Booz | Allen | Hamilton®

BOOZ ALLEN HAMILTON

Response to NIST RFI on Artificial Intelligence Standards

Prepared exclusively for NIST

CONSULTING | ANALYTICS | DIGITAL SOLUTIONS | ENGINEERING | CYBER

TABLE OF CONTENTS

1 INTRODUCTION

Artificial Intelligence (AI) is on the cusp of becoming the most significant technology of the information age. The exponential pace of technological advancement has made it more challenging than ever before to address its unintended consequences. Now is the time to anticipate the broad implications of the next frontier of innovations in artificial intelligence.[1] Many government agencies have explored implementing AI policies, but an overarching set of standards is needed. A fundamental and common AI standard would further reduce domain related risks in organizations as programs move from the development phase to the operational phase and provide a common basis for evaluating performance. Addressing these issues will ultimately lead to driving further adoption of AI.

AI STANDARDS SCOPE

Although AI and Machine Learning (ML) loom large in the public mind as fundamentally new technologies, the design, training, implementation and security considerations around AI continue to fit within current software development methodologies. An emerging concept in this area is that of DevSecOps[2] (Development/Security/Operations). DevSecOps represents the integration of recognized development, security, and operations practices into a single iterative software development pipeline. This pipeline provides a broad framework that can incorporate the new concepts and standards required for the safe deployment of AI in production systems. These practices should be considered across two primary dimensions: their place within this software development pipeline, and their relationship to specific users and use cases.

2 DEVELOPMENT

The standard for the development for ML models should move to a tangible engineering approach rather than a research-based approach. Traditionally, ML model development incorporates a broad range of coupled processes including initial data exploration and labeling; data extraction, transformation and loading (ETL) processes; model design; model training and tuning; and testing and integration into a broader software base. As a result of data dependencies between each step, this cycle has often been approached as an exploratory process closer to scientific research rather than as a software engineering process. The scientific research approach stimulates fast iteration but provides

[1] Knowledge@Wharton. (2019). The AI Boom: Why Trust Will Play a Critical Role. Retrieved from https://knowledge.wharton.upenn.edu/article/coming-ai-breakout-need-rules-road-now/

[2] Carter, K (2017). *Francois Raynaud on DevSecOps*. [online] ieeexplore.ieee.org. Available at: https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=8048652

limited guarantees on the safety and reliability of the final system. This same approach, and the interconnectivity of the individual steps, has also limited the emergence of a consensus concerning the software development process for AI-based systems. New formats and standards are necessary to constrain the representations that developers use in these systems and to provide a framework for organizations and regulators to assess the trustworthiness, reliability and risk associated with AI systems.

CONSIDERATIONS FOR COMMUNITY WIDE STANDARDS

As a broader ecosystem around ML capabilities evolves, system level standards for both data and model representation must be defined. Due to the complexity of the space, these standards should focus on versioning, model testing, and system interfaces while limiting constraints on the overall ML pipeline and internal model structure to the greatest extend possible. Models should have well defined requirements for testing, metadata, and system interfaces that do not require information from the original developer. These standards should be comprised of APIs for testing of model components, introspection capabilities, and the necessary metadata for model use. All deployed models should have this information directly incorporated into their binary representations with standardized methods for reading the data.

Additional considerations that may impact standards development revolve around the architecture and training of ML systems. Both introspection capabilities and modularity of systems can vary dramatically depending on the type of model being used. Changes in the level of introspection and system modularity will impact the approach to testing and verification. Privacy and security considerations may also influence the model structure and impact the available output. A model using Private Aggregation of Teacher Ensembles[3] (PATE) or other privacy inducing methods may require alternate approaches for data versioning due to privacy concerns. The same is true in the case of federated learning. In this context, information concerning the distributed data may be spread amongst multiple owners with different barriers to communication[4]. Any standards specification should consider the impact of these approaches and provide alternative standards where necessary.

DEVELOPMENT STANDARDS, GOVERNANCE & RISK CONTROL

The primary goal of any standards development must be to improve the ability to safely operate, evaluate, and test the system of interest. To circumvent the 'black box problem', AI systems must be built to be auditable, reviewable as to how the system was built, including areas such as model type, training data and any bias that exists within the training data and performance accuracy. Systems should also be explainable to any end user, regardless of technical acumen, and should be able to convey how the ML system arrived at a certain decision. The explainability challenge can be supplemented with documentation tailored to the end user's needs.

When it comes to ML, guaranteeing system safety requires the ability to identify when an error occurs, diagnose its cause, and subsequently determine appropriate measures to return the system to a trusted state. These evaluation and testing steps require knowledge of how the system was built, introspective capabilities for evaluation of different system components, and the means for altering individual components that limit the rebuilding or retraining necessary to address functional gaps. Any process

[3] Papernot, N., et al. (2018). *Scalable Private Learning with PATE.* [online] arxiv.org. Available at: https://arxiv.org/abs/1802.08908

[4] Geyer, R., Klein, T., and Nabi, M. (2017). *Differentially Private Federated Learning: A Client Level Perspective.* [online] arxiv.org. Available at: https://arxiv.org/abs/1712.07557

that meets these requirements must include version control across all components of the system and a standard representation for each component.

Software development processes have been well established for tracking changes to source code over time. However, unlike traditional software, ML systems have extensive dependences[5] on:

- Model parameters
- Data used for training and testing
- Data labels, descriptions and assumptions
- Intermediate model states during the training procedures
- Metrics and cost functions used to evaluate the model
- Random seeds used to define the model's initial conditions

These components are often non-deterministic. Their output can vary dramatically with relatively small changes in both the data that was used to generate them, and the hyperparameters that were used when training the system.

MODEL VERSIONING

As ML and data science have become more prevalent, several groups have started to develop versioning approaches that capture the breadth of concerns related to their development and application. In this context there are two main areas for expansion of scope for version control. The first area focuses on more traditional approaches to the versioning of ML models and their associated data. There are several different organizations providing offerings in this space such as Pachyderm,[6] Comet.ml,[7] ModelHub,[8] and DVC.[9] These approaches combine code, models, and data into a single versioned instance. The second area focuses on versioning and reproducibility of exploratory data analysis. In addition to considering the intersection between code and data, these packages focus more on the exploratory, data-driven and iterative nature of the machine learning process. Some tools in this space are Yellowbrick[10] and Gigantum.[11] For example, Yellowbrick is a package which attempts to standardize approaches to ML visualization while Gigantum is a platform for versioning hypothesis-driven experimental work. While these tools are less focused on software development, they allow consistency and reproducibility in the analysis of complex datasets.

In addition to versioning, the ML field is starting to converge on general paradigms for model representation such as the estimator-transformer approach that is used by both Scikit-Learn[12] and Spark

---

5 Sculley, D., et al. (2015). *Hidden Technical Debt in Machine Learning Systems.* [online] nips.cc. Available at: http://papers.nips.cc/paper/5656-hidden-technical-debt-in-machine-learning-systems.pdf

6 Pachyderm home page, Available at: https://www.pachyderm.io/

7 Comet.ml home page, Available at: https://www.comet.ml/

8 Miao, H., Li, A., Davis, L., and Deshpande, A. (2017). *ModelHub: Deep Learning Lifecycle Management*. [online] ieee.org. Available at: https://par.nsf.gov/servlets/purl/10041785

9 DVC home page, Available at: https://dvc.org/

10 Yellowbrick package description, Available at: https://www.scikit-yb.org/en/latest/

11 Gigantum home page, Available at: https://gigantum.com/

12 Scikit-Learn pipeline description, available at: https://scikit-learn.org/stable/data_transforms.html

MLlib.[13] In the neural network setting, the Open Neural Network Exchange Format[14] (ONNX) standard has been developed for model representation. These packages represent frameworks for measuring and communication consistency in the ML field, but their focus is predominantly on the individual developer as opposed to communication between developers across a development team.

SYSTEM TESTING

ML systems only work when there is some target or loss function explicitly designated for the learning algorithm. Common choices include cross entropy for classification problems and the mean squared error for regression problems. Throughout the training process, this objective function is explicitly optimized. Most practitioners are accustomed to tracking their models' performance using these and a handful of other common metrics of interest, such as accuracy. However, a model's performance can be assessed on many other measures besides these explicit objective functions. Merely tracking scalar values, such as accuracy, gives an incomplete picture of a models' characteristics when compared to other, high-dimensional measures such as a confusion matrix or ROC curve. In many circumstances, other derived measurements such as false positive rate or specificity may be more relevant to deployment than accuracy. It is important that the information necessary for calculating these other measures be preserved and easily accessible and that their results can be conveyed using standard representations.

All of ML is predicated on the "inductive hypothesis" — the belief that data seen in the future will be sufficiently similar to data seen in the past.[15] For this reason, it is important to track both the distribution of data seen by models in deployment and the difference between the real-world data distribution and that used during model training, validation and testing. Models can be assessed not only using the validation data for optimum performance on the loss function, but also for robustness and sensitivity to changes in the input distribution. Confidence levels of outputs must be tracked in order to detect changes in the data regime at runtime. It is also possible to learn other, secondary models, in parallel with the main model of interest. The job of these secondary models is to conduct anomaly detection on the input data of the main model. This can protect against both incremental and catastrophic data drift. More catastrophic varieties of anomalous data include both "innocent" corruption of data, as well as malicious adversarial attacks against AI systems. Tracking the relationships between primary models and "sentinel" models that ensure consistent input is key to understanding how real-world data changes over time will affect the trust and reliability of deployed models.

MODEL TRANSPARENCY

Combining the capability to track input data distributions with more comprehensive information on model output, enables a significantly greater degree of model introspection. This includes identifying sources of model errors such as bias-vs-variance and model-failure-vs-distributional-drift. These measures permit models to be more explainable. In many contexts, users either desire or demand – perhaps by the force of law — to know why an AI system made a particular decision. A standardized suite of model interpretation techniques and feature importance evaluations can help model implementers and users find a lingua franca for explaining model decisions. As models become more

---

[13] Spark MLlib pipeline description, Available at: https://spark.apache.org/docs/latest/ml-pipeline.html

[14] ONNX home page, Availabe at: https://onnx.ai/

[15] Mitchell, T. 1997. "Machine Learning." McGraw Hill.

complex, model behaviors should be tied to model components. Moreover, a model should have a scale which quantitatively describes its ability to be introspective.

3 SECURITY

Machine Learning can be viewed as an alternative to algorithmic programming. At its most fundamental difference, algorithmic programming is based on designing software to explicitly instruct what a system should do to meet an objective. Alternatively, ML models do not need to be explicitly instructed to meet an objective. Rather they learn insights, by applying analytical processes to historical and labeled data, to improve performance of the system to meet an objective. It is especially useful when the rules that govern an algorithm's behavior are too complex for a programmer to enumerate or even fully understand. More specifically, when using ML instead of specifying an algorithm that takes an input and produces a desired output, we provide input data and output data to a ML algorithm that will produce a model that transforms the input into something that closely approximates the desired output. In many cases, this property of ML allows us to leverage algorithms much more complex than what would have been possible with direct specification. Relying on data and an ML algorithm to specify a model, while very powerful, introduces a large amount of uncertainty into programs and systems that use the learned models. With this uncertainty comes much risk and exposure to danger.

Modern deep neural networks consist of many millions of parameters learned during training. It is not trivial, nor practical, to inspect complex learned models to understand how they behave. As a result, a complex machine learned model can never be fully trusted. However, trust in a machine learned model can be increased with model interpretability. Doshi-Velez and Kim prove an excellent overview and roadmap for the science of interpretable ML.[16] Examples of tools for the interpretation of black box algorithms include Shapley Additive Explanations[17] and Locally Interpretable Model Explanations (LIME).[18] Finally, ML procedures can be designed with interpretability in mind such as with the work of Lloyd et al. in developing automatically generated reports[19] and the work of Grosse et al.[20]

The use of ML to develop an algorithm implies that the data was too complex for a human to fully understand. This unfamiliarity around the data exposes systems to data poisoning attacks. In these cases, an adversary will manipulate the training data to degrade the performance of the learned model and introduce backdoors into learned models. For example, a poisoning attack might seek to cause an object detection algorithm to mis-identify a specific object at deployment time. Training data

[16] Doshi-Velez, F. and Been, K. (2017). *Towards A Rigorous Science of Interpretable Machine Learning*. [online] arxiv.org. Available at: https://arxiv.org/pdf/1702.08608.pdf

[17] Lundberg, S. and Lee, S. (2017). *A Unified Approach to Interpreting Model Predictions*. [online] arxiv.org. Available at: https://arxiv.org/pdf/1705.07874.pdf

[18] Ribeiro, M. and Singh, S. (2016). *"Why Should I Trust You?" Explaining the Predictions of Any Classifier*. [online] arxiv.org. Available at: https://arxiv.org/pdf/1602.04938.pdf

[19] Lloyd, J., Duvenaud, D., Grosse, R., Tenenbaum, J., Ghahramani, Z. (2014). *Automatic Construction and Natural-Language Description of Nonparametric Regression Models*. [online] arxiv.org. Available at: https://arxiv.org/pdfr/1402.4304.pdf

[20] Grosse, R., Salakhutdinov, R., Freeman, W., Tenenbaum, J., (2012). *Exploiting compositionality to explore a large space of model structures*. [online] cs.toronto.edu. Available at: https://www.cs.toronto.edu/-rgrosse/uai2012-matrix.pdf

uncertainty also introduces questions about the appropriateness of the data. Practitioners must ask themselves: is the data provided to the machine learning procedure representative of the intended task? If there is an appropriateness mismatch, model performance will suffer at deployment time. Uncertainty in the training data can be reduced by investigating and tracking data provenance, as well as by verifying that assumptions about the data hold prior to training. Models should not be learned from untrusted data.

We rarely have any control over future inputs to our models. In addition to preventing a guarantee of generalization of learned models, the uncertainty surrounding future data provides camouflage for adversarial attacks on learned models. These attacks usually consist of human-imperceptible changes to an input that drastically change a model's output, often with the goal of causing the model to produce the wrong output with high confidence. Examples of these types of adversarial attacks include Evasion,[21] Perturbed Input,[22] Obfuscated Gradient,[23] and Sensor Directed.[24] Attacks rely often on knowledge of the learned model or at least access to the model outputs. As a result, models and their outputs should be shared with only trusted parties. It is important is to be suspicious of consecutive API calls where the distribution of the input does not change very much, as the purpose of these API calls could be to estimate the behavior of the model for attack design.

Uncertainty around model behavior on future data can be reduced by training models in a robust manner and by testing existing models for robustness. Specifically, this means making sure that models are making decisions holistically, based on many extracted features instead of just a few. Another best practice is to not only rely on accuracy as a metric for model performance but also examine the model's behavior on synthetically generated adversarial datasets.

Research around ML security is still in its infancy, and many important problems remain open. ML security has been a focus area for the Booz Allen ML research team, and several research publications around adversarial machine learning have been authored by Booz Allen staff.[25, 26, 27, 28]

---

[21] Biggio, B., Roli, F. (2018). *Wild Patterns: Ten Years After the Rise of Adversarial Machine Learning*. [online] arxiv.org. Available at: https://arxiv.org/pdf/1712.03141.pdf

[22] Doodfellow, I., Shlens, J., Szegedy, C., (2015). *Explaining and Harnessing Adversarial Examples*. [online] arxiv.org. Available at: https://arxiv.org/pdf/1412.6572.pdf

[23] Athalye, A., Carlini, N., Wagner, D., (2018). *Obfuscated Gradients Give a False Sense of Security: Circumventing Defenses to Adversarial Examples*. [online] arxiv.org. Available at: https://arxiv.org/pdf/1802.00420.pdf

[24] Kurakin, A., Goodfellow, I, Bengio, S., (2017). *Adversarial Examples in the Physical World*. [online] arxiv.org. Available at: https://arxiv.org/pdf/1607.02533.pdf

[25] Nguyen, Andre T, Raff, Edward, (2019). Adversarial Attacks, Regression, and Numerical Stability Regularization. In The AAAI-19 Workshop on Engineering Dependable and Secure Machine Learning Systems.

[26] Raff, E., Sylvester, J., Forsyth, S., McLean, M., (2019). *Barrage of Random Transforms for Adversarially Robust Defense.* To appear in CVPR 2019.

[27] Fleshman, W., Raff, E., Sylvester, J., Forsyth, S., McLean, M., (2019). *Non-Negative Networks Against Adversarial Attacks.* In AAAI-2019 Workshop on Artificial Intelligence for Cyber Security.

[28] Another two papers are currently under blind review at other conferences and workshops.

4 OPERATIONS

Once Artificial Intelligence emerges from the lab, it is deployed to an incredibly diverse set of systems. These may include:

- Hyperscale cloud platforms
- Autonomous vehicles
- Portable personal devices such as phones or watches
- Surveillance devices, such as intelligent cameras
- Sensors such as environmental monitoring devices

This broad collection of operational environments provides a challenge when attempting to develop standards for measuring and evaluating AI system performance. Concerns around AI system performance may be as complex as:

- Degrees of model or system failure and robustness in the face of failure
- Level of trust in the model to accurately complete its assigned task
- Mitigate and respond to environmental variability

The variety of algorithms used to embody AI also represents a challenge. For example, learning styles create a variety of concerns around model maintenance, evaluation, reproducibility and version control.

DEPLOYMENT

The AI industry has recognized the need for standards around AI and ML and has sought to establish standards for model representation so that a system trained using one framework can be deployed and used to make decisions about data (commonly described as performing 'inference') using a different platform or framework. As previously described, the Open Neural Network Exchange (ONNX) format is used to represent deep learning models. ONNX has largely been championed by Microsoft, Facebook, and Amazon to provide interoperability and a standard foundation for implementing hardware optimization. Vendors with specialized hardware can implement supporting software components that read the ONNX model format for model execution on a variety of systems, platforms and devices.[29]

The Data Mining Group has developed Predictive Model Markup Language (PMML) and Portable Format for Analytics (PFS), referred to as "two complementary standards that simplify the deployment of analytics models." [30]

Each of these standards highlight the need for a standard representation of AI artifacts to allow portability of analytic techniques between development and deployment platforms. As they are

[29] https://onnx.ai/

[30] PMML General Structure, Available at: http://dmg.org/pmml/v4-3/GeneralStructure.html

emerging works, they may not be appropriately complete to represent all dimensions for measurement and evaluation included in a robust standards doctrine.

DEPLOYMENT CONCERNS RELATED TO LEARNING STYLES

A series of learning styles are broadly used within the AI community. Each of these bring potential concerns over measurement and validation of models trained using those styles in operational environments.

ONLINE LEARNING[31, 32]

Some models do not involve a discreate training process, but rather learn over time in the field as they are exposed to real world conditions. This process is known as Online Learning (OL). In this context it is either impractical or impossible to retain the entire collection of data used to produce the current state of a model trained via OL or the model itself is constantly evolving. It is difficult to reproduce a deterministically congruent model using a static training approach for a model trained via OL. This also precludes certain exploration of model robustness through data perturbation. Operational considerations highlight the need for mechanisms for identifying functional issues and reliable model versioning to detect and respond to model failure in order to return the deployed system to a known, good state.

ACTIVE LEARNING[33]

Supervised learning approaches suffer from a lack of annotated training data. Certain algorithms will identify cases where additional data is needed to better understand the phenomena they are trying to learn. This class of algorithms, known as Active Learning algorithms, provide feedback to a data source, such as a human annotator, as to which data instances need to be labeled in order to learn appropriate class distinctions in the data. This process can introduce systematic bias into models trained with this approach thus robust mechanisms for baselining and tracking model drift are required. In the long run these will lead to active learning models that generalize well to real-world conditions.

TRANSFER LEARNING[34]

Models can be adapted to tasks beyond that for which they were originally trained. The process of re-using a model as a basis for a new model is known as fine-tuning. This learning technique is typically referred to as transfer learning. In effect this process transfers knowledge from one model to another.

---

31 Bottou, L. (2018). *Online Learning and Stochastic Approximations*. [online] bottou.org Available at: https://leon.bottou.org/publications/pdf/online-1998.pdf

32 Amari, S. (1967). *A Theory of Adaptive Pattern Classifiers*. [online] cbs-ni.riken.jp Available at: https://cbs-ni.riken.jp/modules/xoonips/download.php/027.pdf?file_id=43

33 Atlas, L., Cohn, D., Ladner, R. (1989). *Training Connectionist Networks with Queries and Selective Sampling.* [online] nips.cc Available at: https://papers/nips/cc/paper/261-training-connectionist-networks-with -queries-and-selective-sampling.pdf

34 Pratt, L., Mostow, J., Kamm, C. (1991). *Direct Transfer of Learned Information Among Neural Networks.* [online] aaai.cc Available at: https://www.aaai.org/Papers/AAAI/1991/AAAI91-091.pdf

INCORPORATION AND USE OF EMBEDDED SENSORS[35]

In many industries existing standards provide benchmarks against which sensors can be measured. As AI learning styles change, operational systems must address concerns that arise when models evolve under the influence of new data, especially of those data sources sourced from the addition of new sensors. Variability in manufacturing processes can lead to inconsistent sensors that provide a deployed model data that is considerably different from that on which it was trained. It is essential to develop a system of standards and measures to quantify model change, sensor variability and the uncertainty that arises when new conditions develop in the world.

ARTIFICIAL INTELLIGENCE SYSTEM TRUST LEVELS AND CERTIFICATIONS

It is critical to establish standard measures of trust and reliability for Artificial Intelligence systems. For example, certain AI systems may be certified to be used in a human-assistive capacity, while others certified to operate without direct human oversight. This continuum of validated autonomy may span the breadth of what roles AI systems can adopt, from a simple home assistant to a system controlling components of the nation's critical infrastructure. In some cases, it becomes more appropriate to evaluate an AI system as we would evaluate a Human performing the same task. For example, in our paper, "Dr. AI, Where Did you Get Your Degree?"[36] we suggest that government agencies apply standards traditionally used to certify practitioners in the medical domain as opposed to certifying AI as it were a medical device.

TAXONOMY OF ARTIFICIAL INTELLIGENCE

This discussion of learning styles and trust levels leads us to the need for standardization around a taxonomy of Artificial Intelligence. Formal definitions of different types of AI will allow groups across fields to communicate around issues related to AI and ML applications, concerns, the nature of risk and the suitability of systems for specific use cases. Such a taxonomy may consider the following dimensions.

- Definitions of different uses of AI
    - Supportive - AI used in a capacity where it eliminates human toil such as rote repetitive tasks.
    - Augmenting - AI used in a capacity to expand human capability, where it performs tasks humans are incapable of; however, the AI still requires human oversight.
    - Autonomous - used independently from human operation.
- Definitions of levels of AI risk
    - Certification of the implications of failure – from harmless to the loss of human life
    - Detectability of failure – degrees from easy to observe failure to hard to detect failure.

---

35 McGrath, M., Scanaill, C. (2014). *Regulations and Standards: Considerations for Sensor Technology*. [online] springer.com Available at: https://link.springer.com/chapter/10.1007/978-1-4302-6014-1_6

36 Raff, E., Lantzy, S, Maier, E. (2018). *Dr. AI, Where did you get your degree?* Proceedings of the International Workshop on Artificial Intelligence in Health, pp. 76-83. Available at: ceur-ws.org/Vol-2142/short11.pdf

- Susceptibility to influence – ease of gaming the system or performing adversarial attack and robustness in the face of these attacks.

- Ethical Standards
  - Such as prioritizing human well-being[37] over correctness or failure.

It is clear that these dimensions are incomplete, but serve as an example as to how AI system can be graded and certified. Certain applications may require specific levels of certification in order to have an AI system approved to perform for a given use-case.

5 CONCLUSIONS AND RECOMMENDATIONS

The impact of Artificial Intelligence and Machine Learning (AI/ML) systems on society and commerce will only become more pervasive and will impact the future in ways that cannot be fully imagined today. To influence the positive societal and commercial influence of AI/ML systems, standards must be adopted today to formulate a common understanding of these systems and their capabilities and reduce the risks of unintended consequences or direct manipulation via nefarious actors.

Key Recommendations
As AI/ML systems become more complex, government agencies should adopt standards similar to those traditionally used to certify human practitioners for certification.
A Taxonomy of AI/ML systems should be established that captures characteristics along a number of dimensions such as the learning style, nature of its use, the risk of failure, and ethical implications of the decisions made by the AI/ML system.
Information necessary for calculating the historical accuracy and health state of AI/ML systems should be preserved to reconstruct historical model performance including APIs for testing of model components, introspection capabilities, and metadata descriptions of the model.
AI/ML systems can be made more secure by protecting the provenance of training data, as well as by verifying that data assumptions remain the same as the model is trained.
As AI/ML models become more complex, model behaviors should be traceable to each of the model's components.
AI/ML model should have a scale which quantitatively describes its ability to be introspected.

To that end, Booz Allen Hamilton has delivered many AI/ML-based services and products to the U.S. Government. This experience has provided many opportunities to assess and align common approaches and process improvements. It is our position that setting artificial intelligence standards, based on collective ideals of other organizations, would improve AI technologies overall and we welcome this effort.

[37] https://ethicsinaction.ieee.org

About Booz Allen

For more than 100 years, business, government, and military leaders have turned to Booz Allen Hamilton to solve their most complex problems. They trust us to bring together the right minds: those who devote themselves to the challenge at hand, who speak with relentless candor, and who act with courage and character. They expect original solutions where there are no roadmaps. They rely on us because they know that—together—we will find the answers and change the world. To learn more, visit BoozAllen.com.