Independent market research and competitive analysis of next-generation business and technology solutions for service providers and vendors

HEAVY READING WHITE PAPER

Democratizing AI:
How Acumos Can Fast Track Your AI Development

A Heavy Reading white paper produced for Tech Mahindra

Tech Mahindra

AUTHOR: JAMES CRAWSHAW, SENIOR ANALYST, HEAVY READING

INTRODUCTION

The complexity of communications networks is increasing inexorably with the deployment of new services, such as software-defined wide-area networking (SD-WAN), and new technology paradigms, such as network functions virtualization (NFV). To meet ever-rising customer expectations, communications service providers (CSPs) need to increase the intelligence of their network and service operations.

Heavy Reading believes that artificial intelligence (AI) will be key to automating network operations and enhancing customer experience. In order to move to real-time, closed-loop automation, CSPs need systems that are capable of learning autonomously. That is only possible with AI techniques such as reinforcement learning. Network automation platforms such as the Open Networking Automation Platform (ONAP) will need to incorporate AI to deliver efficient, timely and reliable operations.

Developing AI-based systems today can be a complex and time-consuming process. There are many tools available, including those provided by cloud computing companies that are keen to host the data used by those systems. However, this multiplicity of non-interoperable frameworks for AI development is arguably hampering the development of AI applications for all but the largest and most IT-savvy organizations.

Acumos presents a common platform to help bridge the gap between competing AI frameworks and toolkits. It is an open source initiative for the development, training and deployment of AI models. The platform allows data scientists to publish AI models while shielding them from the chores of custom development required to develop a fully integrated solution. Acumos packages each model into an independent, containerized microservice, which is fully interoperable with any other Acumos microservice. These microservices are easy to integrate into practical applications by regular software developers, with no need for a background in data science or knowledge of specialized AI development tools.

Acumos is modeling-language-agnostic (Java, Python, R, etc.); not tied to any particular toolkit (TensorFlow, Scikit Learn, RCloud, H2O, etc.); and, unlike many proprietary systems, not tied to any single runtime infrastructure or cloud service (e.g., Azure, AWS). Acumos only requires a container management facility, such as Docker, to deploy and execute portable, general-purpose applications. Using Acumos, data science teams can build abstract AI models, using their preferred tools. These models can be adapted to a variety of data formats, using data adaptation libraries. The models are trained on particular data sets and can then be shared with a wide community of users via a marketplace. By eliminating the need for data science expertise, Acumos democratizes the AI development process, lowering the barriers to entry for regular software developers.

Self-organized peer groups across one or multiple companies can securely share information about how AI solutions perform, to apply the crowdsourcing concept to AI development. AI models can be acquired from the marketplace and integrated into complete solutions. Developers can access encapsulated AI models, without knowing the details of how they work, and connect them to a variety of data sources, using data adaptation brokers, to build applications through a simple chaining process.

This white paper discusses the advantage of community-based innovation, explores the potential applications of AI in telecom, assesses some of the challenges in applying AI and describes how the Acumos project can help solve some of these problems.

COMMUNITY-BASED INNOVATION

During much of the 20th century, technology innovation in the telecom industry came from the internal R&D function of the largest operators, exemplified by AT&T's Bell Labs, whose scientists created the transistor, the laser, the Unix operating system and the programming language C, among many other innovations we take for granted today. As telecom networks became globally interconnected, there was a need for common protocols and interfaces to allow communications between operators and across countries. This gave rise to a number of standards organizations, such as the ITU, IETF, IEEE, ETSI and 3GPP, where operators, together with their technology suppliers, would agree on common approaches and specifications. These global standards have enabled the low-cost, highly connected, fixed and mobile voice and data services that we enjoy today.

However, agreeing on standards is a lengthy process, especially when some participants have vested interests in promoting different technology approaches that favor their own intellectual property. Given the ever-faster pace of change in business and technology, telecom industry participants are increasingly looking for more agile approaches to technology innovation.

Although many standards bodies exist to serve the IT industry, open source communities arguably play a more important role most notably the Linux family of operating systems. A number of factors make open source an ideal environment for collaboration and innovation:

1. Potentially Large Developer Base

Popular open source communities include a large number of developers from many different companies, as well as academics and hobbyists. Most open source projects start life within a company or university and are donated to open source in order to exploit the resources that a bigger community can provide. When external participants contribute to an open source project, all of those involved (including the original donor) share these benefits. This provides economies of scale in engineering effort.

2. Code Quality & Security

With open source software, the large number of participants typically means that bugs are identified more quickly than proprietary software projects with smaller teams. Open source projects typically have governance rules in place such that when new code is contributed, it is checked by at least two other non-affiliated participants (i.e., working for different companies). The larger community of developers contributing to open source projects often results in a faster identification and resolution of bugs and security weaknesses. Naturally, the users of open source software must make sure they keep up-to-date with the latest release to ensure that their systems remain secure.

3. No Upfront Licensing Costs

Open source software has no upfront licensing costs, which means it can be downloaded for experimentation, research and use in products and services. In practice, many organizations may need support services to allow them to use open source software, integrate it with their existing systems and keep up-to-date. In some respects, open source is like a gift puppy: no upfront cost, but potentially significant maintenance costs. Nonetheless, for most organizations these maintenance costs will be far less than the cost of developing a proprietary solution just for their own use.

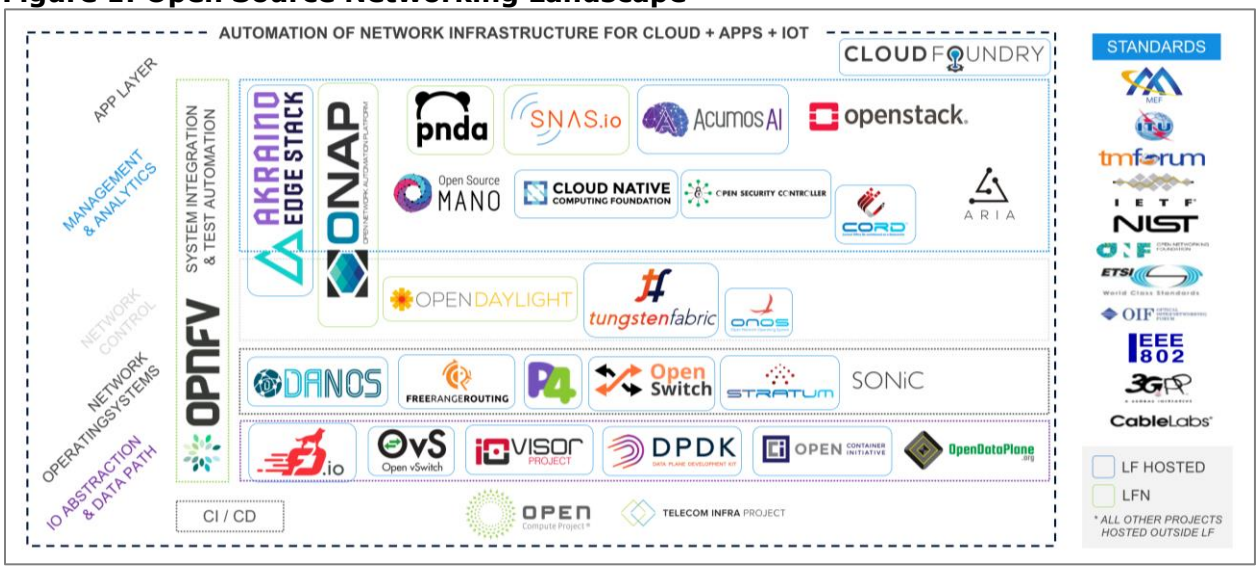
4. Speed of Innovation

Open source software can be modified without the need to ask permission or pay for expensive change requests from a proprietary software vendor. In practice, however, companies may be better off staying with the official distribution and requesting or submitting changes themselves, rather than forking the software. The community will typically make frequent updates to the upstream software with official releases on a regular cadence (e.g., six months). This contrasts with the release cycles of proprietary software, which are typically governed by commercial imperatives, rather than a motivation to have the best software available as soon as possible.

Open Source in Networking

The advantages of open source development and the increasing softwarization of the networking industry, driven by the SDN and NFV paradigms, have led to a proliferation of open source projects in the telecom industry, as summarized in **Figure 1**. One of the most notable projects is ONAP, which is championed by AT&T and China Mobile as the next generation of OSS. Another is OpenStack, which is the dominant telco cloud management system.

Figure 1: Open Source Networking Landscape



Source: Linux Foundation

Open Source AI

The AI field is no different from other software applications and is becoming increasingly dominated by open source software that brings together multiple collaborators and organizations (see **Figure 2** for a few examples). As Ibrahim Haddad writes in *Open Source AI Projects, Insights, and Trends*, "It is increasingly common to see AI as open source projects. This is in part due to its roots in academia, which has historically been a steady source of open source proof-of-concept projects. However, it is also rooted in the cost to build a platform and the realization that the true value is in the models, training data, and the apps. By open sourcing a platform, the host has an opportunity to recruit others (possibly unpaid) contributors, particularly if other organizations can be incentivized to integrate the platform into their own products. As with any technology where talent premiums are high, the network effects of open source are very strong."

Figure 2: Examples of Open Source AI Projects

Project (Creators)	Description
Acumos (AT&T and Tech Mahindra)	A platform and open source framework that makes it easy to build, share, and deploy AI apps. Acumos standardizes the infrastructure stack and components required to run an out-of-the-box general AI environment, freeing data scientists and model trainers to focus on their core competencies, accelerating innovation. As such, the Acumos AI Platform is a complete environment for the full lifecycle of AI and ML application development. The free Acumos Marketplace packages various components as microservices and allows users to export ready-to-launch AI applications as containers to run in public clouds or private environments.
CAFFE2 (Facebook)	A deep learning framework enabling simple and flexible deep learning. Built on the original Caffe, Caffe2 is designed with expression, speed, and modularity in mind, allowing a more flexible way to organize computation.
TensorFlow (Google)	Software library for numerical computation using data flow graphs. Nodes in the graph represent mathematical operations, while the graph edges represent the multidimensional data arrays (tensors) communicated between them. The flexible architecture allows you to deploy computation to one or more CPUs or GPUs in a desktop, server, or mobile device with a single API.
Torch (Ronan Collobert, Koray Kavukcuoglu, Clement Farabet)	Torch is a machine-learning library, a scientific computing framework, and a script language based on the Lua programming language. It provides a wide range of algorithms for deep machine learning (ML).

Source: Open Source AI Projects, Insights, and Trends, Ibrahim Haddad, May 2018

POTENTIAL APPLICATIONS OF AI IN TELECOM

Below we summarize the areas that Tier 1 operators are employing AI today. We have split them into three main categories: networking and IT operations, fraud and security, and customer care. **Figure 3** is not an exhaustive list of all use cases of these companies, nor indeed of all CSPs using AI. Below we discuss each of these categories in more detail.

Figure 3: CSP AI Example Summary

Company	Networking & IT Ops	Fraud & Security	Customer Care
AT&T	✓	✓	✓
Colt	✓		✓
Deutsche Telekom			✓
Globe Telecom	✓	✓	✓
KDDI	✓		
KT	✓		
SK Telecom	✓		
Swisscom			✓
Telefónica	✓		✓
Vodafone	✓		✓

Source: Heavy Reading

Network & IT Operations

Examples of network-centric applications of AI include:

- Anomaly detection for operations, administration, maintenance and provisioning (OAM&P)
- Performance monitoring and optimization
- Alert/alarm suppression
- Trouble ticket action recommendations
- Automated resolution of trouble tickets (self-healing)
- Prediction of network faults
- Network capacity planning (congestion prediction)

AI could support network operations to detect issues – e.g., faults, service-level agreement (SLA) breaches – in real time, diagnose root causes, correlate across multiple event sources, filter out noise (false alarms) and recommend solutions. Although some of these capabilities are built into existing service assurance solutions, they may struggle with the move to 5G, and associated technologies such as NFV, due to increased levels of abstraction in the network design, which complicate correlation analysis.

AI could use clustering to find correlations between alarms that had previously been undetected or use classification to train the system to prioritize alarms. Traditional rule-based alarm correlation suffers from a heavy burden of rule maintenance. With ML we could instead train a system to devise its own rules based on a given set of data inputs (e.g., network telemetry).

AI could be applied to service assurance to automate the resolution of common incidents. The system could be taught by operations staff how to handle these common incidents but still require human approval before taking action (supervised or open-loop operation mode). Longer term, as humans become more comfortable with the ML technology, they may let it operate with increasing autonomy.

Fraud & Security

According to the Communications Fraud Control Association, fraud costs the global telecom industry $38 billion annually, of which roaming fraud accounts for $10.8 billion. CSPs are now using AI to identify revenue leakage (e.g., SIM box fraud) and spot discrepancies between expected results and how events are actually billed.

Traditional security technologies rely on rules and signatures to find threats, but this information can soon become out-of-date. The tactics of adversaries are evolving rapidly, and the number of advanced and unknown threats targeting CSP networks continues to increase. AI-based systems could be trained to adapt to the changing threat landscape, making independent decisions about whether an anomaly is malicious or providing context to assist human experts.

According to Heavy Reading's Telecom Security Market Tracker, AI techniques such as neural networks and ML have already been used for many years to improve the detection of malicious code and other threats within telecom traffic. And AI has the potential to go further,

such as automatically taking remediation actions or presenting a human security analyst with the right type of data on which to base a decision, and perhaps a recommendation.

One recent hot area of activity is in baselining the behavior of devices connected to the Internet of Things (IoT). Here many established vendors and AI startups are developing solutions that will help CSPs to manage IoT devices and services more securely, making use of automatic profiling of those devices.

Customer Care

One of the key applications of AI/ML in the telecom sector to date has been the use of chatbots to augment or replace human call-center agents. For example, Telstra's Kieran O'Meara, Director of Technology Design & Delivery, estimates that 30 percent of inbound calls to a contact center could be resolved by AI chatbots. There is still a role for human agents at Telstra (it has 11,000 today), but with AI assistance O'Meara estimates they can be 20-35 percent more productive. Telstra has around 300 agents managing chatbots on its websites but doesn't expect this number to grow. Instead, it plans to increase the number of agents dealing with customer enquiries directly via messaging apps such as WhatsApp.

Other examples of AI usage in customer service/support include:

- Knowledge portals and AI assistants for human agents
- Contact center optimization and compliance
- Customer voice and text sentiment analysis – Telstra is looking at using text sentiment analysis to enhance the performance of its messaging and chat agents.

AI can also be applied to CRM in areas such as personalized promotions, cross-sell/upsell opportunity identification, and churn prediction and mitigation. Research by Wise Athena investigated the use of deep learning to predict customer churn in a mobile telecom operator. They found the method more accurate than previous methods based on supervised ML classifiers.

CHALLENGES OF BUILDING & DEPLOYING AI MODELS

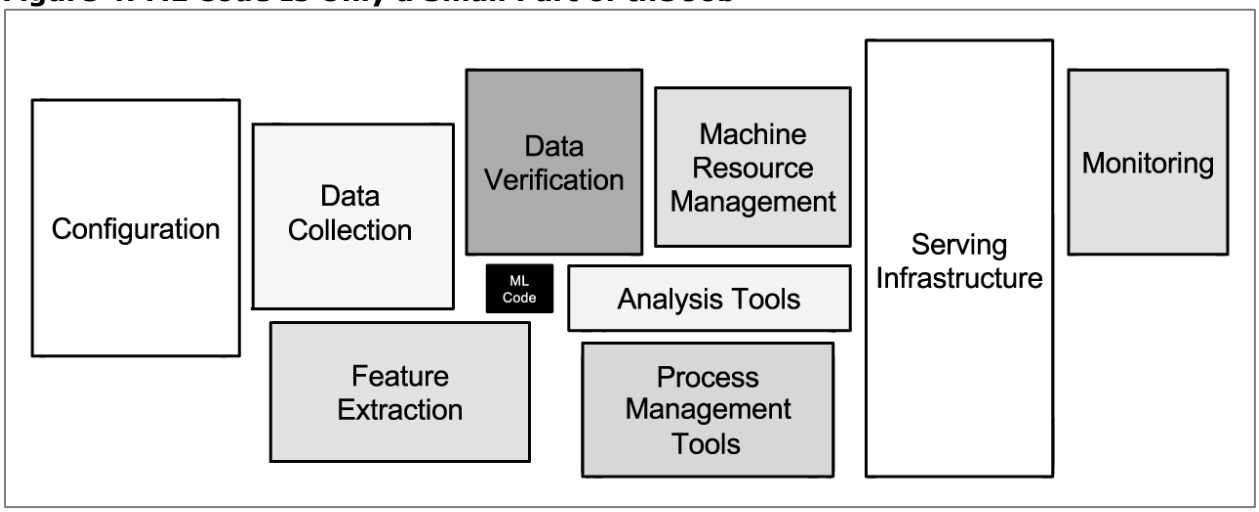
In the paper Big Data Analytics, Machine Learning and Artificial Intelligence in Next-Generation Wireless Networks, the authors highlight the challenges of adopting big data analytics and AI in the next-generation communication system. They note the complexity of managing a huge amount of data, designing algorithms for dynamic and effective processing of large data sets, and then exploiting the insights from the data analytics. Operators are particularly concerned about the effort, skills and headcount needed to manage and operate a big data platform.

Telecom engineers typically don't have backgrounds that include the kinds of mathematical training and experience that are essential in ML. Recruiting people with the right skills is a challenge. The result is that there is a serious skills gap. Deploying ML at scale requires intimate understanding of the mathematics, which is a scarce resource today inside CSPs.

Another challenge is the complex nature of tools available. Only a small fraction of real-world ML systems is comprised of the ML code itself, as represented graphically by the small black

box in the middle of **Figure 4**. The required surrounding infrastructure is vast and complex. As the paper Hidden Technical Debt in Machine Learning Systems reveals, Google finds it is common to incur massive ongoing maintenance costs in real-world AI systems.

Figure 4: ML Code Is Only a Small Part of the Job

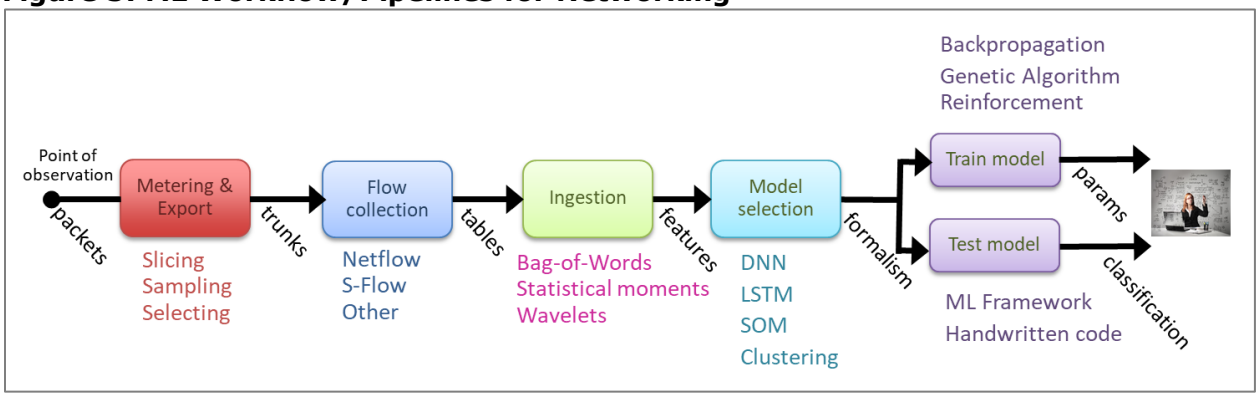


Source: Hidden Technical Debt in Machine Learning Systems, Sculley, et al.

As Michael Lanzetta, Principal Applied ML Scientist at Microsoft, explains in his chapter (ML, DL and AI) of *Artificial Intelligence for Autonomous Networks*, "The majority of any ML project is in data preparation, so building a robust data pipeline is key. This encompasses not just data acquisition but cleansing (and testing/regression of your cleansing process, as well as recleansing as it changes)."

To give a sense of the complexity of building an ML workflow for a networking application, consider **Figure 5**. First we must collect packet data, logs, KPIs, etc., using tools such as Wireshark, NFDUMP. Then the data needs to be ingested, cleaned and normalized. The user then selects a model and trains, tests and deploys the model. Only then does the user actually start to use ML algorithms on the data.

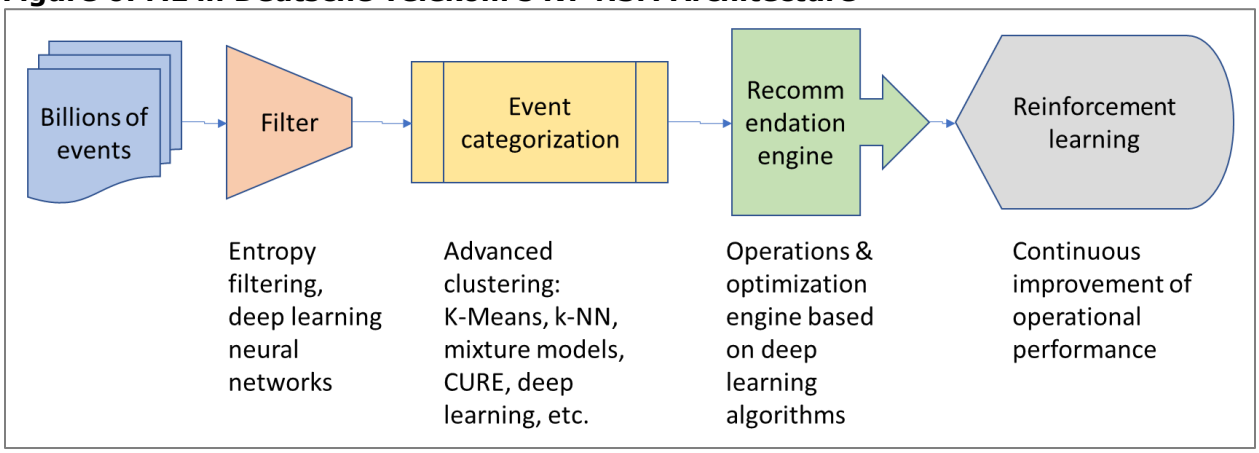
Figure 5: ML Workflow/Pipelines for Networking



Source: David Meyer

A similar workflow is shown in **Figure 6**, courtesy of Deutsche Telekom.

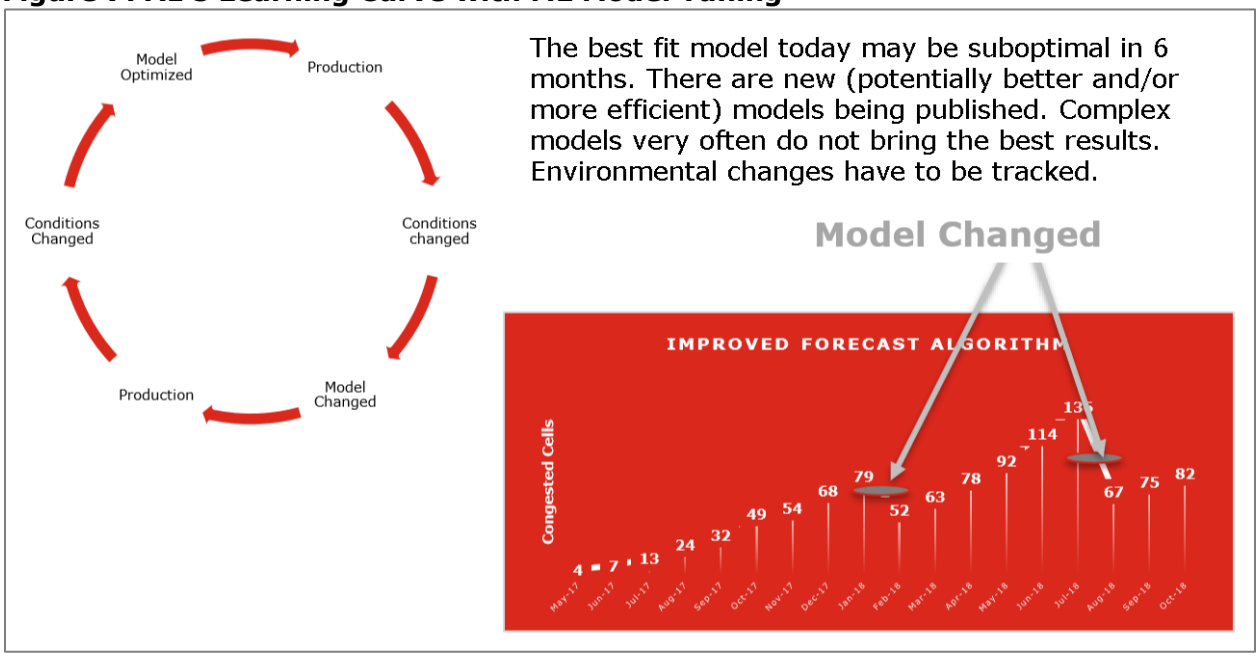
Figure 6: ML in Deutsche Telekom's RT-NSM Architecture



Source: Deutsche Telekom

In addition to the complexity of workflow, we also note the need to review ML models on an ongoing basis, as **Figure 7** from A1 Telekom indicates. In complex areas such as cyber-security bad actors may "get wise" to AI techniques, requiring operators to regularly change their ML algorithms and models in order to stay one step ahead.

Figure 7: A1's Learning Curve with ML Model Tuning



Source: A1 Telekom

THE ACUMOS PROJECT

As discussed above, there are many open source projects related to AI. However, the project that is most relevant to the telecom sector is Acumos, in our view. Acumos is the first project hosted by the Linux Foundation's Deep Learning Foundation, the founding members of which

are (in alphabetical order): Amdocs, AT&T, B.Yond, Baidu, Huawei, Intel, Nokia, Orange, Red Hat, Tech Mahindra, Tencent, Univa and ZTE. Recent additions to this community include Ciena, DiDi and Ericsson.

Acumos began life as an AT&T lab project but has been transferred to the Linux Foundation in order to provide the necessary governance and independence to prosper as an open source community. The beta version of Acumos, released in March 2018, had code from AT&T and Tech Mahindra. The first official release (Athena) was made on November 14, 2018.

The Acumos white paper explains how the development and deployment of AI applications is currently highly time-consuming and requires expensive, specialist talent. Acumos provides a common framework that reduces the need for AI "rocket scientists" (in short supply in the telecom industry) and accelerates development, thereby lowering the barriers to AI not just for CSPs, but for companies in any industry.

Data scientists create AI models with tools such as Google's TensorFlow, Scikit-learn, RCloud or H2O (or directly programmed in Java, Python or R). However, these tools are often tied to a particular execution platform, which makes it hard to integrate them with other components, if custom integration code has to be created. Also, two models on different platforms cannot talk to each other without explicit glue-code. In contrast, Acumos is IT infrastructure-agnostic; you can run it on your laptop, your local server, data center, or on your favorite public cloud platform.

By supporting a large number of toolkits and providing a user-friendly design studio, Acumos can help overcome the integration and complexity challenges of AI development. Acumos packages the AI model into a containerized microservice so that it can be easily integrated into applications by ordinary software developers without the need for data science skills or AI expertise.

By wrapping programs in a container, making each toolkit and language interoperable with the others, the range of available, compatible solutions that Acumos can use is expected to grow over time, and with it the Acumos community. Given the fast pace of change of AI technology, Acumos provides a prudent way to hedge one's bets. By focusing on interoperability, Acumos will help to keep today's AI solutions relevant for future applications.

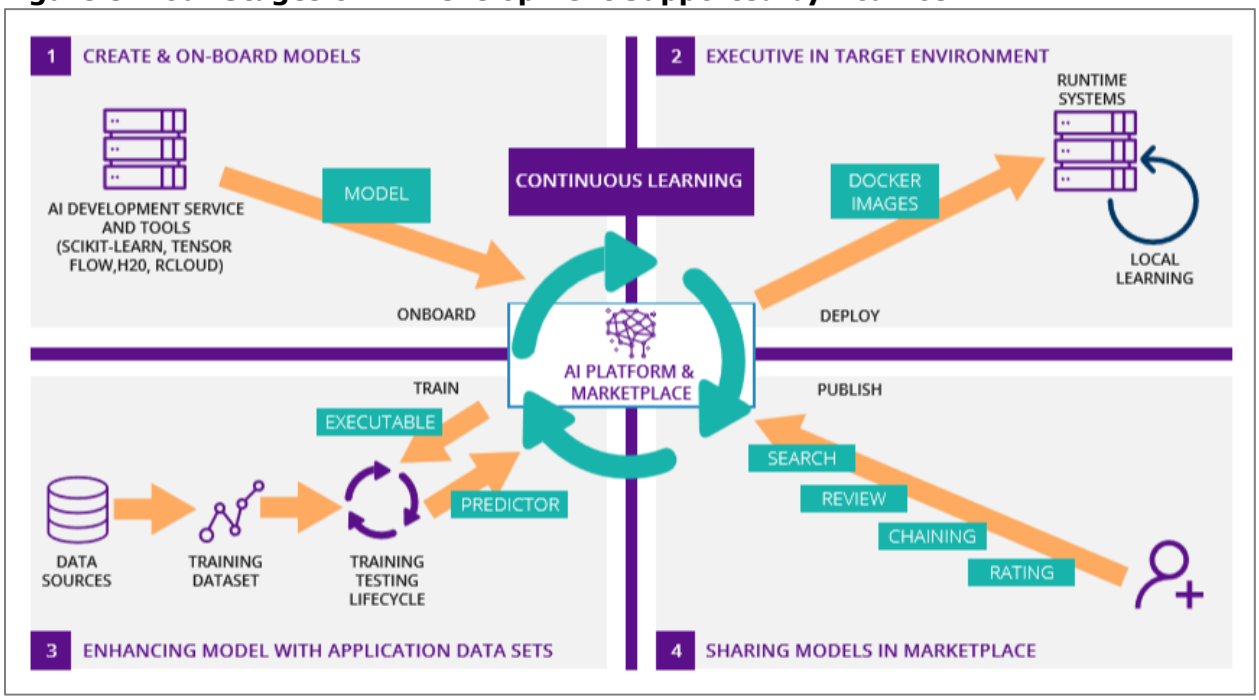
Acumos AI Development Process

Acumos breaks the AI development process into four steps, as shown in **Figure 8**. This approach separates the very specialized functions of model design and data management from the complexities of service development and application lifecycle. By keeping these aspects of AI development distinct, Acumos can make the process quicker, more reliable and open to a much larger community of users.

1. Initially models are onboarded from a data science toolkit (TensorFlow, etc.) and packaged as microservices with a component blueprint describing the microservice API and dependencies.
2. Next the model is packaged into a training application, which can be deployed to a training environment without the need for a developer to change the model in any way. Custom training clients, data access and data caching tools make it easy to assemble a specialized training application for each ML model.

3. Then Acumos provides the training and testing interface needed to turn a basic model into a Predictor that has been trained to perform a specific function. The Predictor is published to a catalog that can be shared across the Acumos community. Other developers can review it and create a full solution by using the Predictor and chaining it with other components using the Design Studio.
4. Finally, the entire solution is packed into a Docker image file that can be deployed to a Docker environment where it can be managed with a container management tool, such as Kubernetes, and executed. Image files can be deployed to Azure, AWS or other popular cloud services, to any corporate data center or any real-time environment as long as it supports Docker or any other lifecycle management tools that are supported in the future.

Figure 8: Four Stages of AI Development Supported by Acumos



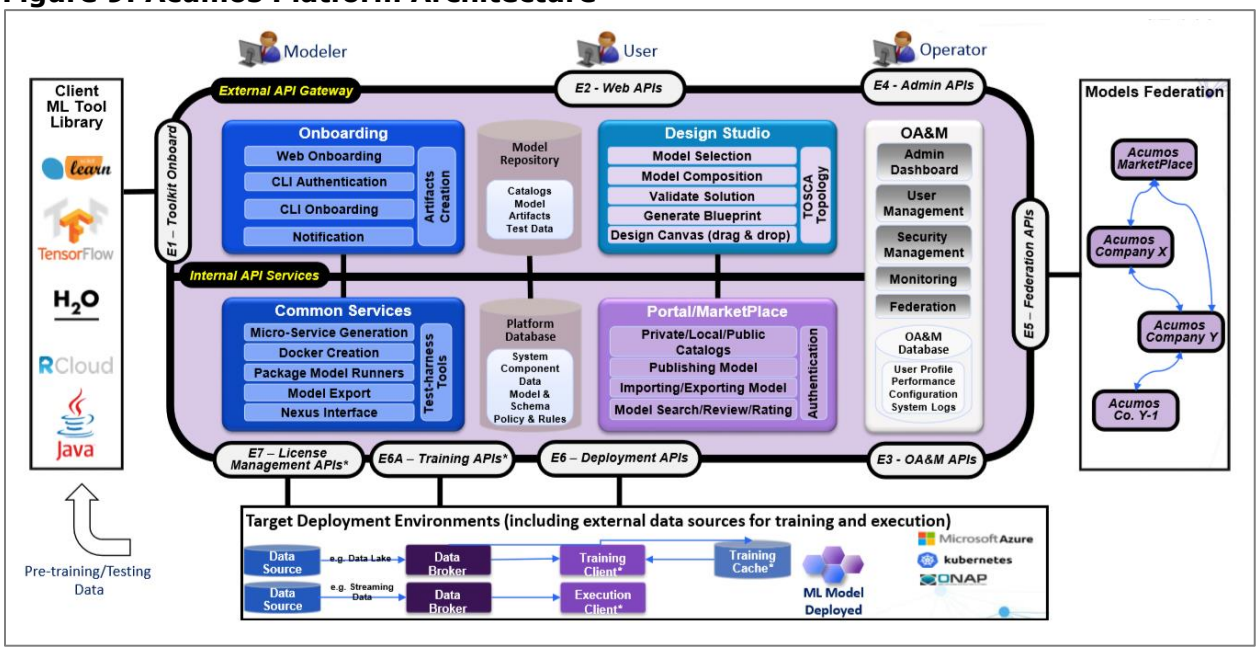
Source: Acumos

Acumos has taken the shrewd decision to leave data storage and processing outside its remit, so that it is not exposed to any data privacy risk issues. CSPs will need to use their existing big data platforms (Kafka, Hadoop, Spark, map reduce, NoSQL, etc.) and feed their chosen models with this data within their own environment.

Acumos Platform Architecture

Figure 9 shows the architecture of the Acumos platform. It comprises a function for onboarding models from ML tools (TensorFlow, etc.), a set of common services for microservice and Docker creation (developers can create and export production-ready AI applications as Docker files), the marketplace function (discussed below), and supporting operations and admin capability. The design studio provides a graphical interface for chaining together multiple models, data translation tools, filters and output adapters into a full end-to-end solution that can be deployed into any runtime environment to accelerate AI applications.

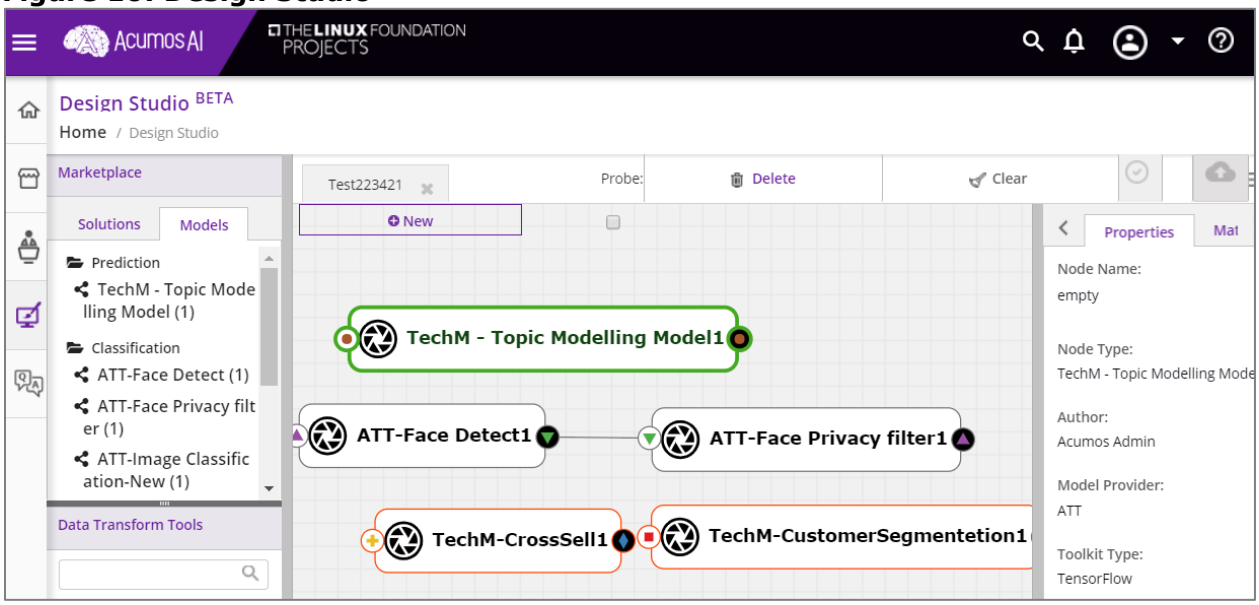
Figure 9: Acumos Platform Architecture



Source: Acumos

The design studio (see **Figure 10**) enables users to compose and stitch individual models into more complex, aggregate models that can be used in ML applications.

Figure 10: Design Studio



Source: Acumos

Acumos Marketplace

Acumos leveraged the community development concept not just to develop its own code base, but also to create a marketplace of models. Borrowing the "app store" concept, Acumos

provides a marketplace where academics and commercial data scientists can sell specialized ML models, customized to address specific industrial use cases. CSPs (and other companies) can experiment with the crowdsourced models on their own datasets and, if they prove useful, pay to use them in production. By minimizing the customization and contextualization needed to use off-the-shelf pre-trained models, users can get insights more quickly and economically.

Some models are available in the marketplace (contributed by different participating organizations) that are generic enough to be downloaded and re-trained for a specific context. These models cover areas such as image classification, face detection, face privacy filtering, image mood classification, topic modeling, cross-selling and customer segmentation.

LF Deep Learning members, including Amdocs, AT&T, Orange, Tech Mahindra and others, plan to contribute more models to the Acumos Marketplace over time, enabling anyone to download, use and customize them. Having a wide variety of models to choose from on the Acumos Marketplace should be of great benefit to business users of AI. As Michael Lanzetta explains, "Since it sometimes takes days or weeks to train a model, picking the right model structure, optimization techniques, learning rates, and other hyperparameters can make the difference between success and failure."

While the basic mathematical principles behind AI are not subject to copyright, models that have been trained on a particular type of data can be copyrighted. As Lanzetta notes, "Currently, the big players are sharing their findings (new neuron types, etc.), but we expect that to become rarer as new neuronal structures become valuable intellectual property." As such, the Acumos Marketplace could provide an opportunity for AI scientists to market their wares and for business users to tap into the expertise of AI specialists.

Athena & Boreas Releases

With the Athena release, users can now deploy models on their own hardware, including servers and virtual machines in a private Kubernetes environment or into a public or private cloud infrastructure for functions such as testing and training. Operations-wise, you can now share models privately within your team, and with specific companies, as well as share them publicly in the marketplace. You can also set up role-based permissions within your organization for accessing and using Acumos.

The next release (Boreas), scheduled for May 2019, will introduce more convenient model training as well as data extraction pipelines to make models more flexible. Boreas will include updates to assist closed-source model developers, including secure and reliable licensing components to provide execution control and performance feedback across the community. Boreas will also include integration with the Linux Foundation's ONAP and Akraino projects and will demonstrate some 5G use cases.

CONCLUSIONS

As Michael Lanzetta explains, "The AI revolution will help to drive a network that is self-healing, self-securing, and adaptive to changing needs. With the revolution in AI systems, networks can detect early patterns of cyberattacks, automatically deploy resources where needed as traffic patterns change and evolve, and improve automated routing around network failures and other issues."

Acumos aims to make AI more accessible for real-world business applications. Data scientists can contribute ML models that ordinary developers can easily use to create their own applications. By making the platform open source, Acumos will tap into the development capabilities of many participants.

Acumos empowers data scientists to publish AI models without forcing them to engage in the custom development of fully integrated solutions that use those models. Acumos is not tied to any one runtime infrastructure or cloud service. It supports many hardware infrastructures in order to maximize the utility of the solutions being deployed. This makes Acumos-compatible solutions portable and flexible.

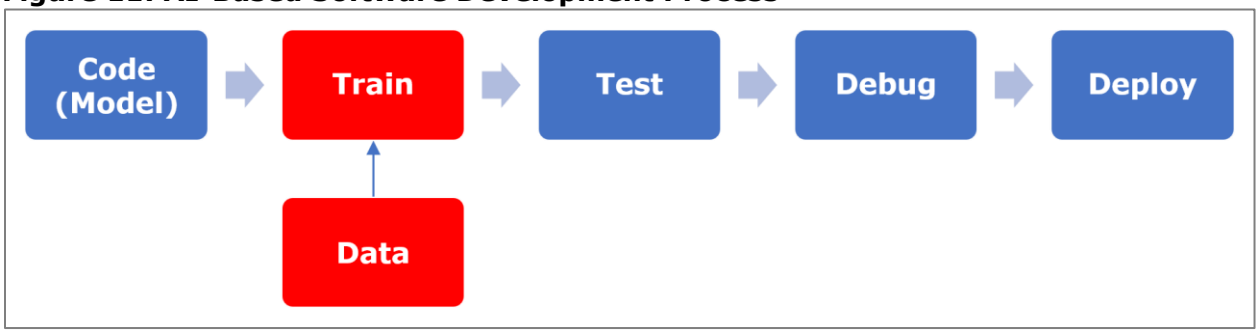
Acumos offers a mechanism for packaging, sharing, licensing and deploying AI models in the form of portable, containerized microservices that are interoperable with one another. It provides a publication mechanism for creating shared, secure catalogs and a mechanism for deployment onto any suitable runtime infrastructure.

With Acumos, the software development process will evolve from a code-writing and -editing exercise into a "code-training" process. Acumos focuses on training models after they are onboarded, published and delivered to a user, rather than as part of the model development process. This offline training involves the evaluation of trained models. End users can then select the best model for the job, and integrate that model into a complete application.

The old process of software development – code, test, debug, retest, deploy – is not sufficient for building ML systems, because it leaves out the training process. With ML, training the software model becomes a key part of the development cycle. Code must be written and trained before it can be fully debugged, making the overall development process more complex.

This changes the focus of software development toward a new set of training-oriented tools which handle things like data acquisition and storage. Data scientists will continue to program the models, but a lot of the heavy lifting will shift to training the models with large quantities of data. The training process and associated data may well be the determining factor in the overall usefulness of the solution, rather than the particular ML algorithm employed. With Acumos, developers will be able to test different algorithms with sample datasets before making a final selection.

Figure 11: AI-Based Software Development Process



Source: Heavy Reading