SCTE•ISBE CABLE-TEC EXPO® VIRTUAL EXPERIENCE » OCTOBER 12-15 2020

2020 Fall Technical Forum SCTE·ISBE • NCTA • CABLELABS®

Employing Neural Networks For Improved Root Cause Analysis In Service Provider Clouds

Pros And Cons, Use Cases, General Approach, And Best Practices

A Technical Paper prepared for SCTE•ISBE by

Helen Zeng, Ph.D.
Staff Consulting Solution Architect
VMware
Palo Alto
hzeng@vmware.com

Robert McIntyre
Product Development
VMware
Atlanta, George
bmcintyre@vmware.com

Table of Contents

List of Figures

1. Abstract

Academics, strategy pundits and entrepreneurs like to talk about the transformative power of artificial intelligence (AI) and the need to accelerate implementations. (AI has been called the “new electricity.”[1]) But service provider operations teams tend to tell a different and more nuanced story. While no one disputes the long-term potential of these technologies, what’s becoming clear is that the complexity of advanced AI, such as neural networking, is difficult to fit into their current paradigms.

The pragmatic way to adopt these technologies is to define a clear use case, start with what you already have, and layer on these technologies in such a way that operations teams can augment existing architectures. A phased approach allows room to focus on learning and enhancing rather than replacing existing network operations (NetOps) practices, including root cause analysis, in order to evaluate costs and benefits. Key to this process is focusing on a better-together approach to provide evidence-based demonstration of value. In this paper, we will introduce neural networking, explain why it is a good fit for today’s networks, and provide use cases from our experience in deploying these technologies in 5G and cable environments. It is more than possible to experiment with this technology, while evolving, rather than reinventing existing skillsets and processes.

2. Introduction

A method of problem solving that moves beyond reactive to proactive management, root cause analysis (RCA) has long been a goal of effective NetOps. As cable networks have become increasingly complex, with service delivery stitched across access networks, optical fabric, IP/MPLS backbones and, increasingly, cloud infrastructure, NetOps has evolved accordingly. In a NetOps 2.0 world, RCA has not only become an expected feature, it has needed to become smarter and more effective.

Implementing a proactive and predictive RCA in complex service provider clouds that span networks comprising more than a million devices is no trivial challenge. How do you determine the network radius of "problem spaces” that are compounded in extent by multi-tenancy, network traffic paths, and physical and logical L2/L3 connections? Neural networking-based algorithmic approaches, using non-deterministic and probabilistic models and automated computation, are one promising approach to this challenge. Employed to multi-tiered problem spaces with temporal and spatial aspects, this brand of AI technology is highly effective at executing RCA on top of non-deterministic anomaly detection, all within context.

The prerequisites are network and service discovery, along with a platform for data collection, streaming and processing. When properly set up and implemented over physical and virtual infrastructure, neural networking can drive improved RCA and its positive business effects across multiple use cases. On the other hand, if inadequately understood or lacking in data, neural networks can exceed their limits or result in less control. As with any AI or machine learning (ML)-driven initiative, upfront knowledge is key.

3. Background

3.1. NetOps, PNM and RCA

There are many aspects to NetOps, from the physical Network Operations Center (NOC) itself to the personnel who work there to the policies employed to the technologies used in managing, monitoring and control any number of discrete or interrelated networks. A common term used in the cable industry used for optimizing NetOps is proactive network maintenance (PNM). As a part of a special initiative covering wired, Wi-Fi and optical technologies, CableLabs has encouraged careful thinking about PNM,

emphasizing the benefits of proactive management.[2] The point of any such exercise, of course, is a higher level of customer service.[3]

The RCA approach to problem solving falls into that broad category of proactive measures, in that it seeks to uncover the underlying cause of faults or incidents affecting network performance; and because the failure to address that cause is likely results in recurring issues. By isolating the fundamental fault associated with a multiplicity of issues, RCA leads to a reduction in immediate number of alarms, as well as a lower number over time. RCA does not by itself remediate problems, but feeds into a process of corrective and preventative action.

3.2. Service Provider Clouds

A typical domain for applying the tools of PNM is the cable access plant, for instance DOCSIS-related problems that occur on that part of the network spanning the CMTS or CCAP device and the related modem at the customer premise. Yet with virtualization, that network now extends beyond the physical plant into cloud infrastructure.

The distributed access architecture (DAA) initiative remains one of the industry's most prominent use cases related to virtualization. Being transformed from purpose-built hardware into software enables a cable operator to run the CCAP in a data center on a private cloud. Other possible cases for MSO virtualization include VOD, network PVR, 5G and Multi-access Edge Computing (MEC).[4] These cases increasingly cross industry lines, many applications all requiring high-performance, low-latency networks. The expansion of these networks has placed a tax on traditional NetOps. MSOs have expressed frustration with monolithic management tools, and delight in those that deliver cross-domain results.[5] An extensive physical and cloud environment makes it difficult to rely on siloed legacy tools with manual processes and scripts. (See Figure 1.)

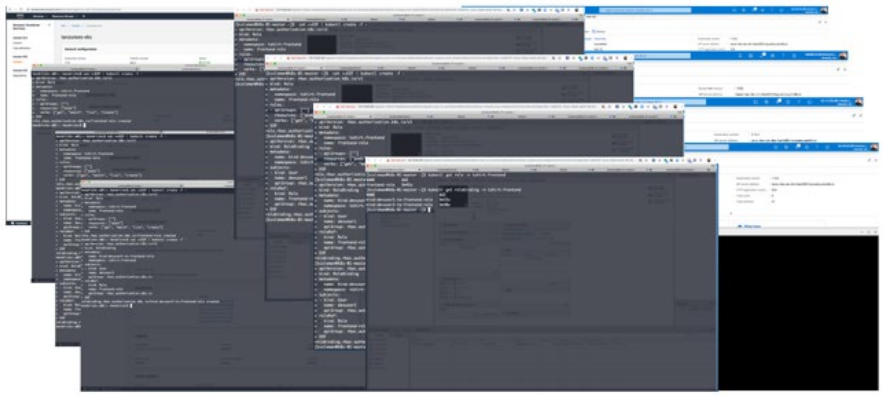

Figure 1 – Multiple Screens Manual Processes

3.3. Neural Networking

In that context, it is fair to ask whether networks can survive without AI. The real question today, however, is what kind of AI or ML serves best. Neural networking is worth considering. It may appear to be a recent addition to this category, but its genesis goes back decades. Building upon theories of biological neural network, i.e. the brain, the seminal paper on artificial neural networks appeared in

1943.[7] By the late 1990s, engineers were looking at possible applications in manufacturing.[8] Driven in part by innovative big-data ingest and storage capabilities, neural networking in recent years has achieved tremendous progress in areas such as natural language processing and image recognition.

In the cable and telecommunications arenas, other kinds of AI and ML have taken the lead. One example is a trial that involved "operational analytics (OA) and machine intelligence (MI)" to predict service impairments in near real-time, which used Support Vector Machine (SVM) classification, spectral clustering, tree-based taxonomy and related tools.[9] Like these uses of advanced data analytics, neural networks can very well approximate the non-deterministic, stochastic nature of various network scenarios.

Neural networking is typically supervised, i.e. it requires setting up and training a data model. A key characteristic is that these networks involve layers of input, hidden and output neural nodes, each connected to others, with certain weighed coefficients. Besides feed-forward networks, there are other kinds of deep-learning models, including recursive neural networks (RNN) and convolutional neural networks (CNN), etc. But neural networking can be unsupervised as well, especially when no labeled data is available. The system tries to find patterns and form clusters in a meaningful way. It involves mapping the continuous random variables to discrete representations, while neural networks are capable of achieving that through their ability to converge.

4. Are Neural Networks a Good Fit?

4.1. Modern Network Characteristics

There are a number of reasons for servicer providers of all stripes to consider the applicability neural networking for key NetOps problems, such as RCA. One overriding reason is that for many, networks are simply not what they were only a few years ago. Nor have they reached an endpoint. Monitoring and analytic tools and platforms simply need to keep pace with advancing network technologies. that now increasingly bear these characteristics:

4.1.1. Complex

A conventional approach to network monitoring and management has involved applying simple thresholds to classifying physical layer HFC performance within normal and abnormal parameters. That remains an effective way to trigger alarms, but insufficient when the goal is to isolate underlying problems affecting services that touch heterogenous networks, each of which may have its own siloed data arranged in unique formats. Assessing and weighing multiple inputs, and becoming even smarter over time, is possible with the computational power and adaptability of a neural network.

4.1.2. Dynamic

The disaggregation of once-unified equipment, such as the CMTS or CCAP device, into core and remote elements, is an indication of not only growing network complexity but also accelerated change. The addition of remote PHY devices (RPDs) entails new configurations and adjusted topology adjustments, while a virtualized CMTS core now depends upon cloud infrastructure that may rapidly scale up and down. Edge wireless, IoT devices and even CPE contribute more unknowns to the mix. Neural networks are adept at handling dynamic change.

4.1.3. Data-rich

The sheer volume and variety of data emanating from today's networks is one of the biggest drivers in the search for new approaches. Only a few years ago, network data mean SNMP queries that delivered a limited amount of information every fifteen minutes, and of that very little was useful. MSOs had tools that could handle those challenges. The data challenges now facing many MSOs are beyond their scope. Neural networks actually work better with inputs provided by big data analytics.

4.1.4. Real-time

Exacting performance in areas such as latency matter to many end customers, whether the scenario is cell backhaul or internet gaming. Data about those services matters, too. But even if MSOs were able to staff their NOCs with the sharpest analysts, the rate at which performance data arrives and the need to process them with utmost speed is beyond human capabilities. Manual review of Syslogs and support cases do not scale, and status-quo analytic tools fall short. To respond to network behavior real-time, service providers need certain automatic ML-based mechanisms, and neural networking is a very good ML model.

4.2. No Free Lunch

Neural networking is not the only advanced AI and ML-based analytic tool available. For supervised classification problems, there are some commonly used ML algorithms, such as SVM, decision tree, logistic regression, etc. For decision tree, rigidness of the model is the common problem which easily leads to over-fitting. Although they can be trained to be accurate, once given new data, they may jump to wrong conclusions. That could happen through creating tree branches that follow certain order of precedence. Ensemble methods, however, can alleviate this problem.

SVM performs data separation either linearly or non-linearly, which highly relies on the choice of kernel function. It makes a model difficult to scale well to a large dataset. While logistic regression tends to underperform when there are multiple or non-linear decision boundaries, it is not flexible enough to naturally capture more complex relationships. Neural networking is more flexible as it has the capability to approximate almost any scenario accurately by adjusting hidden layers and hidden nodes.

A common criticism of neural networks is that they operate like black boxes. While a decision tree is relatively easy to understand, it is hard to explain how a neural network arrives a particular conclusion. Neutral networks do well with large amounts of data, which makes them a good fit for advanced communications networks; but the converse applies, they do less well in scenarios that are data-deprived. Computationally powerful, they can also require more time to train than traditional ML algorithms, which can increase their cost. Likewise, operating them requires some expertise in data science.

Shortchanging knowledge and taking a set-it-and-forget-it approach can be tempting but dangerous. A single-engine propeller aircraft can be operated and maintained manually; whereas a commercial jet airliner requires NetOps 2.0-level systems and procedures. Yet when pilots are unable to override these systems and are unprepared for edge cases that may arise, they cede too much control to the automation that was designed to optimize flight operations.[10]

A similar charge could be leveled against a neural network, which generally speaking is a "non-convex optimization problem," especially when the activation function is non-linear. (Since weights are permutable across layers there are multiple solutions for any minima.) Sometimes, domain knowledge is needed to make sounding judgements when selecting reasonable ML tuning parameters. The takeaway is that it is important, whether building one of these platforms yourself or working with a partner, to understand the technology's potential benefits, its operating characteristics and its limits.

5. Existing Frameworks

How do existing service assurance platforms handle RCA? A crucial prerequisite of any framework is data collection, streaming processing and data modeling. As for RCA more directly, one currently effective approach is to rely on multi-dimensional matrices of deterministic models that create 'signatures' related to symptoms and problems. The idea is to correlate events and alarms with known patterns and signatures to identify where the root of the problem lies. Other best practices include continuous self-updating; adapting to dynamic workloads, network configurations and inventory; and integrating with orchestration tools for auto-remediation and incident management for support workflows.

What status quo platforms typically do not yet have is a continuous flow of anomaly detection, leading to RCA for the anomaly, and finally providing prescription for the problem. A non-deterministic, neural networking engine can extend while enhancing other effective techniques for anomaly detection, RCA and problem prescription. The neural networking ML utilizes streaming processing output, although the training can be done off-line. Then streaming processing engine picks up ML result, through multi-stages to generate real-time results in an automated fashion. This paper focuses on RCA, while the problem prescription is out of scope.

Below is an example of flow using neural networking in radio access network (RAN) throughout the whole data analytics process. (See Figure 2.)

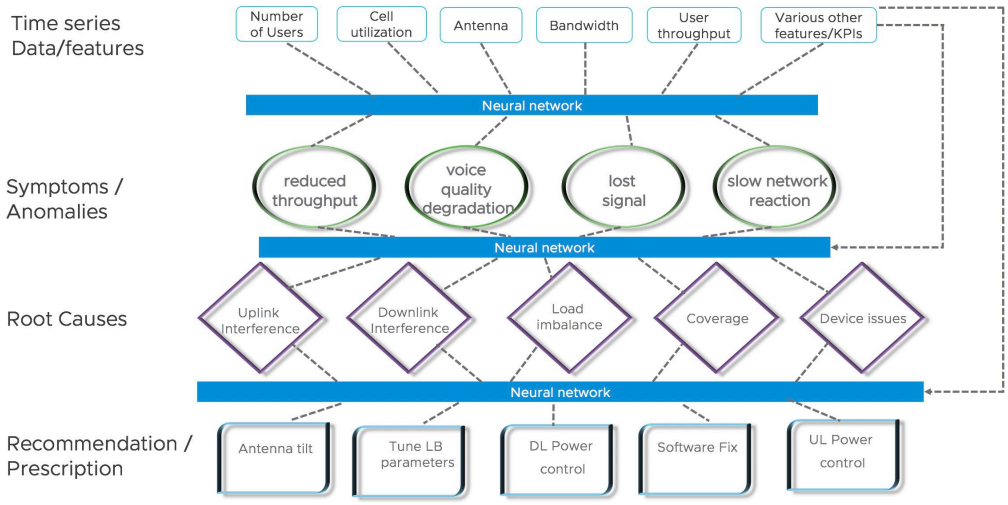

Figure 2 – Continuous ML Flow

6. Use Case Scenarios

6.1. Anomaly Detection, RCA

Neural networking works well for anomaly detection, which is both a powerful technique and a prime use case. The detection of anomalies, or statistical outliers, is applicable in complex, on-prem/off-prem and dynamic environments where operators are interested in long-term trends, aberrations such as spikes, and cross-correlation with other inputs. In the case of having no global formula to define anomaly, neural networking is able to learn the network performance down to the subscriber level over a long-enough period of time and determine the anomaly. All of that information can help drive effective RCA, which

could also be considered a use case. Anomaly detection can be supervised, unsupervised or semi-supervised, depending upon the use of a training data set. Like RCA, it figures in several other use-case scenarios.

6.2. Performance Management

To track the delivery of any number of services across a large subscriber base, a service provider today needs to receive concurrent data feeds from a massive number of network elements, correlate those with user session data, and then calculate real-time key performance indictors (KPIs). The big data platform enables an operator to achieve those tasks. Neural networking is built on top of the platform to combine the KPIs with application-specific inputs and policies to deliver network visibility, anomaly detection, and real-time predictive network intelligence.

6.3. Fault Management and Configuration Management

A fault management system must have a comprehensive picture of the network topology, which is related to configuration management. There are thousands of ways can a network fail, go sideways or skip a beat. Hardware failure, connectivity loss, and power outages are some types of network faults. Let's consider one, the misconfiguration of a network device, such as an RPD. Anomaly detection might first indicate that subscriber data throughout is not matching up with other session-level parameters. A system equipped with neural networks for anomaly detection might detect conditions where average user throughput dropped below a certain level, while the channel utilization was abnormally high, and DOCSIS sub-carrier utilization low. Identifying the condition and the underlying cause then enables a recommendation for device reconfiguration.

Another example is a cellular tower fault. Each cell has neighbors, and if one cell is in trouble, then the user traffic is distributed to the neighbor cells, at which point the neighbor cells will show an abnormal increase of traffic amount and maybe congestion. Some network faults directly impact or even block service delivery. Neural networking can certainly help network fault management to detect/predict fault and further diagnose the source and type of the fault. The subsequent action is to automatically trigger fault correction, or to at least prevent incidents from happening in the future.

6.4. Predictive Maintenance

In its early forms, neural networking was seen as a way to help monitor manufacturing processes, predict failures and schedule maintenance on aging equipment. It likewise applies to telecommunication and IT network equipment, which can experience transient or permanent hardware or software malfunction. One concern involving new proliferating IoT endpoints is battery life.[11] Neural networking could assist in managing this and other aspects of IoT, including the detection of missing data transmission and related RCA, either as part of an MSO's own network or a managed service to businesses.

6.5. Security and Fraud

In the area of intrusion detection, network security experts have combined anomaly detection models with various deep neural networking structures.[12] Models could detect strong login behavior, too many DNS requests or frequent changing of temporary IP or ID during a session. Operators themselves are motivated to prevent telecommunications fraud on their own networks and to deliver high-value managed security services to other businesses. To detect fraud, the data covers several domains, such as network usage in multiple types of networks, financial, personal information, location information, etc. Fraudsters are good at frequently changing their behavior to match the 'normal' pattern of the network to circumvent the security rules, which increases the level of difficulty for fraud detection. In addition to detecting

identity theft and fraudulent activity, network operations intelligence has also had to track the location and time of fraud. The data-mining capability of neural networking has made it a candidate for providing better information to MSOs about unsanctioned use of telecom networks, whether for financial gain, abuse of services, ghosting, tampering, cloning or other fraud.[13]

7. General Neural Networking Approach

A neural network-based service assurance recognizes that the world has changed. Service-related data once arrived slowly and with limited utility. Then services became IP-centric, and it was economically and technically feasible to ingest and store exponentially more data than before. Applying neural networking for anomaly detection, RCA and other uses entails completing a sequence of tasks: data collection (and network discovery), data processing, data modeling and follow-up adjustments. (See Figure 3.)

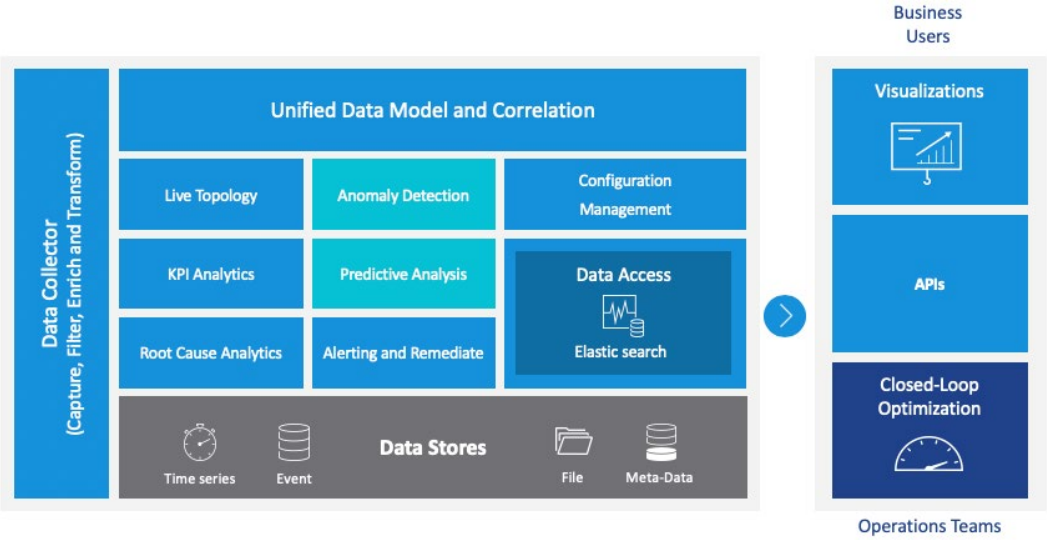

Figure 3 – Data Analytics Model

7.1. Data Collection

This task involves collecting fault and performance metrics for each network component at the user level. There are traces, events as performance metrics. The fields include start and end-time on each of 1000s of pieces of equipment, user ID, traffic type (voice/video/data) IP address and location, throughput, traffic volume, etc. For security and fraud use cases, typically call detail records (CDR) and network management system (NMS) logs are needed. CDR and logs have subscriber-level call details and activity records in a certain period, i.e., once a month. Network devices could use various industry standard data collection methods, such as SNMP, REST, NEP, etc., for streaming data sources. The time stamp is important for model training and prediction. Data collectors may span multiple vendors, with their own implementations and formats, as well across network silos. What generally appears in the NetConf standard are configuration files, counters and information about the state of the device.

7.2. Data Processing Pipeline

After setting up the collectors, the next step is to set up real-time streaming and processing of the time-stamped data into a data pipeline and connecting to a database. The process of data joining can be a

challenge in the streaming domain, with SQL joins being especially slow and difficult. Yet standard big-data and schema-agnostic database formats have helped smooth the preparation phase. Related tasks include filtering, sanitizing the data, as well as running network topology discovery to associate data with device location. There is a Kafka bus to connect various big-data components in the pipeline. Those components exchange data/information through the Kafka bus. Some big data frameworks are Apache Spark, Twitter Heron, Apache Flink, etc.

A common way of storing the processed data is to ingest data via Kafka into a data lake, which will be used for neural networking model training. The streaming processing engine is equipped with a ML module. The module can get the ML model result and uses streaming data to generate prediction results, such as anomaly detection or RCA.

Below is one example of streaming data processing pipeline including neural networking model training engine. (See Figure 4.) Note that data lake is not the only way to store the data; there are other ways which do not require a data lake, such as tiered storage in Kafka. Also, some neural networking ML use cases do not require the results to be fed back to real-time streaming processing engine; they can certainly run separately.

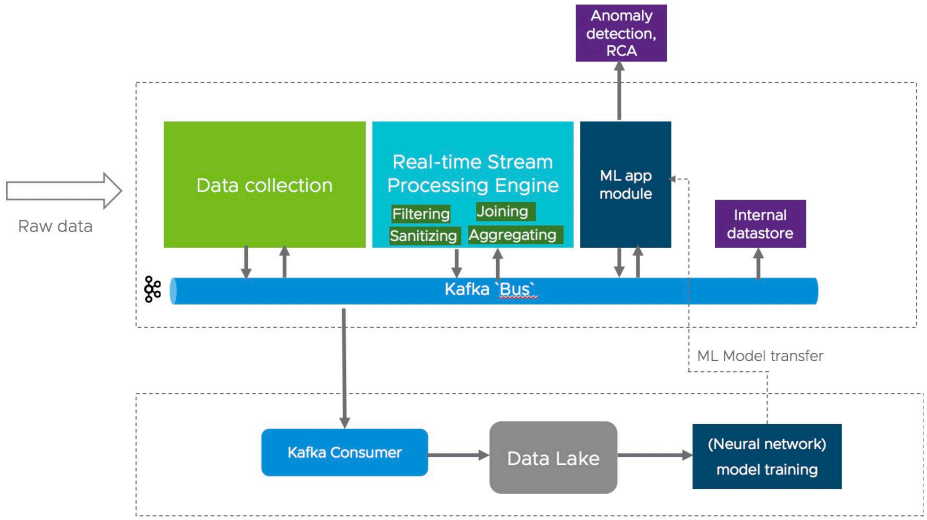

Figure 4 – An Example of Streaming Processing Data Pipeline

7.3. Further Adjustment

Other data management techniques may be required. During data preparation, in the case of missing data, average or default data may be inserted, or the fields could be ignored. Several approaches can be taken in the case of data skewness, such as oversampling, or synthetic minority oversampling technique (SMOTE) to compensate for an imbalance within the set. Neural networks tend to make better predictions when trained on balanced data.

7.4. Data Modeling

Correlating chosen KPIs (containing time stamp, user ID, traffic type, etc.) with different network components is important to establishing an end-to-end view. A critical part of building a model is identifying features, which correspond to parameters or counters in events. These could be throughput, signal strength, transmit power, or whichever ones best align with the use case in question. All of those

features are associated with a single time stamp that represents training data that are fed into the model. Data scientists can optimize the model's "fit" by adjusting the weights and layers of the middle nodes, which may have distinctive properties of their own, such as the ability to do "convolutional" math or look back across the network in a "recursive" manner. If so engineered, the model will deliver outcomes that determine a root cause or detect anomalies. (See Figure 5.) There are various ML framework libraries to draw upon, including TensorFlow, Theano, Deeplearning4j, Keras, sklearn, etc.

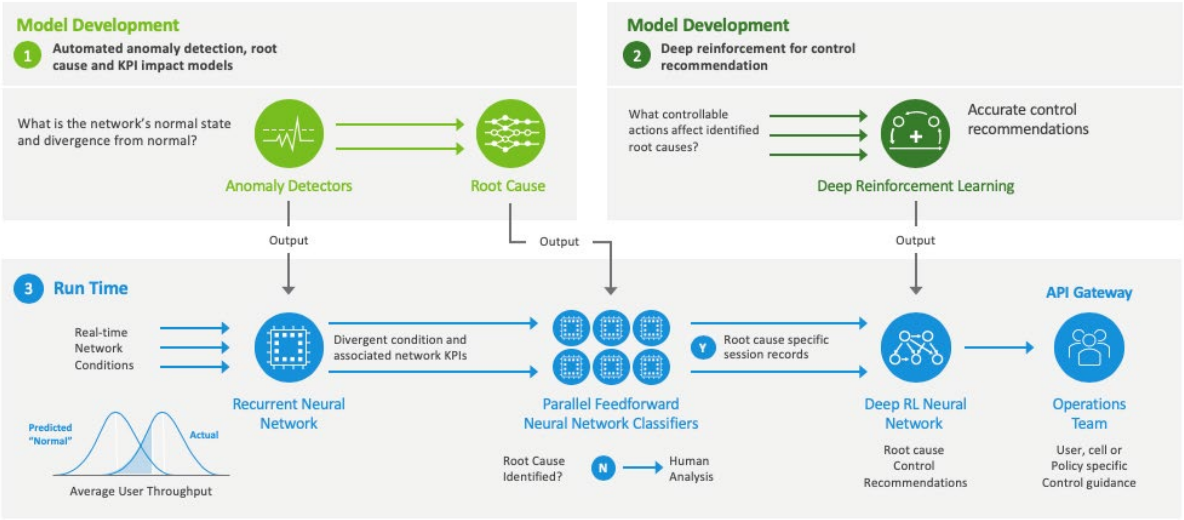

Figure 5 – Data Model Development

7.5. Continuous Learning

The model training is a continuous process. We should proactively compare ML predicted result with ground truth and then adjust the model if necessary. A separate effort can be placed on continuously enhancing the model, and then feedback the new model in the subsequent data analytics flow. (See Figure 6.)

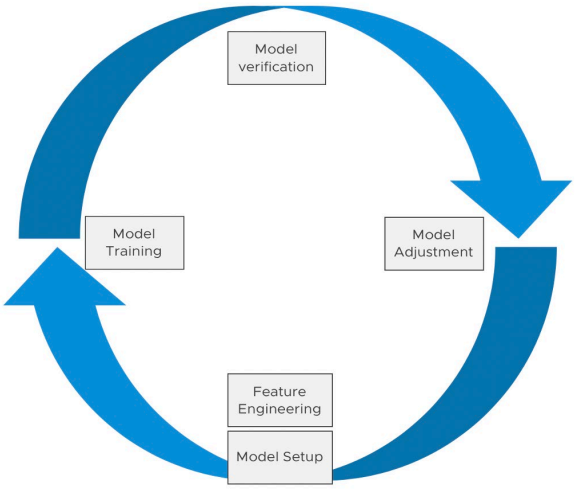

Figure 6 – Continuous Training Cycle

7.6. Model Management

After a neural networking ML model is delivered, the following criteria need to be evaluated:

- ML performance, which can be evaluated by comparing the prediction result with ground truth and quantified by mathematical formula.
- What features are used? Can the feature list be changed?
- Are the training data generalized enough?
- How much training data is enough?
- Is there any over-fitting or under-fitting?
- What are the optimal hyper parameters? Are they generalized or specific to particular scenarios?
- Is human input needed? For example, sometimes human help is needed for training data labeling, or domain knowledge is needed to customize the ML prediction result.

The above considerations should be taken into the model management. Further actions like model versioning, customization, refinement, profile, performance recording, dependency should be incorporated when deploying models:

- Model versioning: If a model is only suitable for old data, a version is used to avoid using the wrong model on new data.
- Model customization: ML models are normally consistent across deployments but can be customized to take into account market-specific conditions. For example, different hyper-parameter lists, or values are used for different markets (due to different RF bands, geographical situation or other factors); different feature lists or weights for different markets or clusters, etc. Another case is that network performance should follow some theoretical rule, but sometimes an AI/ML-driven prediction lands outside of the reasonable boundary; in which case human customization or adjustment is needed.
- Model profile: Certain mechanisms to prevent over-fitting, under-fitting, and ML software framework can be recorded for evaluating ML model performance.
- Model performance: This helps determine how good a model is. It can be evaluated in various ways, such as, F1-score, area under the curve (AUC), receiver operating characteristics (ROC) curve, learning curve, R2-score for regression algorithm, etc. Also the learning curve can help determine how much training-set data are needed or are sufficient.
- Model dependency: When new or novel patterns are detected, the associated event stream can be used as labeled data by domain experts (which suggests human input) for re-training of models using supervised learning. This human-input dependency can be incorporated into model-training software.

7.7. Model Optimization and Automation

Machine learning helps realize a big portion of network operations automation. But the ML model itself can also be optimized and automated. In the case of neural networking, the model performance heavily relies upon the hyper parameters, including the number of hidden layers, the number of nodes in each hidden layer, the learning rate, etc. In a production environment, it is crucial to automate the process of selecting optimum tuning parameters. Bayesian optimization appears to be an efficient choice, as it not only helps find the vector of hyper parameters that result in a neural network with the lowest error, but also reduces considerably the time spent on model tuning. There are other optimization tools or ML libraries available, as well.

8. Other Limitations

As mentioned earlier, while neural networking thrives on large amounts of data, in some cases there may not be enough. Challenges surrounding data joining have also been noted. Mistakes on a key or value could result in the time stamp being off, which would make it difficult to correlate data belonging to the same user session. One solution is to set a slightly wider time range. Finally, there are always going to be false positives and false negatives. The tradeoff between precision and recall cannot be avoided. In setting up the model and use case, operators should take note of the use case and consider any false positives.

9. Best Practices for Adoption of Neural Networking

Experience across the wider service provider industry has revealed an effective way to approach this powerful and challenging technology. If you are motivated to successfully adopt neural networking for RCA, consider the following key principles:

9.1. Start Where You Are

Allow yourself room to experiment. Consider a data-driven before/after business case that can be used to showcase progress and share learnings. We recommend starting with one of the use cases mentioned above, perhaps anomaly detection, being applicable in many scenarios.

9.2. Layer on New Capabilities, Stitched Together at a Workflow-level

Early adopters are thrilled to discover that neural networking can deliver insights that were previously impossible to obtain. Yet the difficulty in explaining how it works can hinder practical and actionable impact. By framing the technology as an augmentation, on a second screen or a dashboard with all anomalies detected by neural networking tagged as an overlay, operations teams can get started with the technology in a way that stages risks and provides ample room for learning and evolving, both in terms of testing and tuning the models, but also to mitigate false positives while the models are being developed.

9.3. Collaborate, Internally and Externally

Given the rate at which these technologies are evolving, and the centrality of model definition and evolution, it pays to collaborate with other teams both internally and externally. In our examples here, many of the models applicable to any network scenario were developed initially for 5G interference scenarios but are equally applicable to DOCSIS environments and LLX deployments. Look for ways to collaborate across silos and share insights.

9.4. Be Metrics-driven: Use Testing to Show Impact over Time

Neural networking is not a reinvention as much as an evolution, and in to demonstrate value and prioritize investments in these technologies, vanguard teams who are seeing the most success with implementations have taken “show me” approaches to proving value. The most straightforward way is to define a clear use case and employ A/B testing to demonstrate the value with before-and-after metrics. Prioritize future steps accordingly.

9.5. Avail Yourself of Expertise When You Need It

Because of the learning curve and pitfalls of developing, training, evolving, and explaining neural networking models, having access to the right level of expertise is crucial in getting these programs off the ground. It can be time-consuming to identify domain experts who are adept at translating the data science principles into the context of RCA. Plan ahead.

Abbreviations

5G	fifth-generation cellular wireless
AI	artificial intelligence
AUC	area under the curve
CCAP	converged cable access platform
CDR	call detail records
CMTS	cable modem termination system
CNN	convolutional neural network
CPE	customer premises equipment
DNS	domain name system
HFC	hybrid/fiber coax
IoT	internet of things
IP	internet protocol
KPI	key performance indicator
LLX	low latency Xhaul
MEC	multi-access edge compute
MI	machine intelligence
ML	machine learning
MPLS	multi-protocol label switching
MSO	multiple systems operator
NETCONF	network configuration protocol
NMS	network management system
NetOps	network operations
NOC	network operations center
OA	operations analysis
PNM	proactive network maintenance
PVR	personal video recorder
RAN	radio access network
RCA	root-cause analysis
REST	representational state transfer
ROC	receiver operating characteristics
RNN	recursive neural network
RPD	remote PHY device
SNMP	simple network management protocol
SMOTE	synthetic minority oversampling technique
SQL	structured query language
SVM	support-vector machine
VOD	video on demand

Bibliography & References

[1] Shana Lynch, “Andrew Ng: Why AI is the New Electricity,” Insights, Stanford Business, March 11, 2017

[2] Jason Rupe, "A General-Purpose Operations Cost Model to Support Proactive Network Maintenance and More," SCTE-ISBE, 2019

[3] Andrew J. Milley, "Proactive Customer Maintenance," SCTE-ISBE, 2019

[4] Andrew Bender, "A Roadmap for Virtualization in HFC Networks," SCTE-ISBE, 2019

[5] Jeff Baumgartner, "Comcast's AI action extends to the network core," Light Reading, July 20, 2020

[6] Claudio Righetti, et al., "Can Future Networks Survive Without Artificial Intelligence," SCTE-ISBE, 2019

[7] Warren McColloch and Walter Pitts, "A logical calculus of the ideas immanent in nervous activity," The bulletin of mathematical biophysics, vol. 5, pp. 115-133, 1943

[8] Tiago A. Piedras Lopes, Antonio Carolos R. Troyman, "Neural Networks on Predictive Maintenance of Turbomachinery," IFAC Proceedings Volumes, Vol. 30, Issue 18, August 1997.

[9] Justin Watson and Roger Brooks, "Predicting Service Impairments from Set-top Box Errors in Near Real-Time and What to Do About It," SCTE-ISBE, 2018

[10] Nadeem, “The Deadly Price of the Automation Paradox,” The Walrus, September 26, 2019

[11] Joe Rodolico, "Methods to Maximize IoT Battery Life," SCTE-ISBE, 2019

[12] Naseer, et al., “Enhanced Network Anomaly Detection Based on Deep Neural Network,” IEEE Xplore, Aug 17, 2018.

[13] Gurunadham, “Identifying Telecommunication Deception using Neural Networks through Data Mining,” International Journal of Engineering and Techniques, Vol. 3, Issue 6, Dec 2017.

