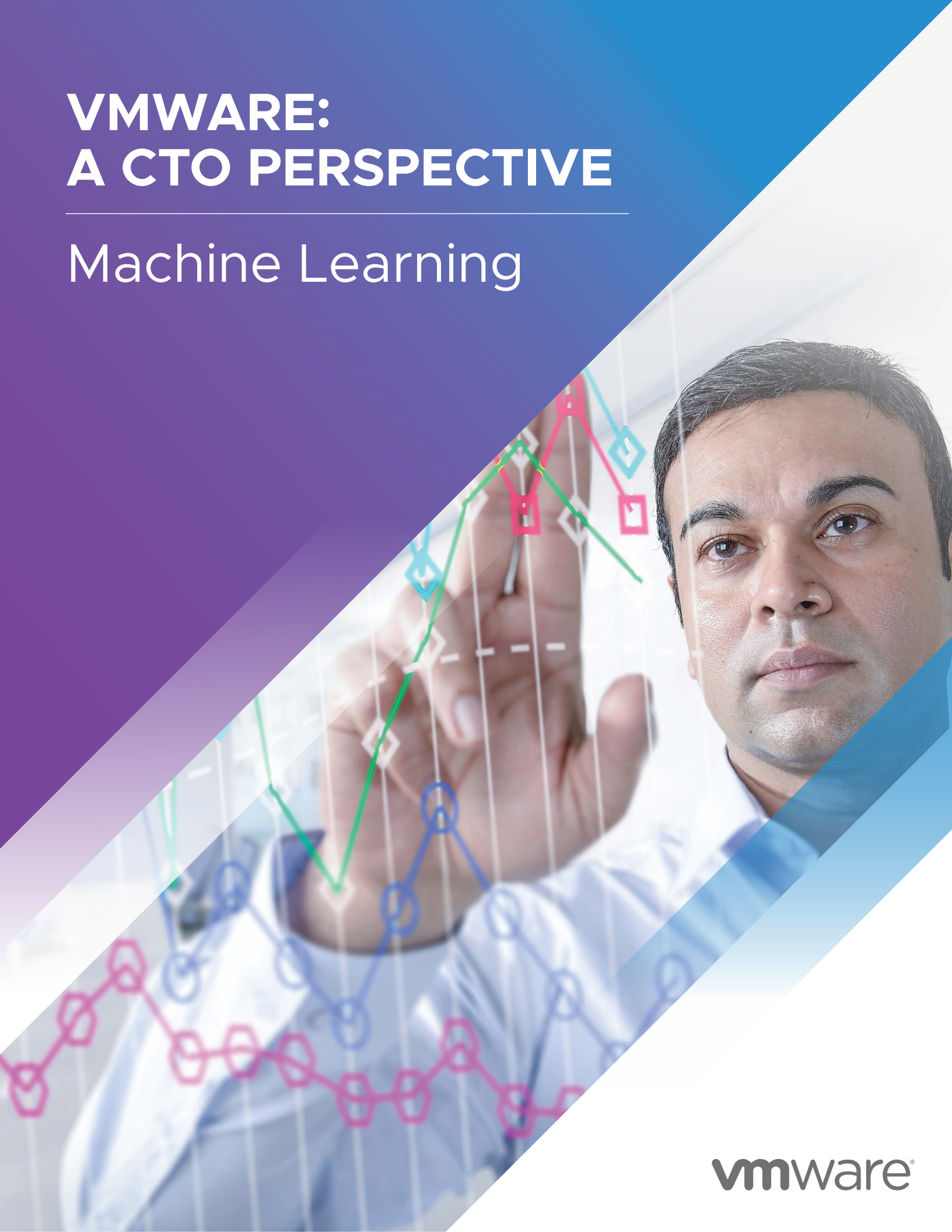
# VMWARE:
# A CTO PERSPECTIVE

## Machine Learning



vmware

AUTHORS:

**DAVID TENNENHOUSE**
SVP & Chief Research Officer, VMware

**&**

**JOSH SIMONS**
Chief Technologist for HPC, VMware

PUBLICATION DATE:

**JANUARY 2019**

# Table of Contents

# Bottom Line

Machine Learning (ML) creates tremendous opportunities for enterprises to detect patterns in the data they collect, and then use those patterns to create new products or services, improve existing offerings, and improve their internal operations. To obtain those benefits, enterprises will need to navigate a number of challenges:

- ML capabilities, such as photo recognition, will become table stakes for many applications. However, only a subset of ML use cases will create competitive advantage.
- Many different types of ML models and development environments are available. It will be important to select technologies that are suited to the application, avoiding those that are overly-complex or that may introduce inherent biases that are not acceptable for enterprise use cases.
- Data scientists are in short supply and require data exploration tools to be effective.
- The deployment of end-to-end ML pipelines requires significant tooling that needs to be skillfully integrated with the automation and orchestration capabilities already in use.
- The choice of location for ML development and deployment will be driven by a number of considerations. Hybrid ML, involving combinations of public cloud, private cloud, and edge elements, will be common.

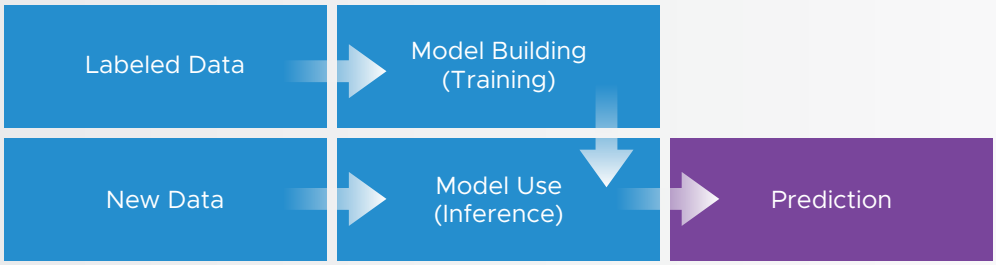
ML is no longer a technology of the future. Enterprises should be taking action now to incorporate ML into their operations, products, and services. While doing so, they should be thoughtful as to what types of ML models they use and how they integrate ML capabilities into their IT infrastructure.

# Machine Learning Basics:

## *Why is it called machine learning?*
## *What are training and inference?*

There has been an explosion of interest in ML. This is especially the case for a variant of ML referred to as Deep Learning (DL) that is powering recent advances in *generic* pattern recognition tasks such as recognizing people in photos and translating between languages. ML can also be used for more *specialized* or *differentiated* tasks such as fraud detection or, in the case of VMware's vRealize Log Insight product, detecting patterns in log messages.

Big Data, i.e., the ability to collect and transform large amounts of data, is a key driver of modern Machine Learning. Typically, data that has previously been collected is used to *train* a machine learning *model*. Ideally, this data is labeled in the sense that it associates raw input data with some information concerning its content. For example, photographs may be labeled to indicate whether an animal appears in them and, if so, what type of animal. This data can then be used to train a model that detects and recognizes specific types of animals in new photos that it hasn't previously seen.




The nice part of all this is that the machine *learns* how to perform this pattern recognition task without being explicitly programmed to do so. There are two key benefits:

1. Instead of laboriously designing and implementing a program that recognizes cats (which we might not even know how to do), we train a model.
2. In addition, the same system (trained on a suitable set of images) can be used to recognize cats, dogs, birds, cars, furniture, etc.

Once a model is built, it is deployed into production to perform *inference* on new data continuously. Inference is the use of the model to assign a label to new data that the model may have never seen before. In contrast to training, the inputs are unlabeled and the outputs are the suggested labels that are inferred by the model.

## Machine Learning is very resource intensive. Training the model often requires very compute-intensive processing of the entire training data set, not just once, but multiple times.

Although each *inference* action, which is often performed online, may be less computationally demanding, the aggregate of inference computations over time – and over a potentially large number of model instances (e.g., every cell phone or automobile) – can be very significant.

*Deep Neural Networks (DNNs)* are a currently popular family of models that are particularly resource intensive. As the neural network part of the name suggests, these are models based on the interconnection ("networking") of parameterized internal compute elements (loosely analogous to neurons). They are referred to as being deep because they have many layers of such compute elements. DNNs require especially large amounts of computing power during training.

While many different types of machine learning models have been developed, they often share a common underlying mathematic representation, typically based on matrix algebra, which is highly parallelizable. As models become larger and more sophisticated, these underlying representations grow larger as well, requiring more and more computational power. This has resulted in a virtuous cycle – the availability of computational power has driven the success of machine learning which, in turn, is now driving demands for increased computational resources.

Since many of the underlying computations in ML are highly parallelizable, they can often benefit from the use of specialized CPU instructions (e.g., SIMD) and/or the scale-out distribution of their processing across multiple CPU cores and/or servers. Some applications, such as DNN training, may also benefit from the use of hardware-based compute accelerators, such as GPUs, that contain thousands of computing elements designed to operate in parallel. Although they are more efficient than a CPU on some operations, the GPU also contains significant complex functionality to support its primary mission – graphics rendering. Seeing an opportunity, a number of companies are developing accelerators that are specifically targeting ML computations. For example, Google has designed their own Tensor Processing Units (TPUs), which are highly optimized for ML processing.

VMware supports the integration of GPUs and other accelerators into our virtualized environment today and we are committed to supporting new accelerators as they become widely available. As with other resources in the data center, pooling and sharing accelerators will be important. For example, during training, a large model may require access to multiple GPUs. Conversely, during inference, it may only require a portion of an accelerator. The efficient utilization of accelerators will require that they eventually be pooled, virtualized and shared across multiple applications that are isolated from each other. Similarly, there will be a need for orchestration and management tools, e.g., to place and monitoraccelerator-dependent applications and to allow them to be suspended and resumed.

# VMware Perspective

At VMware we approach emerging technologies such as machine learning by thinking about the needs of our Enterprise customers. How will our customers integrate machine learning into their operations, products and services, and how can we support them in doing that? How will their needs be different from those of the researchers and others pioneering these technologies?

One key observation is that the locations chosen for ML development and deployment may be different. For example, model experimentation and development may take place in a public cloud, using an elastic offering such as VMware Cloud on AWS, while deployment may be located elsewhere. Furthermore, deployment itself may be hybrid, e.g., training might be implemented in a private or public cloud with online inference located at the edge. The choice will be driven by a number of considerations including *data gravity*, governance, compliance, privacy, cost, latency, ease of development, etc. *Data gravity*, which is particularly relevant to big data applications such as ML, refers to the attractiveness of positioning computation close to where the data is located, so as to avoid the cost, bandwidth, latency, governance, compliance, and privacy implications of moving large datasets across the network.

A second observation is that the tools and techniques that have enabled hyper-scale consumer grade applications, such as photo recognition and language translation, are not always suited to the typical enterprise application. They address well-defined and somewhat *generic* problems for which organizations operating consumer services, such as Google and Microsoft, have collected enormous training sets. They have also created and operate customized ML *pipelines* that are tailored to operate at a very large scale with respect to their data ingestion, the complexity of the models implemented, etc. There are situations, e.g., language translation, in which organizations will want to consume these hyperscale capabilities and re-position them for Enterprise use. There will also be cases where Enterprises will use variants of the hyper-scale ML tools that have been scaled-down and generalized for use on smaller problems. Finally, there will be many cases where simpler ML technologies will be sufficient to address the problem at hand.

In summary, our Enterprise customers will want to use ML to address a wide variety of problems and will typically operate on datasets that are large relative to their own past experience, but much smaller than those used to enable hyper-scale services. They will also want to leverage the agility afforded by hybrid and/or multi-cloud ML deployments. Our focus on Enterprise requirements motivates the observations and recommendations in the remainder of this document.

## How will Enterprises extract long-term competitive advantage from Machine Learning?

Enterprises will integrate *generic* machine learning capabilities (e.g., speech, text and image processing) into their products, services and day-to-day operations. Since the performance of these generic capabilities is very dependent on the size of the training sets, Enterprises will most likely consume them *as-a-service* or by obtaining pre-trained models from third parties and doing their own inference (which also allows them to retain full control over their data).

Adopting *generic* ML will allow enterprises and their customers to garner very significant productivity gains, improvements in quality, etc. Although early adopters may get a head start on their competition, *generic* ML will not provide sustainable competitive advantage since the suppliers will make the same benefits available to the enterprises' competitors.

We believe the path to differentiation will lie in the ability of an organization to leverage its own data and its unique domain knowledge and expertise to field *differentiated* machine learning. This means enterprises will not only need to learn how to consume ML, but also how to choose appropriate ML algorithms and how to train and operate their own models, i.e., how to field ML pipelines. One intriguing compromise with respect to model training is an approach known as *transfer learning* in which a customer acquires a generic pre-trained model from a third party and differentiates it through additional training that leverages the customer's own unique training data.

## What types of big data algorithms have VMware researchers been working on?

Notwithstanding all the buzz about DNN's, our researchers believe there are some other types of algorithms that enterprises will also require to solve important big data problems. One area of interest is detecting patterns in multi-dimensional time series data, i.e., streams of data in which multiple metrics are captured at each point in time. For example, vSphere can capture measurements of many properties of virtual machines in a software-defined data center. So the record for each point of time can be thought of as having thousands of dimensions. If we had better algorithms to handle this type of data we could more efficiently find interesting correlations, e.g., identifying combinations of VM properties that are early indicators that SLAs will not be met, or that a VM migration should be triggered.

A related area in which our researchers have recently made progress is *anomaly detection*, e.g., detecting a pattern of behavior that is indicative of a security exploit. Principal Component Analysis (PCA), which is the gold standard algorithm for anomaly detection, is very resource intensive and involves computations whose storage needs scale quadratically with the number of dimensions being analyzed.

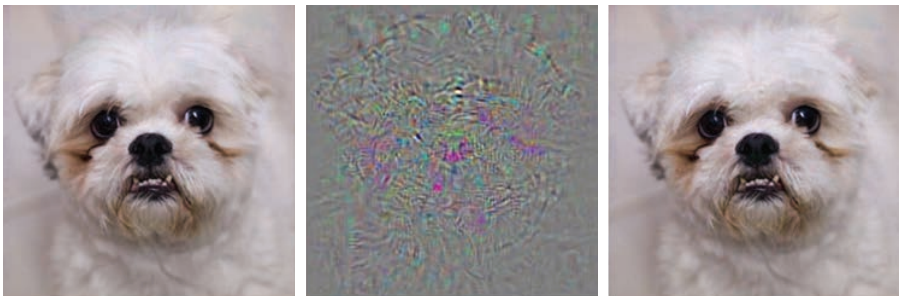
When PCA is used to tease out correlations involving significant numbers of dimensions the storage and computation required quickly exceeds the capacity of modern servers. Our researchers have found a way to use a technique known as sketching to greatly reduce the size of data that needs to be processed. This makes it practical to detect anomalies in datasets that have many more dimensions than can be processed with vanilla PCA. To learn more about Faster Anomaly Detection via Matrix Sketching, visit: **https://arxiv.org/pdf/1804.03065.pdf**

## What other concerns should Enterprises have about ML algorithms? Can they be biased?

Something that is worrisome about certain types of machine learning, especially deep neural networks, is that researchers do not have a thorough understanding of how they work. This may be okay for consumer applications such as recognizing pets in photos but there are many enterprise applications for which the failure to provide a *chain of reasoning* that can *explain* an ML-driven decision will not be acceptable. The absence of such an explanation means we have no way to vet the decisions made by such models.

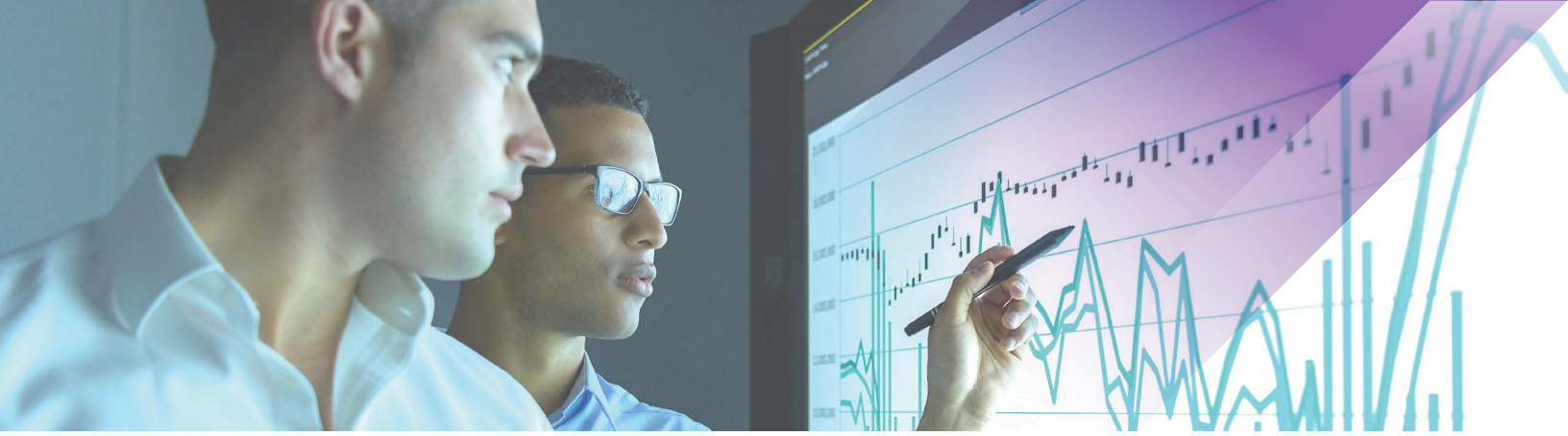
Closely related to *explainability* is the issue of bias. For example, a model designed to detect both cats and dogs whose training data consisted of 99% dog examples, would be very biased towards detection of dogs with much poorer performance on cat images. These biases can arise due to natural biases in training data or through the subtle and deliberate introduction of biased data into the training as a means of attack. These biases are more difficult to detect without explainable models.

In addition to attacks on their training data, models can be susceptible to attacks on their inputs. A small adversarial modification of the input can easily mislead a model, thereby leading to unpredictable behavior. The figure shows an example in which a carefully computed set of pixel modifications have been applied to the image of a dog. While a human easily identifies the tampered image as a dog, the ML model misidentifies the image as an ostrich. While this example may be amusing, researchers have demonstrated that the recognition of a traffic stop sign can be altered by adding unobtrusive stickers to it. Whereas a human viewer would automatically ignore the stickers, the ML algorithm was fooled into interpreting the decorated stop sign as a speed limit sign.



**DOG** **+NOISE** **OSTRICH**

Source: Samim, "Adversarial Machines: Fooling A.Is (and turn everyone into a Manga)." Medium.com Dec. 7th, 2017. https://medium.com/@samim/adversarial-machines-998d8362e996



Today, there are healthcare applications in which certain ML models can, on average and without explanation, produce better diagnostic results than physicians. Similarly, DNN-based models can be used to predict credit worthiness. In the near term, some enterprises may mask reliance on these forms of inadequately understood and potentially biased technology by positioning them as *decision aids* with a human responsible for the ultimate decision. Indeed, the European Union's General Data Protection Requirements (GDPR) states in Article 22 "The data subject shall have the right not to be subject to a decision based solely on automated processing." In the longer term we believe enterprises will gravitate towards *explainable* systems for applications that are mission critical and/or involve significant risk of life and/or property.

The good news is that researchers are actively working to improve the situation and, in the U.S., DARPA has a specific program whose goal is to enable explainable AI[1]. In addition to monitoring these efforts, one of our researchers is working to impose constraints on certain types of DNNs, i.e., to prove logical assertions governing the outputs a model generates[2]. Eventually this may allow us to ensure that the outputs of a model always stay within a range that is known to be safe/acceptable, even if we are not able to explain the reason for the precise output.

## What about the human element?

Data scientists play a key role in the creation of ML models for their organization – from defining the question to be answered to collecting, cleaning & transforming the necessary data. Then selecting, building, evaluating, and tuning appropriate models and algorithms.

To help improve data analyst productivity, VMware researchers have been prototyping Hillview, an open source big data visualization engine that helps data analysts interact with huge data sets that may be distributed over large numbers of servers and storage devices. One way to think of this is as a spreadsheet-like tool that operates on tables that can have trillions of rows and thousands of columns. The tool allows analysts to instantly sample data in order to identify interesting properties, inconsistencies requiring data cleansing, etc.

For more information and a demonstration of Hillview functionality:

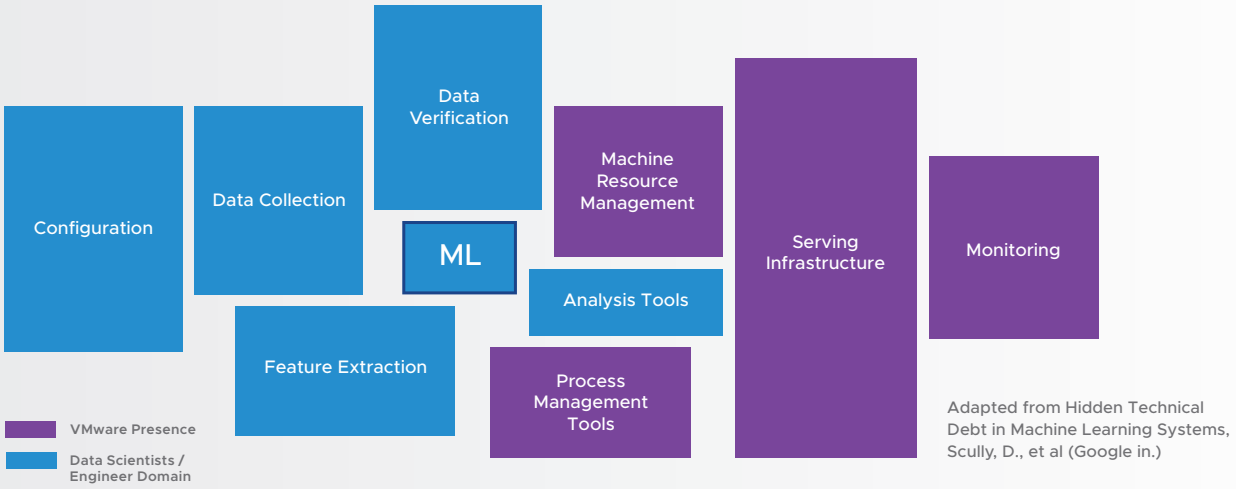
**https://research.vmware.com/projects/hillview**
**https://github.com/vmware/hillview**

[1] https://www.darpa.mil/program/explainable-artificial-intelligence
[2] https://www.aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/16898

## What do Enterprises need to know about deploying end-to-end ML pipelines?

As with traditional applications, there is much more to deploying and operating an ML system than just building an ML model. ML Pipelines are complex as illustrated below.

### End-to-End ML: Core ML algorithms and substantial "surrounding" infrastructure




There are typically a number of steps involved:

**Pre-model evaluation and prototyping:**

- Initial Data Collection and data analysis to assess whether there is any useful *signal* in the data, what data cleansing is required and what type(s) of models should be used. This is where tools such as Hillview can help.
- Build and train a preliminary model and then evaluate its use as an inference engine within the target application. A number of model development environments, such as TensorFlow, Caffe2 and Pytorch, are available to speed this task.

**Development/Operations:**

- Build and operate an automated, recurring training pipeline that collects data, cleanses and verifies it, updates the model, validates the model, and periodically releases the updated model into the inference pipeline. The training frequency may be weeks, days, or near continuous.

- Build and operate an automated inference pipeline that collects operational data, cleanses and verifies it, uses the model to infer results, and monitors the performance of the model. Trained models may be released for use within the data center or pushed to the edge (via IoT management infrastructure) to support low-latency inference near the points of data collection or use. Sometimes the same model can be shared by multiple applications in which case the model can be operationalized as a shared service accessed by applications, each of which has their own data collection, cleansing and monitoring mechanisms.

One important take-away from the diagram is how small the creation of the ML model is in comparison to the surrounding elements. Furthermore, many of the DevOps building blocks (those colored in green on the diagram) can be implemented using traditional workload automation and orchestration tools, such as VMware's vRealize products.

That still leaves a number of new capabilities (those colored in blue on the diagram) that teams fielding ML applications will require. VMware is committed to working with our partners to create validated designs that our customers can use to fill those needs. We are also working with some of the world's leading researchers to create variants of the required functionality that are tailored to the needs of Enterprise and SMB users. For example, we are founding sponsors of the Stanford DAWN project whose mission is to "Empower non-ML experts to conduct production quality domain-specific analytics".

For more information about the Stanford DAWN Project: **https://dawn.cs.stanford.edu/**
For more information about the UC Berkeley RISELab: **https://rise.cs.berkeley.edu/**

### *How is VMware adopting Machine Learning?*

We are working hard to make sure that our customers can use their existing investments in vSphere, VMware Cloud, and VMware Cloud Foundation as a springboard into this exciting ML era. We are also working to make VMware itself an early adopter of ML technology. That means building our ML muscles by investing in our people, processes, products and services. One step we're taking is to incorporate ML capabilities into virtually all of our products and services. The goal is to create *smarter products and services* that deliver increased benefits to our customers. A parallel activity involves incorporating ML into many aspects of our business operations (sales, support, software development, finance, recruiting, HR, etc.) in order to make VMware a *smarter company.* For example, we are exploring the use of text-based ML to help technical support personnel quickly identify similar, previously solved cases so as to speed the resolution of customer issues.

How can you and your team use ML to create a *smarter company* and *smarter products*?

# Recommendations

## Machine Learning offers tremendous opportunities for productivity gains, and enterprises should be aggressive in experimenting with and deploying ML capabilities.

In doing so, they should be thoughtful of the role of data gravity and its impact on where they locate different types of ML functionality. For example, model experimentation might be conducted in the public cloud, training in the private cloud, and inference at the edge.

They should also make conscious choices with respect to how they pursue generic ML functionality, which will be table stakes expected from all players, and differentiated ML, that can create competitive advantage by being grounded in their proprietary data and domain expertise.

As they choose amongst different types of ML models, customers should carefully consider the importance of factors, such as explainability and bias to their business. They should also be aware that some of the models receiving the most attention today, such as DNNs, require extremely large amounts of training data, which might not be readily available. There are many cases in which more established ML models, such as PCA, will be more effective.

Similarly, customers should start thinking of ML training and inference as workloads that need to be incorporated into their operations. As discussed, the models themselves are a relatively small component of the overall ML pipeline. To avoid the proliferation of discrete clusters and operating environments, customers should look for ways to support ML pipelines using orchestration and automation tooling that is common to their other workloads. Doing so will yield operational efficiencies, improved resources utilization, and a consistent approach to compliance and governance. In particular, as enterprises make plans to adopt ML acceleration capabilities, such as GPUs, they should ensure that hardware can be shared across a rapidly changing portfolio of applications, so they don't end up with stranded and under-utilized resources.

Finally, enterprises should be tackling the human component of ML, especially the expanding need for data scientists and the data exploration tools, such as Hillview, that they require. Most companies will find they need to recruit individuals with Big Data skills, as they are in short supply. A good approach is to plan to augment a few new hires by encouraging current employees, especially those with deep domain knowledge, to learn more about data science. There is a generous supply of online courseware and tutorials available from universities, Coursera, Udacity, etc.

# Conclusion

Machine Learning is no longer a technology of the future. Enterprises should be taking action now to incorporate ML into their operations, products and services. While doing so they should be thoughtful as to what types of ML models they use, where they locate them, the data they operate on, and how they integrate ML capabilities into their IT infrastructure.

[3]https://blogs.vmware.com/cloudnative/products/

vmware®



VMware, Inc.
3401 Hillview Ave., Palo Alto, CA 94304
**www.vmware.com**