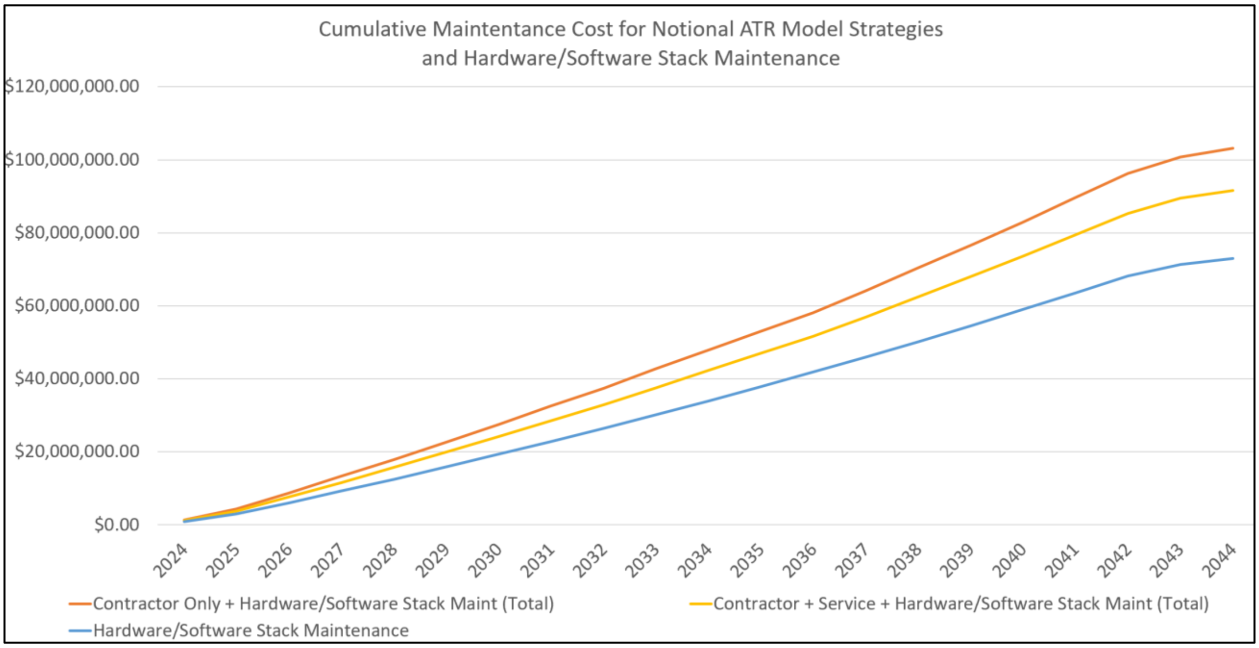

Figure 8: Cumulative Maintenance Cost for Notional ATR Model Strategies Plus Hardware/Software Stack Maintenance

The amount of touch time is required for ML enabled systems is the main cost driver for these systems. As a result, services should rely on some level of organic support to maintain the models and rely on industry for the more technically advanced tasks as highlighted in the background section of this research. A contractor only approach will become cost prohibitive when implementing this new technology across the entire fleet of sensors or systems in all of the branches of service. Our evaluation of the contractor only approach and a hybrid approach of contractor and service support strategies is a framework for program offices to utilize when implementing machine learning model capabilities. Incorporation of these methods for AI-enabled systems early in the planning process will pay dividends later in the service life of programs.

Finally, its important to note that recent innovations in the ML space in generative models, like Large Language Models and Vision Language Models, can alter this maintenance landscape. These models are applicable to a wider array of tasks, without any additional training of the models, but require new forms of interaction from the users with techniques like prompt engineering. As such, these models could alter the maintenance landscape by making certain types of maintenance tasks, like fine-tuning occur almost in stride with the operation of the model which further necessitates that maintenance of the model be capable of being conducted within the services. We leave investigation of the maintenance of these newer generative types of AI to future work.

References

Breck et al., The MLTest Score: ARubric for ML Production Readiness and Technical Debt Reduction. https://storage.googleapis.com/pub-tools-public-publication-data/pdf/aad9f93b86b7addfea4c419b9100c6cdd26cacea.pdf

Moore, A., Hebert, M., Shaneman, S. (2018). The AI stack: a blueprint for developing and deploying artificial intelligence. *Proceedings of SPIE Ground/Air Multisensor Interoperability, Integration, and Networking for Persistent ISR IX*.

Talby, D. (2018, June 5). *Lessons learned turning machine learning models into real products and services.* Retrieved from https://www.oreilly.com/radar: https://www.oreilly.com/radar/lessons-learned-turning-machine-learning-models-into-real-products-and-services/

Treveil, M., Omont, N., Stenac, C., Lefevre, K., Phan, D., Zentici, J., . . . Heidmann, L. (2020). *Introducing MLOps.* New York: O'Reilly.

Visengeriyeva, L., Kammer, A., Bär, I., Kniesz, A., Plöd, M., Eberstaller, S. (2023, March 30) *MLOps Principles*. https://ml-ops.org/content/mlops-principles.

Evidently AI. (2021, July 01). *When to Retrain an Machine Learning Model? Run these 5 checks to decide on the schedule.* Retrieved from KDNuggets: https://www.kdnuggets.com/2021/07/retrain-machine-learning-model-5-checks-decide-schedule.html

Sculley et al. Hidden Technical Debt in Machine Learning Systems. https://papers.nips.cc/paper/2015/file/86df7dcfd896fcaf2674f757a2463eba-Paper.pdf

Sculley et al. Machine Learning: The High Interest Credit Card of Technical Debt. https://research.google/pubs/machine-learning-the-high-interest-credit-card-of-technical-debt/

Zinkevich, Martin. Rules of Machine Learning: Best Practices for ML Engineering. https://developers.google.com/machine-learning/guides/rules-of-ml/.

Cruickshank, I. and Kohtz, S. (2023). Acquiring Maintainable AI-Enabled Systems. *Excerpt From the Proceedings of the Twentieth Annual Acquisition Research Symposium*, 59-67. https://dair.nps.edu/handle/123456789/4852

Ferraris, Patrick. Aided Detection on the Future Battlefield. https://www.army.mil/article/232074/aided_detection_on_the_future_battlefield.

Department of Defense. (2021). *Inflation and Escalation Best Practices For Cost Analysis (Analyst Handbook).* https://www.cape.osd.mil/files/Reports/OSDCAPEEscalationHandbook2021.pdf

Army Financial Management & Comptroller. (2023). *Joint Inflation Calculator*. https://www.asafm.army.mil/Cost-Materials/Cost-Models/#rates-guidance

Department of Defense. (2020). *Operating and Support Cost-Estimating Guide*. https://www.cape.osd.mil/files/OS_Guide_Sept_2020.pdf

Software Engineering Center (SEC). (2022). *Common Sensor Payload (CSP) Product Support Business Case Analysis (PS BCA) For Software Sustainment*. Communications-Electronic Command.

Camm, F., Whitmore, T. C., Weichenberg, G., Sheng, T. L., Carter, P., Dougherty, B., Nalette, K., Bohman, A., Shostak, M. (2021). Data Rights Relevant to Weapon Systems in Air Force Special Operations Command (Report No. RR-4298-AF). RAND. https://www.rand.org/pubs/research_reports/RR4298.html

Department of Defense. (2014). *DOD Product Support Business Case Analysis Guidebook.* https://www.dau.edu/tools/Lists/DAUTools/Attachments/127/Product-Support-Business-Case-Analysis-(BCA)-Guidebook.pdf

Department of Defense. (2016). *Operating and Support Cost Management Guidebook.* https://www.dau.edu/tools/Lists/DAUTools/Attachments/126/Operating-and-Support-Cost-Management-Guidebook.pdf

Department of Defense. (2022). *Defense Budget Overview: United States Department of Defense Fiscal Year 2023 Budget Request.* https://comptroller.defense.gov/Portals/45/Documents/defbudget/FY2023/FY2023_Budget_Request_Overview_Book.pdf

Department of Defense. (2022). *Product Support Manager Guidebook.* https://www.dau.edu/tools/Lists/DAUTools/Attachments/129/Product-Support-Manager-(PSM)-Guidebook.pdf

Evidently AI. (2021, July 01). *When to Retrain an Machine Learning Model? Run these 5 checks to decide on the schedule.* Retrieved from KDNuggets: https://www.kdnuggets.com/2021/07/retrain-machine-learning-model-5-checks-decide-schedule.html

General Services Administration. (2023, March 1). *Defense Federal Acquisition Regulation Supplement (DFARS) 227.7103-5 Government rights.* https://www.acquisition.gov/dfars/227.7103-5-government-rights.

Guariniello, C., Balasubramani, P., DeLaurentis, D. (2021). A System-of-Systems Approach to Enterprise Analytics Design: Acquisition Support in the Age of Machine Learning and Artificial Intelligence. *Proceedings of the Nineteenth Annual Acquisition Research Symposium*, 205-217. https://dair.nps.edu/handle/123456789/4519.

Nagy, B. (2022). Tips for CDRLs/Requirements when Acquiring/Developing AI-Enabled Systems. *Proceedings of the Nineteenth Annual Acquisition Research Symposium*, 218-241. https://dair.nps.edu/handle/123456789/4587.

National Security Commission on Artificial Intelligence. (2021). *Final Report*. https://www.nscai.gov/wp-content/uploads/2021/03/Full-Report-Digital-1.pdf.

Office of the Under Secretary of Defense for Acquisition and Sustainment (OUSD[A&S]). (2021, November 4). *Product Support Management for the Adaptive Acquisition Framework* (Department of Defense Directive 5000.91). Department of Defense. https://www.esd.whs.mil/Portals/54/Documents/DD/issuances/dodi/500091p.PDF?ver=qk1slCU3Y0c1acIDocWyJA%3d%3d.

Moore, A., Hebert, M., Shaneman, S. (2018). The AI stack: a blueprint for developing and deploying artificial intelligence. *Proceedings of SPIE Ground/Air Multisensor Interoperability, Integration, and Networking for Persistent ISR IX*. https://www.ri.cmu.edu/publications/the-ai-stack-a-blueprint-for-developing-and-deploying-artificial-intelligence/.

Patruno, L. (2019, June 10). *The Ultimate Guide to Model Retraining.* Retrieved from mlinproduction.com: https://mlinproduction.com/model-retraining/#:~:text=Quickly%20changing%20training%20sets%20might,require%20monthly%20or%20annual%20retraining.

Schinasi, K. V. (2003). *BEST PRACTICES: Setting Requirements Differently Could Reduce Weapon Systems' Total Ownership Costs.* Government Accountability Office.

Talby, D. (2018, June 5). *Lessons learned turning machine learning models into real products and services.* Retrieved from https://www.oreilly.com/radar: https://www.oreilly.com/radar/lessons-learned-turning-machine-learning-models-into-real-products-and-services/

Treveil, M., Omont, N., Stenac, C., Lefevre, K., Phan, D., Zentici, J., . . . Heidmann, L. (2020). *Introducing MLOps.* New York: O'Reilly.

Visengeriyeva, L., Kammer, A., Bär, I., Kniesz, A., Plöd, M., Eberstaller, S. (2023, March 30) *MLOps Principles*. https://ml-ops.org/content/mlops-principles.