Summary Report for the

SRC/NSF Workshop on Verification, Validation, and Test of Machine Learning (V-TML) Systems

Workshop Dates:	July 24 – 25, 2018
Workshop Location:	Holiday Inn Alexandria at Carlyle, 2460 Eisenhower Avenue, Alexandria VA
Sponsors:	Semiconductor Research Corporation National Science Foundation

Workshop Organizing Committee

- Sankar Basu, NSF
- Noel Menezes, Intel
- Brian Mulvaney, NXP
- Gi-Joon Nam, IBM
- Valeriy Sukharev, Mentor, a Siemens Business
- Tuna Tarim, Texas Instruments
- Li-C Wang, Univ. of California, Santa Barbara
- Yosinori Watanabe, Cadence Design Systems
- David Yeh, SRC and Texas Instruments

Executive Summary

The application of machine learning algorithms and techniques have dramatically increased over the past several years, since it has seemingly limitless potential to change the way systems are optimized and interact with the world around them. Electronic systems with built-in machine learning are just around the corner so the question for the electronic design automation (EDA) community is how to handle the verification, validation, and test of such systems. This workshop will address the current state-of-the-art and formulate the key research needs in charting a path forward towards enabling the safe, reliable, and predictable application of machine learning (ML) to emerging applications.

This document is based on the presentations and discussion at the Workshop on Verification, Validation, and Test of Machine Learning (V-TML) Systems that was held on July 24 – 25, 2018 at Holiday Inn Alexandria at Carlyle, 2460 Eisenhower Avenue, Alexandria VA. Presentations from the workshop may be viewed on the SRC website, https://www.src.org/calendar/e006556/ .

Workshop participants included representatives from industry, academia, and government agencies.

- *Industry:* Cadence, IBM, Intel, Mentor, NXP, Raytheon, Synopsys, and TI
- *Universities:* ASU, KIT, Georgia Tech, MIT, Penn St., Portland St., SMU, Texas A&M, UF, UIUC, UCSB, UT/Dallas, and Wisconsin
- *Government:* NSF and DARPA

The workshop was organized in separate sessions each with presentations covering current state of the art as well as outlining research directions. Following the presentations, discussions with workshop participants were captured by session scribes. The topics of each session are listed below.

- Session 1. Current Research Activities
- Session 2. Data

- Session 3. Explain-ability
- Session 4. Robustness
- Session 5. EDA - Verification
- Session 6. EDA - Validation and Test
- Session 7. Research Needs Discussion

Some identified research directions for exploration:

- Mathematical Basis: formal models for machine learning behavior
- Bounds/Guarantees: how to determine for application specific contexts
- Confidence: how to establish and use it for decisions, including real time
- Control/Decisions: system level stability, controllability, observability
- Trust/Security: interference, noise, malicious use
- Explain-ability/Understandability: models and system-level behavior
- Debug-ability: determining problem origin and remedies
- Verification and testability metrics: guide robust design
- Reliable application-specific data preprocessing: automatic data collection, sanitation, and preparation for machine-learning engines

The research directions are not completely orthogonal to each other and cross areas from theory to practice: mathematics, computer science, and electrical engineering all play a role in advancing the state-of-the art. As with many emerging cross-disciplinary technologies, there is need for research in each of the areas individually as well as a group, since the application of theory into practice is where true impact lies for this type of technology.

The workshop introduced ideas that were ripe for immediate investigation. The following two projects were funded as part of the SRC Global Research Collaboration (GRC) Computer-Aided Design and Test (CADT) program and began on Jan. 1, 2019 that address part of the central theme of the workshop – verifying, validating, and testing ML techniques:

1. Development and Assessment of Machine Learning Based Analog and Mixed-Signal Verification
2. Functional Fault Modeling and Testing of Machine-Learning Hardware

Furthermore, concepts identified as a part by this workshop that are in need of better understanding were included in the 2019 CADT Research Needs document, released on June 10, 2019 with funding anticipated to start on Jan. 1, 2020. These highlight the importance of the topic and how SRC members are already beginning to address the knowledge gap in this area. Even with this start, much more research effort is needed to significantly advance the technology.

Session 1. Current Research Activities

Machine learning techniques have been widely applied in EDA and Test for over a decade. In general there can be two types of applications, one for prediction and the other for interpretation. A prediction model is usually used for improving efficiency or reducing cost. An interpretable model is usually for discovering actionable knowledge. A prediction model can be built if there is sufficient data to enable cross-validation. An interpretable model can be built if the meaningfulness of a model can be learned from an expert. Regardless of which type of model is built, a machine learning model does not guarantee that the outcome is correct or its performance is feasible. Hence, in an application another component is required to safeguard its usage. Today, this safeguard is lacking and without better understanding of this area, the widespread adoption of machine learning techniques will be limited to applications that do not require high levels of reliability and robustness.

The workshop organizing committee developed this set of questions in advance of the workshop to frame the discussion which centers on the applicability of machine learning:

- Why can't ML provide a guarantee on the correctness of its outcome?
- How would one define a robustness metric for a ML model to quantitatively measure this?
- What does it take to build an ML model that ensures a guarantee in terms of the robustness metric?
- Are there differences in ML models and algorithms in terms of difficulty in providing such a guarantee?
- In a practical system with no such guarantee, what is the methodology for users of the system to adopt, in order to ensure they would not accept incorrect outcome from the system?
- How should one safeguard a ML model?

The first presenter in this session was Prof. Li-C. Wang, University of California at Santa Barbara. The title of his presentation was "A Path Toward AI – The System View for Applying Machine Learning." This talk described how machine learning techniques have been applied in design and test over the past several years and the cases which are driven by domain knowledge require a domain expert to validate that the system has acquired the necessary knowledge before autonomous operation may be trusted. There are theoretical and practical challenges in adopting a traditional approach to the machine learning problem formulation in EDA and Prof. Wang described how functional verification and yield optimization issues might be better addressed with an AI system. Included in this presentation was a demonstration of a natural language driven AI system for production yield optimization.

The second presenter in this session was Prof. Elyse Rosenbaum who titled her presentation "Machine Learning of Models Used in EDA." In it, she described ongoing

activities at the Center for Advanced Electronics through Machine Learning (CAEML), which she directs. This effort has activities in the areas on the theory and efficiency of machine learning, the application to optimization, modeling, verification, and security. One of her key points is that the models used in machine learning have limits that are a result of available information, experts' knowledge, and computation resources. Risk of erroneous operation might be best mitigated by judicious checking of models through means such as stability, causality, and passivity analysis, consistency checking with the laws of physics or cross-checking models if they are derived independently.

Key ideas of this session: The EDA community has been applying ML for a decade. Key uses include the verification and testing of IC designs, as well as the development of reduced order models of electrical component or sub-systems. These are areas where data may be labeled and expert practitioners oversee the process. Even so, there are acknowledged limitations in its application and usage. Periodic or even continuous checking of models being used in machine-learning applications is a key approach to mitigate risk of errors, even though this approach may be computationally very expensive. The community has built up expertise and is looking toward overcoming current limitations so that a better understanding would lead to broader adoption of this emerging technology in electronic systems.

The opening session set the stage for many more questions than answers, such as;

- What does it really mean to verify or test a machine-learning system?
- How would a user know that verification or testing approaches are satisfactory?
- Are concepts such as physics-based consistency of ML models useful? What about stability, causality, passivity, etc.?
- Should risk assessment be part of the ML modeling infrastructure?

These questions point to a large gap between the current state-of-the-art and what might be possible.

Session 2. Data

The complexity of hardware systems increases across all facets of the design cycle for every new product generation. The growing complexity and the amount of data being generated, and the complex relation between the data sources calls for data science techniques such as statistics, data visualization, data mining and machine learning to extract the essence of the input data and enhance the capability and/or efficiency of design optimization in various phases of design cycles. Moreover, it is conceivable that system optimization will continue after deployment, such as, adapting to aging or environmental conditions with additional use of machine learning techniques. In this session, we seek techniques and research opportunities to capture, manage and analyze data to make it available to the people who need it by discussing the following questions:

- What are ideal data types/formats for machine learning and how do they vary from applications to applications?
    - Structured data vs. unstructured data
    - Deterministic vs. stochastic data
    - Numerical precision (approximate computing model)
- If necessary, what types of data pre-processing are required? How can we generate labeled data for supervised learning in a more automatic way?
- What are the pros and cons of a stochastic ML system, as opposed to a deterministic ML system? Also, do we need novel data ingestion models for stochastic data from sensory inputs at the edge of computing, for example?

All presenters were asked to provide current practices with concrete examples and discuss how they can be improved further for more reliable, verifiable and explainable machine learning systems.

The first speaker in this session was Tom Guzowski, IBM. The title of his presentation was “Developing a Comprehensive Chip Information Dossier for Design Learning.” This presentation highlighted efforts to leverage historical ASIC and microprocessor design activity to build a comprehensive model of all design activity for open ended analysis. Using a black box model of the design flow some of the questions IBM hoped to answer include if the design project is on track to meet the schedule and what is the optimal process for a general design. Using a set of data acquisition tools, information is kept in repositories for further machine-learning analysis to provide guidance and tracking on progress, comparisons between different designs, as well as projecting needs in the next generation of a design.

The second speaker in this session was Himanshu Kaul, Intel. The title of his presentation was “Reconfigurable Machine Learning Accelerator Circuits.” Here, machine-learning accelerator circuits were presented that had low reconfiguration overhead for supporting different operations, number formats, precisions, and sparsity in the computing needs of the underlying algorithms. On a K-nearest neighbor problem, a 2.7X energy savings for 1.23nJ/query with a 390mV supply voltage. In addition, on a matrix multiply accelerator example, a wide supply voltage range could lead to a 4X higher energy efficiency of 11.3TOPS/W with a 280mV supply voltage.

Key ideas of this section: Two directions are considered: given data of a design process, what can a machine-learning approach provide that wasn’t previously known; and given a machine-learning application, what are ways in which the data and implementation be optimally structured. Subtopics of this are many but also include how the arrival of data over time impacts these two directions, the best ways to handle sensitive data such that confidential information is protected, the structuring of data, the stochastic nature of some data, and pre-processing needs of data.

Session 3. Explain-ability

The effectiveness of current implementations of Machine Learning systems is limited by the inability of the machine to explain its decisions and actions to a human being. For example, a neural network based image recognition system might be trained to recognize certain objects. When presented with a picture of a cat, the system would identify the cat with 90% probability. An explainable version of this machine would be able to identify the cat, and tell you how it arrived at that conclusion (e.g. the object has fur, claws, whiskers, etc.). The machine would tell the human user why it came to a decision, why it did not go down some other path, and why did it declare success. An explainable ML system would enable human users to understand, gain trust, and effectively manage the system.

- How can we provide better explanations on insights learned from ML systems? (rationale for the conclusions; how can we address the opacity of ML systems)
- Why the machine chose the solution it did?
- If the human determines the solution is an error how do you get the machine not to repeat the action?
- How do you automatically cross check the validity of the solution?
- Information-theoretic limits of explain-ability?

The first speaker in this session was David Gunning, DARPA. The title of his presentation was “DARPA’s Explainable Artificial Intelligence (XAI) Program.” The presentation highlighted the ongoing DARPA program and the need for explainable AI. The long-term effectiveness of deploying AI systems will be limited by the machine’s inability to explain its decisions and actions to users. This is an essential part of users trusting a new generation of artificially intelligent partners. Some measureable concepts were identified including machine learning performance, explanation goodness and satisfaction, user-machine task performance, and appropriate trust and reliance.

The second speaker in this session was Manish Pandey, Synopsys. The title of his presentation was “Explainable Design Automation with Machine Learning.” The presentation highlighted the use of machine-learning across the design automation flow, including within individual tools as well as across the flow. Use cases described in detail include formal verification and static signoff but other uses such as automatic test selection, failure triaging, and assessing code risk. The presentation also described how the use thus far has been opportunistic and there is dependency on the available data. Finally, Manish emphasized that explainability was in important consideration for both users as well as developers.

Key ideas in this section: In the design of electronic systems, it is important to determine what information should be provided at each level of the design hierarchy to enable the usage of machine-learning techniques with confidence. This also provides a path for cross-check results at multiple levels and ensure system specifications are met. The concept of reproducibility might be in need of investigation as the stochastic nature of complex decision making could be better understood and affects AI systems. Finally, as

decisions become more complex, the introduction of social sciences might help provide an enhance framework for understanding and explaining the decision making process.

Session 4. Robustness

The robustness of a ML solution can be a major concern when the solution is integrated into a system for performing a certain task where failure of the task has an unacceptable consequence. For example, a ML solution is used to recognize the face of a user to unlock a device (recognize something to trigger a task). Failure can cause a security risk. As another example, a ML solution is deployed into production for quality assurance. Failure can cause unexpected quality degradation. Generally speaking, robustness means the ability to exhibit correct behavior under adverse conditions or rigorous/thorough testing. In plain English, robustness here means that the system would make the same decisions as a person would, applying "common sense." However, defining a robustness metric to evaluate a ML system can be an open-ended question of which the answer is application dependent. For a metric defined at the system level, it might also be required to derive a metric for evaluating the ML solution itself. Take deep neural networks (DNNs) as an example for the ML solution. One current direction to improve the robustness of a DNN model is by making it more immune to so-called adversarial examples. However, it remains unclear what robustness actually means for a DNN model (when such improvement is enough) and how to quantify the system impact when such a DNN-based solution is integrated into a system.

- Is there a mathematical relation between robustness and explain-ability?
    - If there is one, then a mathematical relation might enable a algorithmic approach
    - While it is expected that Robustness ≠ explain-ability, does it follow that a system that is more explainable is more robust?
- How does the depth/quality of data impact robustness and explain-ability?
    - From above, how would more data and higher quality improve robustness?
- Robustness should also comprehend time-dependent effects such as silicon aging. How does ML chip degradation affect its operation and how should this be addressed?

The first speaker in this session was Prof. Somesh Jha, University of Wisconsin-Madison. The title of his presentation was "Towards Semantic Adversarial Examples" in which he showed many examples of how ML algorithms and systems were induced to provide incorrect output. He then described attacks on the ML pipeline with training set poisoning, parameter tampering, and adversarial examples. Included in the presentation were formal definitions of attacks and robustness and concepts of adversarial ML, and the addition of

formal methods to this as an important area for researchers. He points out that adversarial examples are counterexamples to a formal specification and that verifying the system containing a deep neural network may lead to the incorporation of a region of uncertainty in the verification flow. This concept of a region of uncertainty would allow the ML component of the system to be graded such that boundary cases are more easily uncovered and subsequently addressed. He shows this in the case of an image-based automatic braking system. Finally, he mentioned that benchmarks and code for some of the research were available through two website: https://www.robust-ml.org and https://github.com/tensorflow/cleverhans .

The second speaker in this session was Lily Weng and Prof. Luca Daniel, MIT. The title of their presentation was "Certifying Robustness of Neural Networks." In it, they describe three research areas to address robustness to adversarial examples. The first of these is **Attack**, which is how to generate a specific adversarial example of a given network if given some sort of input data. The second is **Defense**, which is how to make a given neural network robust to known attacks and adversarial examples on given input data. The third is **Robustness Quantification**, which is how to quantify the level of robustness of a given neural network to adversarial attacks as well as non-malicious acquisition noise, or even the case where the basic problem statement is ill posed. Open challenges and needs in the Attack and Defense space include that the attack setting be realistic, that the input perturbation be small, as perceived by humans, and that any defense should have some guarantee (or certification) as well as being computationally efficient. On the topic of a Robustness Quantification, most are posed as optimization problems or use formal verification methods. The research approaches should be computationally efficient, be scalable to state-of-the-art problem sizes, and should provide a metric for determining a guarantee. Early work might be a Lipschitz continuity approach. Finally, there is a mention that non-malicious miss-classifications may lead to safety or ethical issues, as well as social injustice.

Key ideas in this section. Is there a way to quantify the operational space for a given system such that design centering ideas may be used to address robustness issues? The adversarial examples show that small differences in the input space give way to large differences in the output space (behavior) and this shows that the system is brittle. This shows that work on metrics for robustness are of critical importance.

Session 5. EDA – Verification

From project kick-off to tape-out, and extending in to the post-tape-out phase, 60% of today's verification effort is spent using the existing computer aided design tools to execute on the design verification plan. The majority of the design verification plan for analog designs are focused on simulating the possible variations in design (corner, PVT, statistical, etc.), while digital verification focuses on identifying the right combination of constraints to target break-the-part tests and corner cases by stressing the design. Due

to the inherent nature of the stimulus and the sheer number of combinations, simulations take a considerable amount of time and can be very taxing on compute resources ultimately impacting project schedules. In addition, simulations may need to be tweaked depending on the type of circuits simulated on case-by-case basis resulting in generation of directed test cases that could easily become a management nightmare. Consequently discussions on efficiency and reusability become very critical to managing the overall verification cycle.

Basic improvements in verification by using machine learning techniques can help with reducing simulation time and therefore reducing project schedules, as well as ensuring reusability, i.e. can we use the same machine learning algorithm on different design types. On the other hand advanced improvements in machine learning algorithms or the creation of new ones to help with computational enhancements will make design verification faster and will also improve the quality of outcome by reducing the probability of error.

- How will designers have confidence in the ML techniques, especially for mixed-signal applications?
- Can the same ML algorithm be used in multiple applications or will they be tailored for each?
- How would someone build a reusable machine learning model?
- Understanding training-data and test-data for machine learning models in verification: can they be the same data set?
- Success metrics for machine learning models: What is the expected outcome of the machine learning model and how do we measure success?
- What are the legal ramifications of supervised & unsupervised learning applied to the model micro-systems in the event of system failure?

The first speaker in this session was Prof. Peng Li, Texas A&M University. The title of his presentation was “Enabling Verification of Analog and Mixed-Signal Circuits Using Machine Learning.” His presentation gave the background and challenge of checking AMS circuits against a given set of design properties and showed that it is possible to develop a statistical design property checking approach to address the AMS design verification challenges. Verification of AMS designs is difficult due to complex operations, which are non-linear and time varying. Using a hybrid formal machine-learning verification approach, he shows it is possible to find errors that were previously missed with existing techniques such as Monte Carlo and Scaled-Sigma Sampling. This shows that it is possible to adopt machine learning for AMS verification. Research avenues for further investigation include data usage, accuracy, and coverage, where methods for identifying dominant parametric dependencies of design properties and failure

mechanisms would be beneficial. Also of benefit would be the development of domain-specific adversarial attack algorithms.

The second speaker in this session was Pradiptya Ghosh, Mentor, a Siemens Business. The title of his presentation was “Potential Application of Computational Intelligence for faster Tape-Out to Yield Ramp-up.” In this talk, he discussed how ML is currently used in the IC design and manufacturing process such as in optical proximity correction and identifying “hotspots” in the layout that are susceptible to electromigration effects. He then offered a vision of a future fabrication environment, such as the European Union project MADEIN4, where the IC factory is viewed as a cyber-physical system and smart automation is enabled by data collection, metrology, and machine learning. This would enable increase the quality of metrology measurements, reduce wafer rework, enable predictive tool maintenance, and provide faster root cause analysis. Advances in the verification of ML systems such as this would be needed to realize the vision of a smart fab.

Key ideas from this session: current verification techniques need enhancements to be applied to future ML applications. Early work on mixed-signal verification show promise but current methods may not be scalable to address the size of coming uses. Storing, filtering, and sampling the data to generate the best models is an area of research. For the application of ML to cognitive assistants and factory automation, root cause analysis of ML failure would be an important area to understand.

Session 6. EDA – Validation and Test

The goal of the human effort during system functionality validation is to understand the correct operation of the system, and identify the root cause of any deviation from correct operation. Same approach should be undertaken for validation and test of autonomous systems with the built-in machine learning (ML) capabilities. ML is a branch of artificial intelligence, which is learning from examples and identifying the structure in a system of random realizations. The large volume of data, which are continuously generated by many sensing and monitoring systems, suggests that the employment of ML in autonomous systems is a promising solution, especially for anomalous behavior detection. The major question is how we can test and validate the functionality of the systems with built-in ML capabilities. Should a large volume of data regarding behavior of such system be collected and analyzed against behavior of a similar but human operated system? For example, post-silicon variation extraction and bug localization through inference, which is performed with built-in ML, can be compared with results of the standard test. A methodology allowing to analyze two sets of data obtained with and without built-in ML and to detect flaws of the employed ML algorithm could be a subject of interests (This methodology might also be used to test machine learning systems both at times 0 and also after deployment). But, first the key questions should be addressed:

- How should one feed data to a data mining (ML) tool?

- Which data mining tool or technique should be used?
- How the results (patterns) can be utilized?
- How many ways are there to detect flaws in a machine learning algorithm?
- Is it possible that an error is known, but that we can't see any difference between the two data sets to explain it?
- How can you trust the insights from ML systems? (Is it even possible to verify / validate the system?)

The first speaker in this session was Prof. Yiorgos Makris, University of Texas at Dallas. The title of his presentation was "Validation and Testing of Analog Machine Learning Systems." In this talk he focuses on analog ML systems and how it has different characteristics that digital ML systems. One observation is that analog ML systems have continuous parameters and requires complex validation such as min/max, ranges, and multiple process corner simulations, which lead to calibration/trimming of each individual chip. This leads to the notion that every analog ML system should be trained individually; that online training with the same inputs will lead to each chip having a slightly different model; and operational noise may make the model evolution non-repeatable for the same IC. Research directions include methods to validate learning capacity of hardware implementations of (analog) ML systems; methods for fast transfer/adaption/consistency checking of weights across chips; methods for extending pre- and post-deployment self-testing and self-tuning procedures to account for learning capacity and acceptable model boundaries; and methods for monitoring and vetting robustness of real-time model evolution.

The second speaker in this session was Prof. Mehdi Tahoori, Karlsruhe Institute of Technology. The title of his presentation was "Machine Learning Techniques for Reliability Monitoring, Mitigation and Adaptation." In this talk he makes the case for incorporating ML techniques into circuit monitoring and testing infrastructure to address whatever adaptation is needed to make the system resilient. He describes his System-Physician-on-Chip (SPOC) that could be the basis for data-driven learning-based adaptive resilient systems, handling process, runtime, environment and usage variability. Areas of research include exploring new definitions of faults, errors, and failures.

Key ideas from this session: Concepts surrounding system-level testing of machine-learning capability should be explored, including acceptable model boundaries and model evolution paths. To guide ML algorithms, more complete understanding of metrics that are useful would be beneficial. The cost of test of ML systems should be better understood as that might be a barrier to widespread adoptions. Furthermore, the area, power, and usage modes of ML infrastructure should be quantified.

Session 7. Research Needs Discussion

During the discussion conducted at the end of each session, many questions were raised and debated. Participants in the workshop came with different research backgrounds and provided diverse perspectives to the same questions. These perspectives may be cast into three levels in view of a machine learning based system (ML-based system): the level concerning the machine learning models, the level concerning the hardware/software system executing and utilizing the machine learning models, and the level concerning the application targeted by the system. In the following, research needs are grouped in these three levels. It is important to note that to achieve an overall requirement, sometime it may require a particular need to be addressed in multiple levels. Also, addressing a need might require considerations in other levels.

Furthermore, the terms “verification,” “validation,” and “test” (VVT) can mean different things in different communities and also in different companies. Hence, it is important to clarify their meanings in the description of the research needs. The general view taken in this document is that: verification concerns the correctness of a conceptual model used to build the system; validation concerns the correctness of a physical realization of the system; test concerns the correctness of the actual manufactured products.

Machine Learning Models:

The assumption is that a machine learning model, such as a deep neural network model, is trained separately and is provided to a system builder as a building block, or piece of intellectual property (IP). In order to verify a ML-based system, the process could be improved with an enhanced or better understanding of the models in use:

- Research is needed to provide a theoretical foundation for quantifying machine learning model behavior. The word “quantifying” is used here because usually, verification quality is quantified through a coverage metric. In general, methods are needed to define how to measure the quality of a ML model and provide an efficient way to measure it. This measure can depend on the requirements to support system-level verification.
- Research is needed to improve explain-ability and understandability of a ML model. Explain-ability and understandability can facilitate debug and diagnosis at the system level. They can also provide guidance to VVT processes. However, limitations on explain-ability are not yet totally clear in theory and in practice.
  - What is the definition of an explainable model?
  - Is there an information-theoretic limit for model explain-ability?
  - Is there any aspect of a model that is unexplainable?
  - How to specify an explain-ability requirement for a system?
- Research is needed to define a measurable robustness metric for a ML model. “Robustness” here means that the model is able to perform correctly under adverse conditions or rigorous testing. Note that robust ML is already recognized as an important research area in the machine learning community, see e.g. https://www.robust-ml.org. The existence of adversarial samples for a deep neural

network model is a well-known fact and many research attempts are under pursued to overcome the issue. Methods are needed to address the important questions regarding the robustness of ML models:
    - What does it take to build an ML model with respect to a given robustness metric?
    - Are certain class of algorithms inherently more robust than others?
    - What is the approach to establish a relation between robustness and explain-ability?
    - How to obtain adversarial examples that actually lead to system-level failures?
    - Can the robustness issue be mitigated through neural network architecture or ML-based system design in order to reduce the need for model and hardware robustness?
    - How is reproducibility related to robustness?
- Research is needed to evaluate the quality of data and its relation to the quality of a ML model. Data are essential for constructing a ML model. In the context of VVT, data are inputs that drive the VVT processes. In this regard, data can be thought of as the "tests" in traditional VVT. In general, methods are lacking that provide reliable data assessment to answer important questions such as:
    - How do we decide that the data at-hand is enough for learning a high-quality model (based on a quality metric)?
    - Is it possible to assess data quality with respect to a learning algorithm without actually running the learning algorithm?
    - Is the data requirement different for learning than for robust learning or explainable learning?
    - How does one assess the data requirement in view of a particular application?
    - How would one incorporate domain knowledge in understanding/processing data?
    - What is the best way to effectively deal with heterogeneous data formats?
    - How do we utilize domain knowledge to compensate the missing information from the data in an application?
    - How does a professor overcome the lack-of-data challenge in academic research?
    - How can one establish a set of benchmarks sufficient to address a research problem?

ML-Based System:

A ML-based system comprises the hardware fabric that executes a ML model and the hardware/software that provides the system functionalities. The needs focus on the new VVT problems introduced by including a ML component in the system. The statistical nature in the operation of a ML component brings a new dimension to the already complex VVT processes. While the high-level objectives in system-level VVT remain very similar to those in today's practice, the specific requirements to achieve the objectives can be new for a ML-based system.

- Research is needed to understand, clarify, and possibly to re-define a variety of system aspects: stability, controllability, observability, reliability, robustness, explain-ability, debug-ability, safety, and trust/security.
    - In a practical system where no guarantee can be provided at the ML model level for a particular aspect, what is the methodology for a system designer to mitigate the issue in order to provide a guarantee at the system level on that aspect?
    - Is there a safe way to safeguard a ML model?
    - How could one assess the requirement for system in view of an application?
- Research is needed to understand, clarify, and possible to re-define the meaning of verification, validation, and test in view of a ML-based system.
    - What are the requirements of VVT, and how should they be specified? How could one assess the impact of a bug or defect?
    - How would one define the notion of coverage in VVT?
    - Is off-line VVT enough? If not, does online VVT require ML as well? Do we need AI or ML to perform VVT of a ML-based system?
    - If the ML-based system includes reinforcement learning, does it change the VVT landscape?
    - Can we use ML in a white-box manner to synthesize a more resilient controller?
    - How could one generate input samples that expose errors in system operation?
    - Does ML acceleration hardware pose a new VVT problem?
- There are other research ideas where the discussion is more centered on the current trend of hardware acceleration for ML:
    - Methods to validate that the learning capacity of an analog hardware implementation is sufficient for accurately executing a given ML model.
    - Methods for fast transfer/adaptation/consistency checking of weight across an accelerator hardware component/IP/chip.
    - Methods for extending pre- and post-deployment self-testing and self-turning procedures to account for acceptable model performance boundaries
    - Methods for monitoring and vetting robustness of real-time model evolution
- Overall, hardware acceleration of ML is gaining traction in the community. SRC held a workshop on this topic on April 15-16, 2019, titled “The Future of Artificial Intelligence Hardware Systems” where aspects of this were discussed.

ML-Based Applications:

The context here can be thought of as that an original equipment manufacturer (OEM) implements its own software for a particular application based on a ML-based system such as an intelligent controller and the software API to support its programming. In this regard, the VVT can be viewed from the perspective of the ML-based system provider, i.e. what to provide, or from the perspective of the OEM, i.e. what to receive.

- Research is needed to define a specification of the requirements on the system providers where it enables an application to be implemented with an assurance on functional safety and/or security.

    - Methods to achieve application-specific bounds and/or guarantees
    - Methods to evaluate confidence on decisions made in an application real time
    - Methods for reliable application-specific data preprocessing: automatic data collection, sanitation, and preparation for machine-learning engines
- Research is needed to develop methods for online verification, validation, and/or testing in view of an application.
    - How VVT can help improve the system over time
    - Methods to assess the level of requirement by an application for online VVT?
    - Is it possible to verify an AI system without a verifiable ML model?
    - How can one provide better guidance to debug an application software?
    - How much transparency can a system provide to an application developer?
    - In an application, how would one propagate a human's input to the underlying ML model?
    - What system features are required to support the implementation of a cross-checking functionality in an application?

In summary, the workshop discussion intended to reach a holistic view on the VVT needs for building a ML-based system. While robustness, explain-ability, and security of a ML model are emerging topics being addressed by researchers in the ML community, the VVT needs discussed in this workshop addresses much broader concerns from a ML system builder and an end-product developer perspective. Research on VVT of a ML accelerator is at its start and yet, it only addresses a small part of the overall system VVT requirements. Overall, research needs discussed in this workshop are unique, fundamental, and critical for implementing a ML-based system and also for deploying it as a safe and reliable end product.