Volume 2: Technical Volume

DeepVerify: Deep neural-net verification framework with provable guarantees

1.0 Identification and Significance of the Problem or Opportunity.

Background: Due to tremendous progress and heightened awareness of AI and Machine Learning, AI applications are being implemented and deployed at a break-neck speed. Specifically, neural networks have recently been deployed in several safety and security critical systems such as self-driving vehicles, autonomous systems/ UAVs, and aircraft collision avoidance, medical devices, cancer detection among others.

However, AI systems are not trustworthy at present as they are opaque, large-scale, nonlinear, non-convex and often incomprehensible. Moreover, the networks lack rigorous testing and inadequate specs. Often times, AI systems demonstrate unexpected or incorrect behaviors in corner cases for several reasons including adversarial attacks, ill-specified goals, biased training data, mismatch between training and deployment environment and overfitting, and/or underfitting of the models. In safety- and security-critical settings, such incorrect behaviors can lead to disastrous consequences. Therefore, there is an urgent need for methods that can provide formal guarantees about the behavioral properties and specifications of AI tools.

The following are the major requirements of DARPA for this solicitation:

- *DARPA is looking for innovative, appropriate and useful verification and validation (V&V) techniques that can provide a framework to define, develop and demonstrate novelty-robust military & commercial AI applications.*
- *DARPA's required framework must be designed to handle different degrees of autonomy and applicable across multiple application domains.*
- *The ultimate outcome of this proposal would be to verify the performance specs of AI primarily in military applications and to ensure unacceptable behaviors will not occur in unknown conditions.*

Current State of Verification & Validation in AI applications

AI/ML industry is nascent: AI/ML industry is in a nascent stage although AI/ML algorithms have been in existence for the last several decades. We have experienced rapid growth and awareness only in the last couple of years. AI/ML developer tools, implementation techniques, debugging tools are all in a state of flux and the media over-hype is making things worse with all the positive and negative publicity. Ian Goodfellow summarizes the current state of AI/ML as follows: "It is clear that testing of naturally occurring inputs is sufficient for traditional machine learning applications, but verification of unusual inputs is necessary for security guarantees. We should verify, but so far we only know how to test."

Complexity of AI frameworks: Traditional V&V methods cannot be applied as is on AI applications due to the very nature of AI/ML. However, there are several valuable lessons and key technologies of conventional V&V can be utilized with certain modifications for the AI applications.

For example, Deep learning (DL) defines a data-driven programming paradigm that automatically composes the system decision logic from the training data unlike a program-based system where the decision logic is embedded in the program source code. Among different architectures of Deep Learning systems, Convolutional Neural Networks (CNN), a feed-forward DL system used heavily in Image recognition applications and Recurrent Neural Networks (RNN) have totally different architecture, use cases and vulnerabilities. In contrast to CNN systems, recurrent neural networks (RNN) follow a very different architectural design, implementing temporal behaviors and "memory" with loops and internal states. Such stateful nature of RNN contributes to its success in handling sequential inputs such as audio, natural languages and video processing. Reinforcement Learning (RL) has no semblance to RNN or CNN. It usually learns a function mapping states to actions, maximizing a reward function for each state-action pair. This learning is done by trial and error. There is no one-size-fits-all solution to the verification of machine learning algorithms. Different domains and different algorithms allow for different verification methods.

V&V efforts are scattered: Currently, there are no proven V&V and development methodologies for AI applications and the impact and implications of AI are barely understood by the developers and the users. Regulated industries like transportation, financial services and medical devices/ medical software, aerospace/aviation/ autonomous vehicles have felt the need and urgency for building robust and safe AI/ML applications and each of these industries are going on their own way of building V&V as their requirements are

unique. Safety standards needed for a medical device or a cancer detection software or a surgical robot has absolutely no relevance to a UAV or a driverless car or credit card application system although each of these applications may be using the same AI/ML programs/ framework. These industries are facing tremendous competitive pressure and they cannot afford to wait for the V&V standards to emerge over the next 10 years. Therefore, each industry is working on its own to build customized V&V for their specific industry requirement. There have been efforts from NIST to come up with set of AI standards, methods and procedures. IEEE has its own efforts on AI standardization. FDA is in the process of building its own AI verification standards for medical devices and medical software called "Software as a Medical Device (SaMD)". Federal Department of Transportation and each State Transportation Agencies are building their own driverless car safety and AI/ML testing procedures. NASA has its own V&V methods for the last 30 years for its home-grown AI/ML neural nets and the US Nuclear Regulatory Commission has its own V&V for systems using traditional old-style AI. The U.S. Army Aviation and Missile Research Development and Engineering Center (AMRDEC) Aviation Engineering Directorate (AED) has funded the Carnegie Mellon® Software Engineering Institute (SEI) to develop a reliability validation and improvement framework. Due to unique requirements of each industry, each industry body is working on its own to build a custom version of a V&V standard.

For example, just for computational simulation segment alone, American Society of Mechanical Engineers (ASME) has formed 6 separate V&V committees for each niche area/sub-segment such as Nuclear Systems, Energy Systems, Medical Devices, Advanced Manufacturing, Fluid Dynamics/heat transfer and Solid Mechanics. The reason is simple. Each one has their own V&V peculiarities which can only be defined within own industry segment.

Therefore, it will be a herculean task to build a single unified all-encompassing V&V methodology for all industries and for all types of AI/ML application environments. However, we can build a common framework with additional sub-framework applicable for specific domains.

Current research is focused on CNN: As articulated above, verifying machine learning applications especially neural networks with its very many variations, is a hard problem, and it has been demonstrated that validating even simple properties about their behavior is an NP-complete problem. AI research community is also not focusing on the one-size-fits-all strategy and there have been efforts to build a robust AI system by solving a very small area within the neural networks domain. Most of the research has been focused on trying to understand the functioning and limitations of a neural network and evaluating the robustness of Convolutional Neural Networks (CNN). Insignificant amount of work is being done on RNN and RL due to their inherent architectural complexity and fundamental differences.

Though we will be addressing both conventional ML and neural-net based ML in this proposal, our focus will be to apply them on neural networks as they have more serious problems, urgent needs and better market opportunity.

Lessons from traditional software V&V: Much work has focused on the generation of AI software but little of that work has utilized the decades of research into software V&V. Two results from Software Engineering (SE) are important here. First, as software grew more complex, software engineering moved away from a simplistic "theorem prover" approach and instead went to "sampling" method that better understand the input and output space of the software. With those methods it is possible to understand (a) the "shape" of the problem and (b) how to best "walk around that shape" to learn the range of system behaviors and (c) how to detect when the system has moved to a different place in that shape and (d) and how to mitigate for those moments. The second important result from SE V&V studies is that when system become complex, we cannot assume that automatic methods can fix all their problems. Hence, we propose a human in the loop system where AI tools repair themselves (when they can) and otherwise call out to humans for help.

2.0 Phase 1 Scope & Technical Objectives

To address the stated needs in this SBIR Phase 1 solicitation, we propose to develop a comprehensive V&V framework called "DeepVerify".

Our solution: *By leveraging and enhancing current opensource tools and technologies,* we are proposing **"**DeepVerify, a suite of tools to verify and certify robustness of different types of AI applications with a specific focus on currently popular neural networks such as deep CNN, reinforcement learning &recurring neural networks. Our tools can be deployed on other types of AI/ML frameworks with necessary modifications. Upon automated in-depth assessment of AI applications and exploration of input data space and risk modeling, DeepVerify will identify areas of concern, exceptions, red flags& bugs and provide a Certificate of Robustness (COR) and a suggested recommendation report for further action. Once the AI applications go into production, DeepVerify's Intelligent Run-time Agent (IRA) will provide a 24x7 run-time monitoring service to evaluate the production system and provide protective measures and insights based on specific use-cases.

DeepVerify is a suite of tools which address specific needs required in verification and validation of different types of neural networks under different circumstances. We will leverage/enhance/modify current available opensource tools into DeepVerify suite.

DeepVerify will develop its robustness validation process using the 3 major software components, namely:

i) DNN Toolkit for CNN, RNN and RL
ii) Bayesian Uncertainty Modeling covering CNN, RNN and RL.
iii) Probabilistic Verifier covering CNN

By utilizing its Risk Framework and Assumptions & Safety Violation Framework, DeepVerify will generate Certificate of Robustness (COR) as the final outcome. DeepVerify will commercialize its offering to enterprise and DOD customers.

By providing all the 3 options, DeepVerify can be applied to a wide range of AI applications. With DNN toolkit, We have the option to be "hard-wired" into the specifics of the AI tool we are testing/ controlling/ adjusting. With probabilistic verifier, an instance-based reasoner (augmented by deep learning) that explores the space of inputs/outputs around an AI tool. Bayesian Uncertainty modeler will enable us how to handle the unknowns. All these 3 modalities will operate under the broad framework of Risk, Assumptions and Safety.

DNN Toolkit for CNN, RNN and RL:

a) DeepVerify-DNN with guarantees: Based on a novel Reinforcement Learning based game approach for safety verification of DNNs, we propose two variants of pointwise robustness, the maximum safe radius problem, which for a given input sample computes the minimum distance to an adversarial example, and the feature robustness problem, which aims to quantify the robustness of individual features to adversarial perturbations. We will rely on Lipschitz constants to provide guaranteed bounds on DNN output for all possible inputs.
For the image classification networks, we will employ the saliency-guided & grey-box approach by W. Ruan, M.Wu, Y. Sun, X. Huang, D. Kroening, M. Kwiatkowska, "Global robustness evaluation of deep neural networks with provable guarantees for the L0 norm". For the feature guided black -box method, we will employ the SIFT object detection technique by D. G. Lowe, Distinctive image features from scale-invariant key points.

b) testRNN-Coverage-guided Testing on Recurrent Neural Networks: testRNN is a tool for testing and debugging LSTM networks. This whitebox testing tool is used to assess the internal structures of an LSTM network, and identify safety and robustness arguments for LSTM networks.

c) ARL Agent for Reinforcement Learning - Standard RL has limitations where the policies synthesized by the agent must satisfy strict constraints associated with the safety, reliability, performance and other critical aspects of the problem. Our proposed solution will not only allow RL verification, it can be extended to mission-critical and safety-critical systems with minor changes. Assured reinforcement learning (ARL) enables an autonomous agent to solve decision making problems under constraints. ARL approach models the uncertain environment as a high-level, abstract Markov decision process (AMDP), and uses probabilistic model checking to establish AMDP policies that satisfy a set of constraints defined in probabilistic temporal logic. These formally verified abstract policies are then used to restrict the RL agent's exploration of the solution space so as to avoid constraint violations.

Bayesian Uncertainty Modeling covering CNN, RNN and RL.

a) DeepVerify-Bayesian Uncertainty modeling: Bayesian deep learning (BDL) is a blend of deep learning and Bayesian probability theory and it provides a deep learning framework with state-of-the-art results which can also model uncertainty. Understanding what a model does not know is a critical part of many machine learning systems specifically in safety-critical and multi-task learning use-cases. Bayesian Uncertainty modeling works well on Safety-critical applications, applications where training data is sparse with small data sets, large data situations, real-time applications or multi-task learning where long computation run-time is not a choice.

Probabilistic Verifier covering CNN

a) DEEPVERIFY-Probablistic Verifier - is an instance-based reasoner (augmented by deep learning for feature detection) that explores the space of inputs/outputs around an AI tool. In this way, our technology can be applied to a wide range of AI tools. It is a novel non-parametric probabilistic programming method which generalizes and explores the synergies between probabilistic, sampling, and optimization approaches for V&V for AI. It approximates non-deterministic AI software as multiple small segments within which it is possible to say "these changes to a,b,c will cause those changes to x,y,z". The features used by DEEPVERIFY to cluster the data will be learned by a deep learner. We will then sample known systems behaviors and see where they occur across all known segments.

Our proposed method will borrow advanced Software Engineering V&V methods to the problem to AI tools and we will also leverage V&V methods successfully deployed by agencies like NASA, JPL and Nuclear Regulatory Commission for previous generation AI applications as well as efforts undertaken by NIST, FDA and ASME. We believe our proposed solution will address the needs of this solicitation.

2.4 Technical Objectives

This section discusses the general objectives of the entire proposal, plus the specific technical objectives of phase 1.

General technical objectives: Our work is guided by the following principles.

- We should try to leverage opensource as much as possible and utilize the key learnings from software engineering V&V as well as V&V from pioneering organizations like NASA, FDA, Nuclear Regulatory Commission etc;
- V&V is closely linked to understanding the risk, assumptions and safety violations. Without understanding these elements, V&V will not be holistic and complete;
- AI V&V tools should have multiple options to handle uncertainties using a bayesian modeler, blackbox /whitebox verification of specific AI tool as well as probabilistic instance-based reasoner which is AI tool independent. This will enable us to cover a wide variety of requirements. Most V&V research efforts are too narrowly focused and simply not viable as a business offering.
- AI V&V should not "just" about testing. Instead, it should be more about finding problems in the current system, identifying the risks, assumption violations and discover (a) segments where we prefer to be and (b) segments where we currently are.(but we don't want to be);
- Data is the nucleus of AI applications. For a successful V&V, Training data sets are the single most important area to focus on.

Specific Technical Objectives:

a) Understand the current state-of-the-art in novelty detection and V&V relating to AI applications;
b) Study the legacy V&V for AI, safety assurance and autonomous systems from pioneering organizations including NASA, FDA and others;
c) Conduct an indepth analysis of challenges in building a V&V for AI applications with specific emphasis on neural networks;
d) Understand the Risk framework and assumption frameworks relating to AI applications;
e) Evaluate multiple options to build a viable software/service product offering;
f) Prepare a road-map for Phase 2 implementation and create a viable prototype that can be commercialized.

3.0 Phase 1 Statement of Work

Over this entire proposal, this research will explore the current state of the art and significantly improve it. Specifically, we will conduct the following work:

Task 1: Conduct a technical survey of existing state-of-the-art techniques & technologies that address novelty and V& V issues in AI frameworks/ applications, identify key learnings from traditional V&V methodologies and apply them to AI frameworks & applications and develop a set of recommendations

Sub-task 1A: Study of V&V methodologies used/proposed by NASA/FDA/JPL etc.

The following are some of the major regulated organizations/ environments that have used neural networks and uncertainty modeling very successfully over the last 2 decades and it is imperative that we study these systems/ V&V procedures to gain deeper understanding and key lessons learned.

- FDA regulations for medical devices/ software that use AI (SaMD): Study the IMDRF risk framework of FDA, Algorithm Change Protocol (ACP) for medical software.
- NASA's use of SOM neural net "Dynamic Cell Structures (DCS)" used in the design of dual neural-nets for its Intelligent Flight Control System (IFCS), an autonomous safety critical adaptive system
- Systems/algorithms validation/ testing procedures used by US Nuclear Regulatory Commission.
- US Federal Reserve regulations SR 11-7 and OCC 2011-12 for model risk management relating to ML models. https://www.federalreserve.gov/supervisionreg/srletters/sr1107.htm
- JPL's Beacon-Based Exception Analysis for Multimissions (BEAM) for Technology for Autonomous Self-Analysis https://tmo.jpl.nasa.gov/progress_report/42-144/144A.pdf
- Gartner report on how to operationalize AI. (https://www.gartner.com/doc/reprints?id=1-5EX8T0M&ct=180906&st=sb)
- Accenture's "Teach and Test" methodology for testing AI applications.
- Canadian government's AI risk management framework https://www.tbs-sct.gc.ca/pol/doc-eng.aspx?id=32592
- UK's The Safety-Critical Systems Club guidelines – https://scsc.uk/

Sub-task 1B: In this task, we will conduct a detailed survey of various verification techniques explored by AI researchers and identify the most relevant techniques that are practical, scalable and can withstand rigorous scrutiny. We will give preference to research concepts that are accompanied by source code for further validation of the research concept.

Survey of various research papers relating to various elements of V&V in AI applications.

- Identifying Unknown Unknowns in the Open World: Representations and Policies for Guided Exploration https://arxiv.org/pdf/1610.09064.pdf
- **MIT's** 'Evaluating Robustness of Neural Networks with **Mixed Integer Programming' tool**, an optimization method used to find a maximum of some objective function, given certain constraints on the variables, and can be designed to scale efficiently to evaluating the robustness of complex neural networks.
- **Google's TCAV:** Testing with Concept Activation Vectors (TCAV) is a new interpretability method to understand what signals your neural networks models uses for prediction.
- Adversarial approaches for uncovering catastrophic failures in safety-critical settings by using RL agents. https://arxiv.org/pdf/1812.01647.pdf
- Stanford's Marabou framework for verification of neural network http://aisafety.stanford.edu/marabou/MarabouCAV2019.pdf
- **Metamorphic testing of ML applications to uncover issues in running applications** https://arxiv.org/pdf/1907.06632v1.pdf
- **Adaptive Stress Testing**: Finding Failure Events with Reinforcement Learning for large systems. https://arxiv.org/pdf/1811.02188.pdf
- **Training verified learners with learned verifiers.** https://arxiv.org/abs/1805.10265
- **Adversarial robustness testing** - Matthias Hein https://arxiv.org/pdf/1812.00740.pdf
- **Imagenet Adversarial – Imagenet-A** https://arxiv.org/pdf/1907.07174.pdf
- **IBM's Adversarial Robustness Toolbox for DNN attacks, defenses and metrics** https://github.com/IBM/adversarial-robustness-toolbox
- Statistical Mechanics Methods for Discovering Knowledge from Production-Scale Neural Networks Charles H. Martin and Michael W. Mahoney, UC Berkeley
- **MIT's lottery ticket algorithm** is a new way to build tiny neural networks for DOD's edge computing use-cases. https://www.technologyreview.com/s/613514/a-new-way-to-build-tiny-neural-networks-could-create-powerful-ai-on-your-phone/
- **Google Vizier Optimization Tool** https://github.com/tobegit3hub/advisor

Task 2: Identify a comprehensive list of technical challenges and uniqueness of AI application verification as compared to other traditional systems and study the risk & assumption framework.

Although V&V techniques and processes are well-established for typical deterministic software, they are either missing or inadequate for ML software. We are yet to build standards and guidance for AI/ML functional and validation specifications, testing standards, model evaluation and performance, model impact, risk assessment and run-time monitoring etc. Moreover, large-scale implementation of AI/ML applications have begun only in the last couple of years. Hence, the V&V in AI/ML industry is non-existent. However, the federally regulated industries are beginning to build their own V&V standards. Industries that are not regulated have not yet found a compelling reason to implement a robust V&V. However, the hype & anxiety created by media on AI is building pressure on large companies to evaluate the risks and dangers of AI. We believe this will lead to urgent efforts in building V&V for AI.

As stated earlier, the AI research community has been making efforts to build a verification model for AI but the efforts are focused largely on CNNs and to a lesser extent in RNN, RL and other types of neural networks. There were two major surveys conducted on various verification methods attempted by the research community are listed below. Based on our study, there are approximately over 100 different methods and approaches to solve the verification challenges in neural networks. However, most are focused on a very small area of CNNs and do not cover the entire gamut of neural networks.

https://arxiv.org/pdf/1810.01989.pdf Verification of ML survey

https://arxiv.org/pdf/1812.08342.pdf survey of safety and trust of ML

The solutions range from Mixed Integer Programming, adaptive testing, game-based verification, classic Knn approach, neuron-based coverage, Lyapunov analysis of network stability, DeepTaylor decomposition, MC/DC coverage, KeYmaeraX theorem prover, RL via policy extraction, Introspection learning, Neural nets into tractable Boolean circuits and the list goes on.

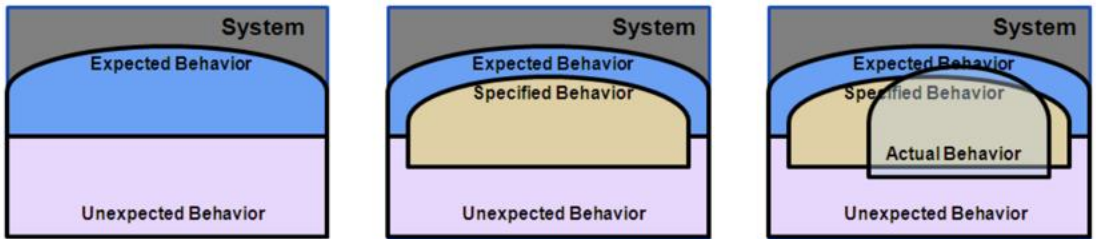
We have shortlisted several interesting ideas proposed in the research publications but we suspect most are not practically viable or scalable in a real production environment. Some of them are simply academic while a few are definitely worth pursuing.

Sub-task 2a: Technical Challenges and Uniqueness of AI applications with reference to V&V:
The difficulties encountered in verification mainly arise from the presence of activation functions and the complex structure of neural networks which are training-data driven. Moreover, neural networks are large-scale, nonlinear, non-convex, and often incomprehensible to humans.

The key challenges for mathematically rigorous V&V are:
(i) There is typically a lack of a physics-based oracle against which to V&V ML results;
(ii) The high-dimensionality of the hypothesis and problem spaces introduces risk in visual support for V&V;
(iii) Nonlinear computations reduce or eliminate the usefulness of standard deterministic software V&V techniques;
(iv) Probabilistic reasoning, with different computations and results from different runs of the same software, requires the ability to mathematically pose and evaluate "correctness" classes based on the hypothesis class and data space and
(v) V&V of massively parallel implementations requires coupling abstractions and probabilistic specifications.

Lack of Data Validation in ML: Currently, there are no mathematical foundations in ML to define data validation (e.g. "Does the data comprise a valid representation of the problem space?") and valid boundaries for the ML problem itself. Unlike a traditional programming language, ML models learn from the data and very data is the basis for evaluating the incoming test or live data.
Training of neural networks: Neural networks are trained over a finite amount of input and output data, and are expected to generalize said data and produce desirable outputs for the given inputs and even for previously unseen inputs. The data-driven nature and lack of efficient methods for analysis of neural networks leads to, in most cases, the treatment of neural networks as black boxes with no assurance in safety.
ML Validation: ML validation has focused on cross-validation and information loss, and while necessary, this is insufficient in that it often disregards data validation and model (and parameter) selection. To date, little R&D has been conducted in mathematically rigorous validation for ML.
Visualization: Mathematically determine the information loss from abstractions required in high dimension visualization and determine the bounds of information loss beyond which the visualization is inadequate for V&V support. Due to complexity and opaque nature of neural networks, there are no significant tools that can visually communicate the information loss from abstractions.

Sub-task 2b: Assumptions & Safety Violation Framework
Assumptions:
Like any other application software, AI applications operate on certain assumptions. During the development of the system, it is imperative that we explicitly state all assumptions made on the system. This will prevent any unexpected violations as we will be able to recognize situations in which the system might behave unpredictably. In short, we should know all known unknowns and have least number of unknown unknowns. Depending on the risk & ripple effect analysis of the application, we will have to build additional precautions to protect the system further.



We can break these assumptions into four major categories.
- Operating environment assumptions including presence of sensors and other environmental factors
  Examples: The gearbox is not used on slopes with a gradient of more than 20%.

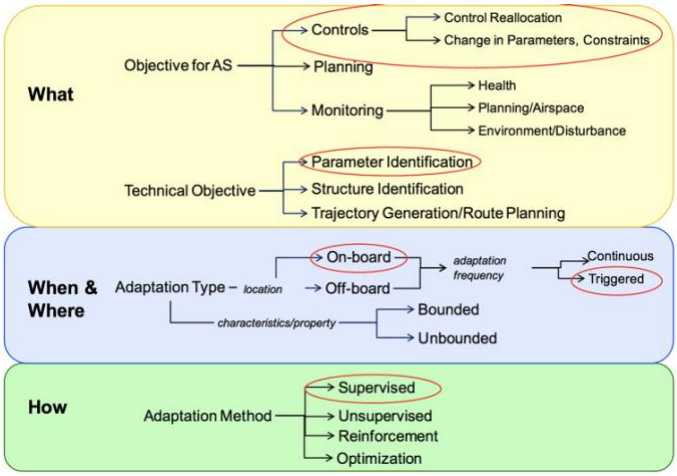
The controller is only used when the car moves forward.

- Platform assumptions including sensor failure assumptions & quality/fidelity assumptions;
  Examples: The engine and gearbox are working as expected.
  The engine can handle RPM between 1000 and 6000.
  The sensor readings have a +/- 5% accuracy.
  The sensors are assumed not to fail (from a software perspective).
- Data assumptions
  Example: The data is subject to the proximity assumption.
  The data is subject to the smoothness assumption.
  No adversarial forces are working against the algorithm.
- Algorithm assumptions.
  Example: Since the algorithm has a formal convergence guarantee, it is assumed to converge

Data assumptions

The training data of a ML algorithm defines the algorithm's functioning and in other words, training data is the ML. Therefore, reasoning about the data becomes extremely important and should be done with care. *Reference: Larry Wasserman. The role of assumptions in machine learning and statistics: Don't drink the koolaid! Technical report, Carnegie Mellon University, 2015*

Some of the data assumptions to be validated include a) evaluation of independent and identically distributed samples b)data proximity, c)data smoothness and d)adversarial behavior.



Algorithm assumptions

- Validate whether the desired functionality can be learned from data.
- Validate whether the chosen algorithm and its structure are able to accurately model the system to be learned.
- Validate whether the chosen cost or goal function is sufficient to get good behavior
- Validate whether the algorithm does not get stuck in a local optimum but reaches an global optimum
- Validate whether the observations made on test data will hold good in the presence of sufficiently bad data.

Interaction among the assumptions

Since all these assumptions refer to the same system, they are not independent of each other and they are intertwined to varying degrees depending application domain. Theoretically this means that a violated environment assumption could cause a cascading violation leading to an assumption violation on the platform, eventually violating an assumption on the algorithm. Therefore, documenting and assessing the risk on interaction among assumptions has to be included as part of assumption violation document.

For offline verification of RL learners were successfully done Goodrich and Barron Associates for their NASA proposal to replace large lookup tables, used in aircraft fuel measurement systems, with neural network. (source: NASA research document referred below). Prof. Sebastian Thrun's Validity Interval Analysis is stated to be another innovative approach to extracting symbolic knowledge from a neural network in a provably correct manner. We are yet to research on these two solutions and we will explore them further during Phase-1 research.

Define Safety Requirements – An example:

- The gearbox controller does not shift down in first gear.

- The gearbox controller does not shift up in fifth gear.
- The gearbox controller does not cause the engine to go over 6000 RPM.
- The gearbox controller does not cause the engine to go under 1000 RPM in any gear but first.
- The controller should not shift up or down immediately twice in a row.
- The gearbox controller should not shift too quickly.
- Convergence should be guaranteed.
- The rate of convergence should be bounded.

List of verifiable elements in a RL neural network

Looking at the example of verification of reinforcement learning it is useful to catalog the aspects of these algorithms that could be verified. A distinction is made between aspects of the system that can be verified offline and those that have to be verified at runtime.

Off-line Verification:

- Off-line Verification elements such as State to action mappings, action sequences, Algorithm properties independent of data and Validating assumptions on training data

Runtime Verification

Many properties that cannot be verified offline can be verified at runtime, although this might not always be feasible with regards to computation time or resource efficiency. Instead of verifying the entire specification, only the affected parts can be verified at runtime, assuming that if the specification was verified at the start of learning, and each change is deemed valid, then the specification is still valid after an arbitrary number of changes.

- State to action mappings,
- Action sequences.
- Monitoring (violations of) assumptions.
- Input instance in training data.

As part of our phase 1 research, we will study the research papers produced by NASA researchers on challenges in the verification of RL algorithms and assessment of System-wide Safety and Assurance Technologies.

Sub-task 2c: Risk Framework

Based on the preliminary analysis after studying FDA and NASA systems, we envisage three classes of V&V stringency for AI applications.

V&V Classes for AI applications:

Class 3	calls for a minimum amount of V&V and is associated with low judged system complexity and low required integrity
Class 2	calls for substantial V&V and is associated with medium-high complexity and medium required integrity
Class 1	entails maximum levels of V&V, being associated with high integrity requirements and high system complexity

We will define performance measures and decision criteria, generic measures and criteria for each of the risk classes.

The three common measures identified for all life cycle phases are:

number of show- stoppers	discovered errors which are so pervasive or general that they bring into question the whole design structure of the system
number of major problems	problems which, while not show stoppers, still require substantial corrective action over a number of elements or modules of the system
number of minor problems	problems which are confined to highly localized parts of the system and which are easily corrected

We will define a standard set of thresholds for AI V&V methods, one set for each of the three Classes of V&V, for each of the three possible outcomes:

(unconditional) Acceptance	Acceptance, requires the developers only to fix any detected errors before continuing to the next activity.
Conditional Acceptance	Conditional Acceptance, requires fixing of errors and then a repeat of appropriate V&V Guideline Package. The last outcome, Rejection, calls for a cessation of further V&V and development until the issues identified by the V&V methods are resolved by management discussion
Rejection	Rejection, requires fixing of all errors before the work can proceed.

The key point in our approach will be driven by two types of risks:

(1) the consequences of failure, based on required integrity and the judged level of complexity, which determines the recommended V&V stringency, and
(2) the risks associated with the faults possible for each life-cycle artifact, which determines the specific V&V method recommended.

Complexity

Is your system a safety system or designed specifically to support or relate to a safety system	High
Does your system control anything or provide real-time advice to an operator to control something, turn on systems or machines or take a decision?	High
Is there any conceivable way that failure/delay, unitended repetition or any other kind of malfunction could directly or indirectly cause damage or harm to anything?	High
Does your system involve near or at-real-time processing which involve complex reasoning, embedded processing and/or connected to a safety system or large number of complex interacting systems?	High
Is your System basically a stand-alone or user-driven system with minimal interaction	Low

Integrity

Evaluation of economic, legal environmental, ethical and business consequences of a failure or incorrect operation of the AI system including:

- Loss of human lives/ human injuries/ long-term health/ discomfort
- Interruption of system service/ disruption of system's mission/ financial loss/ loss of information
- Impact on organization's capability to perform/ Damage to brand image

If the consequences of the above events to be unacceptable, then the AI application integrity must be	High
If the consequences of the above events to be reasonably acceptable, then the integrity must be	Low
If the consequences are in between the above two, then it is	Medium

FDA's risk categorization based on state of healthcare situation or condition was used as one of the references during our preliminary study.

State of healthcare situation or condition	Significance of information provided by SaMD to healthcare decision		
	Treat or diagnose	Drive clinical management	Inform clinical management
Critical	IV	III	II
Serious	III	II	I
Non-serious	II	I	I

Figure 1: SaMD IMDRF risk categorization

Table 4. Proposed Level of Review for Level 1 and Level 2 Precertified Organizations' SaMD in Future Pre-Cert Program

IMDRF Risk Categorization		Level of Review for Level 1 and Level 2 Precertified Organizations' SaMD		
Type	Description	Initial product	Major changes	Minor changes
Type IV	Critical x diagnose/treat	SR	SR	No Review
Type III	Critical x drive		L1 – SR L2 – No Review	
Type III	Serious x diagnose/treat			
Type II	Serious x drive	L1 – SR L2 – No Review		
Type II	Non-serious x diagnose/treat		No Review	
Type II	Critical x inform			
Type I	Non-serious x drive	No Review		
Type I	Serious x inform			
Type I	Non-serious x inform			

Task 3: Develop a conceptual and technical V&V framework for novelty situations in AI applications and create a design document for building & prototyping the proposed framework in phase 2.

In this task, we will expand on the DeepVerify components including DNN Toolkit, Bayesian Uncertainty Modeler and Probabilistic Verifier and build a conceptual and technical framework to cover novelty situations and build a V&V conceptual and technical framework. It has to be noted that we will not build these tools from scratch and we will leverage/enhance existing opensource tools.

Task 3.1 Build a conceptual and technical V&V framework for DNN Toolkit for CNN, RNN and RL:

DeepVerify-DNN with guarantees:
Based on a novel Reinforcement Learning based game approach for safety verification of DNNs, we propose two variants of pointwise robustness, the maximum safe radius problem, which for a given input sample computes the minimum distance to an adversarial example, and the feature robustness problem, which aims to quantify the robustness of individual features to adversarial perturbations. We will rely on Lipschitz constants to provide guaranteed bounds on DNN output for all possible inputs.

Maximum Safety Radius (MSR)
The Maximum Safety Radius (MSR) aims to compute for a given input the minimum distance to an adversarial example, and therefore can be regarded as the computation of an absolute safety radius, within which no adversarial example exists.

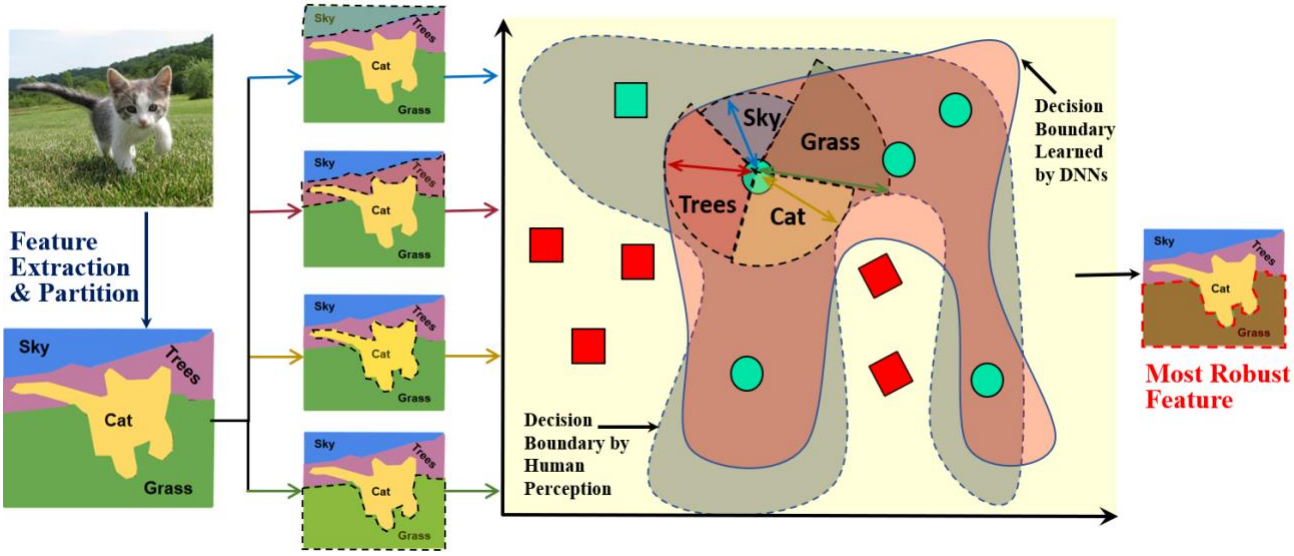


Illustration of the feature robustness (FRA) problem, which aims to find, on an original image, a feature, or a subset of features, that is the most robust against adversarial perturbations. Given a benign image, we first apply feature extraction or semantic partitioning methods to produce a set of disjoint features (`Sky', `Trees', `Cat', etc.), then we find a set of robust features that is most resilient to adversarial perturbations (`Grass' in the above diagram),
which quantifies the most robust direction in a safe norm ball.

Feature Robustness:
The "feature robustness" studies whether the crafting of adversarial examples can be controlled
by restricting perturbations to only certain features (disjoint sets of input dimensions), and therefore can be seen as the computation of a relative safety radius, within which the existence of adversarial examples is controllable.

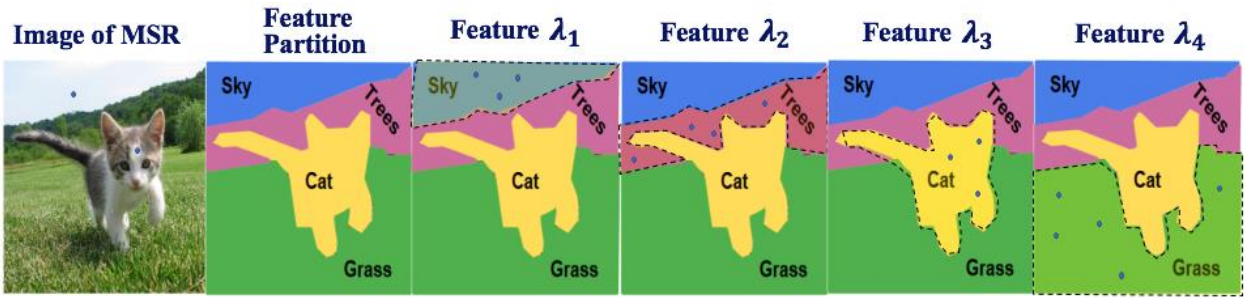


Illustration of the maximum safe radius (MSR) and feature robustness (FRA) problems. From left to right: an adversarial example with two pixel changes, feature extraction of the image, adversarial examples with three changed pixels on features `Sky' and `Cat', four changed pixels on `Trees', and five pixel manipulations on `Grass', respectively.

Intuitively, the search for the most robust feature alternates between maximizing over the features and minimizing over the possible input dimensions within the selected feature, with the distance to the adversarial example as the objective.

Both pointwise robustness problems are formally expressed in terms of nonlinear optimization, which is computationally challenging for realistically-sized networks. We thus utilize Lipschitz continuity of DNN layers, which bounds the maximal rate of change of outputs of a function with respect to the change of inputs, as proposed for neural networks with differentiable layers.

This enables safety verification by relying on Lipschitz constants to provide guaranteed bounds on DNN output for all possible inputs. We work with modern DNNs whose layers, e.g., ReLU, may not be differentiable, and reduce the verification to infinite optimization. More precisely, we prove that under the assumption of Lipschitz continuity, it is sufficient to consider a finite number of uniformly sampled inputs when the distances between the inputs are small, and that this reduction has provable guarantees, in the sense of the error being bounded by the distance between sampled inputs.

Then the finite optimization problems can be computed as the solution of two-player turn-based games, where Player I select features and Player II then performs a perturbation within the selected features. After both players have made their choices, the input is perturbed and the game continues. While Player II aims to minimize the distance to an adversarial example, Player I can be cooperative or competitive. When it is cooperative, the optimal reward of Player I is equal to the maximum safe radius. On the other hand, when it is competitive the optimal reward of Player I quantifies feature robustness. Finally, because the state space of the game models is intractable, we employ an anytime approach to compute the upper and lower bounds of Player I optimal reward. The anytime approach ensures that the bounds can be gradually, but strictly, improved so that they eventually converge.

More specifically, we will apply Monte Carlo tree search algorithm to compute the upper bounds for both games, and Admissible A* and Alpha-Beta Pruning, respectively, to compute the lower bounds for the games.

We will implement the method in a software tool DeepGame2, and conduct experiments on DNNs to show convergence of lower and upper bounds for the maximum safe radius and feature robustness problems. Our approach can be configured to work with a variety of feature extraction methods that partition the input, for example image segmentation, with simple adaptations.

testRNN-Coverage-guided Testing on Recurrent Neural Networks: testRNN is a tool for testing and debugging LSTM networks. This whitebox testing tool is used to assess the internal structures of an LSTM network, and identify safety and robustness arguments for LSTM networks. testRNN uses the test metrics based on the work done by W. Huang, Y. Sun, J. Sharp, and X. Huang, “Test metrics for recurrent neural networks,” 2019. The test metrics are designed to exploit different functional components of the LSTM networks, by considering both the one-step information change and the multi-step information evolution; the information is extracted from gate vectors and hidden state vectors. To refrain from “gaming against criteria”, the test case generation avoids using test metrics as targets, as recommended by Chilensky and Miller. RNNs are abstracted into state machines, based on which quantitative analysis metrics are applied.

testRNN is useful for both model designers and model users. Before putting their LSTM models into practical use, designers can apply coverage-guided testing to find adversarial examples, which can be used for model improvement via e.g., data augmentation. For users, they may refer to testRNN as a useful tool to see if the LSTM applications are sufficiently robust for the usage in critical circumstances.

testRNN provides a lot of useful features for LSTM testing. As an example, it records the number of times a test-condition is covered, according to the generated test suite. Using testRNN, one can understand the learning mechanism behind the LSTM model.

Advantages of testRNN are: a) it generates test cases and can find a number of adversarial examples complying with the specified test-conditions; b) users can use testRNN to compare the robustness of different LSTMs. c) the adversarial examples can be utilized to retrain or improve a model. d) the plots on coverage times can give intuitive explanations on how a model learns from a dataset.

ARL Agent for Reinforcement Learning - Standard RL has limitations where the policies synthesized by the agent must satisfy strict constraints associated with the safety, reliability, performance and other critical aspects of the problem. Our proposed solution will not only allow RL verification, it can be extended to mission-critical and safety-critical systems with minor changes. Assured reinforcement learning (ARL) enables an autonomous agent to solve decision making problems under constraints. ARL approach models the uncertain environment as a high-level, abstract Markov decision process (AMDP), and uses probabilistic model checking to establish AMDP policies that satisfy a set of constraints defined in probabilistic temporal logic. These formally verified abstract policies are then used to restrict the RL agent's exploration of the solution space so as to avoid constraint violations.

Task 3.2 Build a conceptual and technical framework for Bayesian Uncertainty Modeling covering CNN, RNN and RL.

DeepVerify-Bayesian Uncertainty modeling: Bayesian deep learning (BDL) is a blend of deep learning and Bayesian probability theory and it provides a deep learning framework with state-of-the-art results which can also model uncertainty. Understanding what a model does not know is a critical part of many machine learning systems specifically in safety-critical and multi-task learning use-cases. These deep architectures can model complex tasks by leveraging the hierarchical representation power of deep learning, while also being able to infer complex multi-modal posterior distributions. Bayesian deep learning models typically form uncertainty estimates by either placing distributions over model weights, or by learning a direct mapping to probabilistic outputs.
Bayesian Uncertainty modeling works well on Safety-critical applications, applications where training data is sparse with small data sets, large data situations, real-time applications or multi-task learning where long computation run-time is not a choice.

Types of uncertainty
There are actually different types of uncertainty and we need to understand which types are required for different applications. Among them, two are the most important types – epistemic and aleatoric uncertainty.

Epistemic uncertainty
Epistemic uncertainty captures our ignorance about which model generated our collected data. This uncertainty can be explained away given enough data, and is often referred to as model uncertainty. Epistemic uncertainty is really important to model for:

- Safety-critical applications, because epistemic uncertainty is required to understand examples which are different from training data,
- Small datasets where the training data is sparse.

Aleatoric uncertainty
Aleatoric uncertainty captures our uncertainty with respect to information which our data cannot explain. For example, aleatoric uncertainty in images can be attributed to occlusions (because cameras can't see through objects) or lack of visual features or over-exposed regions of an image, etc. It can be explained away with the ability to observe all explanatory variables with increasing precision.

Aleatoric uncertainty is very important to model for:

- Large data situations, where epistemic uncertainty is mostly explained away,
- Real-time applications, because we can form aleatoric models as a deterministic function of the input data, without expensive Monte Carlo sampling.

We can actually divide aleatoric into two further sub-categories:

- Data-dependent or Heteroscedastic uncertainty is aleatoric uncertainty which depends on the input data and is predicted as a model output.
- Task-dependent or Homoscedastic uncertainty is aleatoric uncertainty which is not dependent on the input data. It is not a model output, rather it is a quantity which stays constant for all input data and varies between different tasks. It can therefore be described as task-dependent uncertainty.

Task 3.3 Build a conceptual and technical framework for Probabilistic Verifier covering CNN

DEEPVERIFY-Probablistic Verifier - is an instance-based reasoner (augmented by deep learning for feature detection) that explores the space of inputs/outputs around an AI tool. In this way, our technology can be applied to a wide range of AI tools. It is a novel non-parametric probabilistic programming method which generalizes and explores the synergies between probabilistic, sampling, and optimization approaches for V&V for AI. It approximates non-deterministic AI software as multiple small segments within which it is

possible to say "these changes to a,b,c will cause those changes to x,y,z". The features used by DEEPVERIFY to cluster the data will be learned by a deep learner. We will then sample known systems behaviors and see where they occur across all known segments.

We can detect novelty by building a "certification envelope" around the software (define normal behavior as well has behaviors that have been studied and tested before). Software that executes within that certification envelope has known properties and software that violates that space requires further investigation. In standard practice, learning the features that make up the certification envelope required manual domain engineering. In our approach, we will develop an innovative feature extractor for the certification envelope based on deep learning. The SE testing-as-sampling methods described here do not assume any particular internal model/software format. Rather, they are domain general. Hence, they can be applied across multiple application domains.

In our work, once we have the certification envelope, we will apply learning tools to cluster that space, thus dividing the (harder) problem of analyzing the whole system into many (smaller) problems of monitoring parts of the system. This is important since a known property of software bugs is that they are highly localized. When the real-time performance behavior of the AI application deviates from the performance specs, our framework will either stabilize the application to its normal state or gracefully bring the system to a normal state or hand over the control for human intervention. The core principle is to ensure that the system does not crash if there is an aberration due to an unknown condition.

The premise of this proposal is that "old style" software V&V will not work for AI. However, in this age of agile service-oriented development, the nature of V&V in SE has radically changed. Hence, we assert that "new style" software V&V can be applied to AI. In the following, we will characterize that "new style" as a combination of (a) *probabilistic*, (b) *sampling*, and (c) *optimization-based* approaches. Accordingly, "new style" V&V for SE takes more of a (a) *probabilistic*, (b) *sampling*, and (c) *optimization-based* approach.

(a) ***Probabilistic****:* Researchers into probabilistic programming now represent software as regions (not points) of behavior so V&V becomes probabilistic statements across the space of all possible behaviors[1].

(b) ***Sampling:*** Researchers exploring testing for large (possibly non-deterministic) system now discuss test case prioritization methods where some oracles using domain knowledge to infer what tests might fail fastest, then runs those tests first. Before this approach was applied to Google's three billion-unit tests, it could take up to 80 hours to find a first failing test. After applying these methods, half the tests that fail were found in under 60 minutes[2].

(c) ***Optimization****:* The number of configuration options for modern software is growing exponentially[3] and those options can change the runtime, memory usage, response time of software by orders-of-magnitude[4]. For researchers exploring such systems, "testing" has become an "optimization" problem where poor performance is mitigated at runtime using state-of-the-art incremental optimizers to adjust the configuration options[5]. For example, when testing the software controlling autonomous cars, V&V engineers run a combination of neural nets and decision trees[6] to (firstly) find features that best describe the autonomous software controller then (secondly) divide data from the car into a tree of possibilities then (thirdly) re-run the neural nets within the local regions of data specified by the different branches of the neural net. This process of neural-net (to define the space) then decision tree learners (to divide the space) repeats until it is observed that the autonomous cars have predictable properties within each of the data regions defined by the decision tree branches.

Our proposed DEEPVERIFY Probabilistic Verifier, an extension and generalization of Dr.Tim Menzies' previous work that explores the synergies between probabilistic, sampling, and optimization approaches for V&V for AI. DEEPVERIFY segments and represent the system using a novel non-parametric probabilistic programming method. DEEPVERIFY approximates non-deterministic AI

[1] E. Wagenmakers and M. Lee. 2014. Bayesian Cognitive Modeling: A Practical Course. Cambridge Univesity Press.

[2] Sebastian Elbaum, Gregg Rothermel, John Penix, Techniques for Improving Regression Testing in Continuous Integration Development Environments" IEEE FSE 2014.

[3] T. Xu, L. Jin, X. Fan, Y. Zhou, S. Pasupathy, and R. Talwadker. 2015. Hey, you have given me too many knobs! understanding and dealing with over-designed configuration in system software. In (ESEC/FSE 2015).

[4] Chen, J., & Menzies, T. (2018, July). RIOT: a Stochastic-based Method for Scheduling in the Cloud. IEEE Cloud.

[5] V. Nair, Z. Yu, T. Menzies, N. Seigmund, S. Apel. Finding Faster Configurations using FLASH. IEEE TSE 2018.

[6] R. Abdessalem, S. Nejati, L. Briand, T. Stifter (2018), Testing Vision-Based Control Systems Using Learnable Evolutionary Algorithms, ICSE'18.

software as multiple small *segments* within which it is possible to say "these changes to a,b,c will cause those changes to x,y,z"[7]. The features used by DEEPVERIFY to cluster the data will be learned by a deep learner. We will then sample known systems behaviors and see where they occur across all known segments. These segments have three interesting types:

- *These segments have been rarely used in the past.* Such segments are worrying since it means the AI system has left the regions tested by prior work. In this case, we would adjust configurations in order to drive the system to nearby segments where the behavior is better understood.
- *These segments trend the system towards undesirable outcomes.* Such segments are also worrying since it means the AI system is trending towards sub-optimal behavior.
- *These segments trend the system toward more preferred outcomes.* This is the preferred place to me since when an AI system enters these segments, it is possible to improve its behavior. Note that, in these segments, optimization is possible.

Task 4: Develop a comprehensive & commercially implementable business plan to commercialize the technology. With rapid implementation of AI applications, there is an urgent market need for AI V&V. Defense, Aerospace, Financial Services, Federal and State agencies that provide citizen services, Healthcare and Insurance are prime targets for our proposed product/service.

We plan to offer the services of Certification of Robustness for AI applications using our proposed technology. Initally, our focus will be on AI applications running neural networks which is the fastest growing segment in the AI market. A detailed commercialization plan is outlined in Section 6.0.

Task 5: Feasibility Report/ Deliverables/ Milestone:

Phase 1 Deliverables:

Deliverable	Description	When
Research Report	With details on existing research relating to Novelty Characterizations	At the end of Month 3
Draft Feasibility	With details on the need for researching, developing, customizing scientific techniques to fulfil stated objective.	At the end of Month 4.
Final Feasibility Analysis Report	With results of researching and developing quantitative and qualitative characterizations of novelty, proposed methods for V&V of novelty-robust AI applications in military and commercial domains.	At the end of Month 6.
PI meeting presentation material	As specified by DARPA	When scheduled by DARPA.

Phase 1 milestones (in weeks)

Task	Description	Start Week	End Week
Task 1	Conduct a technical survey of existing state-of-the-art techniques & technologies that address novelty and V& V issues in AI frameworks/ applications, identify key learnings from traditional V&V methodologies and apply them to AI frameworks & applications and develop a set of recommendations	1	6
Task 2	Identify a comprehensive list of technical challenges and uniqueness of AI application verification as compared to other traditional systems and study the risk & assumption framework.	4	8

[7] We approximate AI system as hundreds of small nonparametric segments since a single parametric form covering all variables is hard to find for software systems. This is so since every "if" statement divides software into *segments*, each with different properties.

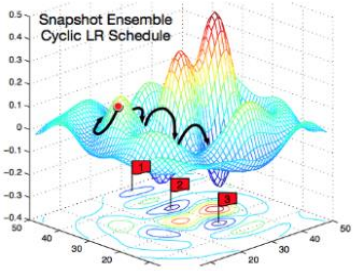
Task 3	Develop a conceptual and technical V&V framework for novelty situations in AI applications and create a design document for building & prototyping the proposed framework in phase 2.	8	22
Task 4	Develop a comprehensive & commercially implementable business plan to commercialize the technology.	18	22
Task 5	Generate & Submit Final Feasibility Report and other deliverables	16	24

4.0 Related Work:

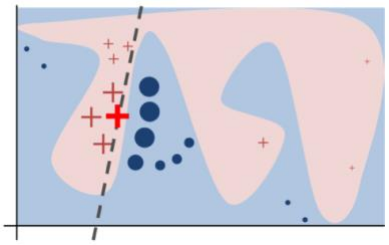
DeepVerify – Probabilistic Verifier

Our initial prototypes suggest there is much value in the DEEP-VERIFY approach. In deep learning networks, initial layers perform feature extraction which subsequent layers use as primitives in their inference process. The whole process is automatic and features can be automatically learned that were never anticipated by human designers. Better yet, if relationships change in the data then the Deep learner should be able to automatically adapt.

Deep learners suffer from long training times (hours to days to weeks) and opaque internal representations (internal nodes do not represent any human-level concept; knowledge is distributed over many weights between many nodes). Both these problems can be solved by segmenting the training of a deep learner, then reasoning locally within those segments. Note that this segmentation+reasoning process operationalizes the non-parametric sampling and probabilistic programming described above.



To understand the segmentation+reasoning , we note that control tools like **REN**[8] can speed up the learning via adjusting the learning rates by tracking the learner's progress in goal space (see figure at left), then adjusting (say) DL's stochastic gradient descent. Also, explanation algorithms like **LIME**[9] (a) ask users what conclusions they want explained then (b) building linear models from other examples local to that example. For example, shown at right is a decision boundary found by a Deep Learning. To explain why the red cross is classified pink (not blue), **LIME** builds linear models from the examples near the red cross.



In our own research, we have applied non-parametric sampling and probabilistic programming to improve **LIME** and **REN**. **LIME** suffers from conclusion instability (since it works from randomly selected instances). Also, **REN** changes deep learning runtimes only by a linear factor, but not the orders of magnitude needed to make deep learning more widely used. **LIME2** tame conclusion instability using a sampling method based on spectral clustering[10]. Also, **LIME**'s regression is merely a single goal learner and recently we have much success with generating better explanations for complex phenomena via optimization[11]. Hence, **LIME2** uses multi-objective sequential model-based optimization that builds models incrementally, then uses them to select the next best example to fold into the model. FYI: sequential model-based optimization is a kind of active learner. As to **REN**, this is an instance-based optimizer. Recently we have had much success with subsampling to speed up instance-based optimization[12]. **REN2** would use adaptive learning rates (learned via instance-based methods that cluster data, then only explore a sub-sample of items per cluster) can quickly (in time O(log(N)) accelerate training.

While successful prototypes in their original deployment area, we think **LIME2 and REN2** make too many limiting decisions about the nature of the system they are exploring. LIME2 is hard-wired for deep learners and there is no need for that. Clearly, LIME2 applies to any learner that generates a decision boundary. Also, REN2's clustering+optimize methods should applicable to any sample of data. Hence, in this work, we look to making DEEPVERIFY by refactoring **LIME2 and REN2,** throwing away

---

[8] M. Ren, W. Zeng, B. Yang, R. Urtasun. (2018). Learning to Reweight Examples for Robust Deep Learning. ICML'18.

[9] M. Ribeiro. S. Singh, C. Guestrin (2016), "Why Should I Trust you?" Explaining the Predictions of Any Classifier. KDD'2016.

[10] Nair, V., Menzies, T., Siegmund, N., & Apel, S. (2018). Faster discovery of faster system configurations with spectral learning. *Automated Software Engineering*, *25*(2), 247-277.

[11] V. Nair, Z. Yu, T. Menzies, N. Seigmund, S. Apel. (2018). Finding Faster Configurations using FLASH. IEEE Transactions SE.

[12] J. Chen, T. Menzie (2018), RIOT: a Stochastic-based Method for Workflow Scheduling in the Cloud. IEEE Cloud '18.

the unnecessary domain coupling, and generalizing them to a general system that better reasons about data, how it is performing badly, and how that performance could be made better. Also, rather than clustering or raw attributes (as done by **LIME2),** DEEPVERIFY will cluster on attributes inferred by deep **learning.** Once refactored in this way, DEEPVERIFY will become the V&V tool described above.

Computer Vision project for Picatinny Arsenal:
Quantum Ventura Inc has been recently selected by Picatinny Arsenal to build a computer vision based real-time object detection system for its remote weapon stations by using an on-board hardware which will integrate with optics and weapon system. The application will work on moving vehicles, rough sea conditions and on stationery vehicles. Due to challenging environment, limitations of neural network, issues relating to latency, we have gained immense knowledge on how to exploit the strengths of neural networks and work around the weaknesses/shortcomings of neural network environment.

5.0 Relationship with future work:

By providing an in-depth analysis of the input data space of AI applications, we can mitigate unexpected surprises during the test/development phase itself and allow the designers to take corrective action. In addition, DeepVerify's Intelligent Run-time Agent (IRA) will provide protective measures and insights into the production AI application.

Robustness verification and validation is an important function as more and more AI applications are being implemented. Matters of safety, reliability, trust, fear of bias, fear of exclusion are important factors in ensuring the AI applications are successfully implemented. Third party verification services will become a necessity in the coming years. For example, NASA has an internal rule that any AI application beyond certain size and complexity, would necessarily require external third-party verification. Based on NASA's experience, cost of V&V is twice the cost of development and this is where tools like our proposed DeepVerify will be of great help. DeepVerify automates as much as possible while giving the AI developer/user the ultimate power. Fully automated AI verification services are not there yet and it is very unlikely to have a completely automated verification services in the near future.

Today data scientists are spending enormous amount of time to create or locate data sets. Without large amount of data, prediction rates of neural networks will not be acceptable to users. With low prediction rates, end-users will lose confidence in the machine learning models. If the ML based systems have to succeed, training data is the single most crucial element and it is on par with algorithms. Validating the data and the models for any aberration adds critical business value to an enterprise. Therefore, we see significant value in our offering.

It is important to note the prevailing opensource data sets are catered towards commercial enterprises and very little data relating to defense industry are available. This will force the Defense industry to start generating synthetic data set which will further increase the necessity to evaluate the data even further.

We will be able to reduce the workload for the data scientists and management personnel and increase their productivity many folds. Our conservative estimation is a minimum 80-100% increase in productivity once the AI training system has enough training data and the data and models are validated for its robustness. Data Scientists can focus on refining the model performance and serve the business units. Using our tools, we can derive benefits like increased productivity, reduction of data scientist/ key user fatigue, better response to end-users and satisfied user base.

6.0 Commercialization Strategy:

Our services will cover: a) model robustness certification/ verification b) Data Sanity checking c) Risk framework analysis and suggested Recommendations d) Continuous monitoring service of AI applications. Our business model to commercialize this effort will be to build a cloud-based robustness certification/ verification services using DeepVerify for AI application modules. The proposed DeepVerify modules will be designed to be portable either as on-prem solution or for the commercial/secure GovCloud. Sensitive model and data set related information can continue to be hosted on client's private servers and made available in small chunks to the cloud system upon authentication. We will provide the DeepVerify solution either as a SaaS version or enterprise version or a hybrid cloud version. We will also explore offering our expert service for building the models for clients and we expect a significant service business to emerge out of this effort including training. Our initial target customers will include large government users, enterprise customers and defense users.
Protiviti.com specializes in model validation services for the financial industry and banks. For a comparative study of relevant business models used by model verification companies, we identified Protiviti.com as a good reference point and we intend to refine our business model by studying companies like Protiviti.

Customer Segments:

DeepVerify is relevant both for DOD and non-DOD customers.

DOD Domains:

Military applications of AI to verify performance specs or to verify that unacceptable behaviors will not occur in unknown conditions.

- Target detection systems such as Weapon control/ Firing Control systems (hand-held weapons), Remote Weapon Stations, UAVs
- Decision making tools that run on AIs.

Non-DOD Domains that require regulatory approvals to operate.

- Automobile companies to test the AI in driverless cars for approval to sell for use on public roads
- Companies that require FDA approval for Software/ medical devices that use AI
- Medical diagnostic companies or insurance companies that want to use AI for treatment decisions
- Consumer-facing finance companies using AI for credit decisions
- Aerospace companies requiring certification from FAA and other standards organizations

Contacting Potential Defense Agencies and Large Primes as Partners

As a certified 8A & HubZone Company and a member of National Armaments Consortium, NAMC and SOSSEC, TechFutures Group,a Silicon-valley SBDC, we have made significant contacts and progress in accessing the lucrative defense and aerospace industry. We will seek specific assistance of SBA Business Opportunity Specialists ('BOS"), Procurement Technical Assistance Centers ('PTAC"), Minority Business Development Agencies ("MBDA") and the National Minority Supplier Development Council to connect with and target Federal Agencies and large prime contractors. We continue to have partnership discussions with prime contractors including Lockheed, BAE Systems and others besides receiving letters of support. Our certifications including SBA 8A, HubZone, DOT andDOD NAC OTA consortium memberships will help us to explore sole source contracts, mentor-protégé teaming arrangements and JVs. We will also appoint channel partners and direct sales force to sell the products and services across US and Europe to begin with.

Challenges to the implementation:

We have listed the following anticipated challenges in building a V&V framework for AI applications: We will device a suitable mitigation for each challenge.

a) Legacy V&V took over 30 years to evolve. A single SBIR Phase 1 may be too short a time to build a framework. That is why, we have carefully chosen the most suitable opensource component that can address multiple situations and framework.
b) V&V for AI is still in a state of infancy. Neural networks and AI applications are very new and we are learning as we go. Newer AI-frameworks are mushrooming every day. Therefore, we may face unexpected hurdles and sudden changes in technology. We will be forced to adapt to the changing market conditions.
c) Opensource tools we plan to use are not mature and we may not get the desired results. Without deep understanding of neural networks, we will be able to overcome the challenges and take corrective action.
d) There could be government controls and legislation on AI applications in general. This may affect everyone involved in developing AI applications.
e) Some of the opensource programs may just provide the API without the underlying source code. During our study, we will identify alternative software sources to the maximum extent possible. Some of the opensource tools do not get periodic updates.
f) Some of the algorithms and solutions offered do not scale and purely academic in nature. With our wide business experience, we have developed an expertise on how to spot them early on.
g) We will ensure that no data privacy violations occur while we do AI data set evaluation.
h) Our successful implementation is dependent on transferring the necessary skills to end-user organizations. We will ensure that we transfer the knowledge to the fullest extent.

7.0 Key Personnel:

7.1 Srini Vasan, PI is the President & CEO of Quantum Ventura Inc and CTO of QuantumX, the R&D arm of Quantum Ventura Inc. https://www.linkedin.com/in/srini-vasan-b0269612/ He specializes in AI/ Advanced Machine Learning, large scale system development, system software engineering relating to Linux kernel and system internals. With a Bachelor degree in Math and a Cost Accounting degree from India, he received his management education from MIT Sloan School. As a techno entrepreneur with a unique blend of technology and management background, Srini has been able to seamlessly shift between technology, management, marketing and entrepreneurship at a deep level. Srini is a US citizen.

He is the PI for the current SBIR project for US Army's Defense Health Agency - project for building an intelligent agent with natural language understanding to interact with virtual patient software and medical mannequin.
Computer Vision Project: Recently got selected to deliver an advanced Optical Targeting System for Remote Weapon Stations for ARDEC, Picatinny Arsenal to identify moving objects using advanced computer vision/AI.
Reinforcement Learning Project: Srini Vasan is working with one of the world's leading experts in the emerging of Reinforcement Learning at Bosch Institute of Data Science and Artificial Intelligence to develop a highly scalable, distributed, parallel processing Reinforcement Algorithm on TensorFlow and Caffe frameworks. He is developing a tool in Cognitive Recognition and Verifiable AI with a team of Google trained Data Scientists. He is also currently building an NXP based computer vision/neural net application for identifying personnel movement in a secure environment.

Srini has hands-on development experience in designing and developing NLP, neural networks, logistic regression using Graph theory, SVM, Sentimental analysis techniques for their product social media analytics TwitterPlay. He has a deep understanding of current state of machine learning research in the area of Neural Networks.
Real-time Linux Kernel: Srini Vasan has good hands-on technical and systems knowledge in constructing a modified, custom real-time Linux kernel with Preempt RT which surpasses many RT Linux OS and we are awaiting an opportune time to commercialize this asset.
seL4 - World's Most Secure OS: Srini Vasan is the lead architect in the development of High Assurance Operating System (HAOS) framework on top of seL4. Sel4 is the first mathematically verified OS kernel which provides unsurpassed security and software assurance. In HAOS, all components are verified using SMACK, a mature open-source modular verifier. HAOS verification tool-chain uses co-design of OS code and verification algorithms to scale to the size of practical OS subsystems and mission-critical user-level applications. His earlier company, Compuflex public listed NASDAQ Company which was sold to another NASDAQ entity and Srini has significant contacts in the Private Equity and VC industry in Silicon Valley and New York.

7.2 Tim Menzies (IEEE Fellow, Ph.D., UNSW, 1995) is a full Professor in CS at North Carolina State University where he teaches software engineering, automated software engineering, and foundations of software science. He is the director of the **RAISE lab** (real world AI for SE). that explores SE, data mining, AI, search-based SE, and open access science. He is the author of over 250 referred publications and editor of three recent books summarized the state of the art in software analytics. In his career, he has been a lead researcher on projects for NSF, NIJ, DoD, NASA, USDA (funding totalling 11 million dollars) as well as joint research work with private companies. For 2002 to 2004, he was the software engineering research chair at NASA's software Independent Verification and Validation Facility. Prof. Menzies is the co-founder of the PROMISE conference series devoted to reproducible experiments in software engineering (**http://tiny.cc/seacraft**). He is an associate editor of IEEE Transactions on Software Engineering, Communications of the ACM, ACM Transactions on Software Engineering Methodologies, Empirical Software Engineering, the Automated Software Engineering Journal the Big Data Journal, Information Software Technology, IEEE Software, and the Software Quality Journal. In 2015, he served as co-chair for the ICSE'15 NIER track. He has served as co-general chair of ICSME'16 and co-PC-chair of SSBSE'17, and ASE'12. For more, see his vita (**http://menzies.us/pdf/MenziesCV.pdf** or his list of publications **http://tiny.cc/timpubs**) or his home page **http://menzies.us**.

Professional Preparation

University New South Wales,	Sydney,	BSc Computer Science,	1985
University New South Wales,	Sydney	MCogSc (Cognitive Science)	1989
University New South Wales	Sydney	Ph.D. (AI)	1996

Appointments

Professor, Computer Science	NC State	2014-
Assoc/Professor of CSEE	West Virginia University	2006-2014
NASA SE Research Chair	West Virginia	2001-2003
Research Assist/Professor	West Virginia University	1998 – 2005
OO+ and expert system consultant	Sydney, Australia	1986 – 1995

Relevant related, recent products

- Yu, Z., & Menzies, T. (2019). FAST2: An intelligent assistant for finding relevant papers. Expert Systems with Applications, 120, 57-71.
- Yu, Z., Kraft, N. A., & Menzies, T. (2018). Finding better active learners for faster literature reviews. Empirical Software Engineering, 23(6), 3161-3186.
- Mathew, G., Agrawal, A., & Menzies, T. (2019). Finding Trends in Software Research. IEEE Transactions on Software Engineering, preprint, to appear

- A. Agrawal, T. Menzies, Is “Better Data” is Better than “Better Data Miners”? ICSE 2018.
- JaechangNam, Wei Fu, Sunghun Kim, Tim Menzies, Lin Tan, Heterogeneous Defect Prediction. IEEE Trans. Software Eng.44(9): 874-896 (2018)

Other Significant Products

- Chen, J., Nair, V., Krishna, R., & Menzies, T. (2018). " Sampling" as a Baseline Optimizer for Search-based Software Engineering. IEEE Transactions on Software Engineering. 2018, to appear.
- Krishna, R., & Menzies, T. Bellwethers: A Baseline Method for Transfer Learning. IEEE Transactions on Software Engineering. 2018, to appear
- Choetkiertikul, M., Dam, H. K., Tran, T., Pham, T. T. M., Ghose, A., & Menzies, T. (2018). A deep learning model for estimating story points. IEEE Transactions on Software Engineering. (2018)
- Fu, Wei, Tim Menzies, and Xipeng Shen. "Tuning for software analytics: Is it really necessary." Information and Software Technology 76 (2016): 135-146.
- Krall, J.; Menzies, T.; Davies, M., "GALE: Geometric Active Learning for Search-Based Software Engineering," in IEEE Transactions on Software Engineering, , 2015)

8.0 **Foreign Citizens:** Key members of the project, Srini Vasan is a US Citizen and Dr.Tim Menzies is a US permanent resident (and an Australian citizen).

9.0 **Facilities & Equipment:** We do not anticipate any special facilities/equipment for the phase 1. Quantum Ventura Inc’s team will work from San Jose, CA and Prof. Tim Menzies of NC State University’s team will work from Raleigh, NC.

10.0 **Sub-contractors/ Consultants:** Quantum Ventura Inc has partnered with Dr.Tim Menzies of NC State University, Raleigh, NC. Letter of participation from NC State University enclosed as reference.

11.0 **Prior or pending support for similar applications**: Not applicable.

12.0 **Technical Data Rights** - Not applicable in phase 1 at the present time.

13.0 **Identification of Assertions** - No assertions

Quantum Ventura Inc's Commercialization Plan for DeepVerify

Our offering: **Quantum Ventura Inc** is proposing **"DeepVerify**", a suite of tools to verify and certify robustness of different types of AI applications with a specific focus on currently popular neural networks such as deep CNN, reinforcement learning &recurring neural networks. Quantum will leverage, enhance, integrate and automate current opensource tools and technologies to build DeepVerify and DeepVerify is not a development effort from scratch. Our tools can be deployed on other types of AI/ML frameworks with necessary modifications but our focus will be on neural networks based AI applications.

What we will deliver to a customer: Upon automated in-depth assessment of AI applications and exploration of input data space and risk modeling, DeepVerify will identify areas of concern, exceptions, red flags& bugs and provide a Certificate of Robustness (COR) and a suggested recommendation report for further action. Once the AI applications go into production, DeepVerify's Intelligent Run-time Agent (IRA) will provide a 24x7 run-time monitoring service to evaluate the production system and provide protective measures and insights based on specific use-cases. IRA is an optional, premium service. COR will be offered at Standard, Premium and Supreme levels. Depending on the levels chosen, depth of robustness certification will vary. COR could be a one-time or it could be purchased as a subscription.

Our services will cover: a) model robustness certification/ verification b) Data Sanity checking c) Risk framework analysis and suggested Recommendations d) Continuous monitoring service of AI applications.

Components of DeepVerify: DeepVerify is a suite of tools which address specific needs required in verification and validation of different types of neural networks under different circumstances. DeepVerify will develop its robustness validation process using the 3 major software components, namely:

i) DNN Toolkit for CNN, RNN and RL (typically used for blackbox/ whitebox robustness verification)
ii) Bayesian Uncertainty Modeling covering CNN, RNN and RL. (to be used mainly for handling unknown unknowns)
iii) Probabilistic Verifier covering CNN (framework independent instance-based reasoner)

By utilizing its Risk Framework and Assumptions & Safety Violation Framework, DeepVerify will go through a largely automated process and generate Certificate of Robustness (COR) as the final outcome.

By providing all the 3 options, DeepVerify can be applied to a wide range of AI applications. With DNN toolkit, we have the option to be "hard-wired" into the specifics of the AI tool we are testing/ controlling/ adjusting. With probabilistic verifier, an instance-based reasoner (augmented by deep learning) that explores the space of inputs/outputs around an AI tool. Bayesian Uncertainty modeler will enable us how to handle the unknowns. All these 3 modalities will operate under the broad framework of Risk, Assumptions and Safety.

Technology Readiness Levels: DeepVerify's software modules are largely opensource.

- DNN toolkit (TRL-3 currently and it will be TRL-4 at the end of Phase 2) is available as opensource which requires further changes, enhancements and integration.
- DeepVerify's Intelligent Runtime is yet to be developed but we have the technical architecture in place. By end of Phase 2, IRA will be TRL-3.
- Bayesian Uncertainly Modeling is currentlyTRL-2 and we expect it to be TRL-3 by the end of Phase 2.
- Probabilistic Verifier – TRL 3 and it will be TRL-4 by the end of Phase 2.
- Risk Framework/ Assumptions & Safety Violation framework are yet to be developed and we will comfortably reach TRL-3 at the end of phase 2.

Our business model: Our business model to commercialize this effort will be to build a cloud-based robustness certification/ verification services using DeepVerify for AI application modules. The proposed DeepVerify modules will be designed to be portable either as an on-prem solution or for the commercial/secure GovCloud. Sensitive model and data set related information can continue to be hosted on client's private servers and made available in small chunks to the cloud system upon authentication.

For a comparative study of relevant business models used by model verification companies, we identified Protiviti.com and missinglink.ai as a good reference points and we intend to refine our business model by studying companies like Protiviti and missinglink.ai. Missinglink.ai provides run-time monitoring service and model evaluation of production systems by providing a run-time agent to evaluate the performance of the AI model and suggests recommendation/improvements. Protiviti specializes in financial

algorithm/model validation/ evaluation services for the financial industry and banks. We believe these are two prime examples of successful business model relevant to our offerings.

We will provide the DeepVerify solution either as a SaaS version or enterprise version or a hybrid cloud version. We will also explore offering our expert service for building the models for clients and we expect a significant service business to emerge out of this effort including training.

Target markets: Our initial target customers will include large government users, enterprise customers, healthcare/ medical devices and defense users. Organizations fall under regulated industries will be our first target as regulatory agencies are quite concerned and uneasy about AI-based applications.

Customer Segments:
DeepVerify is relevant both for DOD and non-DOD customers.

DOD Domains:
Military applications of AI to verify performance specs or to verify that unacceptable behaviors will not occur in unknown conditions.

- Target detection systems such as Weapon control/ Firing Control systems (hand-held weapons), Remote Weapon Stations, UAVs
- Decision making tools that run on AIs.
- Defense Personnel evaluation systems
- AI based mission planning, optimization tools, autonomous platoons, autonomous weapon systems among others.
- Cyber systems that use AI.

Non-DOD Domains that require regulatory approvals to operate.

- Automobile companies to test the AI in driverless cars for approval to sell for use on public roads
- Companies that require FDA approval for Software/ medical devices that use AI
- Medical diagnostic companies or insurance companies that want to use AI for treatment decisions
- Consumer-facing finance companies using AI for credit decisions
- Aerospace companies requiring certification from FAA and other standards organizations
- Insurance companies
- Large enterprises that leverage AI for competitive business advantage.

Challenges to the implementation:
We have listed the following anticipated challenges in building a V&V framework for AI applications: We will device a suitable mitigation for each challenge.

i) Legacy V&V took over 30 years to evolve. A single SBIR Phase 1 may be too short a time to build a framework. That is why, we have carefully chosen the most suitable opensource component that can address multiple situations and framework.
j) V&V for AI is still in a state of infancy. Neural networks and AI applications are very new and we are learning as we go. Newer AI-frameworks are mushrooming every day. Therefore, we may face unexpected hurdles and sudden changes in technology. We will be forced to adapt to the changing market conditions.
k) Opensource tools we plan to use are not mature and we may not get the desired results. Without deep understanding of neural networks, we will be able to overcome the challenges and take corrective action.
l) There could be government controls and legislation on AI applications in general. This may affect everyone involved in developing AI applications.
m) Some of the opensource programs may just provide the API without the underlying source code. During our study, we will identify alternative software sources to the maximum extent possible. Some of the opensource tools do not get periodic updates.
n) Some of the algorithms and solutions offered do not scale and purely academic in nature. With our wide business experience, we have developed an expertise on how to spot them early on.
o) We will ensure that no data privacy violations occur while we do AI data set evaluation.

p) Our successful implementation is dependent on transferring the necessary skills to end-user organizations. We will ensure that we transfer the knowledge to the fullest extent.

Business Risks associated with launching DeepVerify.

a) AI/ML industry is literally less than 3-4 years old and lots of new frameworks, shortcomings of the existing framework are emerging daily. Our technical model has to keep pace with the changes or else we will be left behind. Quantum Ventura is actually involved in developing and deploying AI applications. So, we have the direct experience of deploying AI applications and we are continuously improving upon our skills to keep ourselves updated.
b) DeepVerify as a concept may not be accepted as the verification and validation by third parties is done rarely except in the regulated industries. Considering the serious concerns and opaque nature of AI, managements are looking desperately for a solution from independent providers.
c) Quantum may not have adequate financial resources to market the technology. Quantum plans to create partnerships with cloud companies and large system integrators to penetrate into the market. Quantum's Silicon Valley presence will be a major advantage in marketings its latest technology.
d) DeepVerify's technology may face intense competition from AI framework providers like Google(tensorflow) and Facebook (pytorch / caffe) as they may offer similar services. Quantum believes the AI market is large enough to accommodate multiple players. Since the industry is brand new, size and reputation do not matter as long as we deliver a top -quality product at a reasonable price. Quantum continues to follow this principle till date and we have been successful.

Marketing Strategy:

Quantum will market DeepVerify through cloud providers, strategic business partners, large system integrators and global distributors. Direct marketing will be confined to large enterprises and DOD. Cloud providers have a vested interest in selling our technology as it encourages further cloud business. Strategic business partners, large system integrators and global distributors can use the power of DeepVerify to enhance their consulting business as we are offering a valuable much-needed service for the AI industry.

Pricing Strategy:

- We intend to charge a nominal $50,000 per application assuming the total time taken for an application certification is 200 hours @$250/hour. Depending on application complexity and risk rating, we will charge additional fee.
- On-prem installation of our product will be at $250,000 per year.
- Our consulting services will be billed at $250/ hour.

Contacting Potential Defense Agencies and Large Primes as Partners

As a certified 8A & HubZone Company and a member of National Armaments Consortium, NAMC and SOSSEC, TechFutures Group,a Silicon-valley SBDC, we have made significant contacts and progress in accessing the lucrative defense and aerospace industry. We will seek specific assistance of SBA Business Opportunity Specialists ('BOS"), Procurement Technical Assistance Centers ('PTAC"), Minority Business Development Agencies ("MBDA") and the National Minority Supplier Development Council to connect with and target Federal Agencies and large prime contractors. We continue to have partnership discussions with prime contractors including Lockheed, BAE Systems and others besides receiving letters of support. Our certifications including SBA 8A, HubZone, DOT and DOD NAC OTA consortium memberships will help us to explore sole source contracts, mentor-protégé teaming arrangements and JVs. We will also appoint channel partners and direct sales force to sell the products and services across US and Europe to begin with.

Expertise/Qualifications of Team/ Company Readiness:

Quantum Ventura specializes in AI/ML and it is one of their major strengths. Having obtained significant expertise in neural-network based AI applications, we are confident of delivering a world-class product/ service using DeepVerify. Our technical and management team will be led our CEO Srini Vasan who has in-depth expertise in AI/ML and business management. Having been a successful entrepreneur, he will drive the technical and management aspects of DeepVerify. He will be assisted by Aaron Goldberg to take charge of business development and marketing. Howard Messer, CFO will offer strategic management guidance and financial management. We will hire marketing and technical development team to support the development effort. Quantum has the support of Prof Tim Menzies of NC State University as our world class expert in verification and software engineering.

Anticipated Commercialization Results:

	Additional Investment	Sales Revenue	Development Expenses
Phase 2 Award beginning	$1,000,000	0	$900,000
Phase 2 + plus 1 year	$1,500,000	200,000	$1,500,000
Phase 2 + plus 2 years.	$2,000,000	500,000	$1,800,000
Product Launch – year 1	$1,500,000	$2,000,000	$2,000,000
Year 2	$1,500,000	$10,000,000	$4,500,000
Year 3	$2,000,000	$35,000,000	$20,000,000

Letters of Support – Attached.